Statistical Physics for Communications, Signal Processing, and Computer Science

EPFL

Nicolas Macris and Rüdiger Urbanke

Contents

Foreword

Statistical physics, over more than a century, has developed powerful techniques to analyze systems consisting of many interacting "particles." In the last fifteen years, it has become increasingly clear that the very same techniques can be applied successfully to problems in engineering such communications, signal processing, or computer science.

Unfortunately there are several hurdles which one encounters when one tries to make use of these methods.

First, there is the language. Statistical mechanics has developed over the last 150 years with the aim of providing models and deriving predictions for various physical phenomenon, such as magnetism or the behavior of gases. This long history, together with the specific areas of their original application, has resulted in a rich language whose origins and meaning are not always clear to someone just starting in the field. It therefore takes a considerable effort to learn this language.

Second, except for extremely simple models, the "calculations" which are necessary are often long and daunting and frequently use little tricks and conventions somewhat outside the realm what one usually picks up in a calculus class. A good way of overcoming this difficulty is to start with a familiar example, casting it in terms of statistical physics notation, and by then going through some basic calculations.

Third, and connected to the second point, not all methods and tricks used in the calculations are mathematically rigorous. Some of the most powerful techniques, such as the cavity method, currently do not have a rigorous mathematical justification. In the "right hands" they can do miracles and give predictions which are currently not possible to derive with any classical method. But a newcomer to the field might quickly despair in trying to figure out what parts are mathematical rigorous and what parts are "most likely correct" but cannot currently be justified. Both worlds are valuable. The cavity or replica method give predictions which would be very difficult to guess. These predictions can then be used as a starting point for a rigorous proof. But it is important to cleanly separate the two worlds.

Our aim in writing these notes is not to give an exhaustive account of all there is to know about statistical mechanics ideas applied to engineering problems.

Indeed, several excellent books which take a much more in-depth look already exist. We in particular recommend [1, 2].

Our aim was to write the simplest non-trivial account of the most useful statistical mechanics methods so as to ease the transition for anyone interested in this strange put powerful world. Therefore, whenever we were faced with an option between completeness and simplicity, we chose simplicity. On purpose our language changes progressively throughout the text. Whereas at the beginning we try to avoid as much jargon as possible, we progressively start talking like a physicist. Most of the literature uses this language, so you better get used to it.

We decided to structure our notes around three important problems, namely error correcting codes, compressive sensing, and the random K-SAT problem. Although we will introduce basic versions of each of these problems, we only introduce what is necessary for our purpose. It goes without saying that there are myriad of versions and extensions, none of which we discuss. In fact, we hope that the reader is already somewhat familiar with these topics and accepts that these are important problems worth while studying. Using the basic versions of these problems we explain how they can be cast in a statistical physics framework and how standard concepts and techniques from statistical physics can be used to study these problems. This allows us to introduce the necessary terminology step by step, just when it is needed.

The notes are further partitioned into three parts. In the first part, comprised of Chapters 1-4, we introduce the problems, some of the language, and we rewrite these problems in the language of statistical physics. In the first chapter of the second part, namely Chapter 5, we then introduce the main protagonist, a message-passing algorithm which is also know as the *belief-propagation* algorithm. The remaining chapters of the second part, namely Chapters 6-9.5, contain the analysis of the performance of our three problems under this low-complexity algorithm. We will see that, in many cases, even this simple combination yields excellent performance. Finally, in the third part, consisting of Chapters 11-13, we get to the perhaps most surprising part of our story. Our aim will be to study the fundamental behavior of these three problems without the restriction to low complexity algorithms. I.e., how well would these systems work under optimal processing. The surprise is that the same quantities which appeared in our study of low-complexity suboptimal message-passing algorithms will play center stage also for this seemingly completely unrelated question.

Although we follow essentially the same pattern for each of the three problems, we will see that they are not all equally difficult.

Error correcting coding is perhaps easiest, and in principle most of the question one might be interested in can be answered rigorously. In this case we are dealing with large graphically models which are locally "tree like." It is therefore perhaps not so surprising that message-passing algorithms work well in this setting and that the performance can be analyzed.

Compressive sensing follows a similar pattern but introduces a few more wrinkles. In particular, the story of compressive sensing is leading to the so-called

AMP algorithm. The surprising fact here is that message-passing works very well, and that its performance can be predicted, despite that the relevant graphical model is not sparse at all but rather is a complete tree. The key observation is that every single edge contributes very little to the global performance. AMP can still be analyzed rigorously but the required computations are quite lengthy. We will give an outline of the whole story, but we will not discuss every single step in detail. Once the basic idea is clear, the interested reader should be able to fill in missing details by studying the pointers to the literature.

The hardest problem is without doubt the random K-SAT problem. We will only be able to present a partial picture. Many interesting and very basic questions remain open.

Many people have helped us in creating these notes. In the Spring of 2011 we gave a series of lectures on these topics at EPFL to mostly a graduate student population. We would like to thank Marc Vuffray, Mahdi Jafari, Amin Karbasi, Masoud Alipour, Marc Desgroseilliers, Vahid Aref, Andrei Giurgiui, Amir Hesam Salavati for typing up initial notes for some lectures. In addition we would like to thank Mike Bardet who typed up further material as well as Hamed Hassani who has since contributed material to several of the chapters.

Nicolas Macris, Lausanne, 2013
Rüdiger Urbanke

Part I

Models and their Statistical Physics Formulations

1 Models and Questions: Coding, Compressive Sensing, and Satisfiability

We start by introducing three problems: error correcting *coding*, *compressive sensing*, as well as *constraint satisfaction*. Although these three problems are quite different, we will see that essentially the same tools from statistical physics can be used to gain insight into their behavior as well as to make quantitative predictions. These three problems will serve as our running examples.

TO COMPLETE

1.1 Coding

Error correcting codes

Codes are used in order to reliably transmit information across a noisy channel. Let us start with a basic definition. A *binary block code* $\mathcal{C}$ of length n is a collection of binary n-tuples, $\mathcal{C} = \{\underline{x}^{(1)}, \ldots, \underline{x}^{(\mathcal{M})}\}$, where $\underline{x}^{(i)}$, $1 \leq i \leq \mathcal{M}$, is called a codeword, and where the components of each codeword are elements of $\mathbb{F}_2 = (\{0,1\}, \oplus, \times)$, the binary field. The totale number of codewords is $|\mathcal{C}| = \mathcal{M}$ and the *rate* of the code is defined as $\frac{\log_2 |\mathcal{C}|}{n}$.

We will soon talk about various channel models, i.e., various mathematical models which describe how information is "perturbed" during the transmission process. In this respect it is good to know that for a large class of such models we can achieve optimal performance (in terms of the rate we can reliably transmit) by limiting ourselves to a simple class of codes, called linear codes.

A *linear binary block code* is a subspace of $\mathbb{F}_2^n$, the vector space of dimension n over the field $\mathbb{F}_2$. Equivalently, a binary block code $\mathcal{C}$ is linear iff for any two codewords $\underline{x}^{(i)}$ and $\underline{x}^{(j)}$, $\underline{x}^{(i)} - \underline{x}^{(j)} \in \mathcal{C}$. In particular $\underline{x}^{(i)} - \underline{x}^{(i)} = \mathbf{0} \in \mathcal{C}$. Since $\mathcal{C}$ is a subspace, it has a dimension, call it k, $0 \leq k \leq n$. Hence $|\mathcal{C}| = 2^k$, and the rate of $\mathcal{C}$ is equal to $\frac{k}{n}$.

All codes which we consider in this course are binary and linear. Therefore, in the sequel we sometimes omit these qualifiers. It will be convenient to represent a linear binary code $\mathcal{C}$ of length n and dimension k as the kernel (or null space) of an $(n-k) \times n$ binary matrix of rank $n-k$. Such a matrix is called a *parity-check* matrix and is usually denoted by H. Every binary linear code has such a

representation. So equivalently, we may write

$$\mathcal{C} = \left\{ \underline{x} \in \mathbb{F}_2^n : H\underline{x}^\top = \mathbf{0}^\top \right\}$$

for some suitably chosen matrix H. The proof that at least one such matrix exists is the topic of an exercise.

A few remarks might be in order. First, once we have convinced ourselves that there is at least one such matrix, it is easy to see that there are exponentially many (in $n-k$) such matrices since elementary row operations do not change the row space and hence the code defined by the matrix. All these matrices define the same code, and are equivalent in this sense. But the representation of the code in terms of a bipartite graph, which we will introduce shortly, and the related message-passing algorithm, do depend on the specific matrix we choose and so our choice of matrix is important.

Second, and somewhat connected to the first point, rather than first defining a code $\mathcal{C}$ and then finding a suitable parity-check matrix H, we typically specify directly the matrix H and hence indirectly the code $\mathcal{C}$.

It can then happen that this matrix does not have full row rank, i.e., that its rank is strictly less than $n-k$. What this means is that the code $\mathcal{C}$ contains more codewords than 2^k. Since this will happen rarely, and since having more codewords than planned is in fact a good thing, we will ignore this possibility and only count on having 2^k codewords at our disposal.

The factor graph associated to the parity-check matrix H (of a code $\mathcal{C}$)

Assume that we have a code $\mathcal{C}$ defined by the $(n-k) \times n$ binary parity-check matrix H. We can associate to H the following bipartite graph G. The graph G has vertices $V \cup C$, where $V = \{x_1, \ldots, x_n\}$ is the set of n *variable* nodes corresponding to the n bits (and hence to the n columns of H), and where $C = \{c_1, \ldots, c_{n-k}\}$ is the set of $n-k$ *check* nodes, each node corresponding to one row of H. There is an edge between x_i and c_j if and only if $H_{ji} = 1$.

EXAMPLE 1 (Factor Graph) Consider the following parity-check matrix,

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

The factor graph corresponding to H is shown in Fig. 1.1. □

Gallager's ensemble and the configuration model

A common theme in these notes is that instead of studying specific instances of a problem we define an *ensemble* of instances i.e., a set of instances endowed with a probability distribution. We then study the average behavior of this ensemble, and once the average is determined, we know that there must be at least one

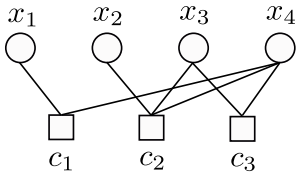


Figure 1.1 The factor graph corresponding to the parity-check matrix of Example 1.

element of the ensemble with a performance at least as good as this average. In fact, in many cases, with a little extra effort one can often show that most elements in the ensemble behave almost as good as the ensemble average.

For coding, we focus on a specific ensemble of codes called the (d_v, d_c)-regular *Gallager* ensemble introduced by Gallager in 1961, [3, 4]. Rather than specifying the codes directly we specify their factor graphs. The ensemble is characterized by the triple of integers (n, d_v, d_c), such that $m = n\frac{d_v}{d_c}$ is also an integer. The parameter n is the length of the code, d_v is the variable node degree, and d_c is the check node degree.

To precisely describe the ensemble we explain how to sample from it. Pick n variable nodes and $n\frac{d_v}{d_c}$ check nodes. Each variable node has d_v *sockets* and each check node has r *sockets*. Number the $d_v n$ variable sockets in an arbitrary but fixed way from 1 till $d_v n$. Do the same with the $d_v n$ check node sockets. Pick a permutation π uniformly at random from the set of permutations on $d_v n$ letters. For $s \in \{1, \dots d_v n\}$ insert an edge which connects variable node socket s to check node socket $\pi(s) \in \{1, \dots d_v n\}$.

If, after construction, we delete sockets (and retain the connections between variable and check nodes) then we get a bipartite graph which is the factor graph representing our code. To this bipartite graph we can of course associate a parity-check matrix H. But note that in this model there can be multiple edges between nodes. A moments thought shows that the parity-check matrix H has a 1 at row i and column j if there are an odd number of connections between variable i and constraint j. Otherwise it has a 0 at this position. In practice multiple connections are not desirable and more sophisticated graph generation algorithms are employed. But for our purpose the typically small number of multiple connections will not play a role. In particular, it does not play a role if we are interested in the behavior of such codes for very large instances.

The above way of specifying the ensemble is inspired by the configuration model of random graphs, see [5]. This is why we call it the *configuration* model. This particular ensemble is a special case of what is called a *low-density parity-check* (LDPC) ensemble. This name is easily explained. The ensemble is *low-density* since the number of edges grows linearly in the block length. This is distinct from what is typically called the Fano random ensemble where each entry of the parity-check matrix is chosen uniformly at random from $\{0, 1\}$, so that the number of edges grows like the square of the block length. It is further

a parity-check ensemble since it is defined by describing the parity-check matrix. We will see that a reasonable decoding algorithm consists of sending messages along the edges of the graph. So few edges means low complexity and, even more importantly, we will see that the algorithm works better if the graph is *sparse*.

For many real systems, LDPC codes are the codes of choice. They have a very good trade-off between complexity and performance and they are well suited for implementations. "Real" LDPC codes are often further optimized. For example, instead of using regular degrees we might want to choose nodes of different degrees and the connections are often chosen with care in order to minimize complexity and to maximize performance. We will ignore these refinements in the sequel. The most important trade-offs are already apparent for the relatively simple regular Gallager ensemble.

Encoding, Transmission, and Decoding

The three operations involved in the coding problem are *encoding*, *transmission over a channel*, and *decoding*. Let us briefly discuss each of them.

Encoding: Given $\mathcal{C}$, a binary linear block code of dimension k, we can *encode* k bits of information by our choice of codeword, i.e., by choosing one out of the 2^k possibilities. More precisely, we have an information word $\underline{u}$, $\underline{u} \in \mathbb{F}_2^k$, and an encoding function g, $g : \mathbb{F}_2^k \to \mathcal{C}$, which maps each information word into a codeword.

Although this function is of crucial importance for real systems, it only plays a minor role for our purpose. This is true since, as we will discuss in more detail later on, for "typical" channels, by symmetry the performance of the system is independent of the transmitted codeword. We therefore typically assume that the all-zero codeword (which is always contained in a binary linear code) was transmitted. Also, in terms of complexity, the encoding operation is not a difficult task. One possible option is to write the linear binary code $\mathcal{C}$ in the form $\mathcal{C} = \{G\underline{u} : \underline{u} \in \mathbb{F}_2^k\}$, where G is the so-called *generator* matrix and where $\underline{u}$ is a binary column vector of length k which contains the information bits. In this form, encoding corresponds to a multiplication of a vector of length k with a $n \times k$ binary matrix and can hence be implemented in $O(k \times n)$ binary operations. In practice the code is often chosen to have some additional structure so that this operation can even be performed in $O(n)$ operations. We will hence ignore the issue of encoding in the sequel.

Transmission over a Channel: We assume that we pick a codeword $\underline{x}$ uniformly at random from the code $\mathcal{C}$. We now *transmit* $\underline{x}$ over a "*channel*". The actual channel is a physical device which takes bits as inputs, converts them into a physical quantity, such as an electric or optical signal, transmits this signal over a suitable medium, such as a cable or optical fiber, and then converts the physical signal back into a number which we can processed, perhaps equal to a voltage

which is measured or the number of photons which were detected. Of course, during the transmission the signal itself is distorted. This distortion is either due to imperfections of the system or due to unpredictable processes such as thermal noise. Instead of considering this potentially very complicated process we use a typically simple mathematical model which describes the end-to-end effect of all these physical processes on the signal. We call this model the "channel model."

Channel Model: Formally, the channel has the input alphabet $\mathcal{X} = \{0, 1\}$ and an output alphabet $\mathcal{Y}$. E.g., two common cases are $\mathcal{Y} = \{0, 1\}$ and $\mathcal{Y} = \mathbb{R}$. We assume that the channel is *memoryless*, which means that it acts on each bit independently. We further assume that there is no *feedback* from the output of the channel back to the input. In this case the channel is uniquely characterized by a transition probability $p(\underline{y} \mid \underline{x})$ where $\underline{y} \in \mathcal{Y}^n$ is the output and where

$$p(\underline{y} \mid \underline{x}) = \prod_{i=1}^{n} p(y_i \mid x_i). \tag{1.1}$$

Note that we get this product form from the assumptions that the channel is memoryless (acts bit-wise) and that we have no feedback.

The following three channels are the most important examples, both from a theoretical perspective, but also because they form the basis of real-world channels: These are the *binary erasure channel* (BEC), the *binary symmetric* channel (BSC) and the *binary additive white Gaussian noise channel* (BAWGNC).

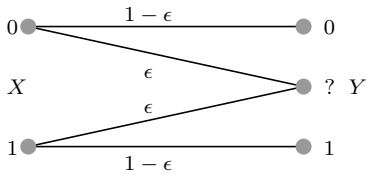


Figure 1.2 Binary erasure and symmteric channels with parameter ϵ.

BEC. The BEC is a very special channel with $\mathcal{Y} = \{0, ?, 1\}$. As depicted in Fig. 1.2, the transmitted bit is either correctly received at the channel output with probability $1 - \epsilon$ or erased by the channel with probability ϵ and thus, nothing is received at the channel output. The erased bits are denoted by "?". For example, if $x = 1$ is transmitted in the BEC, then the set of possible channel observation is $\{1, ?\}$. we may write somewhat formally for the transition probability $p(y|x) = (1 - \epsilon)\delta(y - x) + \epsilon\delta(y - ?)$.

BSC. The output of the BESC is binary $\mathcal{Y} = \{0, 1\}$. As seen on Fig. 1.2 the bit is transmitted correctly with probability $1 - \epsilon$ or flipped with probability ϵ. The transition probability is $p(y|x) = (1 - \epsilon)\delta(y - x) + \epsilon\delta(y - (1 - x))$.

BAWGNC. The output is a real number $\mathcal{Y} = \mathbb{R}$. When $x \in \{0, 1\}$ is sent the received signal is $y = x + z$ with z a Gaussian random number with zero mean and variance σ^2. With these conventions the "signal to noise ratio" is σ^{-2} and the transition probability $p(y|x) = (\sqrt{2\pi}\sigma)^{-1} e^{-\frac{(y-x)^2}{2\sigma^2}}$.

One might wonder if these three simple models even scratch the surface of the rich class of channels that one would assume we encounter in practice. Fortunately, the answer is *yes*. The branch of *communications theory* has built up a rich theory of how more complicated scenarios can be dealt with assuming that we know how to deal with these three simple models.

Decoding: Given the output $\underline{y}$ we want to map it back to a codeword $\underline{x}$. Let $\hat{x}(\underline{y})$ denote the function which corresponds to this *decoding* operation. What decoding function shall we use? One option is to first pick a suitable criterion by which we can measure the performance of a particular decoding function and then to find decoding functions which optimize this criterion. The most common such criteria are the *block error probability* $\mathbb{P}[\hat{\underline{x}}(y) \neq \underline{x}]$, and the *bit error probability* $\frac{1}{n}\sum_{i=1}^{n} \mathbb{P}[\hat{\underline{x}}(y)_i \neq x_i]$. We will come back in Chapter 3 to the precise definition of these error probabilities.

In practice, due to complexity constraints, it is typically not possible to implement an optimal decoding function but we have to be content with a low-complexity alternative. Of course, the closer we can pick it to optimal the better.

Shannon Capacity

So far we have defined codes, we have discussed the encoding problem, the process of transmission, the decoding problem, and the two most standard criteria to judge the performance of a particular decoder, namely the block and the bit error probability.

It is now natural to ask what is the maximum rate at which we can hope to transmit reliably, assuming that we pick the best possible codes and the best possible decoder. Reliably here means that we can make the block or bit probability of error as small as we desire. In fact, it turns out that the answer is the same wheter we use the block error probability or the bit error probability.

In 1948 Shannon gave the answer and he called this maximum rate the *capacity* of the channel. For binary-input memoryless output-symmetric channels the capacity has a very simple form. If the input alpahbet is binary and the output alphabet discrete, and if $p(y \mid x)$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, denotes the transition probabilities, then the capacity of the associated channel can be expressed (in bits per channel use) as

$$H(p(\cdot)) - H(p(\cdot \mid x = 0)) \tag{1.2}$$

where $H(q(\cdot))$ denotes the entropy associated to a discrete distribution $q(y)$, $y \in \mathcal{Y}$. By definition we have

$$H(q(\cdot)) = -\sum_{y\in\mathcal{Y}} q(y)\log_2 q(y). \tag{1.3}$$

Let us illustrate Shannon's formula for the BEC(ϵ). For $q(y) = p(y \mid x = 0)$ we have $q(0) = p(y = 0 \mid x = 0) = 1-\epsilon$, $q(1) = p(y = 1 \mid x = 0) = 0$, and $q(?) = p(y = ? \mid x = 0) = \epsilon$. Further, for $q(y) = p(y) = \frac{1}{2}p(y \mid x = 0) + \frac{1}{2}p(y \mid x = 1)$ we have $p(0) = p(1) = \frac{1}{2}(1-\epsilon)$ and $p(?) = \epsilon$. Hence, $H(p(\cdot)) = 1-\epsilon+h_2(\epsilon)$ and $H(p(\cdot \mid x = 0)) = h_2(\epsilon)$, where $h_2(\epsilon) = -\epsilon\log_2\epsilon - (1-\epsilon)\log_2(1-\epsilon)$ is the so called binary entropy function. We conclude that the capacity of the BEC(ϵ) is equal to $1-\epsilon$. That the capacity is at most $1-\epsilon$ for the BEC is intuitive. For large blocklengths with high probability the fraction of non-erased positions is very close to $1-\epsilon$. So even if we knew a priori which positions will be erased and which will be left untouched, we could not hope to transmit more than $n(1-\epsilon)$ bits over such a channel. What is perhaps a little bit surprising is that this quantity is achievable, i.e., that we do not need to know a priori what positions will be erased and still can transmit reliably at this rate.

The capacities of the BSC and BAWGNC are computed similarly (see exercises).

Questions

Now where we know the basic problem and have discussed the ultimate limit of what we can hope to achieve, the following questions seem natural to investigate.

- What are good and efficient decoding algorithms?
- If we pick a random such code from the ensemble, how well will it perform?
- In particular, is there going to be a threshold behavior so that for large instances the code *works* up to some noise level but *breaks down* above this level as it is indicated schematically in Fig. 1.3? How does this threshold depend on the decoding algorithm?
- Assuming that there is a threshold behavior, how can we compute the thresholds?
- How do these thresholds compare to the Shannon threshold?

We will be able to derive a fairly complete set of answers to all of the above questions.

1.2 Compressive sensing

Basic problem

Here is the perhaps the simplest version of compressive sensing. Let $\underline{x}^{\text{in}} \in \mathbb{R}^n$ representing an "input signal" that we want to capture. We assume that the

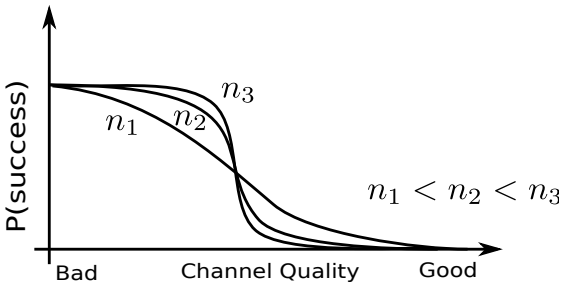


Figure 1.3 The probability of decoding error for a transmitted message versus the channel quality. As the blocklength of the code gets larger, we expect to see a sharper and sharper transition between range of the channel parameters where the system "works" and where it "breaks down."

number of non-zero components $\|\underline{x}^{\text{in}}\|_0 = |\{i|x_i^{\text{in}} \neq 0, i = 1, \ldots, n\}| = k$ of the signal is only a fraction of n; so $k = \kappa n$ with $\kappa < 1$ (and usually much smaller than one). The signal is captured thanks to an $m \times n$ "measurement matrix" A with real entries, $1 \leq m < n$. We set $m = \mu n$ with $\mu < 1$. Let $\underline{y} \in \mathbb{R}^m$ be given by $\underline{y} = A\underline{x}^{\text{in}}$. We think of $\underline{y}$ as the result of m linear measurements, one corresponding to each row of A. Our basic aim is to reconstruct the k-sparse signal $\underline{x}^{\text{in}}$ from the least possible measurements $\underline{y}$.

We know that at least one solution exists, namely $\underline{x}^{\text{in}}$, because the measurements $\underline{y}$ have been produced by this input signal. But since $m < n$, and in fact m is typically *much smaller*, we cannot simply solve the undetermined linear system of equations since the solution will not be unique. But we know in addition that $\underline{x}$ is k-sparse, i.e. has only k non-zero entries entries with $k < n$, (but we do not know which of these entries are non-zero). Therefore, we determine if the set of possible signals, namely

$$\{\underline{x} : A\underline{x} = \underline{y} \text{ and } \|\underline{x}\|_0 = k\}. \tag{1.4}$$

has cardinality one. If this is the case we may in principle be able to reconstruct our signal unambiguously.

One way to ensure the unicity of the solution is to take a measurement matrix A satisfying a *Restricted Isometry Property*. We say that A satisfies the RIP$(2k, \delta)$ condition if one can find $0 \leq \delta < 1$ such that

$$(1 - \delta)\|\underline{x}\|_2 \leq \|A\underline{x}\|_2 \leq (1 + \delta)\|\underline{x}\|_2, \quad \text{for all } 2k\text{-sparse vectors } \underline{x} \in \mathbb{R}^n. \tag{1.5}$$

It is not difficult to see that when this condition is met, then (1.4) has a *unique* solution given by

$$\hat{\underline{x}}_0(y) = \text{argmin}_{\underline{x}:A\underline{x}=y}\|\underline{x}\|_0. \tag{1.6}$$

Indeed, first notice that evidently $A\hat{\underline{x}}_0(y) = y$ so we only have to prove unicity. Suppose $\underline{x}'$ is another solution of (1.4). Then, since both $\underline{x}'$ and $\underline{x}_0(y)$ are k-sparse, their difference is $2k$-sparse. The left hand inequality of the RIP$(2k, \delta)$

condition states $(1-\delta)\|\underline{x}' - \hat{\underline{x}}_0(y)\|_2 \leq \|A\underline{x}' - A\hat{\underline{x}}_0(y)\|_2 = \|\underline{y} - \underline{y}\|_2 = 0$, which of course implies $\underline{x}' = \hat{\underline{x}}_0(y)$.

Solving the optimization problem (1.6) essentially requires an exhaustive search over $\binom{n}{k}$ possible supports of the sparse vectors, which is intractable in practice. One avenue for simplifying this problem is to replace the "ℓ_0 norm" in (1.6) with the ℓ_1 norm. In other words we solve the convex optimization problem,

$$\hat{\underline{x}}_1(y) = \text{argmin}_{\underline{x}:A\underline{x}=y}\|\underline{x}\|_1. \tag{1.7}$$

A fundamental theorem of Candes and Tao states that one can find δ', $0 < \delta' < \delta$, such that if A satisfies RIP$(2k, \delta')$ the solution of this problem is unique and identical to (1.6), [**?**].

This result shows that, for suitable measurement matrices, the ℓ_0 and ℓ_1 optimization problems are equivalent. Thus it suffices to solve the ℓ_1 problem. We will not prove it here but only offer some intuition for it through a simple toy example. Suppose that $n = 2$, so $\underline{x} = (x_1, x_2)^T$, and that we perform a single measurement $y = a_1x_1 + a_2x_2$. This equation corresponds to the line on figure

FIGURE

Figure 1.4 The ℓ_p balls

1.4. We seek to find a point on this line, which minimizes $(x_1^p + x_2^p)^{1/p}$, $p \geq 0$ where the case $p = 0$ is to be understood as the number of non-zero components of (x_1, x_2). As shown on figure 1.4 the solution is found by "inflating" the "ℓ_p-balls" around the origin until the line is touched. It is clear that for a generic line the solution is the same for all $0 \leq p \leq 1$. Note also that for $0 \leq p \leq 1$ the solution only has a single non-zero component, so is "sparse". For $p > 1$ the solution changes with p and both components are non-zero. Note when $p = 1$ there are non-generic measurement matrices corresponding to lines parallel to the faces of the ℓ_1-ball for which the solution is not unique; but as discussed shortly such cases will not bother us because the matrices will be chosen at random.

But what matrices satisfy the RIP condition ? It should come as no surprise that a matrix satisfying teh RIP condition should have a number of lines m at least as large as k. In fact one can show that necessarily $m \geq C_\delta k \log \frac{n}{k}$ for a suitable constant $C_\delta > 0$ [**?**]. It is not easy to make deterministic constructions of "good" measurement matrices approaching such bounds. The same is true with other deterministic conditions yielding equivalence of the ℓ_0 and ℓ_1 optimization

problems. However the toy example suggests that in fact all we migth need are "random measurement matrix". This is indeed a fruitful idea, at least in the asymptotic setting $n, m \to +\infty$ with $\kappa = \frac{k}{n}, \mu = \frac{m}{n}$ fixed, very much in the spirit of random coding. This is the route we will follow.

Ensembles of Measurement Matrices

While deterministic constructions of matrices satisfying the RIP condition are difficult, they can be shown to exist thanks to the probabilistic method [**?**]. The $m \times n$ matrix A will be taken from *the Gaussian ensemble* where the matrix entries are independent identically distributed Gaussian variables of zero mean and variance $1/m$. This normalization is such that each column of A has an expected ℓ_2 norm of 1. As in coding we will consider the assymptotic regime $n, m, k \to +\infty$ with *sparsity parameter* $\kappa = \frac{k}{n}$ and *measurement fraction* $\mu = \frac{m}{n}$ fixed. One can then show that there exists positive numerical constants c_1, c_2 such that for $m \geq c_1 \delta^{-2} k \log(\frac{en}{k})$ matrices from this ensemble satisfy the RIP(k, δ) condition with overwhelming probability $1 - \exp(-c_2 \delta^2 m)$ where the constants c_1, c_2 are numerical constants. More general ensembles are also possible.

The ensemble formulation for the measurement matrices, may also be extended to the signal model. One of the simplest signal distributions assumes that the components x_i are independently identically distributed according to a law of the form

$$p_0(x) = (1 - \kappa)\delta(x) + \kappa \phi_0(x) \tag{1.8}$$

where $\phi_0(x)$ is a continuous probability density. Depending on the model or the application $\phi_0(x)$ is known or unknown. The most realistic assumption for applications is to consider that $\phi_0(x)$ is unknown, and in that case we call $\mathcal{S}_\kappa$ this class of signals.

Noisy measurements and LASSO

A somewhat more realistic version of the measurement model is

$$\underline{y} = A\underline{x} + \underline{z},$$

where $\underline{z}$ is a noise vector, typically assumed to consist of m iid zero-mean Gaussian random variables with variance of σ^2. Again our aim is to reconstruct an k-sparse signal with as few measurements as possible. The matrix A is chosen from the random Gaussian ensemble and the signal from the class $\mathcal{F}_\kappa$.

If we ignored the sparsity constraint then it would be natural to pick the estimate $\hat{\underline{x}}(\underline{y})$ which solves the least-squares problem $\min_{\underline{x}} \|A\underline{x} - \underline{y}\|_2^2$. This problem is easily solved and the solution is well known $\hat{\underline{x}}(\underline{y}) = (A^T A)^{-1} A^T \underline{y}$. But in general this solution will not be k-sparse.

To enforce the sparsity constraint, we can add a second term to our objective

function, i.e., we can solve the following minimization problem,

$$\underline{\hat{x}}_0(\underline{y}) = \text{argmin}_{\underline{x}}(\|A\underline{x} - \underline{y}\|_2^2 + \lambda\|\underline{x}\|_0), \tag{1.9}$$

for a properly tuned parameter λ. Unfortunately this minimization problem is intractable, again because it requires an exhaustive search over the $\binom{n}{k}$ possible supports of the sparse vectors.

We saw in the noiseless case that replacing the "ℓ_0 norm" by the ℓ_1 norm is a fruitful idea. We follow the same route here and consider the following minimization problem

$$\underline{\hat{x}}_1(\underline{y}) = \text{argmin}_{\underline{x}}(\|A\underline{x} - \underline{y}\|_2^2 + \lambda\|\underline{x}\|_1). \tag{1.10}$$

This estimator is called the *Least absolute Shrinkage and Selectio Operator* (LASSO). Again λ has to be chosen appropriately. This estimator can in principle be calculated by standard convex optimizaton techniques, which is already a big improvement over exhaustive search.

Although the LASSO estimator is popular, its a priori justification is not so straightforward. Our discussion suggests that in the noiseless limit it reduces to the pure ℓ_1 estimator which we know gives for a certain range of parameters the correct solution of the ℓ_0 problem. This is one possible justification. Interestingly, the analysis of the LASSO in Chapter **??** the exact frontier for the ℓ_0-ℓ_1 equivalence in the (κ, μ) plane. This frontier is known as the Donoho-Tanner curve which they originally derived by completely different methods. In Chapter 3 we also discuss a somewhat more Bayesian justification of the LASSO in a setting where the signal distribution is not known, but only the parameter κ is assumed to be known. All this is ample justification for studying the LASSO in detail.

Graphical representation

As for coding one can set up a graphical representation for the measurement matrix. We associate to A a bipartite graph G with vertices $V \cup C$, where $V = \{x_1, \ldots, x_n\}$ is the set of *variable* nodes corresponding to the n signal components and $C = \{c_1, \ldots, c_m\}$ is the set of *check* nodes each node corresponding to a row (a measurement) of A. There is an edge between x_i and c_j if an only if $A_{ji} \neq 0$. For the random measurement matrices discussed above this will essentially always be the case and therefore the graph is simply the *complete bipartite* graph depicted on figure 1.5.

If one wishes one may atribute a "random weight" to the edges, but we will seldom need to do so. Therefore, unlike coding, here the graph is always the same. At this point this graphical construction may seem slightly trivial and arbitrary, but it will turn out to be a very useful way of thinking. The reason is that, much as in coding theory, we will develop iterative algorithms exchanging messages along the edges in order to reconstruct the signal. For example, this immediately suggests that the complexity of these algorithms scales like $O(n^2)$

FIGURE

Figure 1.5 The factor graph corresponding to the random gaussian 2×4 measurement matrix

because there are $nm = n^2\mu$ edges. Nevertheless each edge has a random weight of order $\pm 1/\sqrt{n}$ and this will allow us to reduce the complexity to $O(n)$.

Questions

Consider the regime where n tends to infinity and $\kappa = k/n$, $\mu = m/n$ constant.

- For given κ what fraction μ of measurements do we need so that with high probability we can recover $\underline{x}^{in}$ from the measurement $\underline{y}$ if we have no limitations on complexity?
- If we restrict ourselves to the low-complexity LASSO algorithm, how many measurements do we need then?
- Are there ways of designing compressive sensing schemes which achieve the theoretical limits under low-complexity algorithms?

1.3 Satisfiability

SAT problem

Suppose that we are given a set of n Boolean variables $\{x_1, \ldots, x_n\}$. Each variable x_i can take on the values 0 and 1, where 0 means "false" and 1 means "true". We define a *literal* to be either a variable x_i or its negation $\bar{x}_i$. A *clause* is a disjunction of literals, e.g.,

$$c = x_1 \vee x_2 \vee \bar{x}_3$$

where the operation "$\vee$" denotes the Boolean "or" operation. An *assignment* is an assignment of values to the Boolean variables, e.g., $x_1 = 0$, $x_2 = 1$, and $x_3 = 0$. Such an assignment will either make a clause to be *satisfied* or *not satisfied*. For example the clause $x_1 \vee x_2 \vee \bar{x}_3$ with assignment $x_1 = 0$, $x_2 = 1$, and $x_3 = 0$ evaluates to 1, i.e., the clause is satisfied. A SAT formula, call it F, is a conjunction of a set of clauses. For example, consider the SAT formula

$$F = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_2 \vee \bar{x}_4) \wedge x_3.$$

where "$\wedge$" is the Boolean "and" operation.

The basic SAT problem is defined as follows. Given a SAT formula F, determine the satisfiability of F, i.e., determine if there exists an assignment on $\{x_1, \ldots, x_n\}$ so that F is satisfied. This is the SAT *decision* problem. If such an assignment exists we might also want to find an explicit solution.

Why on earth would anyone be interested in studying this question? Perhaps surprisingly, many real-world problems map naturally into a SAT problem. For example designing circuits, optimizing compilers, verifying programs, or scheduling can be phrased in this way. The bad news is that Cook proved in 1973 that it is unlikely that there exists an algorithm which solves all instances of this problem in polynomial time (in n). More precisely, the SAT decision problem is NP-complete.

We say that a formula F is a K-SAT formula if every clause involves exactly K literals. E.g., $(x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_2 \vee x_3 \vee \bar{x}_4)$ is a 3-SAT formula. The following facts are known. The 2-SAT decision problem is easily solved in a polynomial number of steps. Problem 1.6 discusses a simple algorithm called unit-clause propagation which solves a 2-SAT decision problem in at most $2n$ steps and produces a satisfying assignment if one exists. On the other hand for $K \geq 3$ the K-SAT decision problem is NP-complete.

Graphical representation of SAT formulas

Given a SAT formula F, we associate to it a bipartite graph G. The vertices of the graph are $V \cup C$, where $V = \{x_1, \ldots, x_n\}$ are the Boolean variables and $C = \{c_1, \ldots, c_m\}$ are the m clauses. There is an edge between x_i and c_j if and only if x_i or $\bar{x}_i$ is contained in the clause c_j. Further we draw a "solid line" if c_j contains x_i and a "dashed line" if c_j contains $\bar{x}_i$.

EXAMPLE 2 (Factor Graph of SAT Formula) As an example, the graphical presentation of $F = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_2 \vee x_3 \vee \bar{x}_4)$ is shown in Fig. 1.6. □

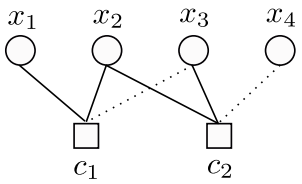


Figure 1.6 The factor graph corresponding to the SAT formula of Example 2.

Ensemble of random K-SAT Formulas

Just like in the coding and compressed sensing problems, rather than looking at individual SAT formulas, we will define an *ensemble* of such formulas and we will then study the probability that a formula from this ensemble is satisfiable. In particular, we will stick to the behavior of random K-SAT formulas.

The ensemble $\mathcal{F}(n, m, K)$ is characterized by 3 parameters: K is the number of literals per clause, n is the number of Boolean variables, and m is the number of clauses. Notice that with K variables we can form $\binom{n}{K}2^K$ clauses by taking K variables among $x_1, \ldots, x_n$ and then negating them or not. We define $\mathcal{F}(n, m, K)$ by showing how to sample from it. To this end, pick m clauses $c_1, \ldots, c_m$ independently, where each clause is chosen uniformly at random from the $\binom{n}{K}2^K$ possible clauses. Then form F as the conjunction of these m clauses. In other words, the ensemble $\mathcal{F}(n, m, K)$ is the uniform probability distribution over the set of all possible formulas F constructed out of n Boolean variables by choosing m clauses. The cardinality of this set is $\binom{\binom{n}{K}2^K}{m}$.

Threshold behavior

Now let us consider the following experiment. Fix $K \geq 2$ (e.g., $K = 3$) and draw a formula F from the $\mathcal{F}(n, m, K)$ ensemble. Is such a formula satisfiable with high probability? It turns out that the most important parameter that affects the answer is $\alpha = \frac{m}{n}$. This ratio is called the *clause density*. Like in coding and compressed sensing we are interested in the asymptotic regime where $n, m \to +\infty$ and α is fixed.

Fig. 1.7 shows the probability of satisfiability of F as a function of both n and α. As we see from this figure, as n becomes larger the transition of the probability of satisfiability becomes sharper and sharper. This is a strong indication that there exists a threshold behavior, i.e., there exists a real number $\alpha_s(K)$ such that

$$\lim_{n\to\infty} \mathbb{P}\left[F \text{ is satisfied}\right] = \begin{cases} 1, & \alpha < \alpha_s(K), \\ 0, & \alpha > \alpha_s(K). \end{cases} \tag{1.11}$$

Here $\mathbb{P}\left[-\right]$ is the uniform probability distribution of the ensemble $\mathcal{F}(n, m, K)$.

As the density α increases one has more and more clauses to satisfy, so it intuitively quite clear that the probability of satisfaction decreases as a function of α. However the existence of a sharp threshold is much less evident, let alone its computation. Such a threshold behavior was conjectured nearly two decades ago based on experiments []. For many years this was proved only for $K = 2$ for which $\alpha_s(2) = 1$. For $K \geq 3$ Friedgut proved that there exists a sequence $\alpha_s(K, n)$, $n \in \mathbb{N}$, such that for all $\epsilon > 0$

$$\lim_{n\to\infty} \mathbb{P}\left[F \text{ is satisfied}\right] = \begin{cases} 1, & \alpha < (1-\epsilon)\alpha_s(K, n), \\ 0, & \alpha > (1+\epsilon)\alpha_s(K, n). \end{cases} \tag{1.12}$$

This result leaves open the possibility that the sequence of thresholds $\alpha_s(K, n)$ does not converge to a definite value as $n \to +\infty$. The proof of a sharp threshold behavior (1.11) was proved recently in [] for K large enough (but finite), but for small K's (except $K = 2$) a proof is still a challenging problem.

The underpinnings of this proof for large K's rest on the statistical mechanics methods which also give the means to compute $\alpha_s(K)$ (for example it is known

that $\alpha_s(3) \approx 4.259$ to three decimal places). As we will see these methods yield much more information than just the threshold value. We will uncover various other threshold behaviors, related not only to the satisfiability of random formulas, but also to the nature of the solution space. Understanding the nature of these threshold behaviors in K-SAT is an order of magnitude more difficult than in coding theory and compressed sensing, and forms part of the more advanced material in chapters 15, 16.

Random max-K-SAT

In the K-SAT decision problem, one is given a formula and is asked to determine if this formula is satisfiable or not. An important variation on this theme is the *max-K-SAT* problem. In this problem one is interested in determining the *maximum possible number of satisfied clauses* where the maximum is taken over all possible 2^n assignments of variables $x_1, \ldots, x_n \in \{0,1\}^n$. Of course it is equivalent to determine the *minimum possible number of violated clauses* where the minimum is taken over all assignments of variables. In later chapters we will adopt this perspective which makes the contact with traditional statistical mechanics questions clearer.

We will be interested in the random version of max-K-SAT which we know formulate more precisely. Take a formula at random from the ensemble $\mathcal{F}(n, m, K)$. This formula contains m clauses labelled $c_1, \ldots, c_m$. If we let $\mathbb{1}_c(\underline{x})$ be the indicator function over assignments that satisfy clause c (i.e the function evaluates to 1 if $\underline{x}$ satisfies c and 0 if $\underline{x}$ does not satisfy c) then the maximum possible number of satisfied clauses is

$$\max_{\underline{x}} \sum_{i=1}^{m} \mathbb{1}_{c_i}(\underline{x})$$

In the random max-K-SAT problem we want to compute

$$\lim_{m\to+\infty} \frac{1}{m} \mathbb{E}\big[\max_{\underline{x}} \sum_{i=1}^{m} \mathbb{1}_{c_i}(\underline{x})\big] \tag{1.13}$$

where the expectation is taken over the ensemble $\mathcal{F}(n, m, K)$ (the existence of teh limit has been proven by methods that we will study in Chapter **??**). Equivalently we want to compute the average of the minimum possible number of violated clauses

$$e(\alpha) \equiv \lim_{m\to+\infty} \frac{1}{m} \mathbb{E}\big[\min_{\underline{x}} \sum_{i=1}^{m} (1 - \mathbb{1}_{c_i}(\underline{x}))\big] \tag{1.14}$$

We define the *max-K-sat threshold* as

$$\alpha_{s,\max}(K) = \sup\{\alpha | e(\alpha) = 0\} \tag{1.15}$$

We will give a non-rigorous computation of (1.14) and (1.15) in chapters 15,

16. In fact, the proof methods [] for the sharp threshold behavior (1.11) have their origin in such statistical mechanics computations.

Intuitively one expects that $\alpha_{s,\max}(K) = \alpha_s(K)$. It is clear that one must have $\alpha_s(K) \leq \alpha_{s,\max}(K)$. However the converse bound is not immediate because one could conceivably have a finite interval $]\alpha_s(K), \alpha_{s,\max}(K)[$ where $e(\alpha) = 0$ but at the same time a sublinear fraction of unsatisfied clauses. Nevertheless it is widely believed this does not happen and that $\alpha_s(K) = \alpha_{s,\max}(K)$. At least we know that this is true for $K = 2$ and for large enough (finite) K [] .

Questions

Here is a set of questions we are interested in:

- Does this problem exhibit a threshold behavior?
- If so, can we determine this threshold α_K?
- Are there low-complexity algorithms which are capable of finding satisfying assigments, assuming such assignments exist?
- If so, up to what clause density do they work with high probability?

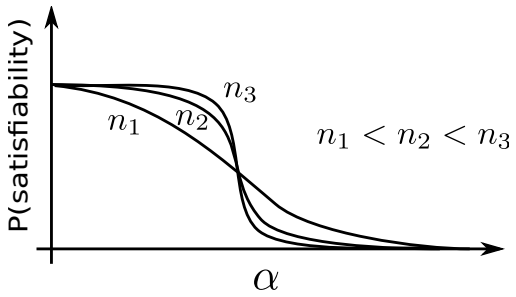


Figure 1.7 The probability that a formula generated from the random K-SAT ensemble is satisfied versus the clause density α.

Perhaps surprisingly, many of the above questions do not yet have a rigorous answer and the satisfiability problem is by far the hardest of our three examples. Nevertheless we will have non-trivial things to say about this problem and if one admits non-rigorous methods, the problem is fairly well understood.

1.4 Overview of coming attractions

TO DO

1.5 Notes

Here we should put some further historical info as well as reference to the literature.

Problems

1.1 Capacity of the BSC and BAWGNC. Apply formula (1.2) to compute the Shannon capacity of the two channels.

1.2 Configuration Model. The aim of this problem is to write a program that can sample a random graph from the configuration model. Your program should take as input the parameters n, m, d_v, and d_c, it should then check that the input is valid, and finally return a bipartite graph according to the configuration model. Think about the data structure. If we run algorithms on such a graph it is necessary to loop over all nodes, refer to edges of each node, be able to address the neighbor of a node via a particular edge and store values associated to nodes and edges.

1.3 Norms and pseudo-norms. Let $\|\underline{x}\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$ for $p > 0$. Let also $\|\underline{x}\|_0 = \sharp(\text{non zero } x_1, \ldots, x_n)$ and $\|\underline{x}\|_\infty = \max_i |x_i|$. Show first that $\|\underline{x}\|_0 = \lim_{p\to 0} \|\underline{x}\|_p$ and $\|\underline{x}\|_\infty = \lim_{p\to+\infty} \|\underline{x}\|_p$. Explain why $\|\cdot\|_p$ is a norm for $1 \le p \le +\infty$ and is *not* a norm for $0 \le p < 1$ (this is why for $0 \le p < 1$ we call it a pseudo-norm). *Hint*: refer to the figure 1.4.

1.4 Least square estimator. Show that the minimizer of $\|\underline{y} - A\underline{x}\|_2^2$ is the least square estimator $\underline{\hat{x}}(\underline{y}) = (A^T A)^{-1} A^T \underline{y}$.

1.5 Poisson Model. An important model of bipartite random graphs is the *Poisson model*. For example the random K-SAT problem is often formulated on this graph ensemble. Pick two integers, n and m. As before, there are n variable nodes and m check nodes. Further, let K be the degree of a check node. For each check node pick K variables uniformly at random either with or without repetition and connect this check node to these variable nodes. For each edge store in addition a binary value chosen according to a Bernoulli(1/2) random variable.

This is called the Poisson model because the node degree distribution on the variable nodes converges to a Poisson distribution for large n. This is also the case for the formulation in 1.3. The two formulation are equivalent in the asymptotic limit.

Write a program that takes n, m, K as input parameters and outputs a graph instance from the Poisson model. Again, think of the data structure.

1.6 Unit Clause Propagation for Random 3-SAT Instances. The aim of this problem is to test a simple algorithm for soving SAT instances. Generate

random instances of the Poisson model. Pick $n = 10^5$ and let $K = 3$. Let α be a non-negative real number. It will be somewhere in the range $[0, 5]$. Let $m = \lfloor \alpha n \rfloor$. For a given α generate many random bipartite graphs according to the Poisson model. Interpret such bipartite graphs as random instances of a 3-SAT problem. This means, the variables nodes are the Boolean variables and the check nodes represent each a clause involving 3 variables. Associate to each edge a Boolean variable indicating whether in this clause we have the variable itself or its negation.

For each instance you generate, try to find a satisfying assignment in the following greedy manner. This is called the *unit clause propagation* algorithm:

(i) If there is a check node in the graph of degree one (this corresponds to a *unit*-clause), then choose one among such check nodes uniformly at random. Set the variable to satisfy it. Remove the clause from the graph together with the connected variable and remove or shorten other clauses connected to this variable (if the variable satisfies other clauses they are removed while if not they are shortened).
(ii) If no such check exists, pick a variable node uniformly at random from the graph and sample a Bernoulli(1/2) random variable, call it X. Remove this variable node from the graph. For each edge emanating from the variable node do the following. If X agrees with the variable associated to this edge then remove not only the edge but the associated check node and all its outgoing edges. If not, then remove only the edge.

Continue the above procedure until there are no variable nodes left. If, at the end of the procedure, there are no check nodes left in the graph (by definition all variable nodes are gone) then we have found a satisfying assignment and we declare success. If not, then the algorithm failed, although the instance itself might very well be satisfiable.

Plot the empirical probability of success for this algorithm as a function of α. You should observe a threshold behavior. Roughly at what value of α does the probability of success change from close to 1 to close to 0?

2 Basic Notions of Statistical Mechanics

Gibbs distributions play a fundamental role in the analysis of the models introduced in Chapter 1. These distributions can be viewed as purely mathematical objects which arise quite naturally in the context of coding, compressed sensing and satisfiability, as we will see in Chapter 3. However, much insight and useful analogies can be gained by understanding why Gibbs distributions are natural and ubiquitous for macroscopic *physical* systems. It is the goal of this chapter to expound on the second point. This will also enable us to introduce some of the language and standard notions and settings of statistical mechanics.

Statistical mechanics describes the *macroscopic* (large scale) behavior of systems that are composed of a very large number of "elementary" degrees of freedom. For example condensed matter systems are composed of around 10^{23} atoms, molecules, magnetic moments or spins, etc. Similarly, we are interested in the behavior of our models when the number of transmitted bits, of signal components or literals is very large.

In physical systems a precise knowledge and description of the microscopic dynamics of each degree of freedom (say solving 10^{23} Newton differential equations for the positions and velocities of molecules) in a macroscopic system is impossible. Fortunately this is not required for the understanding of the macroscopic properties of the system. The general approach of statistical mechanics is to replace the full microscopic dynamical description by a probabilistic one based on appropriate probability distributions. It also turns out that the precise nature of teh microscopic dynamics is largely irrelevant (for example whether it is deterministic or random) except for the existence of quantities that are conserved under the dynamics (e.g. the energy). In fact even the existence of a dynamics is not needed, or at least it is not explicitely needed. This is important because in our models no dynamics is a priori given, and if for some reason we would choose one, presumably this choice would not be unique.

Let us briefly warn the reader that this approach also has its limits. For physical systems the "universal" probabilistic description - given by Gibbs distributions - is valid only once the so-called *thermodynamic equilibrium* is reached.[1]

[1] It is not easy to precisely define thermal equilibrium but intuitively this means the temperature is homogeneous so that there are no heat currents, the pressure is homogeneous so that there are no mechanical stresses, and the chemical potential is homogeneous so that there are no particle currents and chemical reactions.

Systems that are not in thermodynamic equilibrium are said to be *out of equilibrium*. Their fundamental probabilistic description(s) (assuming it exists) is not yet elucidated. Such systems range from the simplest stationary heat or electric flows all the way to living systems!

Thermodynamic equilibrium can somehow be defined as a state of "maximal disorder" but still compatible with whatever "conserved quantity" which might be relevant. This gives us a clue into the nature of the Gibbs distributions: these are the distributions that maximize an entropy functional (Shannon's entropy) under the constraints provided by the conserved quantities. The notion of conserved quantity might not be familiar to the reader. This should not be a problem because the most important one - and the only one that is relevant to us - is the *energy function* or *Hamiltonian* of the system. The engineer or the computer scientist may think of this quantity as some sort of *cost function*. We already encountered one such cost function in the max-K-SAT problem, namely the minimum possible number of violated clauses. In compressed sensing the mean square errors penalized or not by the ℓ_0 or ℓ_1 norms are also cost functions.

To lay the foundations on a concrete footing we will first describe "toy models" of statistical mechanics, which have turned out to be among its most important paradigms. Then we give the simplest possible derivation of the Gibbs distribution from a "maximum entropy principle". We then introduce the standard notions of free energy, marginals, correlation functions, thermodynamic limit and briefly discuss the concept of phase transition. There is no unique way to introduce Gibbs distributions and the main body of this chapter goes along a short path. But one should note that this path uses the notion of Shannon entropy which itself is not an obvious primary object for physical systems. The founding fathers of statistical mechanics deduced Gibbs distributions from more primary principles. The interested reader will find a derivation along such lines in the last section; but the impatient reader can skip this section without harm.

2.1 Lattice gas and Ising models

The lattice gas and Ising models - or more generally *spin systems* - are very simple to formulate but have taught us surprisingly much about statistical mechanics and their importance cannot be understated. There is an immense body of theory that is known about such systems which we will completely omit here (some of it is briefly reviewed in Chapter 4, Sect. 4.7). These models will serve us well to get to rapid and concrete derivation of the Gibbs distribution. This section introduces the Hamiltonians first in the traditional language of statistical mechanics; then a factor graph representation is also discussed.

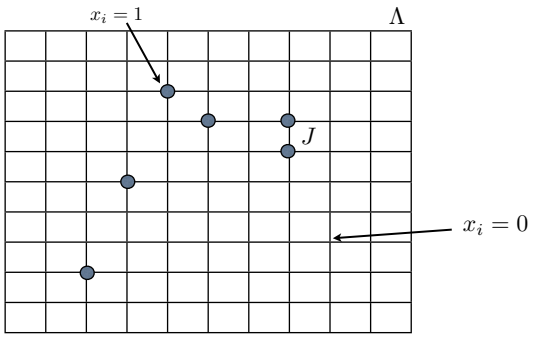


Figure 2.1 Left: a particle configuration in the lattice gas model. Full circles represent occupied sites $x_i = 1$ and empty circles unocupied sites $x_i = 0$. At most one particle occupies a lattice site. Right: a magnetic configuration in the Ising model. Positive signs indicate "up spins" $s_i = +1$ and negative signs "down spins" $s_i = -1$.

Lattice gas model

Consider a discrete d-dimensional grid (see Fig. 2.1; naturally, $d = 3$ is an important case but other values of d are of also of great relevance both theoretically and practically). Particles (e.g. atoms) occupy the vertices of this grid and at most one atom can be present on any single vertex. We call V the set of vertices and E the set of edges. The configuration of the system is described by a vector $\underline{x} = (x_1, \cdots, x_{|V|})$ where $x_i = 1$ if an atom is present at vertex i and $x_i = 0$ if vertex i is empty. To describe the system, let us introduce an energy function. In physics it is usually called the *Hamiltonian*, in computer science it is more common to say *cost* function. We define

$$\mathcal{H}(\underline{x}) = -\sum_{\{i,j\}\in E} J_{ij} x_i x_j - \sum_{i\in V} \mu_i x_i. \tag{2.1}$$

Each edge $\{i, j\}$ is counted once in the sum. Here only neighboring atoms interact and that the interaction "energy" is $-J_{ij}$.

In the canonical model $J_{ij} = J$ and $\mu_i = \mu$ are constant, with $J < 0$ corresponding to repulsive interaction and $J > 0$ to attractive interaction between neighboring atoms. The real number μ is an energy cost associated to the presence or absence of a particle (this might be a chemical affinity or a chemical potential; or for example if a two dimensional grid models the surface of some material which absorbs some vapour one may think of μ as a binding energy between the atoms of the vapour and the surface).

Canonical Ising model

The canonical Ising model is one of the oldest models and one of the best studied. We will refer to it frequently. In this model the degrees of freedoms describe "magnetic moments" localized at the sites of a crystal. For our case these sites are the vertices of the square lattice. The magnetic moments are modeled by so-called *Ising spins* $s_i = \pm$, $i \in V$, which are binary variables taking values in $\{+1, -1\}$. More precisely, the Hamiltonian is

$$\mathcal{H}(\underline{s}) = -\sum_{\{i,j\}\in E} J_{ij} s_i s_j - \sum_{i\in V} h_i s_i. \tag{2.2}$$

where $\underline{s} = (s_1, \ldots, s_{|V|})$. Again in the canonical Ising model $J_{ij} = J$ and $h_i = h$ are constant throughout the lattice. For $J > 0$ neighboring spins have a tendency to align in the same direction (ferromagnetic interaction) while for $J < 0$ they have a tendency to be in opposite directions (antiferromagnetic interaction).

Mathematically speaking the lattice-gas and Ising models are equivalent. One can go from one to the other simply by performing the change of variable

$$x_i = \frac{1}{2}(1 - s_i), \qquad \text{or} \qquad s_i = 1 - 2x_i$$

and redefining the interaction constants.

General Ising models

It is common to formulate the Ising model on general graphs $G = (V, E)$ with vertex set V, $|V| = n$ and edge set E (vertices will be denoted by i, $1 \le i \le n$, and edges by (i, j)). Motivations for such a generalisation are diverse and rich. In statistical or condensed matter physics the graph may be a regular grid or lattice representing an underlying crystalline structure. It may also represent an approximation of continuous space in various dimensions. But there are also important applications of the model in other disciplines e.g. image processing, social networks, neural networks, learning. For such applications the graph does not necessarily have a spatial structure and may be just arbitrary. A general Ising model has the Hamiltonian 2.2 where now the vertex and edge sets, V and E, refer to the general graph G.

EXAMPLE 3 The canonical *Ising* model has $G = \mathbb{Z}^d \cap B$, where d is the dimension, and B is a box of some finite side-length. Here the edges $(i, j) \in E$ of the graph consist of all nearest neighbord pairs, $|i - j| = 1$. Further, $J_{ij} = J$ for $(i, j) \in E$. The model is called ferromagnetic when $J > 0$ and anti-ferromagnetic when $J < 0$.

EXAMPLE 4 The *Curie-Weiss* model has for G the complete graph on n vertices. There are $n(n-1)/2$ edges with an associated interaction constant $J_{ij} = J/n$, with $J > 0$ constant. In addition the (external) magnetic field is

FIGURE

Figure 2.2 Left: factor graph of the canonical Ising model. Right: factor graph of a spin system with pair and plaquette interactions.

taken constant $h_i = h$. This an important exactly solvable toy model which we treat in detail in Chapter 4.

EXAMPLE 5 The Ising model on a tree has for G a tree graph i.e. a graph without loops. of finite depth D with vertex degree k for all vertices except those at the leafs (which have degree one). An exactly solvable case that we analyze in Chapter 4 is defined on a regular (constant degree) symmetric tree of finite depth, with constant interaction and magnetic field strengths.

General binary spin systems

One can also go beyond the hypothesis of pairwise interactions and consider multispin interactions. For example on a square grid the four spins of elementary plaquettes may interact through terms of the form $-\sum_{(i,j,k,l)\in P} J_{ijkl} s_i s_j s_k s_l$ where P is the set of all elementary plaquettes of the square grid and J_{ijkl} is the "plaquette interaction strength".

The most general binary spin Hamiltonian can be cast in the form

$$\mathcal{H}(\underline{s}) = -\sum_{A\subset V} J_A \prod_{i\in A} s_i \tag{2.3}$$

where $J_A \in \mathbb{R}$ and the sum over $A \subset V$ carries over all possible subsets of V (the power set with $2^{|V|}$ elements). The most general lattice gas has a similar Hamiltonian. The Ising models then corresponds to the choice $J_A = h$ for $A = \{i\}$, $i \in V$; $J_A = J$ for all $A = \{i, j\} \in E \subset V \times V$ and $J_A = 0$ otherwise.

The factor graph representation is a convenient representation for such systems. Here the factor graph is a bipartite graph with variable nodes associated to spin variables $s_1, \ldots, s_n$ (or lattice gas variables $x_1, \ldots, x_n$) and clause nodes associated to subsets $A \subset V$ with $J_A \neq 0$. The factor graphs associated to the Ising and lattice gas models on a grid, as well as the one with plaquette interactions added are is shown on Fig. 2.2. Note that in general the factor graph itself does not represent the underlying physical lattice but rather is a summary of the various interactions present in the system.

In Chapter 3 it will become apparent that the LDPC codes and K-SAT models have cost functions that are of the form 2.3.

2.2 Gibbs distribution from maximum entropy

The Gibbs distribution dates back to the very beginning of the 20th century (see Section 2.7). But in the decade following Shannon 1948 paper, Jaynes, Bril-

louin and others [**?**], [**?**] showed that one can derive Gibbs distributions from a "maximum entropy principle".

Let $p(\underline{x})$ (or $p(\underline{s})$) be a probability distribution supposed to describe the thermal equilibrium state of a macroscopic system with degrees of freedom ($\underline{x} = (x_1, \ldots, x_n)$ (or ($\underline{s} = (s_1, \ldots, s_n)$). Here one may keep in mind the lattice gas, Ising or generalized spin systems for concreteness (with $|V| = n$), but it will soon be clear that the development here is very generic. The question is: how do we choose the probability distribution?

This probability distribution should describe typical configurations of the degrees of freedom. If the system were to be completely isolated from the rest of the universe then certainly its energy would be conserved. There could also be other relevant conserved quantities depending on the nature of the system but for our purposes we can ignore more general cases. In reality the system has reached thermal equilibrium through its interactions with the environment, so it is not isolated and the energy is not strictly conserved. However in thermal equilibrium there are no macroscopic fluxes between the system and its environnement, and we can assume that the *average energy is fixed*. Thus $p(\underline{x})$ should satisfy

$$\sum_{\underline{x}} p(\underline{x})\mathcal{H}(\underline{x}) = E \tag{2.4}$$

where E is the average total energy. Of course there remain energy fluctuations due to random exchanges between the system and the environnement but these are expected to be of order $m^{(d-1)/d}$.

Now, we postulate that the state of thermal equilibrium is a maximaly disordered state (since e.g. there are no density or temperature gradients or no electric currents etc) which maximizes the entropy but still satisfies the constraint (2.4). For the entropy we take Shannon's functional

$$S(p(\cdot)) = -\sum_{\underline{x}} p(\underline{x}) \ln p(\underline{x}) \tag{2.5}$$

We use the letter S instead of H because the logarithm is neperian as is traditional in statistical mechanics.

This "guess work" leads us to the following principle: the distribution that describes the thermal equilibrium state is the one that maximizes

$$S(p(\cdot)) - \beta \sum_{\underline{x}} p(\underline{x})\mathcal{H}(\underline{x}) \tag{2.6}$$

Here β is a Lagrange multipier enforcing the constraint (2.4).

The Shannon entropy is a concave functional and other term is linear, therefore the whole functional is concave so it has a unique maximizer. To find it we must recall that there is one more constraint to enforce, namely $\sum_{\underline{x}} p(\underline{x}) = 1$ so we introduce one more Lagrange multplier γ and maximize

$$S(p(\cdot)) - \beta \sum_{\underline{x}} p(\underline{x})\mathcal{H}(\underline{x}) + \gamma \sum_{\underline{x}} p(\underline{x})$$

Setting the derivative with respect to to $p(\underline{x}')$ (for any fixed $\underline{x}'$) to zero we find

$$p(\underline{x}) = e^{\gamma-1}e^{-\beta\mathcal{H}(\underline{x})}$$

The constant γ is fixed by the normalization condition and we find for the maximizer of (2.6)

$$p_G(\underline{x}) = \frac{e^{-\beta\mathcal{H}(\underline{x})}}{Z} \tag{2.7}$$

where

$$Z = \sum_{\underline{x}} e^{-\beta\mathcal{H}(\underline{x})} \tag{2.8}$$

The distribution (2.7) is called the *Gibbs distribution* and Z the *partition function* (or sometimes the sum over states).

What is the interpretation of of the Lagrange multiplier β? For physical systems $\beta^{-1} = k_B T$ where T is the temperature of the system and k_B a constant (called the Boltzmann constant) such that $k_B T$ has units of energy. We briefly explain why in the next paragraph. But of course for our problems (coding, compressed sensing, SAT) there is no "physical temperature" so the reader may well think of β as a mathematical Lagrange parameter enforcing the constraint (2.4). As we will see in Chapter 3 this parameter often has a natural interpretation specific to each problem.

We define the *Gibbs entropy*

$$S(\beta) \equiv S(p_G(\cdot)) = -\sum_{\underline{x}} p_G(\underline{x}) \ln p_G(\underline{x}) \tag{2.9}$$

and the *internal energy*

$$\mathcal{E}(\beta) \equiv -\sum_{\underline{x}} p_G(\underline{x})\mathcal{H}(\underline{x}). \tag{2.10}$$

as functions of β. A remark is in order here: we use an abuse of notation (as is traditional in statistical mechanics and thermodynamics) and the argument of S and $\mathcal{E}$ tells us whether we view them as functional, or functions of β or as we will shortly see E. Note the relation

$$S(\beta) = \ln Z + \beta\mathcal{E}(\beta) \tag{2.11}$$

Obviously then the Gibbs entropy is $S(\beta) = \beta\mathcal{E}(\beta) + \ln Z$; but to make contact with the temperature we have to look at the entropy as a function of the average energy E,

$$S(E) = \beta(E)E + \ln Z(\beta(E)) \tag{2.12}$$

where $\beta(E)$ is computed by inverting the relation $\mathcal{E}(\beta) = E$. Differentiating

(2.12) with respect to E,

$$\begin{aligned}\frac{d}{dE}S(E) =& \beta + \big(\frac{d\beta}{dE}\big)E + \big(\frac{d}{d\beta}\ln Z\big)\frac{d\beta}{dE} \\ &= \beta + \big(\frac{d\beta}{dE}\big)E - \mathcal{E}(\beta(E))\frac{d\beta}{dE} \\ &= \beta \end{aligned} \tag{2.13}$$

We have derived the relation $\frac{d}{dE}S(E) = \beta$, and comparing with "thermodynamic identity" $\frac{d}{dE}S(E) = \frac{1}{k_B T}$ (T the temperature in degree Kelvin and k_B Boltzmann's constant in Joules per degree Kelvin), we get the interpretation of $\beta = 1/k_B T$. One commonly says that β is the "inverse temperature".

2.3 Free energy and variational principle

On the way of our derivation of the Gibbs distribution we have encountered a few important facts that we highlight in this section. But first we introduce a notation that is standard in statistical mechanics.

Bracket notation

Let $A(\underline{x})$ be any function of the configurations $\underline{x}$ of the system (these functions are sometimes called observables). The average with respect to $p_G(\underline{x})$ is denoted by the bracket $\langle - \rangle$,

$$\langle A(\underline{x}) \rangle \equiv \frac{1}{Z}\sum_{\underline{x}} A(\underline{x}) e^{-\beta \mathcal{H}(\underline{x})} \tag{2.14}$$

The normalization factor in such averages is always given by the partition function (2.8). It will become apparent in the next Chapter how convenient it is to have a reserved notation for the Gibbs average $\langle - \rangle$, and distinguish it from expectations $\mathbb{E}$ over other random objects.

Free energy

A notion of paramount importance is the *free energy* defined by

$$F(\beta) = -\frac{1}{\beta}\ln Z \tag{2.15}$$

We have the important relationship[2] (equivalent to (2.11))

$$F(\beta) = \mathcal{E}(\beta) - \beta^{-1} S(\beta) \tag{2.16}$$

[2] This allows an interpretation of the free energy as the amount of energy that is not in a disordered form, i.e in the form of heat. It is the amount of mechanical work that can be extracted from the system, hence the name free.

Computating, exactly or approximately, the free energy is often a major goal and when this is possible we learn a great deal about the model or system at hand. In particular, from the free energy we deduce the *internal energy* by differentiating $\beta F(\beta)$ with respect to β,

$$\begin{aligned}\mathcal{E}(\beta) &= \langle \mathcal{H}(\underline{x}) \rangle \\ &= -\frac{d}{d\beta} \ln Z = \frac{d}{d\beta}(\beta F(\beta)).\end{aligned} \tag{2.17}$$

Also, we can compute the Gibbs entropy by differentiating $F(\beta)$ with respect to $1/\beta$. Indeed,

$$\begin{aligned}S(\beta) &= -\langle \ln p_G(\underline{x}) \rangle \\ &= \ln Z - \beta \langle \mathcal{H}(\underline{x}) \rangle) = \beta F(\beta) - \beta \frac{d}{d\beta}(\beta F(\beta)) \\ &= -\beta^2 \frac{d}{d\beta} F(\beta) = \frac{d}{d(1/\beta)} F(\beta)\end{aligned} \tag{2.18}$$

The "energy fluctuations" are obtained by differentiating twice $\ln Z$. We leave the derivation of the following identity to the reader,

$$\langle \mathcal{H}(\underline{x})^2 \rangle - \langle \mathcal{H}(\underline{x}) \rangle^2 = \frac{d^2}{d\beta^2}(\beta F(\beta)) \tag{2.19}$$

Gibbs variationnal principle

The free energy satisfies an important variational principle. Recall that we deduced the Gibbs distribution as the one which maximizes the functional (2.6). This is the content of the so-called "Gibbs variationnal principle" which is usually formalized as follows. Define the *Gibbs free energy functional* as

$$\mathcal{F}(p(\cdot)) \equiv \sum_{\underline{x}} p(\underline{x}) \mathcal{H}(\underline{x}) - \beta^{-1} S(p(\cdot)) \tag{2.20}$$

This is a convex functional and for any distribution we have the lower bound

$$\mathcal{F}(p(\cdot)) \geq F(\beta) \tag{2.21}$$

with equality attained for $p(\cdot) = p_G(\cdot)$. This principle is often used to compute lower bounds to the free energy by taking "trial distributions" for $p(\cdot)$. These lower bounds sometimes turn out to be useful approximations or may even be sharp.

It is instructive to cast the variational principle in a language that is familiar in information theory or statistics. The *Kulback-Leibler divergence* between two distributions $p(\cdot)$ and $q(\cdot)$ is

$$D_{KL}(p||q) \equiv \sum_{\underline{x}} p(\underline{x}) \ln\left(\frac{p(\underline{x})}{q(\underline{x})}\right) \tag{2.22}$$

This functional satisfies $D_{KL}(p||q) \geq 0$ with equality when $p = q$ (see exercises). Now, note that for $q = p_G$ we have (using (2.7), (2.15) and (2.20))

$$\begin{aligned} D_{KL}(p||p_G) &= \sum_{\underline{x}} p(\underline{x}) \ln\Big(\frac{p(\underline{x})}{p_G(\underline{x})}\Big) \\ &= -S(p) - \sum_{\underline{x}} p(\underline{x}) \ln p_G(\underline{x}) \\ &= -S(p) + \beta \sum_{\underline{x}} p(\underline{x}) \mathcal{H}(\underline{x}) + \ln Z \sum_{\underline{x}} p(\underline{x}) \\ &= \beta \mathcal{F}(p(\cdot)) - \beta F(\beta) \end{aligned} \tag{2.23}$$

The "free energy difference" between a trial distribution and the Gibbs distribution is equal (up to a factor β) to the Kullback-Leibler divergence. Also, $\mathcal{F}(p(\cdot)) \geq F(\beta)$ and $D_{KL}(p||p_G) \geq 0$ are one and the same inequality. It is fitting that sometimes $D_{KL}(p||q) \geq 0$ is called the "Gibbs inequality".

2.4 Marginals, correlation functions and magnetization

Assume that a system is described by a Gibbs distribution. In practice, in order to answer many basic questions, it is often sufficient to compute (exactly or approximately) the first few marginals or even only the averages of a few important observables. In this section we collect a few related definitions and remarks.

Marginals

The definition of marginals is just the usual probabilistic one. More precisely the "first order" marginal, is defined as

$$\nu_i(x_i) = \sum_{\sim x_i} p_G(\underline{x}) \tag{2.24}$$

where $\sum_{\sim x_i}$ means that we sum over all x_j for $j = 1, \ldots i-1, i+1, \ldots n$. In other words we sum over all variables *except* x_i. The "second order" marginal is

$$\nu_{i,j}(x_i, x_j) = \sum_{\sim x_i, x_j} p_G(\underline{x}). \tag{2.25}$$

where we sum over all variables *except* x_i, x_j. Note that the marginals are normalized probability distributions.

To illustrate the use of marginals, suppose that in the lattice gas model we want to compute the averages of the total number of particles $\sum_{i \in V} x_i$ and energy $\mathcal{H}(\underline{x})$. If the marginals are known we use (the reader should check these identities)

$$\langle x_i \rangle = \sum_{x_i} x_i \nu_i(x_i), \qquad \langle x_i x_j \rangle = \sum_{x_i, x_j} x_i x_j \nu_{i,j}(x_i, x_j) \tag{2.26}$$

and once these averages are determined we easily get the averages of the two observables

$$\sum_{i\in V}\langle x_i\rangle, \qquad \text{and} \qquad \mathcal{E}(\beta)=\sum_{\{i,j\}\in E} J_{ij}\langle x_i x_j\rangle - \sum_{i\in V} h_i\langle x_i\rangle. \tag{2.27}$$

Correlation functions

In the previous section we saw that the internal energy, energy fluctuations and entropy can be computed by differentiating the free energy. Something similar is also true for the averages (2.26). Consider the following perturbation of the Hamiltonian where we add "source terms"

$$\mathcal{H}(\underline{x}) \to \mathcal{H}(\underline{x}) + \sum_{i=1}^{n} \lambda_i x_i \tag{2.28}$$

with λ_i "small" real numbers. It is sometimes the case that if we know how to compute the free energy for the unperturbed Hamiltonian then we can also compute it for small values of λ_i's. When this optimistic situation is met, such perturbations may be turned into a useful theoretical tool. Suppose we have access to $\ln Z(\underline{\lambda})$, $\underline{\lambda}=(\lambda_1,\ldots,\lambda_n)$. We have

$$\langle x_i\rangle = \frac{\partial}{\partial\lambda_i}\ln Z(\underline{\lambda})|_{\lambda=0}, \qquad \langle x_i x_j\rangle - \langle x_i\rangle\langle x_j\rangle = \frac{\partial^2}{\partial\lambda_i\partial\lambda_j}\ln Z(\underline{\lambda})|_{\lambda=0}. \tag{2.29}$$

It is a general fact that higher order derivatives yield higher order cumulants. In statistical mechanics these are called "truncated correlation functions". The covariance $\langle x_i x_j\rangle - \langle x_i\rangle\langle x_j\rangle$ is the "two-point" truncated correlation function, and the average $\langle x_i\rangle$ is sometimes called the "one-point" function. It is a good exercise to compute the third order derivative (with respect to $\lambda_i,\lambda_j,\lambda_k$) to see what kind of correlation function is obtained.

Note that for binary variables (i.e $x_i\in\{0,1\}$ or $s_i\in\{+1,-1\}$ as is the case for a lattice gas, an Ising spin system, coding or SAT) the marginals $\nu_i(x_i)$ can be recovered from the averages $\langle x_i\rangle$. For example, for $x_i\in\{0,1\}$ we have $\langle x_i\rangle = 0.\nu_i(0)+1.\nu_i(1)=\nu_i(1)$ and from the normalization condition $\nu_i(0)=1-\langle x_i\rangle$. For $s_i\in\{+1,-1\}$ we have $\langle s_i\rangle=\nu_i(1)-\nu_i(-1)$ and $1=\nu_i(1)+\nu_i(-1)$, thus $\nu_i(1)=\frac{1}{2}(1+\langle s_i\rangle)$, $\nu_i(-1)=\frac{1}{2}(1-\langle s_i\rangle)$. Similarly one can reconstruct $\nu_{i,j}(x_i,x_j)$ from one and two-point correlation functions (see exercises).

Magnetization

An observable that plays a specially important role in Ising spin systems is the magnetization of a spin configuration $m(\underline{s})=\frac{1}{n}\sum_{i\in V}s_i$. The *average magnetization* (also simply called magnetization) is the expectation with respect to the Gibbs distribution.

$$\langle m(\underline{s})\rangle = \frac{1}{n}\sum_{i\in V}\langle s_i\rangle. \tag{2.30}$$

According to the remarks of the previous paragraph, when the Hamiltonian contains a term $h\sum_{i\in V} s_i$ the magnetization can be obtained as a derivative of the free energy with respect to the magnetic field,

$$\langle m(\underline{s})\rangle = -\frac{1}{\beta}\frac{\partial}{\partial h}\ln Z = -\frac{\partial}{\partial h}f(\beta) \tag{2.31}$$

In general one can always add an infinitesimal magnetic field to the Hamiltonian, differentiate the free energy, and then take the additional magnetic field to zero.

As a last remark we note that for certain models with a symmetry between sites it is often the case that $\langle s_i\rangle$ is independent of i, so that $\langle m(\underline{s})\rangle = \langle s_i\rangle$. For example if we replace the square grid by a complete graph in the Ising model and take interaction constants independent of edges and vertices we have a permutation symetry between sites, so $\langle s_i\rangle$ is obviously independent of i. This is the Curie-Weiss model treated in chapter 4.

2.5 Thermodynamic limit and notion of phase transition

The regime of validity of statistical mechanics is the asymptotic limit of large systems where the number of degrees of freedom tends to infinity, $n \to +\infty$. This is also the regime of interest in these notes for the coding, compressed sensing and SAT problems. In the language of statistical mechanics this regime is called the *thermodynamic limit*. This is also the limit in which *phase transitions* are well defined. Here a first rather informal discussion of these concepts. They will be defined more precisely on a case by case basis in later chapter.

Thermodynamic limit

For the models of interest here we expect that $\ln Z$, $S(\beta)$ and $\langle\mathcal{H}(\underline{x})\rangle$ all scale like n, for large n. Such quantities are called *extensive*. Their thermodynamic limit, if it exists, is defined as

$$f(\beta) \equiv \lim_{n\to+\infty}\frac{1}{n}\ln Z, \qquad s(\beta) \equiv \lim_{n\to+\infty}\frac{1}{n}S(\beta), \qquad e(\beta) \equiv \lim_{n\to+\infty}\langle\mathcal{H}(\underline{x})\rangle \tag{2.32}$$

Taking the limit of (2.11) we obtain that these quantities are related by

$$f(\beta) = e(\beta) - \beta^{-1}s(\beta) \tag{2.33}$$

Relations (2.17), (2.18), (2.19) are also true for the limiting quantities scaled by $1/n$, *provided* one can permute $d/d\beta$ and $\lim_{n\to+\infty}$. This is the case as long as $f(\beta)$, $s(\beta)$ and $e(\beta)$ are "sufficiently smooth" functions of β. The issue here is a real one and is connected to the subject of *phase transitions* to which we will come back.

Let us now discuss the issue of thermodynamic limit for the correlation functions and the Gibbs distribution. One cannot simply use the definition (2.7) in

the limit $n \to +\infty$ since the numerator and denominator both tend to infinity (generically exponentially fast). So what is the meaning of the Gibbs distribution in the thermodynamic limit? One way to proceed would be to compute the limits of the marginals, e.g.

$$\lim_{n\to+\infty} \nu_i(x_i), \ \lim_{n\to+\infty} \nu_{i,j}(x_i, x_j), \ \lim_{n\to+\infty} \nu_{i,j,k}(x_i, x_j, x_k), \ \dots \tag{2.34}$$

and define the "infinite volume" Gibbs distribution as the distribution with this set of marginals. Because of phase transition phenomena such limits are *not* always defined in a unique way.

Phase transitions

Let us now say a few words about phase transitions, a subject to which we will come back in due course. The free energy $f(\beta)$ is always a *continuous* and *convex* function of β. To see this note that for finite n, $F(\beta)/n$ is analytic as a function of β, and also that $F(\beta)/n$ is convex as can be seen from the positivity of the variance of the Hamiltonian in (2.19). The limit of a continuous convex function is continuous and convex, thus $f(\beta)$ is continuous and convex. Values of β where differentiability fails are called *phase transition points*. Points where the first derivative of $f(\beta)$ has a jump are called *first order* phase transition points; those where the first derivative is continuous but the second derivative is discontinuous are called *second order* phase transition points (such points form a set of measure zero by a theorem of Alexandrov). Phase transitions of higher order are also possible: a phase transition of n-th order is one where the $n-1$-th derivatives of $f(\beta)$ are all continuous and the n-th one is discontinuous. This classification of phase transitions is due to Ehrenfest [**?**]. We stress that this is not the only classification, nor the most modern one, but one that will suit us. Temperature is not the only parameter with respect to which the free energy can be non-differentiable. For example in the canonical Ising model (with $h_i = h$ constant) there are phase transitions with respect to the magnetic field h. This helps us understand the statement made above about the non-unicity of the Gibbs distribution in thermodynamic limit. Indeed we saw that the magnetization is obtained as derivative of the free energy with respect to h; thus if at a first order phase transition point this derivative can take two distinct values this means that one should define two one-point marginals and hence two Gibbs distributions, in thermodynamic limit. In Chapter 4 we solve explicitly a useful toy model - the Curie-Weiss model - which will allow us to discuss phase transitions more concretely. A mini-review of the phase transitions in the Ising and lattice gas models is found as an aside at the end of that Chapter 4.

2.6 Spin glass models - random Gibbs distributions

In the next chapter we will see that our three problems coding, compressive sensing and satisfiability can be formulated as a particular type of statistical mechanics models, the so-called *spin glass models*. In this paragraph we briefly explain what spin glass models are in general.

One of the ambitions of statistical mechanics is to describe the great variety of "phases" of condensed matter (a non-exhaustive list: gases, liquids, crystalline solids, metals, insulators, semi-conductors, superconductors, superfluids, magnetic materials, liquid crystals, polymers, glasses, emulsions etc). One of the oldest known but still badly understood and intriguing phase is "glass". Ordinary glass is an amorphous material where the geometrical arrangement of atoms is frozen as in a solid but at the same time is irregular as in a liquid; it is believed that in a sense ordinary glass is a "frozen liquid" with such a huge viscosity that it does not flow for all practical purposes. There also exist magnetic materials whose magnetic degrees of freedom interact through irregular interactions with varying signs and have a glassy behaviour. Here we will not say more about the physical concept of "glass" which is often a mater of debate.

Spin glass models are Ising or generalized spin systems, see (2.2), (2.3), with *random interaction constants*. Such models where first introduced by Anderson and Edwards in the 1970's in an attempt to capture the properties of magnetic materials with interactions of "varying" intensity and sign.

EXAMPLE 6 The usual Edwards-Anderson (EA) spin glass model has $G = \mathbb{Z}^d \cap B$, where d is the dimension, and B is a box of some finite side-length, and has *random* iid coupling constants $J_{ij} = \pm J$, where the sign is iid Bernoulli($\frac{1}{2}$) and $h_i = h$ is constant. The analysis of this model is still far from understood nowadays.

EXAMPLE 7 The random field Ising model (RFIM) also has $G = \mathbb{Z}^d \cap B$, where d is the dimension, and B is a box of some finite side-length, has constant $J_{ij} = J$ and random iid magnetic field $h_i = \pm h$ with Bernoulli(1/2) signs. This is also a very non-trivial model with many open questions.

EXAMPLE 8 The Sherrington-Kirkpatrick (SK) model has for G the complete graph on n vertices with $n(n-1)/2$ edges. The coupling constants J_{ij} are iid Bernoulli(1/2) in $\{-\frac{J}{\sqrt{n}}, +\frac{J}{\sqrt{n}}\}$ or Gaussian $\mathcal{N}(0, \frac{J^2}{n})$, and the magnetic field is generally taken constant $h_i = h$. The analysis of this model in Chapter 7 will play a somewhat important role for compressed sensing.

Variants of these models use other distributions for the interaction constants, for example Gaussians. One can also take more complicated models with more general interactions, e.g. J_A's in (2.3) may be random variables, or also replace the regular grids by a random graph. The study of spin glass models has turned out be very non-trivial and has been a source of many fundamental concepts in statistical mechanics of so-called *disordered systems*. Fortunately, the spin glass

models that will be relevant for our three problems are defined on complete or locally tree-like graphs and as we will see the absence of "low dimensional geometry" makes them somehow much easier to study than the EA and RFIM. easier to study. This is already the case for non-random versions as we will see in Chapter 4.

The Gibbs distribution associated to a spin glass Hamiltonian has two levels of randomness. First we have the randomness of the Hamiltonian itself, i.e. the interaction constants or the underlying grid. Once they are sampled from a specified ensemble we have a fixed instance of a Gibbs distribution which is a probability distribution over the spin or lattice gas variables. So the study of spin glass models is the study of *ensembles of random Gibbs distributions*. A word about a terminology that comes from the manufacturing processes of materials and has become standard is in order here. The random interaction constants of the Hamiltonian are called *quenched variables* because once the instance (or the sample) is specified they are fixed or "frozen" once for all. The spin or lattice gas degrees of freedom are sometimes called *annealed variables* because they "adapt" themselves into their typical configurations. A word about notation is also in order. It is very convenient to have two separate notations to distinguish averages with respect to quenched and annealed variables. The expectations with respect to the Gibbs distribution are always denoted by the same bracket $\langle - \rangle$ and those with respect to the quenched variables by $\mathbb{E}$ with possible subscripts describing the ensemble. Thus if $A(\underline{x})$ is an observable (say the magnetization) the average over the annealed and quenched variables is $\mathbb{E}[\langle A(\underline{x}) \rangle]$. The reader should convince himself that it would be meaningless to permute the two expectations.

The quenched randomness is ubiquitous in many engineering problems where one has to deal with particular instances that belong to a model ensemble. This is the point of view that we took in the definition of the coding, compressive sensing and satisfiability problems. As we will see in the next Chapter once an instance of the ensemble is specified the Gibbs distribution appears more or less naturally in the mathematical formulation. So in a sense the connections between our models and the statistical mechanics of spin glasses is not surprising but just very natural. In fact such connections have been with us since the 1970's for various computer science problems such as the travelling salesman or graph partionning problems and also in neural networks (see references [**?**]).

2.7 Gibbs distribution from Boltzmann's principle

This section is not needed for the main development of these notes and can be skipped in a first reading.

We will derive the Maxwell-Boltzmann or Gibbs distributions from two basic principles. We first discuss these principles and then derive the Gibbs distribution in the next section. We point out that there is not only *one* way of deriving Gibb's

distributions and not only *one* set of generally agreed upon principles which lead to them. Rather, as with any physical law, is has to be "gussed" from a variety of experiments, plausible assumptions and models, which all lead to a conclusion that is then validated by experiments.

For concreteness the reader may keep in mind the lattice gas model in the arguments of this section. We suppose that the particles have a dynamics with "trajectories" $x_i(t)$, $i = 1, \ldots, n$ on the lattice parametrized by time t. As we will see the rpecise nature of the dynamics will not concern us except for an "ergodicity hypothesis".

Uniform microcanonical measure

Let $[0, T]$ be the time interval over which we measure an observable quantity $A(\underline{x}(t))$ and let τ be a characteristic microscopic time scale, for example the time scale on which a single particle jumps from a position to a neighboring one. In practice we have $T \gg \tau$. We assume that a measurement returns an average over time

$$\frac{1}{T}\int_0^T dt\, \phi(\underline{x}(t)), \tag{2.35}$$

and that in the state of thermodynamic equilibrium this average is independent of T for $T \gg \tau$, and independent of the origin of time and initial condition (in other words we can shift $[0, T] \to [s, s+T]$ and the average is independent of s).

During the measurement interval the state of the system $\underline{x}(t)$ will wander across the energy surface $\Gamma_E \subset \{0,1\}^{|V|} = \{\underline{x} \mid \mathcal{H}(\underline{x}) = E\}$. Let $t(\underline{x})/T$ be the fraction of time it spends in state $\underline{x}$.

Our first principle states that *for an isolated system*, when $T \gg \tau$, the fraction of time $t(\underline{x})/T$ spent in state $\underline{x}$, is given by the uniform distribution on the energy surface Γ_E. In other words for $t(\underline{x})/T$ we take,

$$\mu_E(\underline{x}) = \frac{\mathbb{1}(\underline{x} \in \Gamma_E)}{W(E)} \tag{2.36}$$

where the normalization factor is

$$W(E) = \sum_{\underline{x} \in \{0,1\}^{|V|}} \mathbb{1}(\underline{x} \in \Gamma_E). \tag{2.37}$$

This distribution is called the *microcanonical distribution*. In words this assumption states that if the system is isolated it spends an equal time in all states.

A fundamental consequence is that we can replace the time average (2.35) by a configurational average,

$$\frac{1}{T}\int_0^T dt\, A(\underline{x}(t)) \approx \sum_{\underline{x} \in \{0,1\}^{|V|}} \mu_E(\underline{x}) A(\underline{x}), \qquad T \gg \tau \tag{2.38}$$

Often equ. (2.38) is formalized and called the *ergodic hypothesis*. The ergodic hypothesis states that the dynamics exactly satisfies this identity in the limit

$T \to +\infty$, for almost all initial conditions $\underline{x}(0)$ (note that the right hand side does not depend on the initial condition) and all observables $A(\underline{x})$.

This ergodic hypothesis has played a very important historical role but has never been proved for macroscopic systems, and its physical relevance has often been debated.[3] In fact its precise validity is not so important, and ultimately we just postulate that averages of a class reasonable of observables in an isolated system can be computed from the uniform distribution.

Boltzmann's principle

Consider the normalization of the microcanonical measure, $W(E)$. Generically this has exponential behavior in the number of degrees of freedom. It is therefore to introduce the *Boltzmann entropy* as

$$S_B(E) = \ln W(E). \tag{2.39}$$

We stress that this is a priori a purely combinatorial quantity: more about it later.

EXAMPLE 9 Let us consider the lattice gas model introduced in the previous example for the non-interacting case $J = 0$. Since the energy surface consists of $\Gamma_E = \{\underline{x} \mid \sum_{i\in V} x_i = E/\mu\}$ there must be E/μ lattice nodes with $x_i = 1$ among $|V| = n$ of them (and the rest with $x_i = 0$). Hence

$$W(E) = \binom{n}{E/\mu} \simeq \exp\left(n h_2\Big(\frac{E}{\mu n}\Big)\right), \tag{2.40}$$

where $h_2(\cdot)$ is the binary entropy function. In the infinite size limit we have

$$s(e) = \lim_{\substack{n\to\infty \\ E/n=e}} \frac{1}{n} S_B(E) = h_2\left(\frac{e}{\mu}\right), \tag{2.41}$$

where $e = E/n$ and $h_2(u) = -u\ln u - (1-u)\ln(1-u)$ the binary entropy function. Note that this is a concave function (for physically sensible Hamiltonians the Boltzmann entropy is a concave function of e; this is not always the case in computer science and coding problems with hard constraints).

There is a purely thermodynamic (and experimentally measurable) notion of entropy elucidated in the 19-th century (along with the notions of heat and work) by Carnot, Clausius, Joule, Helmholtz, Kelvin and others. For a system at thermodynamic equilibrium with homogeneous temperature and pressure T and p, the thermodynamic entropy $S_{\text{thermo}}(E, V)$ is a function of the total energy E and volume V satisfying

$$\frac{\partial}{\partial E} S_{\text{thermo}} = \frac{1}{T}, \qquad \frac{\partial}{\partial V} S_{\text{thermo}} = \frac{p}{T}\,. \tag{2.42}$$

[3] It should be noted that this hypothesis is at the origin of a deep branch of mathematics, "ergodic theory", and has been proven to hold for systems with a few particles such as billiard balls [**?**]

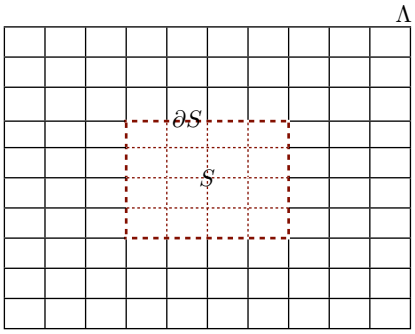


Figure 2.3 The system S is embedded in a thermal bath V. The total system V is considered as an isolated system and its total energy E is conserved. We compute the induced measure on S.

From T and p one can in principle recover S_{thermo}. Note that the unit of S_{thermo} are Joules per dgree Kelvin.

Boltzmann's principle postulates equality of the thermodynamic and Boltzmann entropies. The former is a physically measurable quantity and later is a mathematical combinatorial quantity that can in principle be calculated. So,

$$S_{\text{thermo}} = k_B S_{\text{B}}, \tag{2.43}$$

Here, k_B is Boltzmann's constant with units of Joules per degree Kelvin. If we combine this identitywith the first equation in (2.42) then we get

$$\frac{\partial \mathcal{S}_{\text{Boltz}}}{\partial E} = \frac{1}{k_B T}. \tag{2.44}$$

This fundamental principle makes the connection between statistical mechanics and thermodynamics. In the next paragraph we will see that it is a crucial ingredient in the derivation of the Gibbs distribution.

Derivation of the Gibbs distribution

The microcanonical distribution described earlier, only characterizes an isolated system. However, real macroscopic systems are not isolated. One should also notice that in practice, in order to reach thermal equilibrium it is necessary to put systems in contact with a *thermal bath*, an infinite reservoir which is at a constant temperature.

For simplicity, we take the lattice gas as our big reservoir and suppose it is isolated with total energy E. The real system of interest is *a much smaller but still*

macroscopic system $\Sigma \subset V$ (see Figure 2.3). We label the degrees of freedom in Σ as $(x_1, \dots, x_m)$ and those outside Σ by $(x_{m+1}, \dots, x_n)$. The regime of interest is $1 \gg m \gg n$. We are interested in computing *only* averages of observables which depend on the degrees of freedom of the smaller system Σ, $A(x_1, \dots, x_m)$. Of course we can compute them with the microcanonical distribution

$$\mu_E(x_1, \dots, x_n) = \frac{\mathbb{1}((x_1, \dots, x_n) \in \Gamma_E)}{W(E)}. \tag{2.45}$$

but clearly, since A depends only on $x_1, \dots, x_m$, we only need the marginal of this distribution over the degrees of freedom of Σ.

We now show that the marginal of (**??**) is the Gibbs distribution with inverse temperature $\frac{1}{k_B T} = \frac{\partial}{\partial E} S_B(E)$.

The marginal distribution for Σ reads systems is $x_1, \dots, x_m$ reads

$$\begin{aligned} \mu_{\text{ind}}(x_1, \dots, x_m) &= \sum_{x_{m+1}, \dots, x_n} \mu_E(x_1, \dots, x_n) \\ &= \frac{\sum_{x_{m+1}, \dots, x_n} \mathbb{1}((x_1, \dots, x_n) \in \Gamma_E)}{\sum_{x_1, \dots, x_n} \mathbb{1}((x_1, \dots, x_n) \in \Gamma_E)}. \end{aligned} \tag{2.46}$$

The total energy E is a sum of the energy inside Σ, the energy outside Σ and an interaction part between the inside and the outside,

$$\begin{aligned} E &= \mathcal{H}(x_1, \dots, x_n) \\ &= \mathcal{H}_\Sigma(x_1, \dots, x_m) + \mathcal{H}_{V \setminus \Sigma}(x_{m+1}, \dots, x_n) + \mathcal{H}_{\text{int}}, \end{aligned}$$

Generically $\mathcal{H}_\Sigma$ is of the order of m (the volume of Σ), $\mathcal{H}_{V \setminus \Sigma}$ is of order $n-m$ (the volume of the outside of Σ) and $\mathcal{H}_{int}$ is of order the surface of Σ. In d dimensions the surface of Σ is of order $m^{(d-1)/d} << m << n - m$, thus neglecting the interaction term we conclude that if $(x_1, \dots, x_n)$ belongs to the energy surface Γ_E then $(x_{m+1}, \dots, x_n)$ belongs to the energy surface $\Gamma_{E - \mathcal{H}_\Sigma(x_1, \dots, x_m)}$. With these remarks we obtain

$$\begin{aligned} \mu_\Sigma(x_1, \dots, x_m) &= \frac{\sum_{x_{m+1}, \dots, x_n} \mathbb{1}((x_{m+1}, \dots, x_n) \in \Gamma_{E - \mathcal{H}_\Sigma(x_1, \dots, x_m)})}{\sum_{x_1, \dots, x_m} \sum_{x_{m+1}, \dots, x_n} \mathbb{1}((x_{m+1}, \dots, x_n) \in \Gamma_{E - \mathcal{H}_\Sigma(x_1, \dots, x_m)})} \\ &= \frac{\exp(S_B(E - \mathcal{H}_S(x_1, \dots, x_m))}{\sum_{x_1, \dots, x_m} \exp(S_B(E - \mathcal{H}_\Sigma(x_1, \dots, x_m))} \\ &= \frac{\exp(S_B(E) - \mathcal{H}_\Sigma(x_1, \dots, x_m) \frac{\partial}{\partial E} S_B + \dots)}{\sum_{x_1, \dots, x_m} \exp(S_B(E) - \mathcal{H}_S(x_1, \dots, x_m) \frac{\partial}{\partial E} S_B + \dots)} \\ &= \frac{\exp\left(-\mathcal{H}_\Sigma(x_1, \dots, x_m)/k_B T\right)}{\sum_{x_1, \dots, x_m} \exp\left(-\mathcal{H}_\Sigma(x_1, \dots, x_m)/k_B T\right)}, \end{aligned}$$

The second equality follows from the definition of the Boltzmann entropy. The third equality uses a Taylor expansion to first order since $E \gg \mathcal{H}_\Sigma(x_1, \dots, x_m)$ since $n \gg m$). The last equality is the point where Boltzmann's principle is used. The final result is exactly the Gibbs distribution for Σ.

2.8 Notes

If you visit Boltzmann's grave in Vienna you will see the inscription $S = k \ln W$. Austrian physicist and philosopher. He was a professor of mathematics in Vienna. He hanged himself.

Problems

2.1 Gibbs distribution. Give the details of the derivation leading to (2.7) and (2.8).

2.2 Energy fluctuations. Derive relation (2.19).

2.3 Positivity of Kullback-Leibler divergence. Prove in two different ways that $D_{KL}(p||q) \geq 0$ with equality if and only if $p(\underline{x}) = q(\underline{x})$ for all $\underline{x}$. Hint: use $\ln u \leq u - 1$ for $u > 0$ and also the convexity of $f(u) = u \ln u$.

2.4 Correlation functions from derivatives of partition function. Check the formulas (2.29) and also

$$\frac{\partial^3}{\partial\lambda_i\lambda_j\lambda_k} \ln Z(\underline{\lambda})|_{\underline{\lambda}=0} = \langle x_i x_j x_k\rangle - \langle x_i x_j\rangle\langle x_k\rangle - \langle x_j x_k\rangle\langle x_i\rangle - \langle x_i x_k\rangle\langle x_j\rangle + 2\langle x_i\rangle\langle x_j\rangle\langle x_k\rangle$$

2.5 Marginals for Ising spins. Consider any spin system with binary variables $s_i \in \{+1, -1\}$. Express the marginals $\nu_i(s_i)$ and $\nu_{i,j}(s_i, s_j)$ in terms of the averages $\langle s_i\rangle$, $\langle s_j\rangle$ and $\langle s_i s_j\rangle$.

2.6 Ising model in one dimension: transfer matrix method. The aim of this problem is to solve the one-dimensional Ising model by the transfer matrix method. The Hamiltonian of the one-dimensional Ising model *on a ring* is

$$\mathcal{H} = -J \sum_{i=-\frac{n}{2}}^{\frac{n}{2}-1} s_i s_{i+1} - h \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} s_i - J s_{-\frac{n}{2}} s_{\frac{n}{2}}$$

The last term accounts for the fact that the sites are on a ring. Consider the *transfer matrix*

$$T = \begin{pmatrix} e^{K+h} & e^{-K} \\ e^{-K} & e^{K-h} \end{pmatrix}$$

(i) Show that the partition function can be expressed as $Z_N = tr\,(T^n)$ where tr is the sum over eigenvalues (the trace).

(ii) Find the eigenvalues of T and show that the free energy per spin is in the

thermodynamic limit $n \to +\infty$

$$f = -\beta^{-1} \ln[e^{\beta J} \cosh(\beta h) + (e^{2\beta J} \sinh^2(\beta h) + e^{-2\beta J})^{1/2}].$$

(iii) Compute the *magnetization* from the thermodynamic definition: $m = -\frac{\partial}{\partial h} f$ and plot the curve m as a function of h for various values of β. Convince yourself both on the plot and from the analytic formula that there is *no* sharp phase transition for any temperature $T > 0$.

2.7 Ising model in one dimension: message passing method. In this problem we solve the one-dimensional Ising model by a "message passing" or "iterative" method. We consider the model on an *open* chain, which means that the Hamiltonian is

$$\mathcal{H} = -J \sum_{i=-\frac{n}{2}}^{\frac{n}{2}-1} s_i s_{i+1} - h \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} s_i$$

We want to compute the average $\langle s_i \rangle$ in the thermodynamic limit $n \to +\infty$. For simplicity we consider the middle spin $\langle s_0 \rangle$ (it can be checked that $\lim_{n\to+\infty} \langle s_i \rangle$ is independent of i, for i fixed).

(i) In the Gibbs average for $\langle s_i \rangle$ perform explicitely the sum over the two end spins $s_{-n/2}$ and $s_{n/2}$. Show that this leads to a new model on a shorter chain with new Hamiltonian

$$\begin{aligned}\beta \mathcal{H}^{(1)} = &-J \sum_{i=-\frac{n}{2}+1}^{\frac{n}{2}-2} s_i s_{i+1} - h \sum_{i=-\frac{n}{2}+2}^{\frac{n}{2}-2} s_i \\ &- \beta^{-1}(h + \tanh^{-1}(\tanh(\beta J) \tanh(\beta h)))(s_{-\frac{n}{2}+1} + s_{-\frac{n}{2}-1})\end{aligned}$$

(ii) Repeat this calculation to show that

$$\lim_{N\to+\infty} \langle s_0 \rangle = \tanh(\beta h + 2 \tanh^{-1}(\tanh(\beta J) \tanh(\beta u)))$$

where u is the solution of the fixed point equation

$$u = \beta h + \tanh^{-1}(\tanh(\beta J) \tanh \beta u)$$

(iii) Show that the solution of this fixed point equation is unique (so that there is no ambiguity in this result).

(iv) Check that the result agrees with the expression for m found in the first problem. Hint: use the identity $\tanh(x+y) = (\tanh x + \tanh y)/(1 + \tanh x \tanh y)$

3 Formulation of Problems as Spin Glass Models

We will reformulate the three problems introduced in Chapter 1 in a statistical physics language. Both the coding as well as the compressive sensing problem are inference problems, and in this context Gibbs distributions appear quite naturally. The random K-SAT problem is not an inference problem and the Gibbs distribution does not appear in a completely straightforward way. A simple and natural distribution is the uniform one over the set of satisfying assignments. In a sense this distribution is akin to the microcanonical measure introduced in Sec. 2.7. But, given a formula, the set of satisfying assignments is not known, typically we dont even know if it is empty or not, and in any case it is difficult to get a handle on the uniform distribution. Instead, we will take a Gibbs distribution which is always well defined on all possible assignments and get a good approximation to the uniform distribution when the inverse temperature β tends to infinity.

In all cases we end up with *spin glass* models. What do we mean by this? Take for example the coding or satifiability examples. Instead of talking about physical degrees of freedoms (e.g. magnetic spins), we can think of the bits which are to be transmitted or the Boolean variables and which can take one of two possible values as *spins*. This explains why we talk about *spin* systems. In compressed sensing the signal components are continuous and this model falls in the class of continuous spin systems. But where is the glass? In coding the way we have defined our code ensemble, a check interacts with a random subset of the bits so the graph and interactions are random. The same is true for satisfiability. In compressed sensing the measurement matrices are random which results in random interaction constants between the continuous spins. Note that in compressed sensing the graph itself is bipartite complete and is therefore not a random object. In all our models this type of randomness is quenched: once we pick an instance from the appropriate ensemble we have a fixed Gibbs distribution. In this sense our models fall in the general category of spin glasses.

To summarize, our reformulations will lead us to *random Gibbs distributions*. For each problem we will identify a Hamiltonian function over "spins" with underlying graphs and interaction constants belonging to a random ensemble.

3.1 Coding as a spin glass model

Let $\mathcal{C}$ be a code from Gallager's (d_v, d_c) ensemble of block length n. Recall that d_v is the degree of variable nodes, and that d_c is the degree of check nodes. Further, n is the block length, i.e., it is the number of variable nodes. We have $nd_v = md_c$ where m is the number of parity checks.

Assume that we transmit the codeword $\underline{x} = (x_1, \ldots, x_n)$ through a binary, memoryless symmetric channel without feedback, and let $\underline{y} = (y_1, \ldots, y_n)$ be the received word. We will use the spin variable notation for the codebits. This means that we write $s_i = (-1)^{x_i}$ (or $s_i = 1 - 2x_i$). The channel is described by transition probabilities

$$p(\underline{y}|\underline{s}) = \prod_{i=1}^{n} p(y_i|s_i) \tag{3.1}$$

The three examples to which we will refer most often are the BEC, the BSC, and the BAWGNC.

We will always assume that the transmitted (input) codeword $\underline{s}^{\text{in}}$ is selected uniformly at random, thus the joint distribution for $(\underline{s}, \underline{y})$ is $p(\underline{y}|\underline{s}) \times \frac{\mathbb{1}(\underline{s} \in \mathcal{C})}{|\mathcal{C}|}$. We call $p(\underline{s} \mid \underline{y})$ be the posterior probability distribution of $\underline{s}$ given the received word $\underline{y}$.

MAP decoding

The *bit-MAP estimate* ((MAP means maximum a posteriori) is,

$$\hat{s}_i(\underline{y}) = \text{argmax}_{s_i}\ \nu_i(s_i|\underline{y}), \tag{3.2}$$

where $\nu_i(s_i|\underline{y})$ is the marginal of the posterior $p(\underline{s}|\underline{y})$. This estimator is optimal in the sense that it minimizes the bit probability of error.

Since $\underline{s}^{\text{in}}$ is picked uniformly at random from the code, the probability that bit i is wrongly decoded is

$$\frac{1}{|\mathcal{C}|} \sum_{\underline{s}^{\text{in}} \in \mathcal{C}} \mathbb{P}[\hat{s}_i(\underline{Y}) \neq s_i^{\text{in}}]$$

Thus the *average bit probability of error* is defined as

$$\mathbb{P}_b[\text{error}] = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|\mathcal{C}|} \sum_{\underline{s}^{\text{in}} \in \mathcal{C}} \mathbb{P}[\hat{s}_i(\underline{Y}) \neq s_i^{\text{in}}] \tag{3.3}$$

We will see that bit-MAP decoding has a very natural statistical mechanical interpretation in terms of the magnetization of a spin glass model.

Although we will not be deal much with it, we mention the *block-MAP estimate* $\hat{\underline{s}}(\underline{y}) = \text{argmax}_{\underline{s}}\ p(\underline{s} \mid \underline{y})$ and the associated the block probability of error $\mathbb{P}_{\text{B}}[\text{error}] = \frac{1}{|\mathcal{C}|} \sum_{\underline{s}^{\text{in}} \in \mathcal{C}} \mathbb{P}_B[\hat{\underline{s}}(\underline{Y}) \neq \underline{s}^{\text{in}}]$. We will see that the block-MAP decoding is equivalent to finding the minimum energy states of a Hamiltonian; and that

there is a "finite temperature" decoder which interpolates between the bit-MAP and block-MAP decoders.

The posterior distribution as a spin glass model

We now show that the posterior distribution $p(\underline{s} \mid \underline{y})$ is a random Gibbs distribution. Recall that a code is represented by a bipartite factor graph with variable nodes $i = 1, \ldots, n$ and checks[1] $a = 1, \ldots, m$; like in Fig. 1.1. We call ∂a the set of variable nodes connected to check a. A code word $\underline{x}$ has to satisfy all parity check constraints $\sum_{i \in \partial a} x_i = 0$ for all checks. In spin language are equivalent to $\prod_{i \in \partial a} s_i = 1$ for all checks. Thus the prior distribution over codewords can be written as

$$p_0(\underline{s}) = \frac{\mathbb{1}(\underline{s} \in \mathcal{C})}{|\mathcal{C}|} == \frac{1}{|\,\mathcal{C}\,|} \prod_{a=1}^{m} \frac{1}{2}(1 + \prod_{i \in \partial a} s_i). \tag{3.4}$$

Using Bayes law and the channel law (3.1),

$$\begin{aligned} p(\underline{s}|\underline{y}) &= \frac{p(\underline{y}|\underline{s})p_0(\underline{s})}{p(\underline{y})} \\ &= \frac{p_0(\underline{s}) \prod_{i=1}^{n} p(y_i|s_i)}{\sum_{\underline{s}} p_0(\underline{s}) \prod_{i=1}^{n} p(y_i|s_i)} \end{aligned} \tag{3.5}$$

Now we divide the numerator and denominator by $\prod_{i=1}^{n} p(y_i| - 1)$ and use

$$\frac{p(y_i|s_i)}{p(y_i| - 1)} = e^{h_i s_i + h_i} \tag{3.6}$$

where we have introduced the half-loglikelihood variable associated to channel observation y_i

$$h_i = \frac{1}{2} \ln \frac{p(y_i| + 1)}{p(y_i| - 1)}, \tag{3.7}$$

and obtain

$$p(\underline{s}|\underline{y}) = \frac{p_0(\underline{s}) \prod_{i=1}^{n} e^{h_i s_i + h_i}}{\sum_{\underline{s}} p_0(\underline{s}) \prod_{i=1}^{n} e^{h_i s_i + h_i}}. \tag{3.8}$$

Finally using (3.4) we arrive at the expression

$$p(\underline{s}|\underline{y}) = \frac{1}{Z} \prod_{a=1}^{m} \frac{1}{2}(1 + \prod_{i \in \partial a} s_i) \prod_{i=1}^{n} e^{h_i s_i} \tag{3.9}$$

where the normalizing factor in the denominator is

$$Z = \sum_{\underline{s}} \prod_{a=1}^{m} \frac{1}{2}(1 + \prod_{i \in \partial a} s_i) \prod_{i=1}^{n} e^{h_i s_i}. \tag{3.10}$$

It equivalent to describe the channel outputs by $\underline{h}$ or $\underline{y}$, and we will sometimes

[1] We will usually denote variable nodes by letters $i, j, k, \ldots$ and checks by $a, b, c, \ldots$

interchange them in our notations when this does not lead to ambiguities. So for example we can write $p(\underline{s}|\underline{y}) = p(\underline{s}|\underline{h})$ for the posterior. But for the transition probability of the memoryless channel we have to be more careful. In terms of half-loglikelihood variable we denote it $c(h_i|s_i)$, and formally $p(y_i|s_i)dy_i = c(h_i|s_i)dh_i$. In the exercises you compute explicitely $c(h_i|s_i)$ for the BEC, BSC and BAWGNC.

The posterior (3.9) is a *random Gibbs distribution*, also called a *spin glass model*. Here the word random relates to the randomness of the channel outputs as well as the choice of code. For each channel realization $\underline{h}$ and each code $\mathcal{C}$ picked from the Gallager ensemble we have a distribution over the spins $\underline{s} \in \{-1, +1\}^n$. In the terminology of physics the randomness associated with the code (or factor graph) and channel realisations is called "quenched randomness". This is because in a given experiment (here the transmission and reception of a message) the code and channel realisations are fixed, or frozen. The spins on the other hand are called annealed variables because they fluctuate and adapt themselves into their typical configurations.

What are the distributions of the quenched randomness? The distribution over the codes is the uniform distribution over Gallager's ensemble. In the configuration model introduced in Chapter 1 this is the uniform distribution over all permutations among nd_v sockets. Averages with respect to codes are denoted $\mathbb{E}_{\mathcal{C}}[-]$. The channel outputs are distributed according to $c(\underline{h}|\underline{s}^{\text{in}})$ and corresponding averages $\mathbb{E}_{\underline{h}|\underline{s}^{\text{in}}}[-]$.

This is a good point to recall that averages with respect to the Gibbs distribution, in other words with respect to the spins, are denoted by the bracket $\langle - \rangle$, and are distinguished from averages over quenched variables generically denoted $\mathbb{E}$. Note also that Gibbs brackets depend on $\underline{h}$ so $\langle - \rangle$ and $\mathbb{E}$ cannot be interchanged.

We explained in Chapter 2 that a crucial feature of Gibbs distributions, which plays a fundamental role in their analysis, is their "locality". We see that this is the case here because each term in the products in (3.9) and (**??**) depend on a finite number of spins. This is the essential reason why statistical mechanics methods can be applied.

Bit-MAP decoder and magnetization

The bit-MAP decoder has a natural relation to the magnetization of the spin glass. The definition (3.2) is equivalent to

$$\begin{aligned} \hat{s}_i(\underline{h}) &= \text{sign}(\nu_i(s_i = 1|\underline{h}) - \nu_i(s_i = -1|\underline{h})) \\ &= \text{sign}(\sum_{s_i} s_i \nu_i(s_i|\underline{h})) = \text{sign}\langle s_i \rangle, \end{aligned} \tag{3.11}$$

So the bit-MAP estimate for the i-th bit i is given by the sign of the local magnetisation $\langle s_i \rangle$,

$$\begin{aligned}\langle s_i \rangle &= \frac{1}{Z} \sum_{\underline{s}} s_i \prod_{a=1}^{m} \frac{1}{2}(1 + \prod_{i \in \partial a} s_i) \prod_{i=1}^{n} e^{h_i s_i} \\ &= \frac{\partial}{\partial h_i} \ln Z \end{aligned} \tag{3.12}$$

Using $\mathbb{P}[\hat{s}_i(\underline{h}) \neq s_i^{\text{in}}] = \mathbb{E}_{\underline{h}|\underline{s}^{\text{in}}}[\mathbb{1}(\hat{s}_i(\underline{h}) \neq s_i^{\text{in}})$ the average bit probability of error (3.3) becomes

$$\mathbb{P}_b[\text{error}] = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|\mathcal{C}|} \sum_{\underline{s}^{\text{in}} \in \mathcal{C}} \frac{1}{2}\big(1 - \mathbb{E}_{\underline{h}|\underline{s}^{\text{in}}}\big[s_i^{\text{in}} \text{sign}(\langle s_i \rangle)\big]\big). \tag{3.13}$$

The BEC, BSC and BAWGNC have a special symmetry property which allows to simplify this expression. In the next section we show that for a general class of *symmetric channels* the terms in the sum (3.13) are independent of the input word (see Equ. (3.20)). For such channels there is no loss in generality to assume that the transmitted word is $s_i^{\text{in}} = 1$, $i = 1, \ldots, n$, or $\underline{x} = 0$ the "all-zero codeword". To simplify the notations we set $c(\underline{h}|\underline{1}) = c(\underline{h})$ and $\mathbb{E}_{\underline{h}|\underline{1}^{\text{in}}} = \mathbb{E}_{\underline{h}}$. For symmetric channels the average bit error probability is given by

$$\mathbb{P}_b[\text{error}] = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2}\big(1 - \mathbb{E}_{\underline{h}}\big[\text{sign}(\langle s_i \rangle)\big]\big). \tag{3.14}$$

Interpolating between bit-MAP and MAP decoders

What is the Hamiltonian corresponding to distribution (3.9)? To answer this question it is enough rewrite this expression as $e^{-\beta \mathcal{H}(\underline{s})}/Z_\beta$. If we set $\beta = 1$ we have[2]

$$\mathcal{H}(\underline{s}) = \sum_{a=1}^{m} \frac{1}{2}(1 - \prod_{i \in \partial a} s_i) - \sum_{i=1}^{n} h_i s_i \tag{3.15}$$

So the posterior distribution used in bit-wise MAP decoding can be though as a Gibbs distribution with inverse temperature set to the special value $\beta = 1$.

From this point of view it is natural to try other decoders based on the Gibbs distribution for arbitrary values of the inverse temperature parameter,

$$p_\beta(\underline{s}|\underline{h}) = \frac{1}{Z_\beta} e^{-\beta \mathcal{H}(\underline{s})} = \frac{1}{Z_\beta} \prod_{a=1}^{m} \frac{1}{2}(1 + \prod_{i \in \partial a} s_i) \prod_{i=1}^{n} e^{\beta h_i s_i} \tag{3.16}$$

with the partition function Z_β the sum over all $\underline{s} \in \{-1, +1\}^n$ of the numerator. The *general temperature decoder* is defined as

$$\hat{s}_i(\underline{h}; \beta) = \text{argmax}\, p_\beta(s_i|\underline{h}) = \text{sgn}\langle s_i \rangle_\beta \tag{3.17}$$

[2] Setting β to a different value would amount to scale the Hamiltonian by the inverse of that value.

where the bracket $\langle - \rangle_\beta$ is the average with respect to (3.16). Obviously $\beta = 1$ this is the bit-wise MAP decoder. Taking the limit $\beta \to +\infty$ it is not difficult to see that $\mathrm{sgn}\langle s_i \rangle_\beta \to \mathrm{argmin}\, \mathcal{H}(\underline{s})$. This also equals $\mathrm{argmax}\, p(\underline{s}|\underline{h})$, thus in the zero temperature limit we recover the block MAP decoder. For $1 \leq \beta \leq +\infty$ the general temperature decoder interpolates between the bit-wise and block MAP decoders.

3.2 Channel symmetry and gauge transformations

A binary input channel is said to be *symmetric* when the transition probability satisfies $p(y_i|s_i) = p(-y_i|-s_i)$. Using (3.7) and (**??**) one shows that this is equivalent to $c(h_i|s_i) = p(-h_i| - s_i)$. We show below that without loss of generality one can assume $s_i^{\mathrm{in}} = 1$, so it is useful to also notice that

$$c(-h_i) = c(h_i) e^{-2h_i} \tag{3.18}$$

EXAMPLE 10 For the BEC, BSC, BAWGNC we check explicitly that $p(y_i|s_i) = p(-y_i| - s_i)$. One also computes $c(h_i) = c(h_i|1)$ from (3.7) and (**??**) and finds

$$\begin{aligned}
c(h) &= (1-\epsilon)\delta_{+\infty}(h) + \epsilon\delta(h), && \mathrm{BEC}(\epsilon) \\
c(h) &= (1-p)\delta(h - \ln\frac{1-p}{p}) + p\delta(h - \ln\frac{p}{1-p}), && \mathrm{BSC}(p) \\
c(h) &= \frac{1}{\sqrt{2\pi\sigma^{-2}}} e^{-(h-\frac{1}{\sigma^2})^2/\frac{2}{\sigma^2}}, && \mathrm{BAWGNC}(\sigma^2)
\end{aligned}$$

The identity (3.18) is explicit on these expressions.

As a first application of channel symmetry let us prove (3.14). Consider first $\mathbb{E}_{\underline{h}|\underline{s}^{\mathrm{in}}}\left[s_i^{\mathrm{in}} \mathrm{sign}(\langle s_i \rangle)\right]$. The expectation $\mathbb{E}_{\underline{h}|\underline{s}^{\mathrm{in}}}$ is an integral over h_i's and the bracket $\langle - \rangle$ contains sums (in a numerator and denominator) over s_i's. In the inetgarls and sums we may perform the change of variables

$$s_i \to \tau_i s_i, h_i \to \tau_i s_i, \qquad i = 1, \ldots, n \tag{3.19}$$

for a *code word* $\underline{\tau} \in \mathcal{C}$. Now we note two crucial facts. First, under this transformation the posterior (3.9) remains *invariant*, and therefore $\langle s_i \rangle \to \tau_i \langle s_i \rangle$, where $\langle - \rangle$ is the *same* expectation on both sides of the equality. Second, because of channel symmetry $\mathbb{E}_{\tau_i h_i | s^{\mathrm{in}}} = \mathbb{E}_{h_i | \tau_i s^{\mathrm{in}}}$. Thus

$$\mathbb{E}_{\underline{h}|\underline{s}^{\mathrm{in}}}\left[s_i^{\mathrm{in}} \mathrm{sign}(\langle s_i \rangle)\right] = \mathbb{E}_{\underline{h}|\underline{\tau}\star\underline{s}^{\mathrm{in}}}\left[\tau_i s_i^{\mathrm{in}} \mathrm{sign}(\langle s_i \rangle)\right] \tag{3.20}$$

where we find it convenient to use $\underline{v} \star \underline{u}$ for a vector with components $v_i u_i$, $i = 1, \ldots, n$. Now, since the code is linear $\underline{\tau} \star \underline{s}^{\mathrm{in}}$ is also a code word, and therefore the sum over $\underline{s}^{\mathrm{in}}$ is independent of τ. This proves (3.14).

The idea of using a transformation such as $s_i \to \tau_i$, $h_i \to \tau_i s_i$ with $\underline{\tau}$ a code word, turns out to be very useful in the present framework. Since codewords $\underline{\tau} \in \mathcal{C}$ form a group, the set of such transformations also forms a group. Moreover

these transformations are local in the sense that for each i the variables get multipied by different factors. Transformations with these two properties are called *gauge transformations*. The invariance of the Gibbs distribution under such transformations together with channel symmetry allows to derive a number of useful consequences and identities. We will have the occasion to derive them as we proceed with the theory. The independence of the error probability on the transmitted codeword is one of them.

It is important to note that the invariance of the Gibbs distribution under gauge transformations is a consequence of the linearity of the code. For non-linear codes such an invariance would typically not be present. Also, for the random K-SAT problem where the constraints are "non-linear" we do have (or know) any useful gauge transformations. This is one of the reasons why this problem is a much harder one.

3.3 Conditional entropy and free energy in coding

Without loss of generality we assume from now on that the all all-zero codeword is transmitted. We recall the equivalent notation $\mathbb{E}_{\underline{Y}|\underline{1}} = \mathbb{E}_{\underline{Y}}$, $\mathbb{E}_{\underline{h}|\underline{1}} = \mathbb{E}_{\underline{h}}$.

We explained in Chapter 2 that a lot can be learned from the free energy $-\frac{1}{n}\ln Z$ (recall here we have $\beta = 1$). For example differentiating with respect to h_i yields the magnetization $\langle s_i \rangle$ (see Equ. (3.12)). For spin glass models the free energy is random but usually concentrates in the thermodynamic limit $n \to +\infty$. in the thermodynamic limit and, although this can be non-trivial, we do have examples where this can be proven. Such proof techniques will be studied in Chapter 13. We therefore consider the *average free energy* $-\frac{1}{n}\mathbb{E}_{\underline{h}}[\ln Z]$. We will now show an important relation to the conditional entropy $H(\underline{X}|\underline{Y})$, i.e. the average entropy of the posterior $p(\underline{s}|\underline{y})$,

$$H(\underline{X}|\underline{Y}) = -\mathbb{E}_{\underline{Y}}\left[\sum_{\underline{s}} p(\underline{s}|\underline{y}) \ln p(\underline{s}|\underline{y})\right] \tag{3.21}$$

This relation shows that computing the average free energy or the conditional entropy is basically equivalent. In part III we will develop powerful methods to compute the free energy. This will automatically allow us to compute the conditional entropy and in particular the MAP threshold.

For transmission over a symmetric channel and any fixed linear code (not necessarily an LDPC code) we have

$$\frac{1}{n}H(\underline{X}|\underline{Y}) = \frac{1}{n}\mathbb{E}_{\underline{h}}[\ln Z] - \int_{-\infty}^{+\infty} dh\, c(h) h. \tag{3.22}$$

Observe that the last term in (3.43) depends only on the channel. For the BSC it is equal to $(1-2p)\ln\frac{1-p}{p}$ and for the BAWGNC $1/\sigma^2$. For the BEC there is a little ambiguity here. Formally $\int_{-\infty}^{+\infty} dh\, c(h) h$ is infinite, but this infinity is

cancelled with another infinity in $\ln Z$. Indeed the weight factors $e^{h_i s_i}$ in Z diverge when $s_i = 1$ and $h_i = +\infty$. However we can redefine the partition function replacing $e^{h_i s_i}$ by $e^{h_i s_i - h_i}$, so that the new Z is finite and the last term in (3.43) is not present. This should in principle be done for any channel having a non-zero weight on $h_i = +\infty$, but is not real problem.

The proof of this relation will be a good occasion to illustrate once a again the use of gauge transformations and channel symmetry. Replacing (3.9) in (3.21)

$$\begin{aligned} H(\underline{X}|\underline{Y}) &= \mathbb{E}_{\underline{Y}}[\ln Z(\underline{y})] - \mathbb{E}_{\underline{Y}}\Big[\sum_{\underline{s}} p(\underline{s}|\underline{y}) \ln \prod_{c\in\mathcal{C}} \frac{1}{2}(1+\prod_{i\in c} s_i)\Big] \\ &\quad - \mathbb{E}_{\underline{Y}}\Big[\sum_{\underline{s}} p(\underline{s}|\underline{y}) \sum_{i=1}^n h_i s_i\Big] \\ &= \mathbb{E}_{\underline{h}}[\ln Z] - \sum_{i=1}^n \mathbb{E}_{\underline{h}}[h_i \langle s_i\rangle] \end{aligned} \tag{3.23}$$

To get the last equality we noticed that the second expectation vanishes because $p(\underline{s}|\underline{y})$ is supported on code words and $\ln 1 = 0$. Finally we replaced $\mathbb{E}_{\underline{Y}}$ by $\mathbb{E}_{\underline{h}}$. It remains to show the identity

$$\mathbb{E}_{\underline{h}}[h_i \langle s_i\rangle] = \mathbb{E}_{\underline{h}}[h_i] \tag{3.24}$$

This is part of a whole class of relationships, called Nishimori identities, which follow from gauge invariance and channel symmetry. We will encounter a number of them in susequent chapters. Using a gauge transformation $s_i \to \tau_i s_i$, $h_i \to \tau_i h_i$ and the channel symmetry in the form $c(\tau_i h_i) = c(h_i) e^{h_i \tau_i - h_i}$ we have

$$\begin{aligned} \mathbb{E}_{\underline{h}}[h_i \langle s_i\rangle] &= \mathbb{E}_{\underline{\tau}\star\underline{h}}[h_i \langle s_i\rangle] \\ &= \mathbb{E}_{\underline{h}}[h_i \langle s_i\rangle \prod_{j=1}^n e^{h_j \tau_j - h_j}] \end{aligned} \tag{3.25}$$

Summing over all code words $\underline{\tau} \in \mathcal{C}$,

$$\begin{aligned} \mathbb{E}_{\underline{h}}[h_i \langle s_i\rangle] = \frac{1}{|\mathcal{C}|} &= \frac{1}{|\mathcal{C}|} \mathbb{E}_{\underline{h}}[Z h_i \langle s_i\rangle \prod_{j=1}^n e^{-h_j}] \\ &= \frac{1}{|\mathcal{C}|} \mathbb{E}_{\underline{h}}[h_i \sum_{\underline{s}} s_i \prod_{c=1}^m \frac{1}{2}(1+\prod_{i\in\partial c} s_i) \prod_{j=1}^n e^{h_j s_j - h_j}] \\ &= \frac{1}{|\mathcal{C}|} \sum_{\underline{s}} s_i \prod_{c=1}^m \frac{1}{2}(1+\prod_{i\in\partial c} s_i) \mathbb{E}_{\underline{h}}[h_i \prod_{j=1}^n e^{h_j s_j - h_j}] \\ &= \frac{1}{|\mathcal{C}|} \sum_{\underline{s}} s_i \prod_{c=1}^m \frac{1}{2}(1+\prod_{i\in\partial c} s_i) \mathbb{E}_{\underline{h}}[h_i e^{h_i s_i - h_i}] \prod_{j\neq i} \mathbb{E}_{\underline{h}}[h_i \prod_{j=1}^n e^{h_j s_j - h_j}] \end{aligned} \tag{3.26}$$

The result then follows from the two identities

$$\mathbb{E}_{\underline{h}}[e^{h_i s_i - h_j}] = 1, \qquad \mathbb{E}_{\underline{h}}[h_i e^{h_i s_i - h_i}] = s_i \tag{3.27}$$

because $\sum_{\underline{s}} s_i \prod_{c=1}^{m} \frac{1}{2}(1 + \prod_{i \in \partial c} s_i) = |\mathcal{C}|$. These two identities simply amount to the normalization of $c(h)$ when $s_i = 1$. When $s_i = -1$ it is elementary to see that they follow from $c(-h_i) = c(h_i)e^{-2h_i}$.

3.4 Compressive Sensing as a spin glass model

Recall that we are considering the model

$$\underline{y} = A\underline{x} + \underline{z}, \tag{3.28}$$

where the measurement matrix A is an $m \times n$ real valued matrix with iid zero mean Gaussian entries with variance $1/m$, the noise $\underline{z}$ consists of m iid zero-mean Gaussian entries of variance σ^2, and where the signal $\underline{x}$ consists also of n iid entries distributed with the prior $p_0(x)$. We will assume this prior belongs to the *sparse* class, $p_0 \in \mathcal{F}_\kappa$, that is

$$p_0(x) = (1 - \kappa)\delta(x) + \kappa\phi_0(x) \tag{3.29}$$

where ϕ_0 is a continuous positive and normalized density. So the expected number of non-zero entries in the signal is $k = \kappa n$.

The conditional probability of observing y given x is

$$p(\underline{y} \mid \underline{x}) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} e^{-\frac{1}{2\sigma^2}\|\underline{y} - A\underline{x}\|_2^2}, \tag{3.30}$$

and the joint distribution, taking the prior into account, has the form

$$p(\underline{x}, \underline{y}) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} e^{-\frac{1}{2\sigma^2}\|\underline{y} - A\underline{x}\|_2^2} \prod_{i=1}^{n} p_0(x_i). \tag{3.31}$$

We discuss two scenarios. In the first one *the prior is known* (so here $\phi_0(x)$ is known) and in the second scenario which is more realistic *the prior is not known* and one only knows that it belongs to $\mathcal{F}_\kappa$. In other words κ is assumed to be known but not ϕ_0.

Known prior: MMSE estimator

When the prior is known a reasonable way to estimate the signal is to use the Minimum Mean Square Estimator (MMSE). This estimator is optimal in the sense that it minimizes the Mean Square Error (MSE). The MSE is the functional over the space of estimators $\hat{\underline{x}}(\underline{y}) : \mathbb{R}^r \to \mathbb{R}^n$

$$\mathrm{MSE}[\hat{\underline{x}}] = \mathbb{E}[(\hat{\underline{x}}(\underline{Y}) - \underline{X})^2] \tag{3.32}$$

Here the expectation is with respect to the joint distribution (3.31) and the iid Gaussian entries of A. A standard exercise shows that the minimum is attianed by the MMSE,

$$\hat{x}_i(\underline{y}) = \mathbb{E}_{\underline{X}|\underline{y}}[X] = \int d^n\underline{x}\, x_i\, p(\underline{x} \mid \underline{y}), \qquad i = 1, \dots, n. \tag{3.33}$$

In this expression $p(\underline{x}|\underline{y})$ is the posterior distribution associated to (3.31), and we have adopted the notation $d^n\underline{x} = \prod_{i=1}^n dx_i$. Analogously to the case of coding, we will interpret the posterior as a Gibbs distribution and the MMSE as a "magnetization".

Unknown prior: LASSO estimator

We will almost exclusively concentrate on this situation which is more realistic. A popular choice for the estimator is the LASSO, (**??**)

$$\underline{\hat{x}}_1(\underline{y}) = \mathrm{argmin}_{\underline{x}} \left\{ \frac{1}{2}\|\underline{y} - A\underline{x}\|_2^2 + \lambda\|\underline{x}\|_1 \right\}. \tag{3.34}$$

where the real parameter λ has to be chosen suitably. Since the prior is unknown it is natural to choose the best possible λ for the worse possible prior. Formally we solve a minimax problem,

$$\inf_{\lambda\in\mathbb{R}} \sup_{p_0\in\mathcal{F}_\kappa} \frac{1}{n}\mathbb{E}[(\underline{\hat{x}}_1(\underline{y}) - \underline{x})^2] \tag{3.35}$$

The expectation is again here over the joint distribution (3.31) and the random matrix ensemble. Solving teh minimax problem amounts to find the best possible parameter λ when the signal distribution $p_0(x)$ is the worst possible. The value given by (3.35) is sometimes called teh LASSO minimax risk and will constitute our performance measure.

As explained in Chapter 1 it is not so easy to unambiguously justify a priori the choice of this estimator. We will be able to solve exactly this problem in Chapter **??** and we will find that the minimax-MSE is finite in the same region of parameters for which l_1-l_0 equivalence holds. In the region were l_1-l_0 equivalence does not hold the minimax-MSE diverges. In this sense LASSO is as good as pure l_1 minimization for the noiseless problem, and this justifies the use of Lasso a posteriori. We will shortly give a different, somewhat more phenomenological, justification which does not require to develop the whole theory. We will see that the Lasso estimator can also be considered as a zero temperature limit of a "finite temeprature MMSE" with a Laplacian prior modelling the unknown distribution p_0.

MMSE and LASSO as spin glass models

The posterior entering in the MMSE estimator (3.33) is derived from 3.31,

$$p(\underline{x} \mid \underline{y}) = \frac{1}{Z} \prod_{a=1}^{m} e^{-\frac{1}{2\sigma^2}(y_a - A_a^T \underline{x})^2} \prod_{i=1}^{n} p_0(x_i), \tag{3.36}$$

where y_a, $a = 1, \ldots, m$ are the components of $\underline{y}$ and A_a is the column vector equal to the a-th row of the matrix A. Thus $A_a^T \underline{x} = \sum_{i=1}^{n} A_{ai} x_i$. The explicit expression of the normalisation factor is

$$Z = \int d^n \underline{x} \prod_{a=1}^{m} e^{-\frac{1}{2\sigma^2}(y_a - A_a^T \underline{x})^2} \prod_{i=1}^{n} p_0(x_i) \tag{3.37}$$

The interpretations in terms of spin-glass concepts are analogous to the case of coding. The posterior (3.36) can be though of as a random Gibbs distribution and (3.37) as a partion function. This time the "spin variables" $x_i \in \mathbb{R}$ belong to a continuous alphabet, and one often speaks of "continuous spins". The distribution is random because of the measurement matrix A and the observations $\underline{y}$. These are the quenched variables.

The MMSE estimator (3.33) is the average with respect to the Gibbs distribution and in statistical mechanics notation is written as the bracket $\langle x_i \rangle$. One can interpret it as a "magnetization" for the continuous spins. Note that in order to compute it all we need in principle is the marginal $p(x_i|\underline{y})$ given by integrating (3.36) over all spin variables except x_i. To sum up we have,

$$\hat{x}_i(\underline{y}) = \langle x_i \rangle = \int d^n \underline{x}\, x_i\, p(\underline{x} \mid \underline{y}) = \int dx_i x_i\, p(x_i \mid \underline{y}), \tag{3.38}$$

We saw in Chapter 2 that Gibbs distributions are of the form $e^{-\beta \mathcal{H}}/Z$ where $\mathcal{H}$ is a Hamiltonian. What are the Hamiltonian and the inverse temperatrure here? A natural answer to this question is to take $\beta = 1$ and

$$\mathcal{H}(\underline{x}) = \frac{1}{2\sigma^2} \sum_{a=1}^{m} (y_a - A_a^T \underline{x})^2 + \sum_{i=1}^{n} \ln p_0(x_i) \tag{3.39}$$

In coding where we discussed a 'finite temperature decoder" and noticed that it interpolates between the bit-MAP and block-MAP decoders. Once we have the Hamiltonian view it is immediate to do something similar here. Let

$$p_\beta(\underline{x}|\underline{y}) = \frac{1}{Z_\beta} e^{-\beta \mathcal{H}(\underline{x})} = \frac{1}{Z_\beta} \prod_{a=1}^{m} e^{-\frac{\beta}{2\sigma^2}(y_a - A_a^T \underline{x})^2} \prod_{i=1}^{n} (p_0(x_i))^\beta \tag{3.40}$$

with Z_β the correct normalization factor given by the integral over all x_i's of the numerator. We define a "finite temperature estimator" as the magnetization at inverse temperature β,

$$\hat{x}_{i,\beta}(\underline{y}) = \langle x_i \rangle_\beta = \int d\underline{x}\, x_i\, p_\beta(\underline{x} \mid \underline{y}) = \int dx_i x_i\, p_\beta(x_i \mid \underline{y}). \tag{3.41}$$

For $\beta = 1$ this simply the usual MMSE estimator. In the limit of zero temperature

$\beta \to +\infty$ the integral is concentrated on the spin configurations that minimize the Hamiltonian, in other words

$$\begin{aligned}\lim_{\beta\to+\infty} \hat{\underline{x}}_\beta(\underline{y}) &= \operatorname{argmin}_{\underline{x}} \mathcal{H}(\underline{x}) \\ &= \operatorname{argmin}_{\underline{x}} \big(\frac{1}{2\sigma^2} \|\underline{y} - A\underline{x}\|_2 + \lambda \sum_{i=1}^{n} \ln p_0(x_i) \big) \end{aligned} \tag{3.42}$$

This is analogous to the usual least square estimator but penalized by a term $\ln p_0(x)$ coming from the prior distribution.

Now we can see why the LASSO can be viewed as a zero temperature limit of a finite temperature MMSE. When the prior is unknown but it is only known that the signal is sparse the Laplacian prior $p_0(x) = e^{-\frac{\lambda}{\sigma^2}|x|}$ is a simple, and as it turns out, tractable model for the ensemble of possible priors. This ensemble is parametrized by a single parameter λ and its optimal value as a function of κ is determined from the minimax principle. In a sense, this point of view naturally leads to the AMP algorithm developed in Chapter 8.

3.5 Free energy and conditional entropy in compressive sensing

Assume that the prior is known and consider the Gibbs distribution associated to the MMSE estimator. There is a relation between the average free energy and conditionnal entropy that is perfectly analogous to the one for coding in section 3.3. Consider $-\mathbb{E}_{\underline{Y}}[\frac{1}{n} \ln Z]$ the average free energy where the average is only over $\underline{Y}$ and the measurement matrix is fixed. We have

$$H(\underline{X}|\underline{Y}) = \mathbb{E}_{\underline{Y}}[\ln Z(\underline{y}] + H(\underline{X}) + \frac{n}{2} \tag{3.43}$$

It is pleasing to see that the free energy is directly related to the the mutual information $H(\underline{X}) - H(\underline{X}|\underline{Y})$. Note also that $H(\underline{X}) = nH(X) = \kappa H(\phi_0(\cdot))$.

The derivation is easier than in coding and is a matter of simple algebra. By definition

$$H(\underline{X} \mid \underline{Y}) = -\mathbb{E}_{\underline{X},\underline{Y}}[\ln p(\underline{X} \mid \underline{Y})] \tag{3.44}$$

The logarithm of the posterior distribution is equal to

$$-\frac{1}{2\sigma^2} \|\underline{y} - A\underline{x}\|_2^2 + \sum_{i=1}^{n} \ln p(x_i) - \ln Z(\underline{y}) \tag{3.45}$$

The last term contributes $\mathbb{E}_{\underline{Y}}[\ln Z]$ to the conditional entropy (3.43). The contribution of the second term to (3.43) is also very easy to assess

$$-\mathbb{E}_{\underline{X},\underline{Y}}\left[\sum_{i=1}^{n} \ln p(X_i)\right] = -\sum_{i=1}^{n} \mathbb{E}_{\underline{X}}[\ln p(X_i)] = H(\underline{X}) \tag{3.46}$$

To derive the contribution of the first term it is convenient to write down explicitely the integrals,

$$\begin{aligned}\frac{1}{2\sigma^2}\int d\underline{x}\int d\underline{y}\, p(\underline{x},\underline{y})\|\underline{y}-A\underline{x}\|_2^2 \\ &= \frac{1}{2\sigma^2}\int\prod_{i=1}^{n} dx_i p_0(x_i)\int d\underline{y}\,\|y\|_2^2\frac{e^{-\frac{1}{2\sigma^2}\|\underline{y}\|_2^2}}{(2\pi\sigma^2)^{n/2}} \\ &= \frac{n}{2}\end{aligned} \tag{3.47}$$

The second line is obtained by a shift $\underline{y}\to\underline{y}+A\underline{x}$ in the $\underline{y}$-integral for each fixed $\underline{x}$.

3.6 K-SAT as a spin glass model

Recall the formulation of the random max-K-sat problem of Chapter 1. We take a formula at random from the ensemble $\mathcal{F}(n,K,M)$. The formula corresponds to a biparttite factor graph with dashed and full edges, see Fig. 1.6. As for coding and compressed sensing we adopt the notation that letters $i,j,k,\dots$ are variable nodes and $a,b,c,\dots$ are constraint nodes. In the max-K-sat problem we consider the number of violated clauses for an assignment $\underline{x}$, then we take the best possible assignment that minimizes the number of violated clauses and average over the random formulas,

$$e(\alpha)=\lim_{m\to+\infty}\frac{1}{m}\mathbb{E}\big[\min_{\underline{x}}\sum_{a=1}^{m}(1-\mathbb{1}_a(\underline{x}))\big]. \tag{3.48}$$

In Chapter 13 we study mathematical methods allowing the proof of existence of this limit.

The problem here is not directly formulated in terms of a Gibbs distribution, but a natural and fruitful idea is to one consider the Gibbs distribution associated to the cost function

$$\sum_{a=1}^{m}(1-\mathbb{1}_a(\underline{x})). \tag{3.49}$$

In particular, by studying the Gibbs distribution for very low temperatures we can get hold of $e(\alpha)$ and much more also.

Hamiltonian formulation

We will work in the spin language, so we set $s_i=(-1)^{x_i}$. Furthermore if clause c_a contains the literal x_i (resp. $\bar{x}_i$) we associate a weight $J_{ai}=+1$ (resp.$J_{ai}=-1$) to the edge ai of the factor graph. Thus, for example on Fig. 1.6 full edges have $J_{ai}=+1$ and dashed edges have $J_{ai}=+1$. Moreover the J_{ai} are bernoulli 1/2

random variables. With these convention we see that the i-th variable satisfies clause a when $s_i = -J_{ai}$ and does not satisfy it when $s_i = -J_{ai}$. Therefore

$$\mathbb{1}_a(\underline{x}) = \prod_{i \in \partial a} \left(\frac{1 - s_i J_{ia}}{2}\right) \tag{3.50}$$

and the cost function, also called the Hamiltonian of K-sat, takes the form

$$\mathcal{H}(\underline{s}) = \sum_{a=1}^{m} \prod_{i \in \partial a} \left(\frac{1 + s_i J_{ia}}{2}\right) \tag{3.51}$$

By expanding the product in each term we see that this Hamiltonian involves "multispin interactions" of the form (2.3). This Hamiltonian is random in the sense that the underlying factor graph is random, and this randomness is frozen because once the formula has been chosen from the ensemble it is fixed. This is a *spin-glass Hamiltonian*. Of course we have

$$e(\alpha) = \lim_{m \to +\infty} \frac{1}{m} \mathbb{E}\big[\min_{\underline{s}} \mathcal{H}(\underline{s})\big]. \tag{3.52}$$

The spin assignments that minimize the Hamiltonian (3.51) are often called "ground states" and one of the problems that will be discussed in later chapters will be to understand their geometric organization in the "Hamming space" $\{-1, +1\}^n$. Ground states with zero energy (zero cost) are solutions of the K-sat formula. An important problem is to count them. This amounts to evaluate

$$\mathcal{N}_0 = \sum_{\underline{s}} \prod_{a=1}^{m} (1 - \prod_{i \in \partial a} \left(\frac{1 + s_i J_{ia}}{2}\right)) \tag{3.53}$$

We will also see that it is often useful to take a larger view and count the number of spin assignment of energy (or cost) E,

$$\mathcal{N}_E = \sum_{\underline{s}} \mathbb{1}(\mathcal{H}(\underline{s}) = E) \prod_{a=1}^{m} (1 - \prod_{i \in \partial a} \left(\frac{1 + s_i J_{ia}}{2}\right)) \tag{3.54}$$

Finite temperature formulation

The set of solutions of a K-sat formula, equivalently the set of ground states, is not easy to determine. One way to approach this problem would be to sampel from this space at random thanks to a simple distribution. The simplest distribution one could imagine is the uniform one over solutions, so formally $\mathbb{1}(\mathcal{H}(\underline{s}) = 0)/\mathcal{N}_0$. We immediately face a proble here because some formulas from $\mathcal{F}(n, K, M)$ will not have any solution (and for high enough α this happens with overwhelming probability when n is large) so the uniform distribution is not well defined.

From the point of view of statistical mechanics there is a very natural regularisation of the uniform distribution. Namely one takes the Gibbs distribution

at finite inverse temperature $\beta < +\infty$,

$$p(\underline{s}) = \frac{1}{Z} e^{-\beta \mathcal{H}(\underline{s})} = \frac{1}{Z} \prod_{a=1}^{m} e^{-\beta \prod_{i \in \partial a} \left(\frac{1+s_i J_{ia}}{2} \right)} \tag{3.55}$$

with the partition function

$$Z = \sum_{\underline{s}} \prod_{a=1}^{m} e^{-\beta \prod_{i \in \partial a} \left(\frac{1+s_i J_{ia}}{2} \right)} \tag{3.56}$$

In the zero temperature limit $\lim_{\beta \to +\infty} Z = \mathcal{N}_0$ and formally $p(\underline{s}) \to \mathbb{1}(\mathcal{H}(\underline{s}) = 0)/\mathcal{N}_0$.

From the average free energy $F(\beta) = -\frac{1}{\beta} \mathbb{E}[\ln Z]$ at finite temperature, we can recover the average ground state energy per clause,

$$e(\alpha) = \lim_{m \to +\infty} \lim_{\beta \to +\infty} \frac{1}{m} \mathbb{E}[F(\beta)]. \tag{3.57}$$

To see this we simply note that $\frac{1}{\beta} |\ln Z| \leq C$ uniformly with respect to β, thus by dominated convergence

$$\begin{aligned} \lim_{\beta \to +\infty} \mathbb{E}[F(\beta)] &= -\mathbb{E}[\lim_{\beta \to +\infty} \frac{1}{\beta} \ln Z] \\ &= \mathbb{E}[\min_{\underline{s}} \mathcal{H}(\underline{s})] \end{aligned} \tag{3.58}$$

Recall also that from formula (**??**) we get the Gibbs entropy as a function of the inverse temperature. Here we define a "ground state entropy" per variable by taking the zero temperature limit (assuming the limit exists)

$$s(\alpha) = \lim_{n \to +\infty} \lim_{\beta \to +\infty} \frac{1}{n} \mathbb{E}[\frac{d}{d(1/\beta)} F(\beta)]. \tag{3.59}$$

The ground state entropy is nothing else than the growth rate of the number of solutions in the sat phase,

$$s(\alpha) = \begin{cases} \lim_{n \to +\infty} \frac{1}{n} \mathbb{E}[\ln \mathcal{N}_0], & \alpha < \alpha_s(K), \\ 0, & \alpha > \alpha_s(K). \end{cases} \tag{3.60}$$

3.7 Notes

The prototypical Gauge symmetry of physics is an invariance of the Maxwell equations under a group of local transformations. Gauge symmetry is a fundamental principle underlying all known fundamental forces.

Problems

3.1 Nishimori identities for coding. Use the technique of gauge transformations to prove the identities $[\langle s_i \rangle^{2p-1}] = [\langle s_i \rangle^{2p}]$ for all integers $p \geq 1$.

3.2 Special identities for a Gaussian channel. In the case of a BAWGNC identity (**??**) specializes to $\mathbb{E}_{\underline{Y}}[h_i\langle s_i\rangle] = \sigma^{-2}$. We want to explore a proof that is special to this channel.

(i) First check by explicit calculation that $\sigma^2 c(h)h = -\frac{\partial}{\partial h}c(h) + c(h)$.

(ii) Then use integration by parts and the Nishimori identity of the previous exercise (for $p = 1$) to derive $\mathbb{E}_{\underline{Y}}[h_i\langle s_i\rangle] = \sigma^{-2}$.

3.3 Derivation of the MMSE. Consider the MSE functional (3.32) and show that it is minimized by the MMSE (3.33).

3.4 LASSO for the scalar case. Let $y = x + z$ where z is a Gaussian scalar variable with zero mean and variance σ^2. Compute explicitly the LASSO estimator $\hat{x}(y) = \mathrm{argmin}_x(\frac{1}{2}(y-x)^2 + \lambda|x|)$. The result is called the "soft thresholding estimator".

3.5 Crude upper bound on the sat-unsat threshold α_s Below $\mathbb{P}$ and $\mathbb{E}$ are with respect to the random ensemble $\mathcal{F}(n, K, M)$. Consider the partition function Z of the microcanonical ensemble.

(i) Show the Markov inequality $\mathbb{P}[F \text{ satisfiable}] \leq \mathbb{E}[Z]$.

(ii) Show that $\mathbb{E}[Z] = 2^n(1 - 2^{-K})^M$.

(iii) Deduce the upper bound $\alpha_s < (\ln 2)/\ln(1 - 2^{-K})$. For $K = 3$ this yields $\alpha_s(3) < 5.191$. It is conjectured that $\alpha_s(3) \approx 4.26$: this value is the prediction of the highly sophisticated cavity method of spin glass theory. The asymptotic behavior of this simple upper bound for $K \to +\infty$ is $2^K \ln 2$, which is known to be tight. However, the large K corrections obtained by this bound are not tight.

4 Curie-Weiss Model

Before we start analysing our three running examples, it is instructive to consider a very simple model for which the analysis can be carried out explicitly with fairly little effort. This way we will encounter many concepts in their simplest incarnation. This separates the concepts and notions, and why they are important, from the computational difficulties which we will encounter when we carry out the same analysis for our problems.

We will consider the *Curie-Weiss* model. This is a specific version of the so-called *Ising* model and it is defined on a *complete graph*. This model is admittedly special, but it has two advantages. First, it has an explicit solution. Secondly, and equally important, it still displays many of the interesting features of more complicated models such as variational expressions for the free energy, fixed point equations, and phase transitions.

A second exactly solvable model is the Ising model on a *tree*. This is the subject of the problems. You will see that the solution of the Ising model on the tree can be phrased in terms of *message passing* quantities, another of our favourite themes.

Analogous, but more complicated solutions occur in coding, compressive sensing and K-SAT. It is natural that the solutions of these models share common features with the ones of the Curie-Weiss and Ising model on a tree, because these models are defined on locally tree like graphs (coding and K-SAT) or complete graphs (compressed sensing). However the situation is also considerably more complicated and interesting. One of the reasons is that in coding and K-SAT the graphs are locally tree like but have loops. One other reason is that the Gibbs distributions are random, i.e. the models are non-trivial spin glasses.

We introduced the standard Ising model on a regular grid $\mathbb{Z}^d$ in Chapter 2. This model is not only of considerable historical value for the development of statistical mechanics, but its study has led to many of the fundamental concepts in the theory of phase transitions, and it is still the subject of fascinating mathematical investigations. Models with a low dimensional regular underlying graph have geometrical features that are absent in our three running examples, and their solutions and the mathematical methods of analysis do not quite share similar features (although some aspects are still similar). Nevertheless there is some value in reviewing a few basic properties of the Ising model on $\mathbb{Z}^d$, and this is briefly done in section for completeness in (**??**). One concept that turns out to be

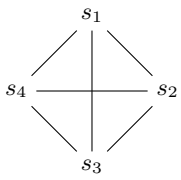


Figure 4.1 A complete graph with 4 nodes.

quite important in more advanced topics such as the cavity method in Chapter 15, is the notion of *pure state* or *extremal measure*. Let us also point out that the Ising model on $\mathbb{Z}^d$ with $d \to +\infty$ becomes equivalent to the Curie-Weiss model and also to the Ising model on a tree with "infinite" vertex degree.

4.1 Curie-Weiss model

The Curie-Weiss model is an Ising spin system defined on a complete graph. A complete graph on a set V of n vertices, is a graph in which the set E of edges is constituted by *all* $n(n-1)/2$ pairs of nodes. An example is shown in Figure 4.1. The Hamiltonian of the Curie-Weiss model is

$$\mathcal{H}(\underline{s}) = -\frac{J}{n} \sum_{\{i,j\}\in E} s_i s_j - h \sum_{i\in V} s_i \tag{4.1}$$

where $J > 0$ (ferromagnetic case) and $h \in \mathbb{R}$. In the first sum $\langle i, j\rangle$ is an unordered pair so each edge is counted only once. Note that the interaction constant is scaled by n, i.e., we have the constant J/n in front of the first sum. With this scaling both terms in the Hamiltonian scale linearly in the system size: this necessary in order to have an interesting thermodynamic limit.

The Gibbs distribution has the form

$$p(\underline{s}) = \frac{1}{Z} e^{\frac{\beta J}{n} \sum_{\langle i,j\rangle\in E} s_i s_j + \beta h \sum_{i\in V} s_i} \tag{4.2}$$

with the partition function given by the sum over all spin configurations $\underline{s} \in \{-1, +1\}^n$

$$Z = \sum_{\underline{s}} e^{\frac{\beta J}{n} \sum_{\langle i,j\rangle\in E} s_i s_j + \beta h \sum_{i\in V} s_i} . \tag{4.3}$$

Recall from Chapter 2, $\beta = 1/k_B T$ where T is the temperature and k_B Boltzman's constant, so the behaviour of the Gibbs distribution depends on the (dimensionless) ratios $J/k_B T$ and $h/k_B T$. More precisely, what is important is the ratio $\mathcal{H}(\underline{s})/k_B T$ of the energy of a spin configuration compared to a "background" energy $k_B T$. For example, if we take $h = 0$ for simplicity, at high temperatures, $k_B T >> J$, we get an almost uniform measure, whereas in the low temperature case, $k_B T << J$, only configurations of minimum energy count. Not surprisingly, we will see that $k_B T \approx J$ is a regime of great interest.

We will first calculate the free energy and then the magnetization. This will allow us to study the singularities of these functions, i.e. the phase transitions displayed by the model.

4.2 Variational expression of the free energy

Recall that the free energy in the thermodynamic limit is given by

$$f(\beta J, \beta h) = -\lim_{n\to+\infty} \frac{1}{n\beta} \ln Z. \tag{4.4}$$

On a complete graph we have the identity,

$$\sum_{\{i,j\}\in E} s_i s_j = \frac{1}{2}\big(\sum_{i\in V} s_i\big)^2 - \frac{1}{2}n. \tag{4.5}$$

Introducing the "magnetisation of a spin configuration" $m_n(\underline{s}) = \frac{1}{n}\sum_{i\in V} s_i$, we can express the Hamiltonian as

$$\mathcal{H}(\underline{s}) = -n\big(\frac{J}{2}(m_n(\underline{s}))^2 + h m_n(\underline{s})\big) + \frac{J}{2}. \tag{4.6}$$

Thus

$$Z = e^{-\frac{\beta J}{2}} \sum_{\underline{s}} e^{n\beta\left(\frac{J}{2} m_n(\underline{s})^2 + h m_n(\underline{s})\right)}. \tag{4.7}$$

The partition function can be computed by first summing over all spin configurations with a fixed magnetization m_n and then by summing over all magnetizations $m_n = \{\frac{j}{n} | j = -n, -n+1, \ldots, n-1, n\}$. We get

$$Z = e^{-\frac{\beta J}{2}} \sum_{m_n} \mathcal{N}(m_n)\, e^{n\beta\left(\frac{J}{2} m_n^2 + h m_n\right)}. \tag{4.8}$$

where $\mathcal{N}(m_n)$ is the cardinality of the set $\{\underline{s} : \sum_{i=1}^n s_i = n m_n\}$. This is easily computed (see Example 3 in Chapter 2 for an analogous calculation). Given m_n, let n_+ and n_- be the number of positive and negative spins respectively. Since $n_+ + n_- = n$ and $n_+ - n_- = n m_n$ we have $n_+ = \frac{1+m_n}{2} n$ and therefore

$$\mathcal{N}(m_n) = \binom{n}{\frac{1+m_n}{2} n} \approx e^{n h_2(\frac{1+m_n}{2})}, \tag{4.9}$$

where $h_2(p) = -p\log_2 p - (1-p)\log_2(1-p)$ the binary entropy function. The last approximation is asymptotically exact for $n \to +\infty$ and is obtained using Stirling's formula. This leads to

$$Z \approx e^{-\frac{\beta J}{2}} \sum_{m_n} e^{n\beta\left(\frac{J}{2} m_n^2 + h m_n + \beta^{-1} h_2(\frac{1+m_n}{2})\right)}. \tag{4.10}$$

Recall that $m_n = \{\frac{j}{n} | j = -n, -n+1, \ldots, n-1, n\}$. So this is a Riemann sum which tends for $n \to +\infty$ to

$$Z \approx e^{-\frac{\beta J}{2}} n \int_{-1}^{+1} dm\, e^{n\beta\left(\frac{J}{2}m^2 + hm + \beta^{-1} h_2(\frac{1+m}{2})\right)}. \tag{4.11}$$

The integrand has the form $e^{-n\beta f(m)}$ thus for $n \to +\infty$ the integral can be evaluated by the Laplace method: the value is dominated by the contribution of a small neighborhood of that value of m where $f(m)$ takes on its minimum. Since for the free-energy computation we take the logarithm of Z, divide by n, and take the thermodynamic limit, we only need to determine the exponential behavior of the integral, and this is trivially given by the maximum value the exponent takes on. This gives us

$$\begin{aligned} f(\beta J, \beta h) &= \min_{-1 \le m \le 1} \left\{ -\left(\frac{J}{2}m^2 + hm\right) - \beta^{-1} h_2(\frac{1+m}{2}) \right\} \\ &\equiv \min_{-1 \le m \le 1} f(m). \end{aligned} \tag{4.12}$$

With a little bit more effort this formula can be converted into a theorem.

This formula is very important. It says that the free energy is given by the solution of a *variational* problem, i.e., as the solution of a minimization problem. The function $f(m)$ which is minimized has various names in the literature. Here we will call it the *free energy function*. We will see in this course that the free energies of the coding, compressive sensing and K-SAT problems are all given by such variational expressions involving (often complicated) free energy functions or functionals.

4.3 Average magnetization

We saw in Chapter 2 that the *magnetisation* in the thermodynamic limit is defined by the Gibbs average

$$\overline{m}(\beta J, \beta h) = \lim_{n \to +\infty} \langle \frac{1}{n} \sum_{i \in V} s_i \rangle \tag{4.13}$$

Note that by linearity of the Gibbs bracket and the symmetry of the model $\overline{m}(\beta J, \beta h) = \langle s_i \rangle$ for all $i \in V$.

We can compute the magnetisation by repeating the calculations of the previous section. Indeed, first note by definition of the Gibbs bracket

$$\langle \frac{1}{n} \sum_{i \in V} s_i \rangle = \frac{\sum_{\underline{s}} m_n(\underline{s}) e^{-\beta \mathcal{H}(\underline{s})}}{\sum_{\underline{s}} e^{-\beta \mathcal{H}(\underline{s})}} \tag{4.14}$$

We have already found the asymptotic behaviour of the denominator as $n \to +\infty$, namely formula (4.11). It is quite clear that the same arguments applied to the

numerator lead to the asymptotics

$$\langle \frac{1}{n} \sum_{i \in V} s_i \rangle \approx \frac{\int_{-1}^{+1} dm\, m\, e^{-n\beta f(m)}}{\int_{-1}^{+1} dm\, e^{-n\beta f(m)}} \tag{4.15}$$

Now assume that the free energy function $f(m)$ has a *unique* global minimum. Then applying the Laplace method to the numerator and denominator one finds

$$\overline{m}(\beta J, \beta h) = \text{argmin}_{-1 \leq m \leq 1} f(m). \tag{4.16}$$

In section 4.5 we will show that unicity of the global minimiser always holds for all $h \neq 0$. So in this case the magnetisation is unambiguously given by the minimiser of the free energy function.

On the other hand, for $h = 0$ the analysis in section 4.5 shows that, the global minimum is unique and given by $\overline{m}(\beta J, \beta h) = 0$ when $\beta J < 1$, but is doubly degenerate when $\beta J > 1$. In this second case if we would blindly apply the Laplace method with $h = 0$ we would find a weighted average over the two minimisers. However this does not yield the "physically correct" magnetization. In the present case, because $f(m) = f(-m)$ when $h = 0$, this weighted average vanishes, but we will now see that the physically correct result is far more interesting!

The correct definition of the magnetization for $h = 0$ is

$$\overline{m}_{\pm}(\beta J) = \lim_{h \to 0_{\pm}} \overline{m}(\beta J, \beta h) = \lim_{h \to 0_{\pm}} \lim_{n \to +\infty} \langle \frac{1}{n} \sum_{i \in V} s_i \rangle \tag{4.17}$$

In other words the correct way to proceed is to take the limit $h \to 0_{\pm}$ *after* the thermodynamic limit $n \to +\infty$. In that case when we apply the Laplace method in the calculation above, *only one* global minimum is selected. We will show in section 4.5 that for $\beta J < 1$ both limits vanish, but that for $\beta J > 1$ they do not vanish and are opposite (note that when the limits dont vanish they must be opposite because for $h = 0$ the free energy function is even $f(m) = f(-m)$). Thus $\overline{m}(\beta J, \beta h)$ has a jump discontinuity on the line $(\beta J > 1, h = 0)$. This is our first encounter of a phase transition, a theme on which we elaborate in the next section.

There is a good physical reason for the order of the limits in 4.17. In a macroscopic system there always remains a residual infinitesimal magnetic field $h = 0_{\pm}$. When the magnetisation is discontinuous for $h = 0_{\pm}$ (here this happens at low temperatures $\beta J > 1$) we call it a *spontaneous magnetization* and say that there is a *spontaneous symmetry breaking*. The magnetization and symmetry breaking are called "spontaneous" because physically we do not get to choose the orientation of the magnetization: the infinitesimal perturbations in the environment select an orientation.

We conclude this section with a very useful relationship between the free energy $f(\beta J, \beta h)$ and the magnetization $\overline{m}(\beta J, \beta h)$. As we mentioned in Chapter 2,

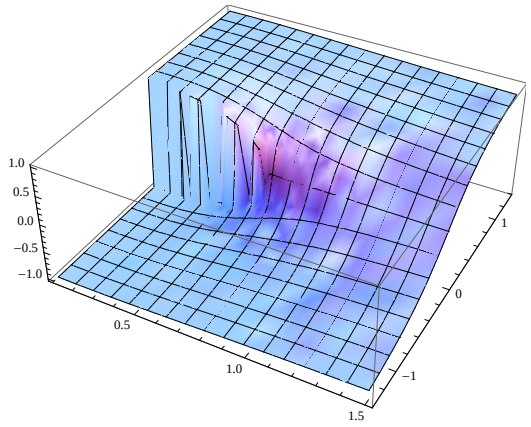


Figure 4.2 The behavior of $\overline{m}(\beta J, \beta h)$ as a function of $(1/(\beta J), \beta h)$, where $1/(\beta J) \in [0, 1.5]$ and $\beta h \in [-1.5, 1.5]$.

Gibbs averages can be obtained by differentiating the free energy, i.e., we have

$$\langle \frac{1}{n} \sum_{i=1}^{n} s_i \rangle = \frac{\partial}{\partial h} \frac{1}{n\beta} \ln Z_n. \tag{4.18}$$

Taking the limit $n \to +\infty$ one finds the important relation

$$\overline{m}(\beta J, \beta h) = -\frac{\partial}{\partial h} f(\beta J, \beta h). \tag{4.19}$$

The careful reader will notice that we have interchanged the limit $n \to +\infty$ and the partial derivative. We do not prove it here, but this is permitted except at phase transition points, i.e. except on the line $(\beta J \geq 1, h = 0)$.

4.4 Phase diagram and phase transitions

Consider the *free energy function* $f(m)$ and look at the minimiser $m(\beta J, \beta h)$. As already mentioned in the previous section for $h \neq 0$ this minimizer is unique and there is no ambiguity, so we think of this case. Instead of plotting $\overline{m}(\beta J, \beta h)$ as a function of $\beta J > 0$ and βh, we will plot $\overline{m}(\beta J, \beta h)$ as a function of $1/(\beta J) = k_B T/J$ (on the T-axis) and $\beta h = h/k_B T$ (on the h-axis).

Figure 4.2 shows the resulting plot. Why are we interested in this figure? As we discussed in the previous section this function represents the average magnetization, i.e., it represents a quantity describing the global behavior of the system as a function of the parameters. For some values of the parameters

$(\beta J, \beta h)$, the system behaves smoothly when we perturb the parameters. But for some other parameters the system behavior changes abruptly. These are so-called *phase transitions*.

A look at the figure already reveals two different forms of behavior. For parameters on the line segment $(0 < 1/(\beta J) < 1, h = 0)$, when we move along the h-axis, the magnetization $m(\beta J, \beta h)$ jumps. At the tip of this line segment $(1/(\beta J) = 1, h = 0)$ the magnetization is continuous but not differentiable. For example if we move along the T-axis or along the h-axis across the point $(1/(\beta J) = 1, h = 0)$, $m(\beta J, \beta h)$ changes in a continuous fashion, but its derivative (wrt to T or h) jumps. Finally, for all other points, $\overline{m}(\beta J, \beta h)$ changes smoothly and is in fact analytic (i.e., infinitely differentiable with an absolutely convergent Taylor expansion).

We call the first behavior a phase transition of *first order* and the second behaviour a phase transition of *second order*. To understand the terminology here, recall Equ. (4.19). At a first order transition the magnetization jumps and equivalently the first derivative of the free energy is discontinuous. At a second order phase transition the magnetization is continuous but its first derivative is discontinuous and equivalently the second derivative of the free nergy is discontinuous.

For a sligthly different perspective, let us replot Figure 4.2 but this time let us consider the picture "from the top," i.e., we only show the $1/(\beta J)$ and βh axis. This is shown in Figure 4.3. The different ways to change parameters leading to the various phase transitions are indicated. The segment indicated in blue, given by $(0 < 1/(\beta J) < 1, h = 0)$ is called the *co-existence line*. This name is easily explained. If we approach this line from the top or the bottom, i.e., we consider the limit $h \to 0_{\pm}$, then we get two opposite values $\pm\overline{m}_{\pm}(\beta J)$. So "on the line" we can think of having two possible "co-existing" phases. This line terminates terminates at the *critical point* $(\beta J = 1, h = 0)$ where the magnetization is continuous but not differentiable.

Going down one further dimension by fixing a value of $1/(\beta J) < 1$ and only varying h, or by fixing $h = 0$ and varying $1/(\beta J)$ across $\beta J = 1$, Figure 4.4 explicitly shows phase transitions of first and second order.

Let us sum up with a few general remarks about phase transitions.

The variational expression (4.12) of the free energy implies that it is a continuous and concave function of βJ and βh. In particular this means that the function itself does not jump, only its derivatives might. Here we have seen that two types of singularities occur in the phase diagram. The first derivative is discontinuous when the coexistence line is crossed, this is a first order phase transition. The second derivative is discontinuous when the critical point is crossed, this is a second order phase transition.

Continuity and concavity of the free energy is a general requirement in thermodynamics, and a general property of well behaved statistical mechanical models. Only the derivatives may have jumps. If the n-th derivative is discontinuous one speaks of a phase transition of order n. We point out there exist models with

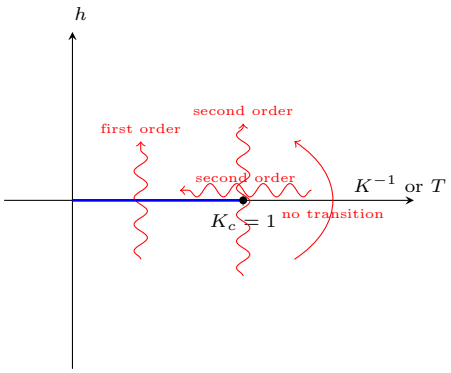
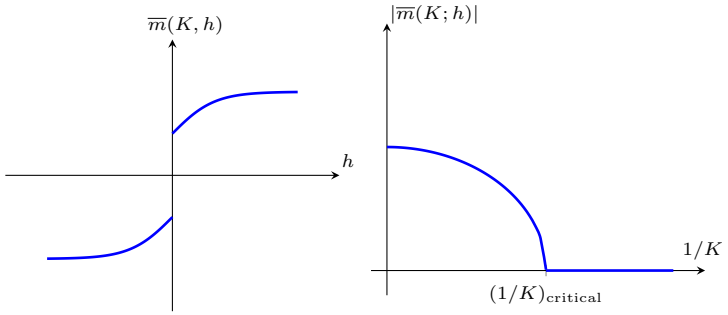


Figure 4.3 The blue line is called *coexistence line* because two thermodynamic phases (e.g. water/ice) coexist for parameters on it. Crossing the thick line is a first order phase transition. This line is terminated by the *critical point*. Crossing the critical point is a second order phase transition. There are many ways to cross it.



(a) First order (b) Second order

Figure 4.4 A phase transition of first and second order.

phase transitions of "infinite order" where the free energy is non-analytic but all its derivatives are continuous are known to exist. This classification of phase transitions due to Ehrenfest is not the only one. The more modern view point is to distinguish between continuous and discontinuous transitions and to classify them according to the type of symmetry change. These issues will not concern us in this course and ehrenfest's classification is good enough for our purposes.

Phase transitions related to singularities of the free energy are sometimes called "static" or "thermodynamic" phase transitions. We will encounter also other types of phase transitions that are called "dynamical" in the sense that they are related to a sudden change of the behaviour of algorithms but the free energy stays perfectly analytic.

4.5 Analysis of the fixed point equation

We have plotted the three-dimensional picture of $\overline{m}(\beta J, \beta h)$ and from this we can in principle see all phase transitions. But there is value in rederiving our conclusions in a more classical way by using calculus. By doing so, not only will we be able to add details to our picture, but we will also encounter some notions which will reappear throughout the course.

Curie-Weiss fixed point equation

Let us solve the variational problem (4.12) by differentiating the free energy function

$$f(m) \equiv -(\frac{J}{2}m^2 + hm) - \beta^{-1}h_2(\frac{1+m}{2}). \tag{4.20}$$

Explicitly $f'(m) = 0$ yields,

$$\beta(Jm + h) + \frac{1}{2}\ln\frac{(}{1-m}1 + m) = 0. \tag{4.21}$$

Using the identity

$$\tanh(\frac{1}{2}\ln\{\frac{1+m}{1-m}\}) = m, \tag{4.22}$$

we obtain the Curie-Weiss *fixed point equation*

$$m = \tanh(\beta(Jm + h)). \tag{4.23}$$

Of course this equation may have many solutions, and one has to select the ones which minimizes $f(m)$. If no solution is present then the minimum is attained at $m = \pm 1$. However this case does not concern us too much because it happens only for $\beta = +\infty$ ($T = 0$).

Equ. (4.23) is also called the *mean field equation*. Let us explain teh terminology here. Equation (4.23) expresses the magnetization as the one of an hypothetical single spin submitted to a magnetic field $Jm + h$. Indeed Hamiltonian of this single spin would be $-(Jm + h)s$ and its magnetization

$$m = \langle s\rangle = \frac{\sum_{s=\pm 1} se^{-\beta(Jm+h)s}}{\sum_{s=\pm 1} e^{-\beta(Jm+h)s}} = \tanh(\beta(Jm + h)) \tag{4.24}$$

One can think of $Jm + h$ as the effective average magnetic field felt by a each single spin on the complete graph.

This way of thinking is at the basis of the "mean field theory" of magnetism pioneered by Curie-Weiss and also at the basis of the generic "mean field approximations" for Ising spin systems. In the Curie-Weiss model it turns out that the mean field equation is exact. For Ising models on low dimensional regular grids such equations are not exact but often give a valuable first insight. However as briefly explained in section **??** they can also lead to qualitatively wrong predictions and care must be exercised. Even when mean field equations are "good"

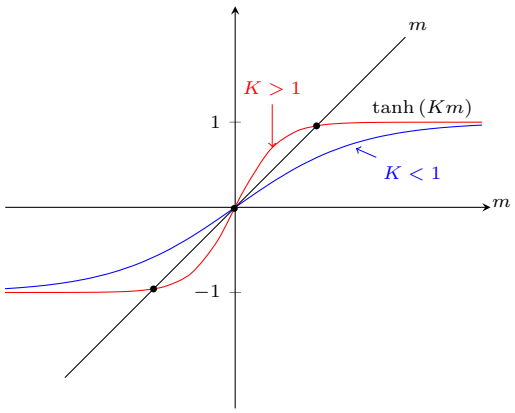


Figure 4.5 Curie-Weiss fixed points, $h = 0$

or exact it must not be thought that they are easy to derive. We will see that the solutions of our problems are intimately related to mean field equations but these are considerably more subtle to derive, let alone assess wheter they are exact or not.

Analysis of the Curie-Weiss equation and of the phase transitions

Now our task is to find solutions of the Curie-Weiss equation and select the ones that minimize $f(m)$. The solutions of (4.23) can be determined graphically. In the discussion below we distinghuish the cases $h = 0$, $h > 0$ and $h < 0$.

Case $h = 0$. The fixed points and free energy function $f(m)$ are shown in Figure 4.5 and Figure 4.6. In the "high temperature phase" $\beta J < 1$ there is a unique fixed point $\overline{m}(\beta J, 0) = 0$ and $\beta f(\beta J, 0) = \ln 2$. In the "low temperature phase" $\beta J > 1$ there are three fixed points $\{\overline{m}_-, 0, \overline{m}_+\}$ with $\overline{m}_\pm$ the global minimizers of $f(m)$ and $\overline{m} = 0$ a local maximum. As explained before, the magnetisation of a physical system will choose between two possible values $\overline{m}_-$ or $\overline{m}_+$ because there is always an infinitesimal $h = 0_\pm$ in the environnement. This is called "spontaneous symmetry breaking".

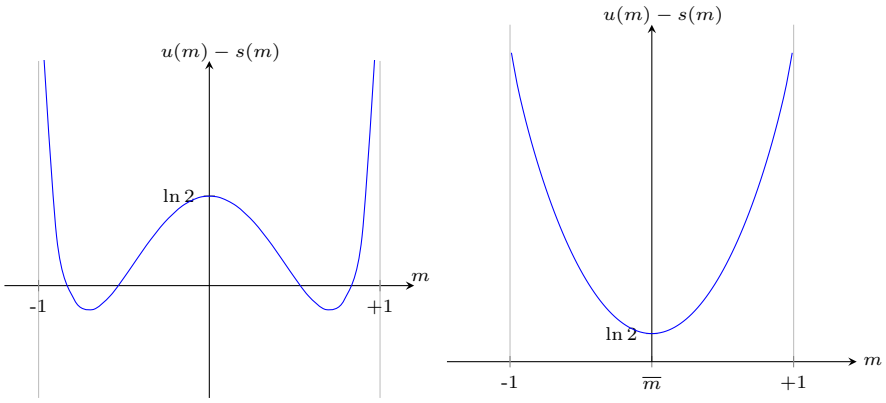
Let us look more closely at the behaviour of the magnetization for $h = 0$ as a function of $1/(\beta J)$ is shown in Figure 4.4. For βJ close to $\beta J = 1$ we can expand the Curie-Weiss equation around $\overline{m} = 0$,

$$m = \tanh \beta J m \approx \beta J m - \frac{(\beta J)^3}{3} m^3$$

Besides $\overline{m} = 0$ we have two other solutions

$$\overline{m}_\pm \sim \pm 3(\beta J - 1)^{1/2}$$

The exponent $1/2$ is called a *critical exponent*. Remarkably the critical exponent



(a) $K > 1$ (b) $K < 1$

Figure 4.6 Free energy functional

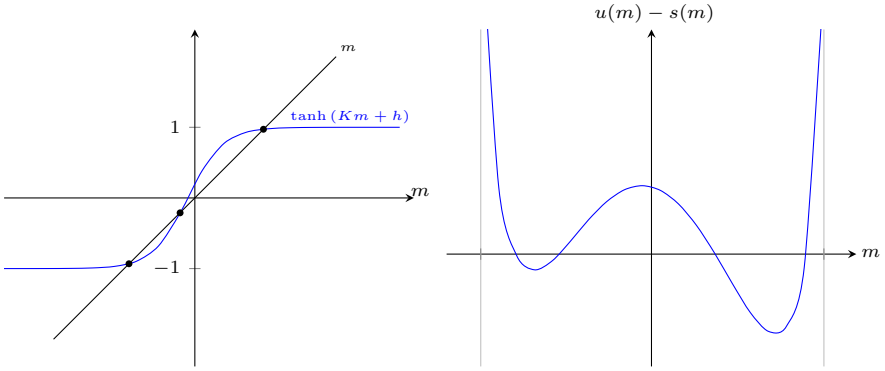
often does not depend on the details of the Hamiltonian but only on the dimensionality of the system (here $d = +\infty$), and the underlying symmetries of the Hamiltonian (here the Hamiltonian is invariant under $s_i \to -s_i$ for $h = 0$). For example in the exercises you will see that the ising model on a tree has the same critical exponent (in some sense the tree is an infinite dimensional graph). The magnetisation remains continuous but its derivative jumps. This means that the free energy has discontinuous second derivative and according to the Ehrenfest classification the transition is called second order. One also refers to such transitions as continuous transition because of the continuity of the magnetisation.

Cases $h > 0$ and $h < 0$. Fixed points and free energy function $f(m)$ are shown in Figures 4.7 and 4.8 for $h > 0$ (h not too large), $\beta J > 1$ and for $h > 0$, $\beta J < 1$. Note that there is always a unique global minimizer $\overline{m} > 0$. The situation for $h < 0$ is symmetric with a global minimizer $\overline{m} < 0$.

It is of interest to discuss what happens when h is infinitesimal, $h \to 0_{\pm}$. For $\beta J < 1$, $\overline{m}(\beta J, \beta h)$ is continuous and differentiable (even analytic) and there is *no* phase transition. For $\beta J > 1$, $\overline{m}(\beta J, \beta h)$ is discontinuous at $h = 0$. This is called a *discontinuous phase transition* or a *first order phase transition* (because the first derivative of the free energy jumps). See figure (4.4). At the critical point $(\beta J = 1, h = 0)$ the jump disappears and

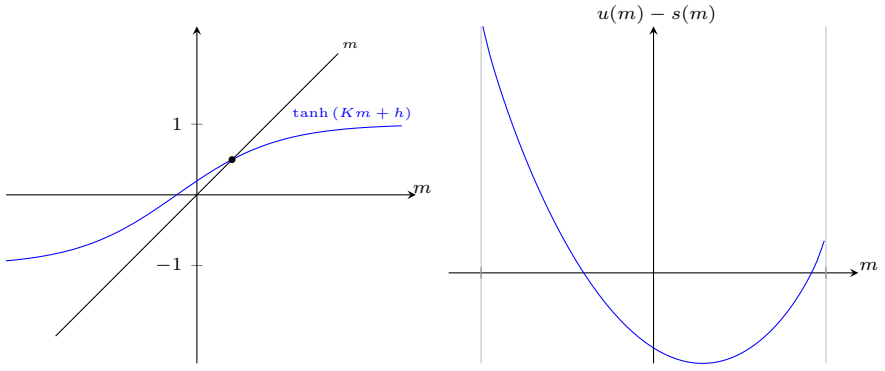
$$\overline{m}(\beta J = 1, h) \sim \pm |h|^{\frac{1}{3}}, \qquad h \to 0_{\pm} \tag{4.25}$$

This is again an example of second order phase transition thsi time with critical exponent $\frac{1}{3}$ (exercise: show this by expanding the Curie-Weiss equation for small h when $\beta J = 1$.)



(a) Fixed points (b) Free energy

Figure 4.7 Curie-Weiss fixed points, $h > 0, K > 1$



(a) Fixed points (b) Free energy

Figure 4.8 Curie-Weiss fixed points, $h > 0, K < 1$

4.6 Ising model on a tree

TO DO (transfer from exercises)

4.7 Phase transitions in the Ising model on $\mathbb{Z}^d$

This section is not needed for the main development of these notes and can be skipped in a first reading.
TO COMPLETE

4.8 Notes

Problems

4.1 Definition of the Ising model on a tree.

In problems of Chapter 2 you proved that the Ising model in one dimension ($d = 1$) does not have a phase transition for any $T > 0$. On the grid $\mathbb{Z}^d$ there is a non trivial phase diagram with first and second order phase transitions for any $d \geq 2$. This is also the case on the complete graph (as shown in the lectures) which morally corresponds to $d = +\infty$. Another graph that in a sense, corresponds to $d = +\infty$, is the q-ary tree for $q \geq 3$. Indeed on $\mathbb{Z}^d$ the number of lattice sites at distance less than n from the origin scales as n^d. On the q-ary tree it scales as $(q-1)^n$ which grows faster than n^d for any finite d (for $q \geq 3$). Of course $q = 2$ corresponds to $\mathbb{Z}_+$.

The goal of the three exercises below is to solve for the Ising model on a q-ary tree and show that it displays first and second order phase transitions (with similar qualitative properties than on a complete graph).

Consider a finite rooted tree and call the root vertex o. All vertices have degree q, except for the leaf nodes that have degree 1. We suppose that the tree has n levels (the root being "level 0"). The thermodynamic limit corresponds to $n \to +\infty$. The Hamiltonian (multiplied by β) is

$$\mathcal{H}_n = -J \sum_{(i,j)\in E_n} s_i s_j - h \sum_{i\in V_n} s_i \tag{4.26}$$

were $J > 0$, $h \in \mathbb{R}$, V_n is the set of vertices and E_n the set of edges for the tree with n levels. We are interested in the magnetization *of the root node* in the thermodynamic limit:

$$m(J,h) = \lim_{n\to+\infty} < s_o >_n = \frac{\sum_{\{s_k, k\in V_n\}} s_o \, e^{-\beta \mathcal{H}_n}}{Z_n} \tag{4.27}$$

The formula $\operatorname{atanh} y = \frac{1}{2} \ln \frac{(}{1+y} 1 - y)$ might be useful.

4.2 Recursive equations. Perform the sums over the spins attached at the leaf nodes and show that

$$< s_o >_n = \frac{\sum_{\{s_k, k\in V_{n-1}\}} s_o \, e^{-\beta \mathcal{H}'_{n-1}}}{Z'_{n-1}} \tag{4.28}$$

where E_{n-1} and V_{n-1} are the edge and vertex sets of a tree with with $n-1$ levels and the new Hamiltonian is

$$\beta \mathcal{H}'_n = -J \sum_{(i,j)\in E_{n-1}} s_i s_j - h \sum_{i\in V_{n-1}} s_i - (q-1) \tanh^{-1}(\tanh \beta J \tanh \beta h) \sum_{i\in \text{level } n-1} s_i \tag{4.29}$$

Iterate this calculation and deduce

$$< s_o >_n = \tanh(\beta h + q \tanh^{-1}(\tanh \beta h \tanh u_n)) \tag{4.30}$$

where

$$u_{k+1} = \beta h + (q-1)\tanh^{-1}(\tanh \beta J \tanh u_k), \qquad u_1 = \beta h \tag{4.31}$$

Check that for $q = 2$ you get back the recursion found in one dimension in Chapter 2.

4.3 Analysis of the recursion. We want to analyze the fixed point equation obtained in the preceding question for $q \geq 3$,

$$u = \beta h + (q-1)\tanh^{-1}(\tanh \beta J \tanh u) \tag{4.32}$$

Plot the curves $u \to u - h$ and $u \to (q-1)\tanh^{-1}(\tanh \beta J \tanh u)$ and show that:

- for $\beta J \leq \frac{1}{2}\ln(\frac{q}{q-2})$, (4.32) has a unique solution, and that the iterations (4.31) converge to this unique solution.
- for $\beta J > \frac{1}{2}\ln(\frac{q}{q-2})$:
  - for $|h| \geq h_s$, (4.32) has a unique solution (you do not need to compute h_s explicitly although it is possible to find its analytical expression) and that the iterations (4.31) converge to this unique solution.
  - for $|h| < h_s$, (4.32) has three solutions $u_-(h) < u_0(h) < u_+(h)$. Check graphically that for $h > 0$ the iterations (4.31) with initial condition $u_1 = h$ converge to $u_+(h)$. Similarly for $h < 0$ they converge to $u_-(h)$. Check also graphically that the fixed point $u_0(h)$ is unstable whereas $u_\pm(h)$ are stable.

4.3 Phase transitions. Now we want to discuss the consequences of the results in the previous problem for the phase diagram. On a tree the magnetization is defined as the average spin of the root. More precisely for $h \neq 0$

$$m(\beta J, \beta h) = \lim_{n \to +\infty} \langle s_o \rangle_n, \tag{4.33}$$

and we define the "spontaneous magnetization" as $m_\pm(\beta J) = \lim_{h \to 0_\pm} m(\beta J, \beta h)$. You will show that in the $((\beta J)^{-1}, h)$ plane there is a first order phase transition line $((\beta J)^{-1} \in [0, (\frac{1}{2}\ln(\frac{q}{q-2}))^{-1}[\,, h = 0)$ terminated by a critical point $(\text{atanh}(q-1)^{-1})^{-1}$. Outside of this line $m(\beta J, \beta h h)$ is an analytic function of each variable.

- Deduce from the analysis in problem 2 that for $\beta J \leq \frac{1}{2}\ln(\frac{q}{q-2})$, $m_+(\beta J) = m_-(\beta J) = 0$.
- Deduce that for $\beta J > \frac{1}{2}\ln(\frac{q}{q-2})$, $m_+(\beta J) \neq m_-(\beta J)$ (jump discontinuity or first order phase transition) and that for $\beta \to +\infty$ $m_\pm \to \pm 1$.
- Show that for $\beta J \to \frac{1}{2}\ln(\frac{q}{q-2})$ from above, $m_\pm(\beta J) \sim (\beta J - \frac{1}{2}\ln(\frac{q}{q-2}))^{1/2}$. So on the line $h = 0$, as a function of βJ, the spontaneous magnetization is continuous but not differentiable at $\frac{1}{2}\ln(\frac{q}{q-2})$ (second order phase transition).

- Now fix $\beta J = \frac{1}{2}\ln(\frac{q}{q-2})$ and show that $m(\frac{1}{2}\ln(\frac{q}{q-2}), \beta h) \sim |\beta h|^{1/3}$. As a function of h the spontaneous magnetization is continuous but not differentiable at teh critical point (second order phase transition).

Hint: for the last two questions you can expand the fixed point equation to order u^3.

Remark 1: Note that the exponents $1/2$ and $1/3$ are the same than for the model on a complete graph. This is also the case for all $d \geq 4$ and is not the case for $d = 2, 3$.

Remark 2: On a tree the definition of the magnetization above is *not equivalent* to minus the derivative of the free energy with respect to h. In fact there is a fine point: $-\frac{1}{n}\ln Z_n$ is dominated by the contributions of leaf nodes and is not the "physically meaningful" definition of free energy. Rather the "physically meaningful" definition is given by an integral, with respect to h, of the magnetization at the root.

Part II

Analysis of Message Passing Algorithms

5 Marginalization and Belief Propagation

We have seen that computing the marginals of the Gibbs distributions is a central problem. For example in coding and compressed sensing the tasks of decoding and signal estimation can both be reduced to the determination of a "magnetization" which in turn is easy to obtain once we know the marginals. Unfortunately, for general Gibbs distributions this is an intractable problem. Nevertheless all is not lost, much to the contrary. Indeed, we have seen in Chapter 1 that the factor graphs of our models are always either locally tree like (coding and K-SAT) or complete (compressive sensing); and in Chapter 4 we have learned how to exactly solve two simple models, on the tree and the complete graph, which are toy versions of our more ambitious models.

In this chapter we will concentrate on an *efficient* calculation of marginals for the case where the factor graph is a *tree*. The emphasis here is on the word "efficient". We will see that this question has a natural answer in the form of a message-passing algorithm. The message-passing paradigm is the basis for the *low-complexity* algorithms which we will apply to our problems even when the factor graph *is not* a tree. There is a price to pay on non-tree graphs because marginalization is a priori not exact. Therefore our low complexity message passing algorithms are *suboptimal* in the sense that they do not give correct solutions up to the so-called *static thresholds*. For example message passing decoders do not work up to the MAP threshold of the code ensemble; K-SAT solvers based on message passing find solutions only for densities α quite smaller than the SAT-UNSAT threshold α_s. In the analysis of message passing we will find *algorithmic thresholds* which are smaller (i.e. worse) than the static thresholds.

There is a surprise. Message-passing algorithms are also the key for the analysis of the static thresholds and phase transitions of our three examples. A priori it is not obvious that there should be any connection between static thresholds and low-complexity algorithms. For example as we will see static thresholds are non-differentiability points of the free energy (just as for the Curie-Weiss model) but algorithmic thresholds are not visible on the free energy (since away from static thresholds it is analytic). Nevertheless these two worlds are connected as we will see in the third part of our lectures. Quite remarkably one can also go one step further. In Chapter 14 we will consider a class of ensembles - called spatially coupled ensembles - for which the static and dynamical thresholds may

even be equal. For these ensembles the low complexity message passing methods work all the way up to the static thresholds and allow optimal solutions!

So far we have associated a factor graph to the Hamiltonians or cost functions. In the next section this idea is taken a little bit further by associating the factor graph to the Gibbs distribution itself. We then use this representation to help organize the marginalization on trees and derive the message passing algorithm. As we will see on trees marginalization ultimately boils down to an application of a distributive law of multiplication and addition. Finally we illustrate through simple examples how the formalism is applied to our three problems.

5.1 Factor graph representation of Gibbs distributions

One important characteristic of the Gibbs distributions of our three problems is its *factorized form*. Generically

$$p(\underline{x}) = \frac{1}{Z}\prod_c f_c(x_{\partial c}), \qquad Z = \sum_{\underline{x}\in\mathcal{X}^n}\prod_{c=1}^m f_c(x_{\partial c}) \tag{5.1}$$

where $x_{\partial c}$ is the set of variables x_i entering as arguments of the factors f_c.

The simplest incarnation of this factorization occurs in K-SAT (see (3.55)) where in spin language the alphabet $\mathcal{X} = \{-1, +1\}$, $x_i \to s_i$ and the factors are $f_a(s_{\partial a}) = \exp\{-\beta\prod_{i\in\partial a}(\frac{1+s_i J_{ia}}{2})\}$. For coding (see (3.9)) we have two types of factors $f_i(s_i) = e^{h_i s_i}$ and $f_a(s_{\partial a}) = \frac{1}{2}(1+\prod_{i\in\partial a} s_i)$. For compressed sensing (see (3.40)) the alphabet is continuous $\mathcal{X} = \mathbb{R}$ so in (5.1) the sums must be interpreted as integrals $\int d^n\underline{x}$ and there are two types of factors $f_i(x_i) = (p_0(x_i))^\beta$ and $f_a(x_{\partial a}) = e^{-\frac{\beta}{2\sigma^2}(y_a - A_a^T\underline{x})^2}$. Analogous identifications for general Ising models of Chapter 2 and also for the Curie-Weiss model are left as an exercise. Note that the factorization is not unique, but usually it is pretty clear how to find a natural one.

From now on we will focus on a generic factorization (5.1) and come back to specific illustrations in sections 5.4-5.6. We associate with this factorization a *factor graph* which is mildly different from the ones introduced in Chapter 1. For each variable x_i draw a *variable node* (circle) and for each factor f_c draw a *factor node* (square). Connect a variable node to a factor node by an *edge* if and only if the corresponding variable appears in this factor.

EXAMPLE 11 (Simple Example) Let's start with an example. Consider a distribution with factorization

$$p(x_1, x_2, x_3, x_4, x_5, x_6) = \frac{1}{Z} f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5). \tag{5.2}$$

The resulting graph for this distribution is shown on the Figure 5.1. ♢

The factor graph is *bipartite*. This means that the set of vertices is partitioned into two groups (the set of nodes corresponding to variables and the set of nodes

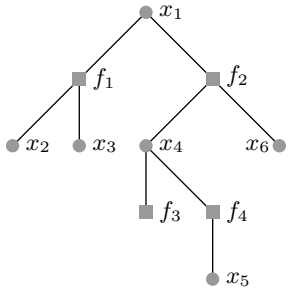


Figure 5.1 Factor graph of f given in Example 11.

corresponding to factors) and that an edge always connects a variable node to a factor node. For our particular example the factor graph is a (bipartite) *tree*. This means that there are no *cycles* in the graph; i.e., there is one and only one path between each pair of nodes.

As we will show in the next section, for factor graphs that are trees marginals can be computed efficiently by *message-passing* algorithms. This remains true in the slightly more general scenario where the factor graph forms a *forest*; i.e., the factor graph is disconnected and it is composed of a collection of trees. In order to keep things simple we will assume a single tree and ignore this straightforward generalization.

5.2 Marginalization on trees

We first remark that in order to carry out the marginalization in practice one can first ignore the partition function Z. Indeed suppose that we want to compute the marginal $\nu_1(x_1)$ (recall definition (2.24)) for (5.1). Let us first compute the "marginal" of the numerator only

$$\mu_1(x_1) = \sum_{\sim x_1} \prod_c f_c(x_{\partial c}) \tag{5.3}$$

Clearly $\nu_1(x_1) = \mu(x_1)/Z$ so the only difference between $\nu_1(x_1)$ and $\mu_1(x_1)$ is a proportionality factor which serves to normalize the marginal. Thus, assuming that we are able to compute $\mu(x_1)$, we simply get the marginal by normalizing

$$\nu_1(x_1) = \frac{\mu_1(x_1)}{\sum_{x_1 \in \mathcal{X}} \mu_1(x_1)}, \tag{5.4}$$

This last step is an easy task that involves only one sum or an integral.

In the sequel and in practice we just deal with the "marginalization" of the numerator and normalize the result in the very last step.

Distributive Law

On trees marginalization can be achieved by a careful application of the distributive law. Let $\mathbb{F}$ be a field (think of $\mathbb{F} = \mathbb{R}$) and let $a, b, c \in \mathbb{F}$. The *distributive law* states

$$ab + ac = a(b + c). \tag{5.5}$$

This simple law, properly applied, can significantly reduce computational complexity: consider, e.g., the evaluation of $\sum_{i,j} a_i b_j$ as $(\sum_i a_i)(\sum_j b_j)$. Factor graphs provide an appropriate framework to systematically take advantage of the distributive law.

Let's start with Example 11. The numerator of p is a function f with factorization

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5). \tag{5.6}$$

We are interested in computing the *marginal* of f with respect to x_1

$$\mu_1(x_1) = \sum_{\sim x_1} f(x_1, x_2, x_3, x_4, x_5, x_6).$$

What is the complexity of a brute force computation? Assume that all variables take values in a finite alphabet, call it $\mathcal{X}$. Determining $\nu(x_1)$ for all values of x_1 by brute force requires $\Theta\left(|\mathcal{X}|^6\right)$ operations, where we assume a naive computational model in which all operations (addition, multiplication, function evaluations, etc.) have the same cost. But we can do better: taking advantage of the factorization, we can rewrite $\nu(x_1)$ as

$$\mu(x_1) = \Big[\sum_{x_2, x_3} f_1(x_1, x_2, x_3)\Big]\Big[\sum_{x_4} f_3(x_4)\big(\sum_{x_6} f_2(x_1, x_4, x_6)\big)\big(\sum_{x_5} f_4(x_4, x_5)\big)\Big].$$

Fix x_1. The evaluation of the first factor can be accomplished with $\Theta\left(|\mathcal{X}|^2\right)$ operations. The second factor depends only on x_4, x_5, and x_6. It can be evaluated efficiently in the following manner. For each value of x_4 (and x_1 fixed), determine $\sum_{x_5} f_4(x_4, x_5)$ and $\sum_{x_6} f_2(x_1, x_4, x_6)$. Multiply by $f_3(x_4)$ and sum over x_4. Therefore, the evaluation of the second factor requires $\Theta\left(|\mathcal{X}|^2\right)$ operations as well. Since there are $|\mathcal{X}|$ values for x_1, the overall task has complexity $\Theta\left(|\mathcal{X}|^3\right)$. This compares favorably to the complexity $\Theta\left(|\mathcal{X}|^6\right)$ of the brute force approach.

Recursive Determination of Marginals

Consider the factorization of a generic function g (e.g. the numerator of a Gibbs distribution (5.1)) and suppose that the associated factor graph is a tree (by definition it is always bipartite). Suppose that we are interested in marginalizing g with respect to the variable z; i.e., we are interested in computing $\mu(z) =$

$\sum_{\sim z} g(z, \dots)$. Since the factor graph of g is a bipartite tree, g has a generic factorization of the form

$$g(z, \dots) = \prod_{k=1}^{K} [g_k(z, \dots)]$$

for some integer K with the following crucial property: z appears in each of the factors g_k, but all other variables appear in *only one* factor. To see this assume to the contrary that another variable is contained in two of the factors. This implies that besides the path that connects these two factors via variable z another path exists. But this contradicts the assumption that the factor graph is a tree.

For the function f of Example 11 this factorization is

$$f(x_1, \dots) = [f_1(x_1, x_2, x_3)]\,[f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)]\,,$$

so that $K = 2$. The generic factorization and the particular instance for our running example f are shown in Figure 5.2. Taking into account that the individual

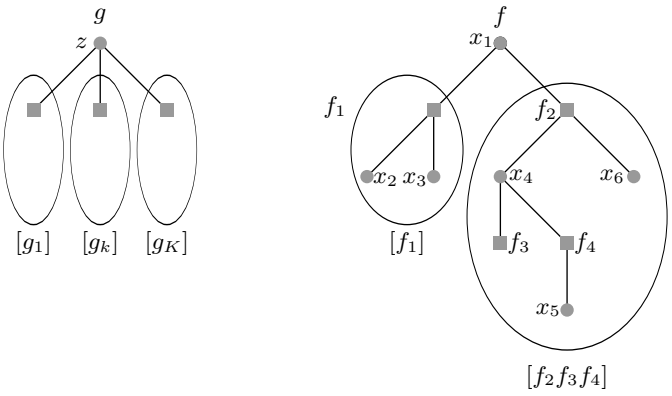


Figure 5.2 Generic factorization and the particular instance.

factors $g_k(z, \dots)$ only share the variable z, an application of the distributive law leads to

$$\mu(z) = \sum_{\sim z} g(z, \dots) = \underbrace{\sum_{\sim z} \prod_{k=1}^{K} [g_k(z, \dots)]}_{\text{marginal of product}} = \underbrace{\prod_{k=1}^{K} \Big[\sum_{\sim z} g_k(z, \dots)\Big]}_{\text{product of marginals}}\,. \tag{5.7}$$

In words, the marginal $\sum_{\sim z} g(z, \dots)$ is the product of the individual marginals $\sum_{\sim z} g_k(z, \dots)$. In terms of our running example we have

$$\nu(x_1) = \Big[\sum_{\sim x_1} f_1(x_1, x_2, x_3)\Big]\Big[\sum_{\sim x_1} f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)\Big].$$

This single application of the distributive law leads, in general, to a non-negligible reduction in complexity. But we can go further and apply the same idea recursively to each of the terms $g_k(z, \dots)$.

In general, each g_k is itself a product of factors. In Figure 5.2 these are the factors of g that are grouped together in one of the ellipsoids. Since the factor graph is a bipartite tree, g_k must in turn have a generic factorization of the form

$$g_k(z, \dots) = \underbrace{h(z, z_1, \dots, z_J)}_{\text{kernel}} \prod_{j=1}^{J} \underbrace{[h_j(z_j, \dots)]}_{\text{factors}},$$

where z appears only in the "kernel" $h(z, z_1, \dots, z_J)$ and each of the z_j appears *at most twice*, possibly in the kernel and in at most one of the factors $h_j(z_j, \dots)$. All other variables are again unique to a single factor. For our running example we have

$$f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5) = \underbrace{f_2(x_1, x_4, x_6)}_{\text{kernel}} \underbrace{[f_3(x_4) f_4(x_4, x_5)]}_{x_4} \underbrace{[1]}_{x_6}.$$

The generic factorization and the particular instance for our running example f are shown in Figure 5.3. Another application of the distributive law gives

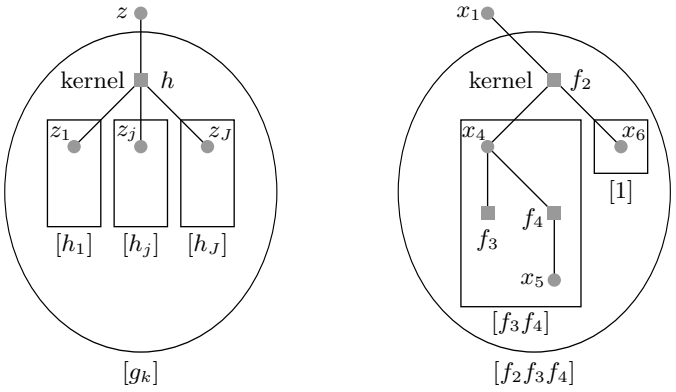


Figure 5.3 Generic factorization of g_k and the particular instance.

$$\begin{aligned} \sum_{\sim z} g_k(z, \dots) &= \sum_{\sim z} h(z, z_1, \dots, z_J) \prod_{j=1}^{J} [h_j(z_j, \dots)] \\ &= \sum_{\sim z} h(z, z_1, \dots, z_J) \underbrace{\prod_{j=1}^{J} \Big[\sum_{\sim z_j} h_j(z_j, \dots) \Big]}_{\text{product of marginals}}. \end{aligned} \tag{5.8}$$

In words, the desired marginal $\sum_{\sim z} g_k(z, \dots)$ can be computed by multiplying the kernel $h(z, z_1, \dots, z_J)$ with the individual marginals $\sum_{\sim z_j} h_j(z_j, \dots)$ and summing out all remaining variables other than z.

We are back to where we started. Each factor $h_j(z_j, \dots)$ has the same generic form as the original function $g(z, \dots)$, so that we can continue to break down the

marginalization task into smaller pieces. This recursive process continues until we have reached the leaves of the tree. The calculation of the marginal then follows the recursive splitting in reverse. In general, nodes in the graph compute marginals, which are functions over $\mathcal{X}$, and pass these on to the next level. In the next section we will elaborate on this method of computation, known as message passing: the marginal functions are messages. The message combining rules at function nodes is explicit in (5.8). And at a variable node we simply perform pointwise multiplication.

Let us consider the initialization of the process. At the leaf nodes the task is simple. A function leaf node has the generic form $g_k(z)$, so that $\sum_{\sim z} g_k(z) = g_k(z)$: this means that the initial message sent by a function leaf node is the function itself. To find out the correct initialization at a variable leaf node consider the simple example of computing $\sum_{\sim x_1} f(x_1, x_2)$. Here, x_2 is the variable leaf node. By the message-passing rule (5.8) the marginal is equal to $\sum_{\sim x_1} f(x_1, x_2) \cdot \mu(x_2)$, where $\mu(x_2)$ is the initial message that we send from the leaf variable node x_2 towards the kernel $f(x_1, x_2)$. We see that to get the correct result this initial message should be the constant function 1.

5.3 Marginalization via Message Passing

In the previous section we have seen that, in the case where the factor graph is a tree, the marginalization problem can be broken down into smaller and smaller tasks according to the structure of the tree.

This gives rise to the following efficient *message-passing* algorithm. The algorithm proceeds by sending messages along the edges of the tree. Messages are *functions* on $\mathcal{X}$, or, equivalently, vectors of length $|\mathcal{X}|$. The messages signify marginals of parts of the function and these parts are combined to form the marginal of the whole function. Message passing originates at the leaf nodes. Messages are passed up the tree and as soon as a node has received messages from all its children, the incoming messages are processed and the result is passed up to the parent node.

EXAMPLE 12 (Message-Passing Algorithm for f of Example 11) Consider this procedure in detail for the case of our running example as shown in Figure 5.4. The top leftmost graph is the factor graph. Message passing starts at the leaf nodes as shown in the middle graph on the top. The variable leaf nodes x_2, x_3, x_5, and x_6 send the constant function 1 as discussed at the end of the previous section. The factor leaf node f_3 sends the function f_3 up to its parent node. In the next time step the factor node f_1 has received messages from both its children and can therefore proceed. According to (5.8), the message it sends up to its parent node x_1 is the product of the incoming messages times the "kernel" f_1, after summing out all variable nodes except x_1; i.e., the message is $\sum_{\sim x_1} f_1(x_1, x_2, x_3)$. In the same manner factor node f_4 forwards to its parent

node x_4 the message $\sum_{\sim x_4} f_4(x_4, x_5)$. This is shown in the rightmost figure in the top row. Now, variable node x_4 has received messages from all its children. It forwards to its parent node f_2 the product of its incoming messages, in agreement with (5.7), which says that the marginal of a product is the product of the marginals. This message, which is a function of x_4, is $f_3(x_4) \sum_{\sim x_4} f(x_4, x_5) = \sum_{\sim x_4} f_3(x_4) f_4(x_4, x_5)$. Next, function node f_2 can forward its message, and, finally, the marginalization is achieved by multiplying all incoming messages at the root node x_1. ◊

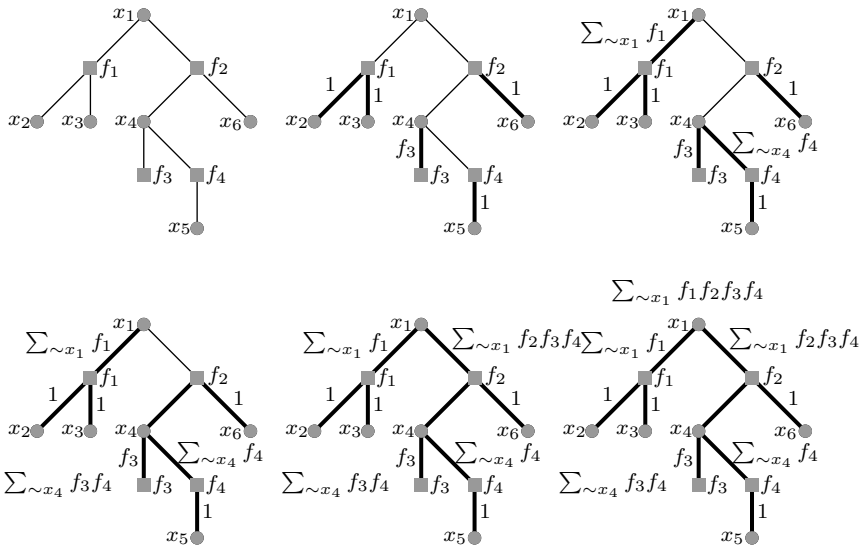


Figure 5.4 Marginalization of function f from Example 11 via message passing. Message passing starts at the leaf nodes. A node that has received messages from all its children processes the messages and forwards the result to its parent node. Bold edges indicate edges along which messages have already been sent.

Complexity of message passing

Before stating the message-passing rules formally, consider the following important generalization. Whereas so far we have considered the marginalization of a function f with respect to a *single* variable x_1 we are actually interested in marginalizing for *all* variables. We have seen that a single marginalization can be performed efficiently if the factor graph of f is a *tree*, and that the complexity of the computation essentially depends on the largest degree of the factor graph and the size of the underlying alphabet. Consider now the problem of computing *all* marginals. We can draw for each variable a tree rooted in this variable and execute the single marginal message-passing algorithm on each rooted tree. It is easy to see, however, that the algorithm does not depend on which node is the root of the tree and that in fact all the computations can be performed simulta-

neously on a single tree. Simply start at all leaf nodes and for every edge compute the outgoing message along this edge as soon as you have received the incoming messages along all *other* edges that connect to the given node. Continue in this fashion until a message has been sent in both directions along every edge. This computes *all* marginals so it is more complex than computing a single marginal but only by a factor roughly equal to the average degree of the nodes. We now summarize this discussion.

Belief propagation equations

Messages flow on edges in both directions. Messages from variables nodes (circles) to function nodes (squares) are denoted $\mu_{i\to c}$, and messages from function nodes to variable nodes $\hat{\mu}_{c\to i}$. As before the letters $a, b, c, \ldots$ are reserved for function nodes and $i, j, k, \ldots$ for variable nodes. Although this may sometimes be redundant notation, in order to avoid confusions it is convenient to reserve μ for messages from variable nodes (circles) to factor nodes (squares) and $\hat{\mu}$ for messages from factor nodes to variable nodes. Marginals, once normalized, will be denoted by ν. Messages and marginals are functions on $\mathcal{X}$ and for finite alphabets it is sometimes useful to think of them as vectors with $|\mathcal{X}|$ components.

Message passing starts at leaf nodes. Consider a node and one of its adjacent edges, call it e. As soon as the *incoming* messages to the node along all *other* adjacent edges have been received these messages are processed and the result is *sent out* along e. This process continues until messages along all edges in the tree have been processed. In the final step the marginals are computed by combining *all* messages which enter a particular variable node. The initial conditions and processing rules are summarized in Figure 5.5. Since the messages represent (unormalized) probabilities or *beliefs*, the algorithm is also known as the *belief propagation* (BP) algorithm. From now on we will mostly refer to it under this name.

We sumarize the BP relations here for further reference

$$\mu_{i\to a}(x_i) = \prod_{b\in\partial i\setminus a} \hat{\mu}_{b\to i}(x_i) \tag{5.9}$$

$$\hat{\mu}_{a\to i}(x_i) = \sum_{\sim x_i} f_a(x_{\partial a}) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(x_j) \tag{5.10}$$

At leaf nodes these are interpreted as $\mu_{i\to c}(x_i) = 1$ and $\hat{\mu}_{c\to i}(x_i) = f_c(x_{\partial c})$. The marginals are obtained as

$$\nu_i(x_i) = \frac{\prod_{a\in\partial i} \hat{\mu}_{a\to i}(x_i)}{\sum_{x_i} \prod_{a\in\partial i} \hat{\mu}_{a\to i}(x_i)} \tag{5.11}$$

$$\nu_a(x_{\partial a}) = \frac{f_a(x_{\partial a}) \prod_{i\in\partial a} \mu_{i\to a}(x_i)}{\sum_{x_{\partial a}} f_a(x_{\partial a}) \prod_{i\in\partial a} \mu_{i\to a}(x_i)}. \tag{5.12}$$

When we compute the marginals it is not important how the messages are normalized. Indeed in (5.11)-(5.12) the normalizations cancel out. We will often

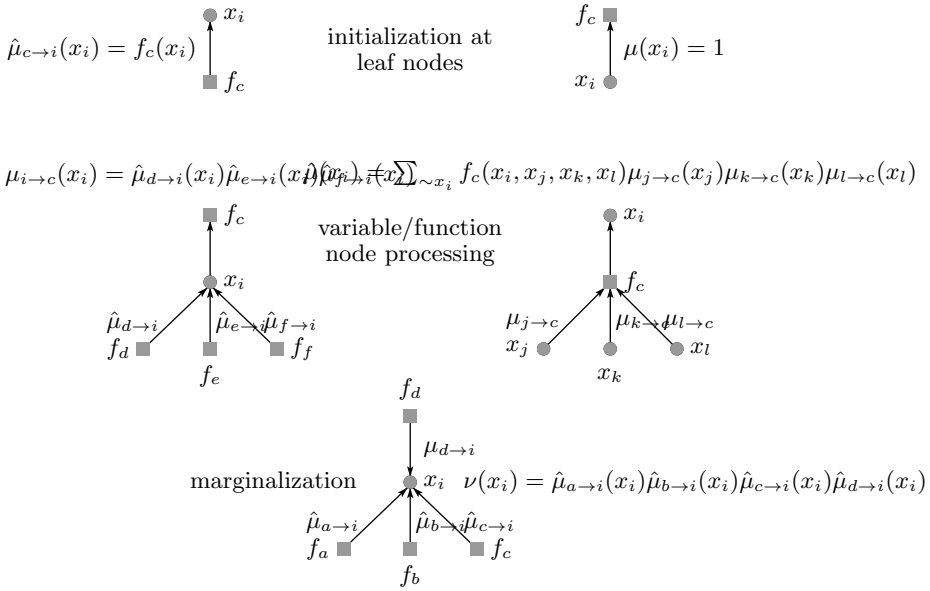


Figure 5.5 Message-passing rules. The top row shows the initialization of the messages at the leaf nodes. The middle row corresponds to the processing rules at the variable and function nodes, respectively. The bottom row explains the final marginalization step.

exploit this fact and write (5.9)-(5.10) as proportionality relations. This often simplifies many calculations.

Algorithmic versus static point of view

As explained in this chapter the BP relations allow to compute exact marginals on trees. By starting the process at leaf nodes we are sure that it converges in a finite number of steps to the exact marginals. On non-tree graphs the situation is not as simple because this process *does not* yield exact marginals. There, the BP relations form the basis of an algorithm which outputs *BP marginals* which are used to make decisions about the decoded bit, signal estimate, etc. To run the algorithm we have to decide on a schedule to compute the messages. The so-called "flooding schedule" is popular. At each time step t one sends in parallel messages $\mu_{i\to c}^{(t)}(x_i)$ from variable nodes to function nodes, and from these one computes messages $\hat{\mu}_{c\to i}^{(t)}(x_i)$ which are sent back in parallel again. One runs these iterations for times $t = 0, \ldots, T$ until some reasonable stopping time, and the BP marginals are estimated thanks to the messages at time T.

In the third part of these notes the BP equations will be used in a "statistical mechanics" non-algorithmic way, namely as fixed point equations. We will see that they also arise when one minimizes the so-called "Bethe free energy" much as

the Curie-Weiss fixed point equation appeared in Chapter 4 when we minimized the free energy function. This point of view will be become key when we relate low complexity algorithms to static thresholds.

5.4 Decoding via Message Passing

Assume we transmit over a binary-input memoryless channel using a linear code. Recall the formulation in Chapter 3: the rule (3.11) for the *bit-wise* maximum a posteriori (MAP) decoder reads $\hat{s}_i(\underline{h}) = \operatorname{argmax}_{s_i \in \{\pm 1\}} p(s_i \mid h) = \operatorname{sign}\langle s_i \rangle$ which is immediate to compute once we have $p(s_i \mid \underline{h})$ the marginal of distribution (3.9). So we have to marginalise the numerator of

$$p(\underline{s} \mid \underline{h}) = \frac{1}{Z} \prod_{a=1}^{m} \frac{1}{2}(1 + \prod_{i \in \partial a} s_i) \prod_{i=1}^{n} e^{h_i s_i}. \tag{5.13}$$

and eventually normalize the resulting function of $s_i \in \{-1, +1\}$. This numerator has a factorized form with two types of factors, $f_i(s_i) = e^{h_i s_i}$ and $f_a(\{s_i, i \in \partial a\}) = \frac{1}{2}(1 + \prod_{i \in \partial a} s_i)$, which are associated to square nodes in the factor graph representation of (5.13). The first factor is attached in the factor graph to a single bit and describes the influence of the channel. The second one is attached to several bits and describes the parity-check constraints.

EXAMPLE 13 (Bit-wise MAP Decoding) Consider the code defined by its parity-check matrix with Tanner graph shown on the left of Fig. 5.6.

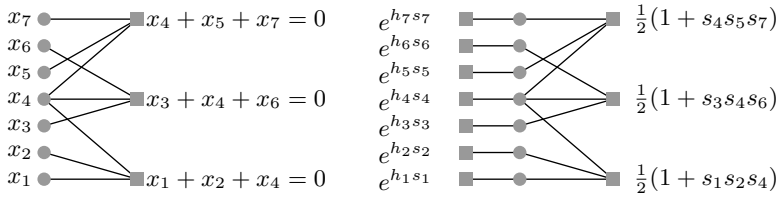


Figure 5.6 Left: graphical representation of the parity check code. Right: factor graph associated to the distribution (5.13) of our running example.

The factor graph corresponding to the distribution (5.13) is shown on the right of this figure. It includes the (Tanner) graph of parity check code, but additionally contains factor nodes which represent the effect of the channel. For this particular case the resulting graph is a tree. We can therefore apply the message-passing algorithm to this example to perform bit-wise MAP decoding. ◊

In principle the messages are uniquely specified by the general message-passing rules and we could simply move on to the next example. Indeed, the real power of the factor graph approach lies in the fact that, once the graph and the factor

nodes are specified, no thought is required to work out the messages. For the current example perhaps the result is quite intuitive and this might seem as no big deal. But in "real" systems substantially more complicated factor graphs are encountered and in such cases without the message passing rules it might be quite difficult to figure out how to correctly combine messages. Despite the fact that we could just blindly follow the rules, it is instructive to explicitly work out a few steps of the belief propagation algorithm for this example.

EXAMPLE 14 (Message passing algorithm for decoding) We give the first three steps of belief propagation for the tree in Figure 5.6. In the first step the initial messages are sent from leaf nodes. Here all leaf nodes are factor nodes whose factor is the prior, thus the initial messages are $\hat{\mu}_{k\to k}(s_k) = e^{h_k s_k}$ for $k = 1, \ldots, 7$. At the second step six variable nodes send messages to factor nodes, namely the variable nodes that participate in only a single parity-check constraints: $\mu_{1\to 1}(s_1) = e^{h_1 s_1}$, $\mu_{2\to 1}(s_2) = e^{h_2 s_2}$, $\mu_{3\to 2}(s_3) = e^{h_3 s_3}$, $\mu_{5\to 1}(s_5) = e^{h_5 s_5}$, $\mu_{7\to 1}(s_7) = e^{h_7 s_7}$. At the third step the three factor nodes have received all their input, except the input from variable node 4. Hence, they can send their messages in direction of node 4. These are

$$\hat{\mu}_{1\to 4}(s_4) = \sum_{s_1,s_2} \frac{1}{2}(1 + s_1 s_2 s_4) e^{h_1 s_1} e^{h_2 s_2},$$

$$\hat{\mu}_{2\to 4}(s_4) = \sum_{s_3,s_6} \frac{1}{2}(1 + s_3 s_4 s_6) e^{h_3 s_3} e^{h_6 s_6},$$

$$\hat{\mu}_{3\to 4}(s_4) = \sum_{s_5,s_7} \frac{1}{2}(1 + s_4 s_5 s_7) e^{h_5 s_5} e^{h_7 s_7}.$$

The sums involved in the messages are easy to compute. For example using $e^{h_i s_i} = \cosh h_i + s_i \sinh h_i$ the first one is equal to

$$\hat{\mu}_{1\to 4}(s_4) = (2\cosh h_1 \cosh h_2)(1 + s_4 \tanh h_1 \tanh h_2)$$

Looking at one more step, note that at this point all incoming messages to variable node 4 are known and so we can compute the "marginal" $\mu_4(s_4)$ (of the numerator) by multiplying all messages incoming into variable node 4. Explicitly,

$$\begin{aligned}\mu(s_4) =& (2\cosh h_4)(1 + s_4 \tanh h_4)(2\cosh h_1 \cosh h_2)(1 + s_4 \tanh h_1 \tanh h_2) \\ &\times (2\cosh h_3 \cosh h_6)(1 + s_4 \tanh h_3 \tanh h_6) \\ &\times (2\cosh h_5 \cosh h_7)(1 + s_4 \tanh h_5 \tanh h_7)\end{aligned}$$

To get the true marginal $\nu_4(s_4) = p(s_4 \mid \underline{h})$ one has to normalize $\mu(s_4)$,

$$p(s_4 \mid \underline{h}) = \frac{\mu(s_4)}{\mu_4(1) + \mu_4(-1)}$$

To compute the other marginals one continues in this fashion with further steps of belief propagation. As a final remark, note that (in the binary case) messages can equivalently be considered as vectors with two components or as Bernoulli distributions. ◊

5.5 Message Passing in Compressed Sensing

Recall the spin glass setting for compressed sensing in Section 3.4. From the marginals $p(x_i|\underline{y})$ of the posterior distribution (3.40)

$$p_\beta(\underline{x} \mid \underline{y}) = \frac{1}{Z_\beta} \prod_{a=1}^{r} e^{-\frac{\beta}{2\sigma^2}(y_a - A_a^T \underline{x})^2} \prod_{i=1}^{n} (p_0(x_i))^\beta, \tag{5.14}$$

we can compute the Gibbs average $\hat{x}_{i,\beta}(\underline{y}) = \langle x_i \rangle_\beta$. To get the MMSE estimate (when the prior is known) we set $\beta = 1$; to get the LASSO estimate (when we only know that the prior is in the sparse class $\mathcal{F}_\kappa$) we take $p_0(x) = e^{-\frac{\lambda}{\sigma^2}|x|}$ and send $\beta \to +\infty$. For compressive sensing marginalization involves integrals instead of discrete sums. Formally, the distributive law (5.5) is replaced by $\int dx\, a(x)b(x) + \int dx\, a(x)c(x) = \int dx\, a(x)(b(x)+c(x))$ but otherwise the marginalization proceeds exactly in the same way as in the discrete case if we simply replace sums by integrals in the message-passing rules (note that in our applications all integrals will remain finite).

To obtain $p(x_i \mid \underline{y})$, it is sufficient to marginalize the numerator in (5.14) and eventually normalize the resulting function of x_i. As in the coding case, this numerator has a factorized form with two types of factors $f_i(x_i) = (p_0(x_i))^\beta$ and $f_a(x_{\partial a}) = e^{-\frac{1}{2\sigma^2}(y_a - A_a^T \underline{x})^2}$. We already associated a "Tanner graph" to the measurement matrix A in Chapter 2. Here we go one step further. In the factor graph representation for the distribution (5.14) we add extra square nodes corresponding to the factors $(p_0(x_i))^\beta$ and attach them to variable nodes. The other square nodes already present in the representation of the measurement matrix are associated to the factors $f_a(x_{\partial a})$. Let us discuss a concrete illustration.

EXAMPLE 15 (Factor graph for compressive sensing) Figure 5.7 shows a factor graph associated to (5.14). Edges are present if and only if $A_{ai} \neq 0$. One may think of $A_{ai} \neq 0$ as the "strength" of an edge. This factor graph contains the graph representing A itself, and has also additional factor nodes which represent the prior for the signal ◊

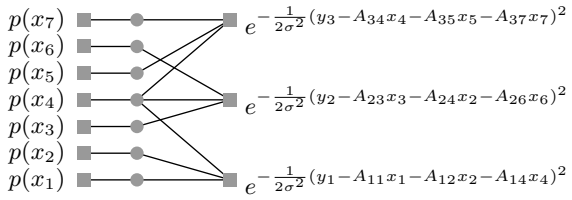


Figure 5.7 Factor graph for compressive sensing. The edges represent the non-zero elements of the measurement matrix. The signal has seven components and there are three measurements.

A few comments are in order. In this example we take a factor graph that is a tree for the purpose of illustration of the message passing rules below. However in

compressive sensing the graph is far from being a tree; it typically is a complete graph. Indeed we assume that the entries of the measurement matrix are iid Gaussian, so the matrix is dense. This is one important difference between the compressive sensing and coding models In coding our analysis will rely heavily on the fact that the graph is sparse and that when we look at very large instances the Tanner graph will "locally" be a tree. At first glance it therefore appears that message-passing techniques which explicitly rely on the Tanner graph being a tree are of no use in the compressive sensing context. But perhaps surprisingly, as we will see, we will still be able to analyze this situation. The key in this case is that despite the fact that we will not face a tree, the influence of each edge vanishes in the limit of large graphs. This relies heavily on the $1/m$ scaling of the variance of the matrix elements A_{ai}.

Let us now discuss belief propagation for the example.

EXAMPLE 16 (Message passing algorithm for compressive sensing) We give the first three steps of belief propagation for the tree in Figure 5.7. As remarked above, the messages are continuous distributions and instead of performing binary sums one has compute integrals; this is the main difference with the coding case. In the first step, the initial messages are sent from leaf nodes: $\hat{\mu}_{k\to k}(x_k) = (p_0(x_k))^\beta$ for $k = 1, \ldots, 7$. At the second step six variables (namely the ones that participate in only one measurement) send messages to factor nodes: $\mu_{1\to 1}(x_1) = (p_0(x_1))^\beta$, $\mu_{2\to 1}(x_2) = (p_0(x_2))^\beta$, $\mu_{3\to 2}(x_3) = (p_0(x_3))^\beta$, $\mu_{5\to 1}(x_5) = (p_0(x_5))^\beta$, $\mu_{6\to 1}(x_6) = (p_0(x_6))^\beta$ $\mu_{7\to 1}(x_7) = (p_0(x_7))^\beta$. At the third step the three factor nodes send messages to variable node 4. These are

$$\hat{\mu}_{1\to 4}(x_4) = \int\int dx_1 dx_2 \, (p_0(x_1))^\beta (p_0(x_2))^\beta e^{-\frac{\beta}{2\sigma^2}(y_1 - A_{11}x_1 - A_{12}x_2 - A_{14}x_4)^2},$$
$$\hat{\mu}_{2\to 4}(x_4) = \int\int dx_3 dx_6 \, (p_0(x_3))^\beta (p_0(x_6))^\beta e^{-\frac{\beta}{2\sigma^2}(y_2 - A_{22}x_2 - A_{23}x_3 - A_{26}x_6)^2},$$
$$\hat{\mu}_{3\to 4}(x_4) = \int\int dx_5 dx_7 \, (p_0(x_5))^\beta (p_0(x_7))^\beta e^{-\frac{\beta}{2\sigma^2}(y_3 - A_{34}x_4 - A_{35}x_5 - A_{37}x_7)^2}.$$

Note that all integrals are certainly convergent as long as the prior $p_0(\cdot)$ is integrable. This time, contrary to the coding example where binary sums could easily be computed, in general the integrals cannot be performed analytically but have to be evaluated numerically. One exception where a complete analytical calculation is easy, is the case where the priors are Gaussians. This leads to messages that are Gaussians throughout the whole belief propagation algorithm. A mixture of Bernoulli and Gaussian priors also leads to explicit although rather complicated formulas. This last case is sometimes considered as a model of a sparse prior in the context of compressive sensing. Note however, that the Laplacian prior $ce^{-\frac{\lambda}{\sigma^2}|x_k|}$ does *not* lead to completely analytically tractable integrals because of the absolute value.

At this point we can compute the marginal $\mu_4(x_4)$. Indeed all messages in-

coming into variable node 4 are known, so

$$\mu_4(x_4) = p_0(x_4)\hat{\mu}_{1\to 4}(x_4)\hat{\mu}_{2\to 4}(x_4)\hat{\mu}_{3\to 4}(x_4)$$

To get the marginal $p(x_4 \mid \underline{y})$ we normalize $\mu_4(x_4)$,

$$p(x_4 \mid \underline{y}) = \frac{\mu(x_4)}{\int dx_4\, \mu(x_4)}.$$

Finally, the computation of other marginals requires further steps of belief propagation. ◊

LASSO estimate and min-sum rules

We remarked in 3.4 that the LASSO estimate can be obtained by taking the prior $p_0(x_i) = e^{-\frac{\lambda}{\sigma^2}|x_i|}$, and letting $\beta \to +\infty$. Taking the $\beta \to +\infty$ limit of the message passing rules developed here leads to the so-called *min-sum* rules. It is instructive to work this out in detail for the current example. To obtain a well defined limit for the message passing rules it is convenient to define

$$\hat{e}_{a\to i} = -\frac{1}{\beta}\ln\hat{\mu}_{a\to i}, \qquad \text{and} \qquad e_{i\to a} = -\frac{1}{\beta}\ln\mu_{i\to a}.$$

Then the initial messages from leaf square nodes to variables are $\hat{\epsilon}_{k\to k}(x_k) = \frac{\lambda}{\sigma^2}|x_k|$ for $k = 1,\dots,7$. At the second step the six variables $k = 1,2,3,5,7$ participating in a single measurement send messages to factor nodes: $\epsilon_{k\to k}(x_1) = \frac{\lambda}{\sigma^2}|x_k|$. At the third step the three factor nodes send messages to variable node 4. These are deduced from the finite β messages by applying the Laplace method to the integrals,

$$\begin{aligned}
\hat{e}_{1\to 4}(x_4) &= \min\{\frac{\lambda}{\sigma^2}|x_1| + \frac{\lambda}{\sigma^2}|x_2| + \frac{1}{2\sigma^2}(y_1 - A_{11}x_1 - A_{12}x_2 - A_{14}x_4)^2\}\\
\hat{e}_{2\to 4}(x_4) &= \min\{\frac{\lambda}{\sigma^2}|x_3| + \frac{\lambda}{\sigma^2}|x_6| + \frac{1}{2\sigma^2}(y_2 - A_{22}x_2 - A_{23}x_3 - A_{26}x_6)^2\},\\
\hat{e}_{3\to 4}(x_4) &= \min\{\frac{\lambda}{\sigma^2}|x_3| + \frac{\lambda}{\sigma^2}|x_6| + \frac{1}{2\sigma^2}(y_3 - A_{34}x_4 - A_{35}x_5 - A_{37}x_7)^2\}.
\end{aligned}$$

The "marginal" for node 4 is

$$e_4(x_4) = \frac{\lambda}{\sigma^2}|x_4| + \hat{e}_{1\to 4}(x_4) + \hat{e}_{2\to 4}(x_4) + \hat{e}_{3\to 4}(x_4)$$

and the LASSO estimate for variable node 4 is simply $\hat{x}_4 = \operatorname{argmin} e_4(x_4)$. These relations constitute the min-sum algorithm.

There is also an alternative route how to derive the min-sum relations. The belief-propagation equations (sometimes also called sum-product algorithm) were derived from the distributed law once we applied it to a factor graph which is a tree. It led to the marginalization of a function. But instead of using the operations of summing and multiplying (leading to the sum-product algorithm) we

can use as basic operations the minimization and summing. The corresponding distributive law for this case reads

$$\min(a+b, a+c) = a + \min(b, c). \tag{5.15}$$

We can now formaly proceed just as in the previous case. A quick way to see this is to use the correspondence $(+, \times) \to (\min, +)$ which transforms $ab+ac = a(b+c)$ to $\min(a+b, a+c) = a+\min(b, c)$. You will derive the min-sum message passing rules from the distributive law in an exercise.

5.6 Message passing in K-SAT

We illustrate message passing for K-SAT with two applications. In the first one we count solutions of a K-SAT formula and in the second we discuss the determination of minimum energy assignments.

Counting solutions through message passing

Recall in the K-SAT model we introduced in Section 3.6 the number of solutions of a $K-SAT$ formula,

$$\mathcal{N}_0 = \sum_{\underline{s}} \prod_{a=1}^{m} (1 - \prod_{i \in \partial a} (\frac{1+s_i J_{ai}}{2})). \tag{5.16}$$

We illustrate here how one could attempt to compute it by message passing methods. Suppose we can count the number of solutions having a fixed value for the i-th variable, namely

$$\mathcal{N}_i(s_i) = \sum_{\sim s_i} \prod_{a=1}^{m} (1 - \prod_{i \in \partial a} (\frac{1+s_i J_{ai}}{2})). \tag{5.17}$$

where the sum carries over all variables except s_i. The total number of solutions is simply obtained as $\mathcal{N}_0 = \mathcal{N}_i(+1) + \mathcal{N}_i(-1)$. The task of computing (5.17) is nothing else than our marginalization problem. The factor graph associated to (5.16) has only one type of factor $(1 - \prod_{i \in \partial a} (\frac{1+s_i J_{ai}}{2}))$ associated to the square nodes. Again, message passing provides an exact solution on a tree-graph. When the graph is not a tree it forms the basis of a solution finding message passing algorithm, called Belief Propagation Guided Decimation (BPGD), which we will study in Chapter 9.5. Let us for now illustrate how the marginalization proceeds on our simple tree graph example.

EXAMPLE 17 (Counting solutions in 3-SAT) Consider the 3-SAT formula shown on Fig. 5.8. Here we keep the signs $J_a i = \pm 1$ associated to the edges open in order to see more clearly the structure of the messages (so we have a set of 2^9 formulas here). The factors associated to each square are the indicator

functions of the clause. For example clause number 1 is *not* satisfied by the assignment $s_1 = J_{11}$, $s_2 = J_{12}$, $s_4 = J_{14}$ and is satisfied by teh 7 other assignments. Note that contraty to coding and compressed sensing there is are no "priors", so no degree-one square nodes with factors attached to variable nodes. Here message

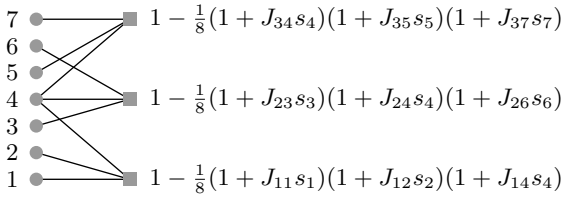


Figure 5.8 Factor graph for the K-SAT counting problem. The graph represents the formula and the factors associated to teh square nodes are the indicator functions of each constraint written in spin language.

passing starts at leaf nodes, namely the variable nodes $1, 2, 3, 5, 6, 7$ which send the trivial initial messages $\mu_{i\to 1}(s_i) = \mu_{i\to 2}(s_i) = \mu_{i\to 3}(s_i) = 1$, $i = 1, 2, 3, 5, 6, 7$. In teh second step all clauses can compute one outgoing message towrads variable node 4 by taking into account their factor and two incomimg messages. In detail,

$$\hat{\mu}_{1\to 4}(s_4) = \sum_{s_1,s_2} \big(1 - \frac{1}{8}(1+J_{11}s_1)(1+J_{12}s_2)(1+J_{14}s_4)\big) \times 1 \times 1,$$

$$\hat{\mu}_{2\to 4}(s_4) = \sum_{s_3,s_6} \big(1 - \frac{1}{8}(1+J_{23}s_3)(1+J_{24}s_4)(1+J_{26}s_6)\big) \times 1 \times 1,$$

$$\hat{\mu}_{3\to 4}(s_4) = \sum_{s_5,s_7} \big(1 - \frac{1}{8}(1+J_{34}s_4)(1+J_{35}s_5)(1+J_{37}s_7)\big) \times 1 \times 1$$

The binary sums are easily performed and yield $\hat{\mu}_{a\to 4}(s_4) = 4 - \frac{1}{2}(1 + J_{a4}s_4)$ for $a = 1, 2, 3$. In the next step we can compute the "marginal" for variable node 4 from the three incoming messages,

$$\mathcal{N}_4(s_4) = \mu_4(s_4) = (4 - \frac{1}{2}(1+J_{14}s_4))(4 - \frac{1}{2}(1+J_{24}s_4))(4 - \frac{1}{2}(1+J_{34}s_4)) \tag{5.18}$$

For example if the formula has $J_{14} = 1$, $J_{24} = 1$ and $J_{34} = -1$ the number of solutions with $s_4 = +1$ equals $\mathcal{N}_4(1) = 3 \times 3 \times 4 = 36$ and the number of solutions with $s_4 = -1$ equals $\mathcal{N}_4(-1) = 4 \times 4 \times 3 = 48$. The total number of solutions is $\mathcal{N}_0 = 36 + 48 = 84$. Note that we obtained this result without going through the remaining marginalization steps. This calculation also teaches us something about the uniform distribution over solutions. Indeed if we sample uniformly among solutions the probabilities that a solution has $s_4 = \pm 1$ are

$\mathcal{N}_4(\pm 1)/\mathcal{N}_0 = 3/7$ and $4/7$. We obtain this result from anaother point of view in the next paragraph. To calculate all such probabilities one has to go through the other marginalization steps. $\Diamond$

Message passing at positive and zero temperatures

Recall the Gibbs distribution in the finite temperature formulation of K-SAT

$$p(\underline{s}) = \frac{1}{Z}\sum_{\underline{s}}\prod_{a=1}^{m}\exp\{-\beta\prod_{i\in\partial a}(\frac{1+s_iJ_{ai}}{2})\}. \tag{5.19}$$

Again we associate a factor graph to this distribution with one type of factor attached to the clauses, namely $f_a(s_{\partial a}) = \exp\{-\beta\prod_{i\in\partial a}(\frac{1+s_iJ_{ai}}{2})\}$. We illustrate message passing on the same tree-like example as before.

EXAMPLE 18 (Belief propagation at positive temperature for 3-SAT) Consider again the 3-SAT formula shown on Fig. 5.8. The factors associated to the square notes are now teh β dependent weights entering in (5.19). Message passing originates at leaf nodes $1,2,3,5,6,7$ which send the trivial initial messages $\mu_{i\to 1}(s_i) = \mu_{i\to 2}(s_i) = \mu_{i\to 3}(s_i) = 1$, $i = 1,2,3,5,6,7$. In the second step all clauses send their message to variable node 4,

$$\hat{\mu}_{1\to 4}(s_4) = \sum_{s_1,s_2}\exp\{-\frac{\beta}{8}(1+J_{11}s_1)(1+J_{12}s_2)(1+J_{14}s_4)\}\times 1\times 1,$$

$$\hat{\mu}_{2\to 4}(s_4) = \sum_{s_3,s_6}\exp\{-\frac{\beta}{8}(1+J_{23}s_3)(1+J_{24}s_4)(1+J_{26}s_6)\}\times 1\times 1,$$

$$\hat{\mu}_{3\to 4}(s_4) = \sum_{s_5,s_7}\exp\{-\frac{\beta}{8}(1+J_{34}s_4)(1+J_{35}s_5)(1+J_{37}s_7)\}\times 1\times 1$$

Using $e^{-\beta n} = 1 + (e^{-\beta}-1)n$ for $n\in\{0,1\}$ we can easily calculate the binary sums. For example

$$\begin{aligned}\hat{\mu}_{1\to 4}(s_4) &= \sum_{s_1,s_2}\left(1+(e^{-\beta}-1)(\frac{1+J_{11}s_1}{2})(\frac{1+J_{12}s_2}{2})(\frac{1+J_{14}s_4}{2})\right)\\ &= 4+(e^{-\beta}-1)(\frac{1+J_{14}s_4}{2}).\end{aligned} \tag{5.20}$$

At this step we can already calculate the "marginal" $\mu_4(s_4)$ by multiplying all messages incoming into variable node 4

$$\begin{aligned}\mu_4(s_4) =&(4+(e^{-\beta}-1)(\frac{1+J_{14}s_4}{2}))(4+(e^{-\beta}-1)(\frac{1+J_{24}s_4}{2}))\\ &\times(4+(e^{-\beta}-1)(\frac{1+J_{34}s_4}{2}))\end{aligned} \tag{5.21}$$

and the true marginal is obtained as usual by normalization $\nu(s_4) = \mu_4(s_4)/(\mu_4(1)+\mu_4(-1))$. For the remaining marginals one has to perform extra message passing steps. $\Diamond$

Given a formula and given that solutions exist for this formula, when we take $\beta \to +\infty$ the Gibbs distribution tends to the uniform distribution over solutions. Therefore in the limit we have

$$\lim_{\beta\to+\infty} \nu_i(s_i) = \frac{\mathcal{N}_i(s_i)}{\mathcal{N}_0} \tag{5.22}$$

This is easily checked explicitely in the example above: using $e^{-\beta} \to 0$ in (5.21)) we find $\nu_4(\pm 1) = 3/7$ and $4/7$.

We now turn to the zero temperature case in more detail. Suppose we want to determine the assigments $\underline{s}$ that minimize the K-SAT Hamiltonian $\mathcal{H}(\underline{s})$ (**??**). When the graph associated to the formula is a tree message passing methods yield an exact solution; while in the non-tree case they form the basis of algorithms for finding solutions that we study at the end of this course (Survey Propagation). As for the LASSO estimator, we can take two alternative routes. We can directly set up the min-sum message passing rules by a proper use of the distributive law (5.15), or we can look at the $\beta \to +\infty$ limit of the BP rules. The second method is somehow more convenient for us since we have allready developped all the finite β formalism. This is illustrated with our running example.

EXAMPLE 19 (Zero temperature limit: min-sum for 3-SAT) We take the same 3-SAT formula as in Fig. 5.8. The correct limiting behavior of messages is captured by the definition (as for LASSO)

$$\hat{e}_{a\to i} = -\frac{1}{\beta}\ln\hat{\mu}_{a\to i}, \qquad \text{and} \qquad e_{i\to a} = -\frac{1}{\beta}\ln\mu_{i\to a}.$$

The initial messages from leaf nodes $1, 2, 3, 5, 6, 7$ are $e_{i\to 1}(s_i) = e_{i\to 2}(s_i) = e_{i\to 3}(s_i) = 0$, $i = 1, 2, 3, 5, 6, 7$. Next, all clauses send a message to variable node 4,

$$\begin{aligned}
\hat{e}_{1\to 4}(s_4) &= \min_{s_1,s_2}\Big(\Big(\frac{1+J_{11}s_1}{2}\Big)\Big(\frac{1+J_{12}s_2}{2}\Big)\Big(\frac{1+J_{14}s_4}{2}\Big) + 0 + 0\Big),\\
\hat{e}_{2\to 4}(s_4) &= \min_{s_3,s_6}\Big(\Big(\frac{1+J_{23}s_3}{2}\Big)\Big(\frac{1+J_{24}s_4}{2}\Big)\Big(\frac{1+J_{26}s_6}{2}\Big) + 0 + 0\Big),\\
\hat{e}_{3\to 4}(s_4) &= \min_{s_3,s_6}\Big(\Big(\frac{1+J_{34}s_4}{2}\Big)\Big(\frac{1+J_{35}s_5}{2}\Big)\Big(\frac{1+J_{37}s_7}{2}\Big) + 0 + 0\Big).
\end{aligned}$$

The minima are easily calculated directly from these expressions. For example testing all four possibilities $(s_1, s_2) = (\pm J_{11}, \pm J_{12})$ yields $\hat{e}_{1\to 4}(s_4) = 0$. This can also be obtained directly from (5.20). Similarly we have $\hat{e}_{2\to 4}(s_4) = \hat{e}_{3\to 4}(s_4) = 0$. The resulting "marginal" for variable node 4 vanishes for both values of $s_4 = \pm 1$, namely

$$e_4(s_4) = \hat{e}_{1\to 4}(s_4) + \hat{e}_{2\to 4}(s_4) + \hat{e}_{3\to 4}(s_4) = 0 \tag{5.23}$$

Since $e_4(s_4) = \min_{\sim s_4} \mathcal{H}(\underline{s})$ we deduce that any there exist zero energy assignments (so assignments that satisfy the formula) with both values $s_4 = \pm 1$.

$\Diamond$

Problems

5.1 Min-Sum Message Passing rules. In class we discussed how to compute the marginal of a multivariate function $f(x_1, \ldots, x_n)$ efficiently, assuming that the function can be factorized into factors involving only few variables and that the corresponding factor graph is a tree. We accomplished this by formulating a message-passing algorithm. The messages are functions over the underlying alphabet. Functions are passed on edges. The algorithm starts at the leaf nodes and we discussed how messages are computed at variable and at function nodes.

Recall from the derivation that the main property we used was the *distributive law*. Consider now the following generalization. Consider the so-called *commutative semiring* of extended real numbers (including ∞) with the two operations min and + (instead of the usual operations + and $*$).

(i) Show that both operations are commutative.

(ii) Show that the identity element under min is ∞ and that the identity element under + is 0.

(iii) Show that the distributive law holds.

(iv) If we formally exchange in our original marginalization + with min and $*$ with +, what corresponds to the marginalization of a function?

(v) What are the message passing rules and what is the initialization?

5.2 Application to the Lasso estimate. The goal of this problem is to show that in case the factor graph associated to the measurement matrix is a tree we can solve the Lasso minimization problem by using the min-sum algorithm. Recall that the Lasso estimate is

$$\underline{\hat{x}}^{\text{lasso}}(\underline{y}) = \text{argmin}_{\underline{x}} \left\{ \frac{1}{2} \|\underline{y} - A\underline{x}\|_2^2 - \lambda \|x\|_1 \right\}.$$

Consider first the minimum cost given that x_i is fixed.

$$E_i(x_i) = \min_{\sim x_i} \left\{ \frac{1}{2} \|\underline{y} - A\underline{x}\|_2^2 - \lambda \|x\|_1 \right\}.$$

where $\min_{\sim x_i}$ denotes minimization of the expression in the bracket with respect to all variables, except x_i which is held fixed. $E_i(x_i)$ is a function of a single real variable whose minimizer yields the i-th component of $\underline{\hat{x}}^{\text{lasso}}(\underline{y})$.

Consider the Tanner graph in Figure 6.7 in the notes and write down the factors associated to factor nodes. Pick your favourite variable, say variable 4, and describe the steps of the min-sum algorithm for the computation of $E_4(x_4)$.

6 Coding: Belief Propagation and Density Evolution

Message passing methods have been very successful in providing efficient and analyzable algorithms for the coding problem. In this and the next chapter we provide an introduction to this analysis. In the last lecture we learned how to marginalize a Gibbs distribution whose factor graph is a tree, by emloying by employing message passing rules. We saw that on trees message passing starts at the leaf nodes and that a node which has received messages from all its children processes the messages and forwards the result to its parent node. On a tree this messsage-passing algorithm is equivalent to MAP decoding since we are computing without any approximation the marginals of the posterior distribution. From now on we will refer to this algorithm as BP and leave the term "message-passing" as a generic term to encompass all local algorithms which follow the basic *message-passing* paradigm, i.e., where an outgoing message along an edge is only a function of the messages incoming at the same time along all *other* edges incident to the node.

If the graph is not a tree then we can still use BP, but we need to define a *schedule* which determines when to update what messages. It is not clear how well such an algorithm will perform. It is the aim of the present and the subequent chapter to clarify these issues. We will carry out the analysis in detail for the BEC and then explain how the general case can be treated. The BEC has the advantage that its analysis can be done by pen and paper. The general case is conceptually not much harder, but there are a signficant number of details which one has to take care of. This makes the analysis more difficult.

6.1 Message-Passing Rules for Bit-wise MAP Decoding

We illustrated the message passing rules for coding on a small coding example in Section 5.4. Recall that the Gibbs distribution has two type of factors: $e^{h_i s_i}$ and $\frac{1}{2}(1+\prod_{j\in\partial a} s_j)$. The first kind of factor is associated to a square nodes $\widehat{i}$ of degree one attached to variable nodes i and generates a message $\mu_{\widehat{i}\to i}(s_i) = e^{h_i s_i}$. The other relevant messages flow from the usual parity checks to variable nodes $\hat{\mu}_{a\to i}(s_i)$ and from variable nodes to usual parity checks $\mu_{i\to a}(s_i)$. Thus for coding

the general BP equations (5.9), (5.10) read

$$\mu_{i\to a}(s_i) = e^{h_i s_i} \prod_{b\in\partial i\setminus a} \hat{\mu}_{b\to i}(s_i) \tag{6.1}$$

$$\hat{\mu}_{a\to i}(s_i) = \sum_{\sim s_i} \frac{1}{2}(1 + \prod_{j\in\partial a} s_j) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(s_j) \tag{6.2}$$

In the binary case of interest here these equations can be simplified by adopting a convenient parametrization of the messages. Indeed we already remarked at the end of Section 5.3 that their normalizations cancel out in the final computation of "marginals". So all that should matter are the half-loglikelyhood ratios

$$l_{i\to a} = \frac{1}{2}\ln\left\{\frac{\mu_{i\to a}(+1)}{\mu_{i\to a}(-1)}\right\}, \qquad \hat{l}_{a\to i} = \frac{1}{2}\ln\left\{\frac{\hat{\mu}_{a\to i}(+1)}{\hat{\mu}_{a\to i}(-1)}\right\} \tag{6.3}$$

which do not involve the normalization. To see the form that the first BP equation (6.1) takes with this parametrization, we write this equation for each value $s_i = \pm 1$, take the ratio

$$\frac{\mu_{i\to a}(+1)}{\mu_{i\to a}(-1)} = e^{2h_i} \prod_{b\in\partial i\setminus a} \frac{\hat{\mu}_{b\to i}(+1)}{\hat{\mu}_{b\to i}(-1)}, \tag{6.4}$$

and then take the logarithm to obtain

$$l_{i\to a} = h_i + \sum_{b\in\partial i\setminus a} \hat{l}_{b\to i}. \tag{6.5}$$

Reducing the second BP equation (6.2) to a form involving only the loglikelihood ratios (6.3) involves a little more algebra. First we write (6.2) for each spin value $s_i = \pm 1$ and take the ratio,

$$\frac{\hat{\mu}_{a\to i}(+1)}{\hat{\mu}_{a\to i}(-1)} = \frac{\sum_{\sim s_i}(1 + \prod_{j\in\partial a\setminus i} s_j)\prod_{j\in\partial a\setminus i}\mu_{j\to a}(s_j)}{\sum_{\sim s_i}(1 - \prod_{j\in\partial a\setminus i} s_j)\prod_{j\in\partial a\setminus i}\mu_{j\to a}(s_j)}. \tag{6.6}$$

Next we divide the numerator and denominator by $\prod_{j\in\partial a\setminus i}\mu_{j\to a}(-1)$ and use the identity

$$\frac{\mu_{j\to a}(s_j)}{\mu_{j\to a}(-1)} = e^{l_{j\to a}(s_j+1)} = (\cosh l_{j\to a})(1 + s_j \tanh l_{j\to a}) \tag{6.7}$$

to obtain

$$\frac{\hat{\mu}_{a\to i}(+1)}{\hat{\mu}_{a\to i}(-1)} = \frac{\sum_{\sim s_i}(1 + \prod_{j\in\partial a\setminus i} s_j)\prod_{j\in\partial a\setminus i}(1 + s_j\tanh l_{j\to a})}{\sum_{\sim s_i}(1 - \prod_{j\in\partial a\setminus i} s_j)\prod_{j\in\partial a\setminus i}(1 + s_j\tanh l_{j\to a})}. \tag{6.8}$$

In order to perform the summations in the numerator and denominator we first expand the products into a sum of monomials of the spin variables

$$\begin{aligned}(1\pm \prod_{j\in\partial a\setminus i} s_j) \prod_{j\in\partial a\setminus i} (1+s_j \tanh l_{j\to a}) \\ = (1\pm \prod_{j\in\partial a\setminus i} s_j) \sum_{J\subset\partial a\setminus i} \prod_{j\in J} s_j \prod_{j\in J} \tanh l_{j\to a} \\ = \sum_{J\subset\partial a\setminus i} \prod_{j\in J} s_j \prod_{j\in J} \tanh l_{j\to a} \pm \sum_{J\subset\partial a\setminus i} \prod_{j\in J^c} s_j \prod_{j\in J} \tanh l_{j\to a} \end{aligned} \tag{6.9}$$

When we sum this expression over spin assignments the only monomials that survive correspond to the subsets $J=\emptyset$ in the first sum and $J^c=\emptyset$ in the second sum. Therefore the ratio (6.4) reduces to the simple form

$$\frac{\hat\mu_{a\to i}(+1)}{\hat\mu_{a\to i}(-1)} = \frac{1+\prod_{j\in\partial a\setminus i}\tanh l_{j\to a}}{1-\prod_{j\in\partial a\setminus i}\tanh l_{j\to a}} \tag{6.10}$$

Finally taking the logarithm and using $\frac{1}{2}\ln\frac{1+x}{1-x} = \operatorname{atanh} x$ we arrive at

$$\hat l_{a\to i} = \operatorname{atanh}\Big\{ \prod_{j\in\partial a\setminus i} \tanh l_{j\to a} \Big\} \tag{6.11}$$

Let us now look at the "marginals" computed from the BP equations. We will call them *BP marginals* and denote them by $\nu_i^{\mathrm{BP}}(s_i)$ to distinguish them from the true marginals $\nu_i(s_i)$ of the Gibbs distribution. As repeatedly pointed out on a tree the BP marginals and true marginals are the same. Adapting (5.11) to the present setting,

$$\nu_i^{\mathrm{BP}}(s_i) = \frac{e^{h_i s_i}\prod_{a\in i}\hat\mu_{a\to i}(s_i)}{e^{h_i}\prod_{a\in i}\hat\mu_{a\to i}(+1) + e^{-h_i}\prod_{a\in i}\hat\mu_{a\to i}(-1)} \tag{6.12}$$

In order to express the BP marginals in terms of the loglikehood ratios we divide the numerator and denominator by $e^{h_i}\prod_{a\in i}\hat m u_{a\to i}(+1)$ and use (6.3) to deduce

$$\begin{aligned}\nu_i^{\mathrm{BP}}(s_i) &= \frac{e^{(h_i+\sum_{a\in\partial i}\hat l_{a\to i})(s_i+1)}}{1+e^{2(h_i+\sum_{a\in\partial i}\hat l_{a\to i})}} \\ &= 1+s_i\tanh(h_i+\sum_{a\in\partial i}\hat l_{a\to i})\end{aligned} \tag{6.13}$$

From this marginal one can compute the *BP magnetization* of the i-th bit (to be distinguished from the true magnetization)

$$m_i^{\mathrm{BP}} = \sum_{s_i=0,1} s_i \nu_i^{\mathrm{BP}}(s_i) = \tanh(h_i+\sum_{a\in\partial i}\hat l_{a\to i}) \tag{6.14}$$

The *BP estimate* for bit i is then

$$\hat s_i^{\mathrm{BP}} = \operatorname{sign}(m_i^{\mathrm{BP}}) \tag{6.15}$$

There is a nice interpretation of (6.14). The BP magnetization is the same as that of a system constituted by a *single spin* with Gibbs distribution (at $\beta = 1$)

$$\frac{e^{-l_i s_i}}{2\cosh l_i}, \qquad l_i = h_i + \sum_{a \in \partial i} \hat{l}_{a \to i} \tag{6.16}$$

In the context of statistical mechanics the estimate l_i, for the total likelihood ratio associated to bit i, is called a *local mean magnetic field* or simply *local mean field*.

Summary of BP equations for coding

To summarize, in the case of transmission over a binary channel the messages can be compressed into a single real quantity. In particular, if we choose this quantity to be the half-loglikelihood ratio (6.3) then the processing rules take on a particularly simple form

$$\begin{cases} l_{i \to a} = h_i + \sum_{b \in \partial i \setminus a} \hat{l}_{b \to i} \\ \hat{l}_{a \to i} = \operatorname{atanh}\left\{ \prod_{j \in \partial a \setminus i} \tanh l_{j \to a} \right\} \end{cases} \tag{6.17}$$

The BP estimate of a bit is given by

$$\hat{s}_i^{\mathrm{BP}} = \operatorname{sign}(\tanh(h_i + \sum_{a \in \partial i} \hat{l}_{a \to i})) \tag{6.18}$$

For the special case of the BEC one can make further simplifications as dicussed in Section 6.3.

6.2 Scheduling on general Tanner graphs

If the Tanner graph is a tree, then message-passing starts from the leaf nodes and messages propagate through the graph until a message has been sent on each edge in both directions. However, cycle-free parity-check codes do not perform well. This is true even if we allowed optimal decoding. Hence we have to use codes whose Tanner graph has cycles.

Given a factor graph with cycles, the order in which messages are computed has to be defined explicitly and in principle different schedules might result in different performance. We call such an order a *schedule*. A naive scheduling which is convenient for analysis of belief propagation is the *flooding* or *parallel* schedule. In this schedule at each step every outgoing message is updated according to the incoming messages in the previous step.

In more details. Every iteration consists of two steps. In the first step we compute the outgoing messages along each edge at variable nodes and we forward them to the check node side. In the second step we then process the incoming messages at check nodes, and compute for every edge at check nodes the outgoing

messsage and send it back to variable nodes. What about the initial condition? At the very beginning, none of the messages except the ones coming from the channel are defined. So in order to get started, we set all "internal" messages to be "neutral" messages. E.g., if we represent messages as log-likelihood ratios, this means that we set all internal messages to 0. One can check that for a tree this prescription reduces to the initial conditions dictated by the theory developped in Chapter **??**.

Let us formalize the above discussion. Iterations are indexed by "time", a discrete integer $t \geq 1$. At iteration t in the first step we have messages flowing (in parallel) from variable to check nodes, $l_{i\to a}^{(t)}$, and in the second step we have messages flowing from check to variable nodes, $\hat{l}_{a\to i}^{(t)}$. They satisfy

$$\begin{cases} l_{i\to a}^{(t)} = h_i + \sum_{b\in\partial i\setminus a} \hat{l}_{b\to i}^{(t-1)} \\ \hat{l}_{a\to i}^{(t)} = \operatorname{atanh}\left\{ \prod_{j\in\partial a\setminus i} \tanh l_{j\to a}^{(t)} \right\} \end{cases} \tag{6.19}$$

The iterative process is initialized with $l_{i\to a}^{(0)} = \hat{l}_{a\to i}^{(0)} = 0$. The total estimated likelihood ratio for bit i at time t is

$$l_i^{(t)} = h_i + \sum_{a\in\partial i} \hat{l}_{a\to i}^{(t)} \tag{6.20}$$

and the BP estimate at time t for the bit is

$$\hat{s}_i^{\mathrm{BP,t}} = \operatorname{sign}(\tanh\, l_i^{(t)}) \tag{6.21}$$

6.3 Message Passing and Scheduling for the BEC

The BEC is a very special binary input memoryless channel. As depicted in Fig. 1.2, the transmitted bit is either correctly received at the channel output with probability $1-\epsilon$ or erased by the channel with probability ϵ and thus, nothing is received at the channel output.[1] The erased bits are denoted by "?". For example, if $s_i = 1$ (resp. $s_i = -1$) is transmitted in the BEC, then the set of possible channel observations is $\{1, ?\}$ (resp.$\{-1, ?\}$). The loglikelihood ratios corresponding to the various channel observations are

$$h_i = \log\left(\frac{p(y_i \mid s_i = 1)}{p(y \mid s_i = -1)}\right) = \begin{cases} \frac{1}{2}\log(\frac{1-\epsilon}{0}) = +\infty & y = 1, \\ \frac{1}{2}\log(\frac{\epsilon}{\epsilon}) = 0, & y = ?, \\ \frac{1}{2}\log(\frac{0}{1-\epsilon}) = -\infty, & y = -1. \end{cases}$$

Now, since the initial condition for the internal messages is $l_{i\to a}^{(0)} = 0, \hat{l}_{a\to i}^{(0)} = 0$ the BP equations (6.19) imply that at later times $l_{i\to a}^{(t)} = 0, \hat{l}_{a\to i}^{(t)}\{\pm\infty, 0\}$. This allows to further simplify the BP equations.

According to the variable-node rule the outgoing message from a variable node

[1] But note that the position of the erased bit is known.

is $+\infty$ (or $-\infty$) if at least one incoming message from one of its neighbors is $+\infty$ (or $-\infty$), otherwise it is equal to 0. Note that it is not possible that a variable node receives both $+\infty$ and $-\infty$ simultaneously. This is due to the fact that by assumption the transmitted word is a valid codeword and that the channel never introduced mistakes.

Since $\tanh l_{i\to a} \in \{\pm 1, 0\}$, we can use $\tanh l_{i\to a} = \operatorname{sign}(l_{i\to a})$ to simplify the updating rule of check nodes to the following equation,

$$\operatorname{sign}(\hat{l}_{a\to i}) = \prod_{j\in\partial a\setminus i} \operatorname{sign}(l_{j\to a}). \tag{6.22}$$

This discussion shows that on the BEC, knowing the sign of all incoming messages is sufficient to compute outgoing messages, thus we can assume that the set of messages is $\{\pm 1, 0\}$ instead of $\{\pm\infty, 0\}$. At check nodes the operation is then simple multiplication. At variable nodes, if at least one of the incoming edges is non-zero, then all non-zero incoming messages must in fact be the same and the outgoing message is this common value. Otherwise, when all incoming messages are 0, the outgoing message is also 0.

For the BEC, but only for the BEC, we can implement the parallel schedule in a more efficient manner. For this channel, some thought shows that the messages emitted along a particular edge can only jump once, namely from 0 to either the value $+1$ or -1. After the value has jumped it stays constant thereafter. Further, the message can only jump if at least one of the incoming messages jumped. Therefore, rather than recomputing every message along every edge in each iteration, we can just follow changes in the messages and see if they have consequences. As a consequence, we have to "touch" every edge only once and so the complexity of this algorithm scales linearly in the number of edges.

6.4 Two Basic Simplifications

To analyze the performance of the (l, r)-regular LDPC ensemble over a channel, we pick a code $\mathcal{C}$ uniformly at random from the ensemble of graphs and run the message passing algorithm. For a given code $\mathcal{C}$ and channel parameter ϵ, let $P_{\mathrm{BP,b}}(\mathcal{C}, \underline{s}^{\mathrm{in}}, \epsilon, t)$ denote the average bit error probability of the message passing decoder for codeword $\underline{s}^{\mathrm{in}}$ at iteration t. Explicitely,

$$P_{\mathrm{BP,b}}(\mathcal{C}, \underline{s}^{\mathrm{in}}, \epsilon, t) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}(1 + \mathbb{E}_{\underline{h}|\underline{s}^{\mathrm{in}}}[s_i^{\mathrm{in}} \hat{s}_i^{\mathrm{BP,(t)}}]) \tag{6.23}$$

where we recall that $\mathbb{E}_{\underline{h}|\underline{s}^{\mathrm{in}}}$ is the expectation with respect to channel outputs given the input word (see Chapter 3). We will study the behavior of $\mathbb{P}_{\mathrm{BP,b}}(\mathcal{C}, \underline{s}^{\mathrm{in}}, \epsilon, t)$ in terms of ϵ and t as a measure of performance of the code $\mathcal{C}$.

For the binary erasure channel, we either can decode a bit correctly, or the bit is still erased at the end of the decoding process. Therefore, in this case

we typically compute the bit erasure probability. If we want to convert this into an error probability, then we can imagine that for all erased bits we flip a coin uniformly at random. With probability one-half we will guess the bit correctly and with probability one-half we will make a mistake. Therefore, the bit erasure and the bit error probability are the same up to a factor of one-half. In our calculations we will always compute the erasure probability for the erasure channel. But our language will sometimes reflect the general case and so we will talk about error probabilities.

Restriction To The All-One Codeword

In Chapter 3 we showed that the bit-wise MAP error probability is independent of the transmitted codeword as long as the channel is symmetric. Something similar holds for the BP decoder. Therefore we can analyze the error probality of the BP decoder assuming that the all-one codeword was transmitted (i.e., the codeword, all of its components are 1, in the spin language where the components are from the set $\{\pm 1\}$). In formulae, we claim that (recall $\mathbb{E}_{\underline{h}} = \mathbb{E}_{\underline{h}|\underline{1}}$)

$$\begin{aligned} P_{\mathrm{BP,b}}(\mathcal{C}, \underline{s}^{\mathrm{in}}, \epsilon, t) &= P_{\mathrm{BP,b}}(\mathcal{C}, \epsilon, t) \\ &= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} (1 - \mathbb{E}_{\underline{h}}[\hat{s}_i^{\mathrm{BP},(\mathrm{t})}]) \end{aligned} \tag{6.24}$$

This is true in a more general setting than the present one. In general, for the statement to hold we need two kinds of symmetry to hold: channel symmetry and decoder symmetry. Decoder symmetry here means that at check nodes the magnitude of the outgoing message is only a function of the magnitude of the incoming messages, and that the sign of the outgoing message is the product of the signs of the incoming messages. At variable nodes, we require that if the signs of all the incoming messages are reversed then the outgoing message also just changes by a reversal of the sign. This is obviously the case for the BP decoder. But often one often implements simplified versions for which the symmetry conditions also hold.

For the BEC and BP decoding it is particularly easy to see why (6.24) is true. If you go back to the message-passing rules for this case, you will see that both at check nodes as well as at variable nodes we can determine if the outgoing message is an erasure or not by only looking how many of the incoming messages are erasures, but we do not need to know the values of the incoming messages. Therefore, the final erasue probability only depends on the erasure pattern created by the channel, but is independent of the transmitted codeword.

The general case is proved by using the two symmetry conditions stated above. The proof is not very difficult and we leave it to the reader.

Concentration

The second major simplification stems from the fact that, rather than analyzing individual codes, it suffices to assess the ensemble average performance. When this is true the individual behavior of elements of an ensemble is with high probability close to the ensemble average. More precisely one can prove the following statement [**?**].

Let $\mathcal{C}$, chosen uniformly at random from the Gallager ensemble LDPC(d_v, d_c, n), be used for transmission over a BMS channel. Then, for any given $\delta > 0$, there exists an $\alpha > 0$, $\alpha = \alpha(d_v, d_c, \delta)$, such that

$$\mathbb{P}\{|P_{\mathrm{BP,b}}(\mathcal{C}, \epsilon, t) - \mathbb{E}\left[P_{\mathrm{BP,b}}(\mathcal{C}, \epsilon, t)\right]| > \delta\} \leq \epsilon^{-\alpha n}. \tag{6.25}$$

where here $\mathbb{P}$ and $\mathbb{E}$ refer to the code ensemble.

In words, all except an exponentially (in the blocklength) small fraction of codes behave within an arbitrarily small δ from the ensemble average. Therefore, assuming sufficiently large blocklengths, the ensemble average is a good indicator for the individual behavior and it seems a reasonable route to focus one's effort on the design and construction of ensembles whose average performance approaches the Shannon theoretic limit.

6.5 Concept of Computation Graph

Message passing takes place on the local neighborhood of a node. At each iteration, variable nodes send their beliefs along their edges toward check nodes and, then, the check nodes compute the outgoing message for each of their edges according to the beliefs of incoming edges and send it back to the variable nodes. Afterwards, each variable node updates the outgoing messages along its edges according to beliefs returned back on its edges.

Therefore, after t iterations, the belief of a variable node depends on its initial belief and the beliefs of all the nodes placed within (graph) distance $2t$ or less. The graph consisting of these nodes is called the computation graph of that variable node of height t. For example, the factor graph of a $(2, 4, 6)$-regular LDPC code is shown in Fig. 6.1(a) and the computation graph of node 1 with height 1 is also depicted in Fig. 6.1(b).

If a computation graph is tree, then no node is used more than once in the graph. Therefore the incoming messages of each node are independent. But note that by increasing the number of iterations, the number of nodes in a computation graph grows exponentially and thus in at most $c \log n$ steps, where c is some suitable constant, some node will necessarily be reused. It is clear that small computation graphs are more likely to be tree-like than large ones and that the chance of having a tree-like computation tree increase if we increase the blocklength.

Let us discuss this last point in more detail. Let T_t denote the computation

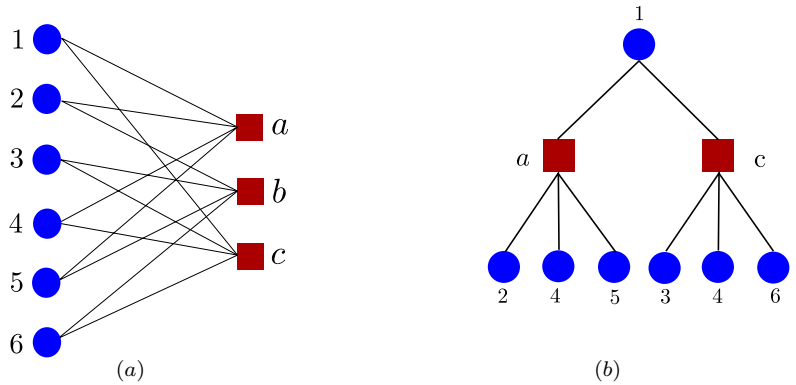


Figure 6.1 (a) The Tanner graph of a $(2, 4)$-regular LDPC code with 6 variable nodes; (b) The corresponding computation graph of node 1 for the first iteration.

graph of a variable node chosen uniformly at random from the set of variable nodes of height t in the (d_v, d_c, n)-regular LDPC ensemble. If the height t is kept fixed then

$$\lim_{n\to\infty} \mathbb{P}\left(T_t \text{ is a tree }\right) = 1. \tag{6.26}$$

We only give a sketch of the proof. We are given the randomly chosen variable node and we construct its computation graph of height t by growing out its "tree" one node at a time, breath first. We use the principle of *deferred decisions*. This means that rather than first constructing a particular code, then checking if the correspoding computation graph is a tree and then averaging over all codes we perform the averaging over all codes at the same time as we grow the tree, i.e., we *defer* the decision of how edges are connected until we look at a particular edge and reveal its endpoints. Note that a computation graph of a fixed height has at most at certain number of nodes and edges in there. At each step when we reveal how a particular edge is connected there are two possible events. The newly inspected edge is either connected to a node which is already contained in the computation graph. In this case we terminate the procedure since we know that the computation graph is not a tree. Or the edge is connected to a new node, maintaining the tree structure. Since not yet revealed edges are connected uniformly at random to any not yet filled slot, the probability of reconnecting to an already visited node vanishes like $1/n$, where n is the blocklength. By the union bound, and since we only perform a fixed number of steps, it follows that the probability that the computation graph is indeed a tree behaves like $1 - O(1/n)$, which proves the claim.

6.6 Density Evolution

We will now show how to compute the bit error probability under BP decoding. Expression (6.24) shows that in principle, given a code $\mathcal{C}$ from the ensemble, and a variable node i selected uniformly at random, we should compute the expectation of $\hat{s}_i^{\mathrm{BP},(t)}$. According to (6.21) we should determine the probability distribution of $l_i^{(t)}$. A priori the difficulty here is that this depends on messages that are not independent. But, fortunately the results in sections **??** and 6.5 allow to by-pass this problem at least in the limit where n grows large and t is fixed (but arbitrarily large).

From the concentration of the error probability (6.25) in the large block-length limit it suffices to compute the average over the code ensemble of the error probability,

$$P_{\mathrm{BP,b}}(d_v, d_c, \epsilon, t) = \lim_{n \to +\infty} \mathbb{E}[P_{\mathrm{BP,b}}(\mathcal{C}, \epsilon, t)] \tag{6.27}$$

Since the computation graph T_t of a random vertex of fixed height t is a tree with probability $1 - O(1/n)$ we get

$$P_{\mathrm{BP,b}}(d_v, d_c, \epsilon, t) = \lim_{n \to +\infty} \mathbb{E}[P_{\mathrm{BP,b}}(\mathcal{C}, \epsilon, t) | T_t \text{ is a tree }] \tag{6.28}$$

Our task is therefore reduced to the computation of the probability distribution of $l_i^{(t)}$ on a tree. This problem can handled quite easily, at least in principle, because the incoming messages to each node of this tree graph are independent.

It is common to refer to the iterative equations governing the probability distributions on the tree as the *Density Evolution* (DE) equations. For the BEC these are a simple set of algebraic (polynomial) equations and we first give their derivation in this simple case. For general BMS channels these are integral equations, but as we will see conceptually their derivation is not much more difficult.

DE equations for the BEC

Consider a computation *tree* T_t with height t. We divide this computation graph to $t+1$ levels, from 0 to t. Level 0 contains the leaf nodes and the 1st level contains the parent check nodes and the grandparent variable nodes of the leaf nodes (Fig. 6.2).

Every variable node at the ℓ-th level is the root of a computation tree with height ℓ. However, its root has degree $d_v - 1$. Consider $\{0, +1, -1\}$ the outgoing message emitted by a variable nodes towards its parent check node in the $\ell+1$-th level. It is equal to either 0 (erasure message) with probability x_ℓ or a known value (± 1) with probability $1 - x_\ell$.

Now consider level $\ell+1$. Each variable node is connected to $d_v - 1$ check nodes and each check node is connected to $d_c - 1$ variable nodes of ℓ-th level. Consider $\{0, +1, -1\}$ the outgoing message emitted by a check node towards its parent

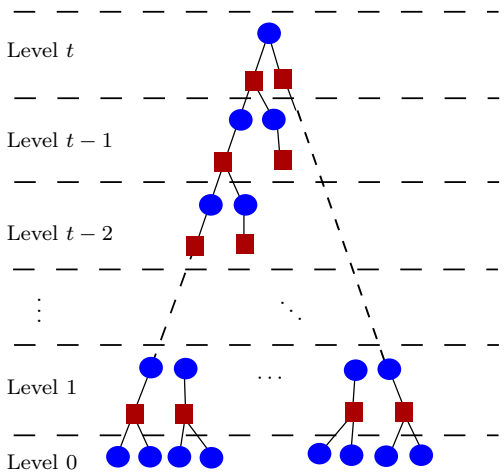


Figure 6.2 A computation graph of $(2,3)$-regular LDPC code with height t. The graph is split to $t+1$ levels.

variable node in the same level. We call y_ℓ the probability that this message is an erasure.

The outgoing message of a check node is an erasure message, if at least one of its incoming messages is 0. Since the incoming messages are independent, then the probability that a check node at level $\ell+1$ sends an erasure message to its parent variable node is

$$y_\ell = 1-(1-x_\ell)^{d_c-1} \tag{6.29}$$

The outgoing message from a variable node of $\ell+1$-th level, i.e. $x_{\ell+1}$, is erasure message if its initial message from the channel is erasure message and all of its children (check nodes) at level $\ell+1$ also send erasure messages. Moreover the incoming messages are independent, hence

$$x_{\ell+1} = \epsilon y_\ell^{d_c-1} \tag{6.30}$$

These are the two DE equations for the BEC, and of course they can be merged into a single one

$$x_{\ell+1} = \epsilon(1-(1-x_\ell)^{d_c-1})^{d_v-1} \tag{6.31}$$

By definition, the outgoing message at level 0 is an erasure with probability $x_0=\epsilon$. Therefore, the erasure probability of the root of T which is connected to d_v check nodes of level t is

$$\mathbb{P}_{\mathrm{BP,b}}(d_v,d_c,\epsilon,t) = \epsilon(1-(1-x_{t-1})^{d_c-1})^{d_v}. \tag{6.32}$$

In section 6.7 we will analyze the DE equation and draw conclusions for the error probability of the BP decoder.

DE equations for general BMS channels

Luckily it turns out that exactly the same type of analysis works for general BMS channels. The DE equations for the BEC (6.29), (6.30) are "polynomial equations" relating probabilities x_ℓ, y_ℓ of the erasure messages. They also involve the channel erasure probability ϵ For the general case, the DE equations are "integral equations" relating two probability distributions for the messages of type $l_{i\to a}$ and $\hat{l}_{a\to i}$ after a certain number of iterations. Besides they involve the channel distribution $c(h)$. we will pretend that all distributions have densities. This is not really true and it is important to take into account probability distributions which are convex combinations of densities and point masses. However, practically, this makes no difference in the formalism except for introducing technicalities that only serve to obscure the picture.

Not very surprisingly, the DE equations will involve two types of "convolution" operations over probability distributions. The first one is the standard convolution. Let l_1 and l_2 be two independent random variables with distributions $a_1(l)$ and $a_2(l)$; then their sum $l = l_1 + l_2$ is distributed as

$$(a_1 \otimes a_2)(l) = \int_{\mathbb{R}^2} dl_1 a(l_1) dl_2 a(l_2) \delta(l - (l_1 + l_2)) \tag{6.33}$$

The second type of convolution is denoted by $\boxplus$ and is given by the distribution of $l = \operatorname{atanh}(\tanh l_1 \tanh l_2)$,

$$(a_1 \boxplus a_2)(l) = \int_{\mathbb{R}^2} dl_1 a(l_1) dl_2 a(l_2) \delta(l - \operatorname{atanh}(\tanh l_1 \tanh l_2)) \tag{6.34}$$

It is clear that $\otimes$ convolution is commutative and associative and that the neutral element is $a(l) = \delta(l)$. We leave it as an exercise to the reader to show that $\boxplus$ is also commutative, associative and that the neutral element is $a(l) = \Delta_\infty(l)$ the unit mass at infinity. However the two operations do not "mix" well together in the sense that $(a_1 \otimes a_2) \boxplus a_3 \neq a_1 \otimes (a_2 \boxplus a_3)$. Finally let us point out that if we are willing to bring all the random variables into a different domain, then again we can write the $\boxplus$ operation as a usual convolution. We will not pursue this further here. For our purpose it sufficies to know that there are computationally efficient ways of computing these convolutions.

We are ready to derive the DE equations. Consider again the computation *tree* T_t with height t, with the division into $t+1$ levels, from 0 to t as before (Fig. 6.2). Look at level $\ell + 1$. At a variable node, the incoming messages are independent (real valued) random variables sent by the $d_v - 1$ children check nodes. Let these messages be $\hat{l}_1, \ldots, \hat{l}_{d_v - 1}$ and their common distribution $y_\ell(\hat{l})$. The BP equations tell us that the outgoing message from the variable node to the check node (both at level $\ell + 1$) is

$$l = h + \hat{l}_1 + \cdots + \hat{l}_{d_v - 1}$$

Let $x_{\ell+1}(l)$ denote the probability distribution of the outgoing message. Since the outgoing random variable is the sum of a fixed number of independent random

variables, the density of the outgoing random variable is the convolution of the densities of the incoming random variables, i.e.,

$$x_{\ell+1} = c \otimes y_\ell^{\otimes d_v - 1} \tag{6.35}$$

Here we use the notation $y_\ell^{\otimes d_v-1}$ for $y_l \otimes \cdots \otimes y_l$ convolved $d_v - 1$ times. This equation is the analog of (6.30). Now we seek an equation for y_ℓ in terms of x_ℓ. At check nodes of level $\ell+1$ the incoming messages are $d_c - 1$ independent random variables coming from the children variable nodes of level ℓ. Call the random messages $l_1, \cdots, l_{d_c-1}$ and denote their probability distribution by $x_\ell(l)$. From the BP equations the outgoing message from check nodes to the variable node (both at level $\ell + 1$) is

$$\hat{l} = \operatorname{atanh}\left(\prod_{i=1}^{d_c-1} \tanh l_i)\right)$$

and we have for the probability densities

$$y_\ell = x_\ell^{\boxplus d_c - 1} \tag{6.36}$$

As above, we use the notation $x_\ell^{\boxplus d_c-1}$ for $x_l \boxplus \cdots \boxplus x_l$ convolved $d_c - 1$ times. This equation is the analog of (6.29).

Equations (7.29) and (7.30) are the DE equations for general BMS channel. Combining them into a single equation yields the so-called *density evolution equation*

$$x_{l+1} = c \otimes (x_\ell^{\boxplus d_c - 1})^{\otimes d_v - 1} \tag{6.37}$$

We can now compute the bit-wise probability of error of the BP decoder. In the final step the BP algorithm computes teh loglikehood ratio associated to the root node as a sum of all messages incoming from d_v children check nodes plus the one coming from the channel

$$l = h + l_1 + \cdots l_{d_v}$$

Since all messages are independent on the computation tree the distribution of l is equal to $c \otimes (y_{t-1})^{\otimes d_v}$, or

$$c \otimes (x_{t-1}^{\boxplus d_c - 1})^{\otimes d_v} \tag{6.38}$$

From (6.24) and (6.21) we see that the errors come from the events $\operatorname{sign}(\tanh l) = -1$, in other words $l < 0$. Thus

$$\mathbb{P}_{\mathrm{BP,b}}(d_v, d_c, \epsilon, t) = \int_{-\infty}^{0} dl\, (c \otimes (x_{t-1}^{\boxplus d_c - 1})^{\otimes d_v})(l) \tag{6.39}$$

6.7 Analysis of DE Equations for the BEC

We have seen that the bit probability of error of the BP decoder (6.32) can be computed from the DE recursions (6.31). We will show here that a threshold

phenomenon appears. Namely there is a noise threshold ϵ_{BP}, called the BP-threshold, such that for $\epsilon < \epsilon_{\mathrm{BP}}$ the limit of $\mathbb{P}_{\mathrm{BP,b}}(d_v, d_c, \epsilon, t)$ when the number of iterations $t \to +\infty$ vanishes, while for $\epsilon > \epsilon_{\mathrm{BP}}$ this limit remains strictly positive.

In order to compute $\lim_{t\to+\infty} \mathbb{P}_{\mathrm{BP,b}}(d_v, d_c, \epsilon, t)$ we have to analyze the recursion $x_t = f(\epsilon, x_{t-1})$ where

$$f(\epsilon, x) = \epsilon(1 - (1-x)^{d_c-1})^{d_v-1} \tag{6.40}$$

and the initial condition is $x_0 = 1$ (or equivalently $x_0 = \epsilon$). We ask whether the sequence $\{x_t\}$ converges to 0 or not. In case it does, the decoding is successful, otherwise it is not.

Note that the function $f(\epsilon, x)$ is increasing in ϵ and x for $x, \epsilon \in [0, 1]$. This is key to prove the following.

LEMMA 6.1 *Let $2 \le d_v \le d_c$ and $0 \le \epsilon \le 1$. Let $x_0 = 1$ and $x_t = f(\epsilon, x_{t-1})$, $t \ge 1$. Then (a) The sequence $\{x_t\}$ is decreasing in t; (b) If $\epsilon \le \epsilon'$ then $x_t(\epsilon) \le x_t(\epsilon')$.*

Proof Let us first show thaat the sequence $\{x_t\}$ is decreasing. We use induction. The first two elements of the sequence are $x_0 = 1$ and $x_1 = f(\epsilon, x_0) = \epsilon$, so $x_0 \ge x_1$. Therefore, for $t \ge 2$, we assume $x_{t-1} \le x_{t-2}$ as the induction hypothesis. Since $f(\epsilon, x)$ is increasing in x, we obtain $f(\epsilon, x_{t-1}) \le f(\epsilon, x_{t-2})$. The left hand side is equal to x_t, and the right hand side to x_{t-1}, and we deduce that $x_t \le x_{t-1}$. To prove the second claim, we use induction once more. Assume that $\epsilon \le \epsilon'$. Then $x_1(\epsilon) = \epsilon \le \epsilon' = x_1(\epsilon)$. The general statement is deduced as follows:

$$x_t(\epsilon) = f(\epsilon, x_{t-1}(\epsilon)) \le f(\epsilon', x_{t-1}(\epsilon)) \le f(\epsilon\prime, x_{t-1}(\epsilon')) = x_t(\epsilon'), \tag{6.41}$$

where the first inequality follows from the fact that $f(\epsilon, x)$ is increasing in ϵ, and the second inequality follows from it being increasing in x, together with the induction hypothesis. □

From the first part of the previous lemma, it follows that $x_t(\epsilon)$ converges to a limit in $[0, 1]$, $\lim_{t\to+\infty} x_t(\epsilon) = x_\infty(\epsilon)$. From teh continuity of the function (6.40) we conclude that the limit of the density eveolution iterations is a solution of the fixed point equation

$$x_\infty(\epsilon) = f(\epsilon, x_\infty(\epsilon)). \tag{6.42}$$

From the second part of the lemma, it follows that if $x_t(\epsilon) \to 0$ for some ϵ, then $x_t(\epsilon') \to 0$ for all $\epsilon' < \epsilon$. Let $x_\infty(\epsilon) = \lim_{t\to\infty} x_t(\epsilon)$. Then $x_\infty(\epsilon)$, as well as the error probability

$$\lim_{t\to+\infty} \mathbb{P}_{\mathrm{BP,b}}(d_v, d_c, \epsilon, t) = \epsilon(1 - (1 - x_\infty(\epsilon))^{d_v-1})^{d_c}, \tag{6.43}$$

are increasing in ϵ as shown in Figure 6.3. Hence we can define the quantity

$$\epsilon^{\mathrm{BP}} = \sup\{\epsilon : x_\infty(\epsilon) = 0\}$$

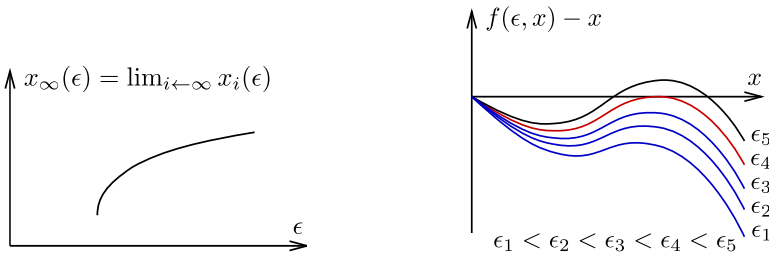


Figure 6.3 *Left*: Monotonicity of x_∞ as a function of ϵ. For $d_v \geq 3$, $d_c > d_v$, x_∞ jumps at the threshold. For $d_v = 2$, $d_c > d_v$ x_∞ changes continuously at the threshold. *Right*:The threshold ϵ_{BP} is the largest channel parameter so that $f(\epsilon, x) - x < 0$ for the whole range $x \in [0, 1]$.

which we call *the BP threshold*.

There is a graphical way to characterize this threshold. Note that $x_\infty(\epsilon)$ is a solution of the fixed point equation $x = f(\epsilon, x)$. Thus, if $f(\epsilon, x) - x < 0$ for all $x \in [0, \epsilon]$, then $x_\infty(\epsilon) = 0$. For the converse, as soon as there is a fixed point $f(\epsilon, x) = x$ in the interval $]0, \epsilon]$, we have that $x_\infty > 0$. In fact it is easy to check that this condition can be further simplified since there never can be a fixed point in $]\epsilon, 1]$ as $f(\epsilon, x) < \epsilon$. Therefore, if $f(\epsilon, x) - x < 0$ for all $x \in [0, 1]$, then $x_\infty = 0$. For the converse, as soon as there is a fixed point $f(\epsilon, x) = x$ in the interval $]0, 1]$, we have that $x_\infty(\epsilon) > 0$. This condition is graphically depicted in Figure 6.3.

EXAMPLE 20 For the $(3, 6)$-regular ensemble, we get $\epsilon_{\text{BP}} \approx 0.4294$. Note that the rate of this ensemble is $R = 1 - \frac{d_v}{d_c} = \frac{1}{2}$. Therefore, the fraction 0.4294 has to be compared to the erasure probability that an optimum code (say, a random linear code) could tolerate, which is $\epsilon_{\text{Shannon}} = 1 - R = \frac{1}{2}$. We conclude that already this very simple code, together with this very simple decoding procedure can decode up to a good fraction of Shannon capacity.

6.8 Analysis of DE equations for general BMS channels

This section is not needed for the main development of these notes and can be skipped in a first reading.

The elementary analysis for the BEC can be extended to the class of general symmetric channels. Although the main ideas are the same, the functional nature

of the DE equation (6.37)

$$x_{t+1} = c \otimes f(c, x_t), \qquad f(c, x) = c \otimes (x^{\boxplus d_c - 1})^{\otimes d_v - 1} \tag{6.44}$$

makes the analysis technically more challenging. Here we give a brief version of the theory, and refer to **??** for a thorough development.

Ordering by degradation of symmetric distributions

The analysis for the BEC rests on the monotonicity in ϵ and x of the function $f(\epsilon, x)$. We will need analogous properties for the functional on the right hand side of the DE recursion (6.37). The key is to introduce a partial order relation between distributions.

We already noted that the DE equations preserve the symmetry property of the initial channel distribution. In other words when we initialize the DE recursion with $x_0(l) = c(l)$, which satisfies the symmetry condition $c(l) = e^{-2l}c(-l)$, we have for all $t \geq 1$, $x_{t+1}(l) = e^{-2l}x_t(-l)$. For this reason, we may restrict ourselves to the space of "symmetric distributions" satisfying $a(l) = e^{-2l}a(-l)$.

Let $M_k(a) = \int dla(l)(\tanh l)^k$. It is not difficult to see that the symmetry condition for a implies

$$\int dla(l)(\tanh l)^{2k-1} = \int dla(l)(\tanh l)^{2k} \tag{6.45}$$

for all integers $k \geq 1$. Symmetric distributions can be entirely characterized by their even moments: if two symmetric distributions a and b have the *sam* set of even moments, $M_{2k}(a) = M_{2k}(b)$, then they must be equal. Indeed, by the symmetry condition their odd moments are also equal, and since all moments are less than 1, Carleman's criterion is satisfied; thus one can reconstruct a unique measure from the set of even moments and $a = b$.

Let us now define *ordering by degradation*. We say that a_2 is degraded with respect to a_1, and write $a_2 \succ a_1$ if an only if $M_{2k}(a_2) \leq M_{2k}(a_1)$ for all $k \in \mathbb{N}^*$. The following example gives the intuitive meaning of this concept.

EXAMPLE 21 Consider the likelihood distribution of the BEC channel $c_\epsilon(h) = \epsilon\delta(h) + (1-\epsilon)\Delta_\infty(h)$. Note that it is symmetric and that the moments are $M_{2k} = M_{2k-1} = 1-\epsilon$ for $k \geq 1$. Take two channels c_{ϵ_1} and c_{ϵ_2} with $\epsilon_2 > \epsilon_1$. According to our definition we have $c_{\epsilon_2} \succ c_{\epsilon_1}$ because $1-\epsilon_2 < 1-\epsilon_1$; in other words "$c_{epsilon_2}$ is degraded with respect to c_{ϵ_1}" means that "$c_{epsilon_2}$ is more noisy than c_{ϵ_1}". We leave it as an exercise to the reader to show that the same interpretation applies to our other basic symmteric channels, the BSC and BAWGNC.

As a side remark note that we can associate a "symmetric channel" to any symmetric distribution a. The idea is to think of the distribution as a the "likelihood distribution" of some channel. Explicitely, The transition probability of the channel can be explicitely calculated through the identities $p(y|+1)dy = a(l)dl$ and $p(y|-1)dy = a(-l)dl$ where $l = \frac{1}{2}\ln\frac{p(y|+1)}{p(y|+1)}$. There is a nice characterization

of the relation $a_2 \succ a_1$ in terms of the associated channels $p_2(y|x)$ and $p_1(y|x)$. Namely there exists a channel $q(y|x)$ such that $p_2(z|x) = \sum_y q(z|y)p_1(y|x)$. In other words the channel associated to a_2 is more noisy than the one associated to a_1.

Ordering by degradation is preserved under the two convolutions operations $\otimes$ and $\boxplus$. More precisely if $a_1 \succ a_2$ and b are symmetric distributions we have: $a_2 \otimes b \succ a_1 \otimes b$, $b \otimes a_2 \succ b \otimes a_1$ and $b \boxplus a_2 \succ b \boxplus a_1$. The proof of these assertions is the subject of an exercise.

Entropy distance, entropy functional and moment expansions

For the BEC, besides monotonicity of $f(\epsilon, x)$, an important ingredient was the continuity of the function with respect to ϵ and x. Here we introduce a suitable distance in the space of symmetric distributions that allows to prove analogous statements. We do not wish to introduce sophisticated topological language here and we proceed in a pedestrian way that will be sufficient for our purposes.

For any two symmetric distributions a and b define

$$d(a,b) = \sum_{k \geq 1} \frac{|M_{2k}(a) - M_{2k}(b)|}{2k(2k-1)} \tag{6.46}$$

It is easy to see that this is a well defined distance, i.e. it is symmetric, satisfies the triangle inequality and vanishes if and only if $a = b$. We call it the *entropy distance* because there is a natural relation with an *entropy functional*.

This entropy functional is defined as

$$H[x] = \int dl\, x(l) \ln(1 + e^{-2l}) \tag{6.47}$$

This is precisely the Shannon entropy $H(Y|X)$ corresponding to a symmetric channel whose likelihood distribution is $x(l)$. Using $\ln(1 + e^{-2l}) = \ln 2 - \ln(1 + \tanh l)$, expanding the logarithm in powers of $\tanh h$, and using the equality of even and odd moments we get the *moment expansion*

$$H[x] = \ln 2 - \sum_{k=1}^{+\infty} \frac{M_{2k}(x)}{2k(2k-1)} \tag{6.48}$$

We now collect a few useful tricks that will allow to efficiently use these quantities in the analysis of the DE recursion. By linearity of this entropy functional

$$H[a - b] = -\sum_{k=1}^{+\infty} \frac{M_{2k}(a) - M_{2k}(b)}{2k(2k-1)} \tag{6.49}$$

In particular when $a \succ b$ we have $M_{2k}(a) < M_{2k}(b)$ and therefore

$$d(a,b) = H[a - b], \quad \text{if} \quad a \succ b. \tag{6.50}$$

The following inequalities are handy; for $a \succ b$ and any x symmetric

$$H[x \otimes (a-b)] \leq H[a-b], \qquad H[x \boxplus (a-b)] \leq H[a-b] \tag{6.51}$$

To prove the second inequality we use the moment expansion and the fact that moments are multiplicative for the $\boxplus$ operation, $M_{2k}(a \boxplus b) = M_{2k}(a)M_{2k}(b)$,

$$\begin{aligned} H[x \boxplus (a-b)] &= -\sum_{k=1}^{+\infty} \frac{M_{2k}(x \otimes a) - M_{2k}(x \otimes b)}{2k(2k-1)} \\ &= \sum_{k=1}^{+\infty} M_{2k}(x) \frac{M_{2k}(b) - M_{2k}(a)}{2k(2k-1)} \\ &\leq \sum_{k=1}^{+\infty} \frac{M_{2k}(b) - M_{2k}(a)}{2k(2k-1)} \\ &= H[a-b] \end{aligned}$$

The first inequality is less strightforward because the moments are not mutiplicative for the usual convolution $\otimes$. But we can use the *duality rule* $H((a-b) \otimes (a'-b')) = -H((a-b) \boxplus (a'-b'))$ (see exercises) as follows

$$H[x \otimes (a-b)] = -H((x-\Delta_\infty) \otimes (a-b)] \tag{6.52}$$

$$= \sum_{k=1}^{+\infty} M_{2k}(\Delta_\infty - x) \frac{M_{2k}(b) - M_{2k}(a)}{2k(2k-1)} \tag{6.53}$$

$$= \sum_{k=1}^{+\infty} (M_{2k}(\Delta_\infty) - M_{2k}(x)) \frac{M_{2k}(b) - M_{2k}(a)}{2k(2k-1)} \tag{6.54}$$

$$\leq \sum_{k=1}^{+\infty} \frac{M_{2k}(b) - M_{2k}(a)}{2k(2k-1)} \tag{6.55}$$

$$= H[a-b] \tag{6.56}$$

Analysis of DE recursion and the BP threshold

Let us first prove that the functional $f(c,x)$ on the right hand side of the DE recursions (6.37), is "increasing" with respect to the distributions c and x. Since ordering by degradation is preserved by convolution we obviously have $f(c_2,x) \succ f(c_1,x)$ when $c_2 \succ c_1$. Now, notice that if $a_2 \succ a_1$ and $b_2 \succ b_1$ then $a_2 \otimes b_2 \succ a_1 \otimes b_2$ and $a_1 \otimes b_2 \succ a_1 \otimes b_1$, so also $\succ a_2 \otimes b_2 \succ a_1 \otimes b_1$. Generalizing, for $a_i \succ b_i$, $i = 1, \ldots, n$ we have $a_1 \otimes \cdots \otimes a_n \succ b_1 \otimes \cdots \otimes b_n$. The same statements are true if we replace $\otimes$ by $\boxplus$. Thus for $x_2 \succ x_1$ we get $x_2^{\boxplus d_c - 1} \succ x_2^{\boxplus d_c - 1}$, and then $(x_2^{\boxplus d_c - 1})^{\oplus d_v - 1} \succ (x_2^{\boxplus d_c - 1})^{\oplus d_v - 1}$, and finaly $f(c,x_2) \succ f(c,x_1)$.

Consider a family of channels c_ϵ parametrized by ϵ (for example a noise level). We say that the *family of channels is ordered by degradation* when $c_\epsilon \prec c'_\epsilon$ for $\epsilon < \epsilon'$. The BEC, BEC or BAWGNC are three such families.

We are now ready to prove the analog of Lemma 6.1

LEMMA 6.2 *Let $2 \leq d_v \leq d_c$ and c_ϵ be family of channels ordered by degradation. Let $x_0 = \delta(\dot{})$ and $x_t = f(c_\epsilon, x_{t-1})$, $t \geq 1$. Then (a) The sequence of distributions $\{x_t\}$ is decreasing in t in the sense $x_{t+1} \prec x_t$; (b) If $c_\epsilon \prec c_{\epsilon'}$ then $x_t(c_\epsilon) \prec x_t(c_{\epsilon'})$.*

Proof We first show the claims by induction. We have $x_0 = \delta(\dot{})$ and $x_1 = f(c, x_0) = c$, so $x_0 \succ x_1$. Therefore, for $t \geq 2$, we assume $x_{t-1} \prec x_{t-2}$ as the induction hypothesis. Since $f(c, x)$ is increasing in x, we obtain $f(c, x_{t-1}) \prec f(c, x_{t-2})$ and we deduce that $x_t \prec x_{t-1}$. To prove the second claim assume that $c_\epsilon \prec c_{\epsilon'}$. Then $x_1(c_\epsilon) = c_\epsilon \prec c_{\epsilon'} = x_1(c_{\epsilon'})$. The general statement is deduced similarly to the case of the BEC: $x_t(c_\epsilon) = f(c_\epsilon, x_{t-1}(c_\epsilon)) \prec f(c_{\epsilon'}, x_{t-1}(c_\epsilon)) \prec f(c_{\epsilon'}, x_{t-1}(c_{\epsilon'})) = x_t(c_{\epsilon'})$. □

From statement (a) of the Lemma of the Lemma says that DE iterations give a "decreasing" sequence of probability distributions $x_0 = \delta(\dot{}) \succ x_1 = c \succ x_2 \succ \cdots \succ x_t \succ \ldots$. This means that for each $k \geq 1$ we have an increasing sequence of moments $M_{2k}(x_0) = 0 < M_{2k}(x_1) = M_{2k}(c) < M_{2k}(x_2) < \ldots M_{2k}(x_t) < \ldots$, and since this sequence is bounded by 1, it converges to a real number in $[0, 1]$. Let m_{2k}^∞ be the limits for each $k \geq 1$. Since even and odd moments are equal, odd moments also converge towards the same set of numbers $m_{2k-1}^\infty = m_{2k}^\infty$. Since $|m_k^\infty|^{-1/k} \geq 1$ Carleman's criterion, namely that $\sum_k \geq 1 |m_k^\infty|^{-1/k} = +\infty$, is satisfied thus the set of numbers $\{m_k^\infty\}$ are the moments of some probability distribution x_∞ with moments $M_{2k-1}(x_\infty) = M_{2k}(x_\infty) = m_{2k-1}^\infty = m_{2k}^\infty$. To summarize, we have $x_t \to x_\infty$ in the sense $d(x_t, x_\infty) \to 0$.

LEMMA 6.3 *The limiting distribution x_∞ is a solution of the DE fixed point equation $x_\infty = f(c, x_\infty)$.*

Proof In the case of the BEC this statement was quite trivially obtained directly from the continuity of $f(\epsilon, x)$. For general channels we use the tools introduced in the previous paragraph. It is sufficient to show $d(x_\infty, f(c, x_\infty)) = 0$ because then all moments of x_∞ and $f(c, x_\infty$ are equal and by Carleman's criterion the two distributios must be equal. By the triangle inequality for any t,

$$d(x_\infty, f(c, x_\infty)) \leq d(x_\infty, x_{t+1}) + d(x_{t+1}, f(c, x_t)) + d(f(c, x_t), f(c, x_\infty)) \quad (6.57)$$

The second term vanishes because $x_{t+1} = f(c, x_t)$. We now argue that the limits of the first and third terms when $t \to +\infty$ vanish. By construction of x_∞, $\lim_{t \to +\infty} M_{2k}(x_t) = M_{x_\infty})$, which implies $\lim_{t \to +\infty} d(x_\infty, x_{t+1}) = 0$ by dominated convergence. To compute the limit of the third term we recall that $x_t \succ x_\infty$

so

$$\begin{aligned}
d(f(c,x_t),f(c,x_\infty)) &= H(f(c,x_t)-f(c,x_\infty)\\
&= H(c\otimes((x_t^{\boxplus d_c-1})^{\otimes d_v-1}-(x_\infty^{\boxplus d_c-1})^{\otimes d_v-1}))\\
&\le H((x_t^{\boxplus d_c-1})^{\otimes d_v-1}-(x_\infty^{\boxplus d_c-1})^{\otimes d_v-1})\\
&= H((x_t^{\boxplus d_c-1}-x_\infty^{\boxplus d_c-1}+x_\infty^{\boxplus d_c-1})^{\otimes d_v-1}-(x_\infty^{\boxplus d_c-1})^{\otimes d_v-1})\\
&= \sum_{p=1}^{d_v-1}\binom{d_v-1}{p}H((x_t^{\boxplus d_c-1}-x_\infty^{\boxplus d_c-1})^{\otimes p}\otimes(x_\infty^{\boxplus d_c-1})^{\otimes d_v-1-p})\\
&\le \sum_{p=1}^{d_v-1}\binom{d_v-1}{p}H(x_t^{\boxplus d_c-1}-x_\infty^{\boxplus d_c-1})\\
&= (2^{d_v-1}-1)H(x_t^{\boxplus d_c-1}-x_\infty^{\boxplus d_c-1})
\end{aligned}$$

Each term of the last sum is estimated thanks to similar tricks,

$$\begin{aligned}
H(x_t^{\boxplus d_c-1}-x_\infty^{\boxplus d_c-1}) &= H((x_t-x_\infty+x_\infty)^{\boxplus d_c-1}-x_\infty^{\boxplus d_c-1})\\
&= \sum_{q=1}^{d_c-1}\binom{d_c-1}{q}H((x_t-x_\infty)^{\boxplus q}\boxplus x_\infty^{\boxplus d_c-1-q})\\
&\le (2^{d_c-1}-1)H((x_t-x_\infty))
\end{aligned}$$

Putting these results together we obtain the simple inequality

$$\begin{aligned}
d(f(c,x_t),f(c,x_\infty)) &\le (2^{d_v-1}-1)(2^{d_c-1}-1)H((x_t-x_\infty))\\
&= (2^{d_v-1}-1)(2^{d_c-1}-1)d(x_t,x_\infty)
\end{aligned}$$

which implies (by an argument above) $\lim_{t\to+\infty} d(f(c,x_t),f(c,x_\infty))=0$. □

From statement (b) of the lemma, it follows that if $x_t(c_\epsilon)\to\Delta_\infty$ (in the sense that $d(x_t,\Delta_\infty)\to 0$) for a channel c_ϵ, then $x_t(c_{\epsilon'})\to\Delta_\infty$ for a less noisy channel $c_{\epsilon'}\prec c_\epsilon$. Hence we can define a *BP threshold* as

$$\epsilon^{\mathrm{BP}}=\sup\{\epsilon : x_\infty(\epsilon)=\Delta_\infty\}$$

Not surprisingly (with a bit more work) one can show that the DE fixed point allows to calculate the probability of error

$$\lim_{t\to+\infty}\mathbb{P}_{\mathrm{BP,b}}(d_v,d_c,\epsilon,t)=\int_{-\infty}^{0} dl\,(c_\epsilon\otimes(x_\infty^{\boxplus d_c-1})^{d_v})(l), \tag{6.58}$$

For $\epsilon<\epsilon^{\mathrm{BP}}$ we have $x_\infty=\Delta_\infty$ which yields a vanishing probability of error. It is also possible to show that above ϵ^{BP} this is an increasing function of ϵ.

Examples

In your homework you will implement DE for the $(3,6)$-ensemble and the AWGNC. You will then be able to compare your prediction to the predictions which

you previously derived by running simulations of the BP algorithm and the BAWGNC.

If we consider e.g., the BSC, then DE predicts a threshold for the $(3,6)$-ensemble of $\epsilon^{\mathrm{BP}} = 0.084$. This means that as long as the channel introduces fewer than 8.4 percent errors, the BP decoder will with high probability be able to recover the correct codeword from the received word. Note that for rate one-half the maximum number of errors which a capacity-achieving code can tolerate is around 11 percent. So we see that, as for the BEC, the simple $(3,6)$-regular ensemble achieves a good fraction of capacity under BP decoding.

6.9 Exchange of limits

At this point you might be slightly worried. We have defined density evolution by looking at the erasure fraction which remains after ℓ iterations when we take the blocklength to infinity. Subsequently we have analyzed DE by looking what happens if we take more and more iterations. In short, we have looked at the limit $\lim_{\ell\to\infty}\lim_{n\to\infty}$.

This is certainly a valid limit, but if the implication is sensitive to the order in which we take the limit then one might worry how well experiments for "practical length" of lets say thousands of bits to hundreds of thousands of bits and "practical number of iterations" lets say dozens to hundreds of iterations might fit the theory. At least for the BEC there is a fairly simple and straightforward analytic answer – the limit is the same regardless of the order and can also be taken jointly as long as both quantities tend to infinity!

We will not prove this result here. The key is to consider the converse limit $\lim_{n\to\infty}\lim_{\ell\to\infty}$ and to prove that it gives the same result. Note that due to the special nature of the BEC, the performance is monotonically decreasing in the number of iterations (things only can get better if we perform further iterations). From this basic observation we can deduce the following: Let $\ell(n)$ be any increasing function so that $\ell(n)$ tends to infinity if n tends to infinity. Then, for any channel parameter ϵ, the error probability under the limit $\lim_{n\to\infty}\lim_{\ell\to\infty}$ is no larger than the error probability under the joint limit when $\ell = \ell(n)$, which in turn is no larger than the error probability under the limit $\lim_{\ell\to\infty}\lim_{n\to\infty}$. If now we can show that the two extreme cases have the same limit, then any joint limit also has this same limit.

For the BEC the limit $\lim_{n\to\infty}\lim_{\ell\to\infty}$ can in fact be analyzed and this is what was done in [6]. The technique is to use the so-called *Wormald* method, a method which we will encounter soon when we will analyze simple algorithms to solve the K-SAT problem.

For the general case the situation is more complicated. Experiments and "computations" show that also in the general case the limit does not depend on the order. But in order to show this rigorously one currently has to impose some further constraints on the ensemble, see **??**.

6.10 BP versus MAP thresholds

This is a good point to make a small digression on issues that are treated in detail in part III. In the language of statistical mechanics the BP threshold corresponds to a *dynamical* phase transition in the sense that we have here a sharp change in behavior of the algorithm. The MAP probability of error also displays a threshold behavior (in the limit of infinite block length), i.e it vanishes for $\epsilon < \epsilon_{\rm MAP}$ and is strictly positive for $\epsilon > \epsilon_{\rm MAP}$. Clearly we always have $\epsilon_{\rm BP} < \epsilon_{rmMAP}$ since the MAP decoder is the one among all decoders that minimizes the error probability. There is an important conceptual difference between the two thresholds. The MAP threshold can also be shown to be a singularity of the (infinite block-length) Shannon conditional entropy $\lim_{n\to+\infty} \frac{1}{n}\mathbb{E}[H(\underline{X}|\underline{Y})]$ (or in view of (**??**)) of the free energy in thermodynamic limit. This entropy is a continuous convex function of ϵ which vanishes for $\epsilon \leq \epsilon_{\rm MAP}$ and is strictly positive for $\epsilon > \epsilon_{\rm MAP}$. In this sense, this threshold coresponds to a *static* phase transition in teh sense introduced in Chapters **??** and **??**. We stress here that the infinite block-length Shannon conditionnal entropy has *no* singularity at the BP threshold: dynamical thresholds related to algorithms are not visible on free energies. Very interestingly, and perhaps surprisingly from the point of view of coding at least, although the MAP and BP phase transitions are of a different conceptual nature, they are deeply related. In particular we will see in Part III that one can also compute the MAP threshold and probability of error from the DE equations!

Problems

6.1 Belief Propagation for $(3,6)$ Ensemble and AWGN Channel. In the first homework you have implemented a program which can generate random elements from a regular Gallager ensemble. We will now use this, together with the message-passing algorithm discussed in class, to simulate transmission over a BAWGN channel.

We will use elements from the $(3,6)$-ensemble of length $n = 1024$. For every codeword we send we generate a new code. This way we get the so called *ensemble average*. As discussed in class last week, when transmitting with a binary linear code over a symmetric channel, we can in fact assume that the all-zero (in 0/1 notation) codeword was sent since the error probability is independent of the transmitted codeword. This simplifies our life since we do not need to implement an encoder. We assume that we send the codeword over a binary-input additive white Gaussian noise channel. More precisely, the input is ± 1 (with the usual mapping). The channel adds to each component of the codeword an independent Gaussian random variable with zero mean and variance σ^2. At the receiver implement the message-passing decoder discussed in class. It is typically easiest to do the computations with likelihoods. Since a random element from the $(3,6)$ ensemble typically does not have a tree-like factor graph the scheduling of the messages is important. To be explicit, assume that we use a *parallel* sched-

ule. This means, we start by sending all *initial* messages from variable nodes to check nodes. We then process these messages and send messages back from check nodes to all variable nodes. This is one *iteration*. For each codeword perform 100 iterations and then make the final decision for each bit.

Plot the negative logarithm (base 10) of the resulting bit error probability as a function of the capacity of the BAWGN channel with variance σ^2. This capacity does not have a closed form but can be computed by means of the numerical integral

$$C(\sigma^2) = \int_{-1}^{1} \frac{\sigma}{\sqrt{2\pi}(1-y^2)} e^{-\frac{(1-\sigma^2 \tanh^{-1}(y))^2}{2\sigma^2}} \log_2(1+y)dy.$$

If the code and the decoder where optimal and the length of the code were infinite, where should you see the phase transition (rapid decay of error probability)?

6.2 Gallager Algorithm A. In class we discussed the BP algorithm which is the "locally optimal" message-passing algorithm. One of its downsides in a practical application is that it requires the exchange of real numbers. Hence, in any implementation messages are quantized to a fixed number of bits. One way to think of such a quantized algorithm is that the message represents an "approximation" of the underlying message that BP would have sent.

Assume that we are limited to exchange messages consisting of a single bit. Recall that for BP a positive message means that our current estimate of the associated bit is $+1$, whereas a negative message means that our current estimate is -1 (the magnitude of the BP message conveys our certainty). So we can think of a message-passing algorithm which is limited to exchange messages consisting of a single bit, as exchanging only the sign of their estimate.

The best known such algorithm (and historically also the oldest) is Gallager's algorithm A. It has the following message passing rules.

We assume that the codewords and the received word have components in $\{0,1\}$.

(i) *Initialization:* In the first iteration send out the received bits along all edges incident to a variable node.

(ii) *Check Node Rule:* At a check node send out along edge e the XOR of the incoming messages (not counting the incoming message along edge e).

(iii) *Variable Node Rule:* At a variable node. Send out the received value along edge e unless all incoming messages (not counting the incoming message on edge e) all agree in their value. Then send this value.

Assume that transmission takes place over the BSC(p) and that we are using a $(3,6)$-regular Gallager ensemble. Write down the density evolution equations for the Gallager algorithm A.

6.3 Density Evolution via Population Dynamics. In class we have seen the density evolution (DE) for transmission over the BEC. This was relatively easy since in this case the "densities" are in fact numbers (erasure probabilities).

For general channels, DE is more involved since it really involves the evolution of densities. These are the densities of messages which you would see at the various iterations if you implemented the BP message-passing decoder on an infinite ensemble for a fixed number of iterations.

An quick and dirty way of implementing DE for general channels is by means of a population dynamics approach. Here is how this works. Assume that transmission takes place over a given BMS channel and that we are using the (l, r)-regular Gallager ensemble. Pick a population size N. The larger N the more accurate will be your result but the slower it will be.

(i) Pick an *initial* population, call it $\mathcal{V}_0$. This set consists of N iid log-likelihoods associated to the given BMS channel, assuming that the transmitted bit is 1 (we are using spin notation here). More precisely, each sample is created in the following way. Sample Y according to $p(y \mid x = 1)$. Compute the corresponding log-likelihood value, call it L.

(ii) Starting with $\ell = 1$, where ℓ denotes the iteration number, compute now the densities corresponding to the ℓ-th iteration in the following way.

(iii) To compute $\mathcal{C}_\ell$ proceed as follows. Create N samples iid in the following way. For each sample, call it Y, pick $r-1$ samples from $\mathcal{C}_{\ell-1}$ with repetitions. Let these samples be named $X_1, \ldots, X_{r-1}$. Compute $Y = 2\tanh^{-1}(\prod_{i=1}^{r-1} \tanh(X_i/2))$. Note, these are exactly the message-passing rules at a check node.

(iv) To compute $\mathcal{V}_\ell$ proceed as follows. Create N samples iid in the following way. For each sample, call it Y, pick $l-1$ samples from $\mathcal{C}_\ell$ with repetitions. Let these samples be named $X_1, \ldots, X_{l-1}$. Further, pick a sample from $\mathcal{V}_0$, call it C. Compute $Y = C + \sum_{i=1}^{l-1} X_i$. Note, these are exactly the message-passing rules at a variable node.

We think now of each set $\mathcal{V}_\ell$ and $\mathcal{C}_\ell$ as a sample of the corresponding distribution. E.g., in order to construct this distribution approximately we might use a histogram applied to the set. Recall, that we assume here the all-zero codeword assumption. Hence, in order to see whether this experiments corresponds to a successful decoding, we need to check whether in $\mathcal{V}_\ell$ all samples have positive sign and magnitude which converges (in ℓ) to infinity.

Implement the population dynamics approach for transmission over the BAWGNC(σ) channel using the $(3, 6)$-regular Gallager ensemble. Estimate the threshold using this method. Plot the threshold on the same plot as the simulation results which you performed for your last homework. Hopefully this vertical line, indicating the threshold, is somewhere around where the error probability curves show a sharp drop-off.

7 Interlude: message passing for the Sherrington-Kirkpatrick Spin Glass

This Chapter applies message passing methods to the Sherrington-Kirkpatrick (SK) model of a spin glass (see Section 2.6). The SK model is a very particular random spin system defined on a complete graph with iid random interaction strength associated to each edge.

The impatient reader can very well jump ahead directly to the next chapter on compressed sensing. But there are good reasons for the present interlude. Certainly, the conceptual and historical role of the SK model in our theoretical understanding of random spin systems cannot be underestimated. For us, the message passing analysis of this model will serve as a steping stone towards the technically involved but related compressed sensing problem. Here we explore message passing methods within the SK model chapter and then apply what we have learned to compressive sensing in Chapter 8.

Both applications are similar to coding in their general initial outline. However there is a fundamental difference: the SK and compressed sensing models are defined on complete graphs (the graph for compressed sensing is biparttite complete). This is as far as one can get from locally tree-like graphs, so one might think that that BP simply should not work very well for such models and that this should be the end of the story. But in fact, perhaps surprizingly, the story is much more complicated and interesting. Belief Propagation works well for compressed sensing and for the SK model in its high temperature phase. For the low temperature phase we will see in part III that message passing methods work if they are suitably "upgraded" to a new level of sophistication.

Because of the denseness of the graph, message passing algorithms a priori involve $\Theta(n)$ messages flowing on edges at each iteration step. From the point of view of complexity this is not very good (recall in coding for sparse graphs this complexity is $\Theta(n)$). However, as we will see, the denseness of the graph in fact allows to simplify the BP equations and bring down this complexity to linear order. In the SK model the simplified equations one ends up with are the celebrated Thouless-Anderson-Palmer (TAP) equations. In statistical mechanics these equations were initially derived quite heuristically by correcting the naive mean field approximation by an Onsager "reaction term". Here we will discover that the Onsager recation term just comes for free in the BP formalism.

The same mechanisms will be encountered again in te framework of compressed sensing, and this is enough motivation for studying the SK model first. But there

is one more reason. For the SK model the degrees of freedom are binary spins, and therefore similarly to coding the messages can immediately be parametrized by real numbers. In compressed sensing the degrees of freedom are signal components, i.e. real numbers in general (continuous spins) and the messages are functions of these real variables. So the practical implementation of BP would be much too costly (the quantization of messages drastically increases teh complexity). Fortunately, a somewhat technical approximation allows to effectively reduce them to a set of real numbers.

An analog of density evolution can be derived from the TAP equations; for fascinating historical reasons this goes under the strange name of *replica symmetric solution* (see the notes). A replica symmetric *fixed point* equation allows the calculation of a threshold behaviour in the temperature-magnetic field plane; the so-called Almeida-Thouless line. From our point of view here, this threshold behaviour is an algorithmic or dynamic phase transition, because it marks a sudden change in the average behavior of the message passing algorithm. It is natural to ask whether or not there is a relation with a static or thermodynamic phase transition? The short answer is that, for the SK model these are one and the same. To avoid any confusion we hasten to say that the replica symmetric solution is not exact in the whole phase diagram but only above the Almeida-Thouless line. It is only in part III that we will have enough tools to prove such statements, but we briefly give a preview of the answers in the last section of the present chapter.

7.1 Sherrington-Kirkpatrick model and belief propagation approach

Sherrington-Kirkpatrick model

The Sherrington-Kirkpatrick model of a spin glass was very briefly introduced in the examples of Section 2.6 and this is the good place to give more details. The model is defined on a complete graph with n vertices. The degrees of freedom are binary spins $s_i = \pm 1$, $i = 1, \dots, n$ attached to each vertex. The Hamiltonian is

$$\mathcal{H}(\underline{s}) = -\sum_{i \neq j}^{n} J_{ij} s_i s_j - h \sum_{i=1}^{n} s_i, \tag{7.1}$$

where h is a constant magnetic field and J_{ij} are $n(n-1)/2$ iid random variables (the "coupling constants") associated to the edges of the complete graph. In popular versions of the model one chooses $J_{ij} \sim \mathcal{N}(0, J^2/n)$ or $J_{ij} = \pm J/\sqrt{n}$ with iid Bernoulli(1/2) signs; $J > 0$ a constant.

Why are the coupling constant scaled by $1/\sqrt{n}$? That this is the right scaling can be seen by looking at the fluctuations of the Hamiltonian. The mean and variance of $\mathcal{H}(\underline{s})$ are respectively equal to $-h \sum_{i=1}^{n} s_i$ and $(n-1)J^2/2$. Thus for general spin assignmenets the energy of a general spin assignment has a standard

deviation of $O(\sqrt{n})$ around a mean $O(n)$ and we expect the thermodynamic limit to make sense and be non-trivial. Later on it will often be useful to explicitely extract the scaling by setting $J_{ij} = \tilde{J}_{ij}/\sqrt{n}$ where $\tilde{J}_{ij} \sim \mathcal{N}(0, J^2)$ or $\tilde{J}_{ij} = \pm J$.

The corresponding Gibbs distribution $(e^{-\beta \mathcal{H}(\underline{s})})/Z$ is itself random. As is usual for random Gibbs distributions, there are two levels of randomness: the first one associated to quenched or frozen variables (here the coupling constants J_{ij}) and the second one corresponding to the spin assignments distributed according to the Gibbs distribution. We refer back to Chapter 2 a more extensive discussion of these two levels of randomness.

One of the major achievements of the theory of random spin system is teh derivation of an exact formula for the average free energy of the SK model, namely $-\lim_{n\to+\infty} \mathbb{E}[\ln Z]/n$, as well as a proof of the concentration of $-(\ln Z)/n$ as $n \to \infty$. But this is a long story spanning more than 25 years of statistical mechanics. Let us warn the newcomer that the similarity of (7.1) with the one of the Curie-Weiss model should not lead to the false impression that the path to the solution is easy, and embarking into it at the present stage would distract us too much from our present goal. As explained in the introduction in the present chapter we concentrate on the message passing approach. A brief comparison of the findings with the exact solution is given in Section 7.4 and aspects of the exact solution will be studied in Part III.

Belief propagation equations

We now look at BP equations for the SK model. It will be clear that these equations are the same for any Ising model with pairwise interactions (as defined in Sect. 2.1). The specificities due to the SK model will really come in the bext section.

To proceed systematically with the formalism of 5, we first set up the factor graph formulation. The vertices $i = 1, \dots, n$ of the initial graph G play the role of variable nodes. On every edge $(i, j) \equiv a$ we place a factor node with kernel $f_a(s_i, s_j) = e^{\beta J_{ij} s_i s_j}$. We then attach extra degree-one factor nodes $\hat{i}$ to each variable node i. The kernel associated to $\hat{i}$ is $f_i(s_i) = e^{\beta h s_i}$.

Further, we let $\hat{\mu}_{a\to i}(s_i)$ denote the message which flows from the factor node a to the variable i. In a similar manner, $\mu_{i\to a}(s_i)$ is the message flowing from variable i to factor node a. There is also a "trivial" message $\mu_{\hat{i}\to i}(s_i) = f_i(s_i) = e^{\beta h s_i}$ = flowing from degree-one factor nodes to variable nodes. Since all messages depend on binary variables $s_i = \pm 1$ we can use the same type of parametrization used for coding in Chapter 6 and set,

$$\hat{h}_{a\to i} = \frac{1}{2\beta} \ln\left\{ \frac{\hat{\mu}_{a\to i}(+1)}{\hat{\mu}_{a\to i}(-1)} \right\}, \qquad h_{i\to a} = \frac{1}{2\beta} \ln\left\{ \frac{\mu_{i\to a}(+1)}{\mu_{i\to a}(-1)} \right\}. \tag{7.2}$$

Up to the factor β^{-1} these are the usual half-loglikelihood variables associated to the messages. In the context of spin systems they are also called *cavity magnetic fields*. The reason comes from their physical interpretation which will shortly

become clear. This interpretation is also the reason why we prefer here the letter "h'' instead of "ℓ'' used in Chapter [**?**].

The general BP equations (5.9), (5.10) read

$$\mu_{i\to a}(s_i) = e^{hs_i} \prod_{b\in\partial i\smallsetminus a} \hat{\mu}_{b\to i}(s_i) \tag{7.3}$$

$$\hat{\mu}_{a\to i}(s_i) = \sum_{\sim s_i} e^{\beta J_{ij} s_i s_j} \prod_{j\in\partial a\smallsetminus i} \mu_{j\to a}(s_j) \tag{7.4}$$

An exercise shows that parametrization (7.2) leads to

$$\begin{cases} h_{j\to a} &= h_j + \sum_{b\in\partial j\setminus a} \hat{h}_{b\to j}, \\ \hat{h}_{a\to j} &= \frac{1}{\beta}\operatorname{atanh}\{\tanh(\beta J_{ij})\tanh(\beta h_{i\to a})\}. \end{cases} \tag{7.5}$$

Note the similarity with (6.19) in coding theory. Equ. (7.5) reduce to such "coding-like" equations by setting $\beta = 1$ and letting $J_{ij} \to +\infty$ in which case the factor nodes correspond to degree two parity checks.

There is a special feature of systems with degree two factors that we have not encounterd yet explicitely. Equs (7.5) can be conveniently reduced to a single one with messages flowing on the *original* graph G. To see this note that because a factor b has degre two, a directed edge $b \to j$ can be identified with a directed edge $i \to j$ on the original graph G where i is the unique vertex in $\partial b \setminus j$. In other words, setting $h_{i\to j} = \hat{h}_{b\to j}$, (7.5) become

$$h_{i\to j} = \frac{1}{\beta}\operatorname{atanh}\big\{(\tanh(\beta J_{ij})\tanh(\beta(h_i + \sum_{k\in\partial i\setminus j} h_{k\to i}))\big\} \tag{7.6}$$

This message passing equation does not refer anymore to the factor graph. Messages flow on teh original graph G (the complete graph in the case of the SK model).

The BP-marginal, $\nu_i^{\mathrm{BP}}(s_i)$, at vertex i is determined from its log-likelihood variable

$$h_i + \sum_{a\in\partial i} \hat{h}_{a\to i}, \qquad \text{or equivalently} \qquad h_i + \sum_{k\in\partial i} h_{k\to i} \tag{7.7}$$

Explicitly, the normalized marginal is

$$\nu_i^{\mathrm{BP}}(s_i) = \frac{e^{\beta(h_i+\sum_{k\in\partial i} h_{k\to i})s_i}}{2\cosh(\beta(h_i + \sum_{k\in\partial i} h_{k\to i}))}. \tag{7.8}$$

The BP estimate for the magnetization, is by definition the average spin computed from the BP-marginal

$$m_i^{\mathrm{BP}} = \sum_{s_i\in\{\pm 1\}} s_i \nu_i^{\mathrm{BP}}(s_i) = \tanh(\beta(h_i + \sum_{k\in\partial i} h_{k\to i})). \tag{7.9}$$

We will call m_i^{BP} the BP-magnetization to distinguish it from the (true) thermal equilibrium magnetization $m_i = \langle s_i\rangle$.

Let us pause a second to give a physical interpretation of these formulas. A single spin s in the presence of a magnetic field h has a Hamiltonian $\mathcal{H}(s) = -hs$ and thus a magnetization $\tanh(\beta h)$ (if you have never checked this simple fact do it immediately please!). Therefore one interprets $h_i + \sum_{k\in\partial i} h_{k\to i}$ as an effective magnetic field felt by spin s_i. This is often called the *local field* or also the *mean field*. The local field is the sum of the external field h_i and a cavity field $h_{i,\mathrm{cav}} \equiv \sum_{k\in\partial i} h_{k\to i}$. The later is called *cavity field* because it is an effective field produced by the rest of the system in a cavity left out when one removes vertex i from the graph. Hence the denomination "cavity fields" for the messages $h_{k\to i}$ (and more generally $\hat{h}_{a\to i}$, $h_{i\to a}$).

Flooding schedule

From the perspective of traditional statistical mechanics one would view the BP equations as fixed point equations and try to find all solutions. When multiple solutions arise the important question is: which one to choose? Such issues are discussed in Part III.

For the moment we are interested in the algorithmic standpoint. Recall from Chapter 5 when the underlying graph is a tree the initial conditions and iterations are clearly determined (note that this is also the situation where the BP equations certainly have a unique solution). But when the graph is not tree-like we have to specify initial conditions and a schedule to solve the equations iteratively.

Just as in coding we adopt the flooding schedule and the initialization is just given by the "prior" that we have about the local field. We therefore set

$$h^t_{i\to j} = \frac{1}{\beta}\,\mathrm{atanh}\big\{(\tanh(\beta J_{ij})\tanh(\beta(h_i + \sum_{k\in\partial i\setminus j} h^{t-1}_{k\to i}))\big\}, \qquad h^0_{i\to j} = 0. \quad (7.10)$$

The BP-estimate of the magnetization at time t is,

$$m^t_i = \tanh\big\{\beta(h_i + \sum_{j\in\partial i} h^t_{j\to i})\big\}. \quad (7.11)$$

What is the complexity of this schedule on a complete graph? The complete graph has $n(n1)/2$ edges so at each iteration step of the flooding schedule we exchange a quadratic number of messages, and the complexity of message passing is $\Theta(n^2)$ times the number of iterations.

7.2 From belief propagation to Thouless-Anderson-Palmer equations

As just noted above because the graph is complete, a single iteration of BP has quadratic complexity which is costly. Fortunately one can simplify the BP equa- tions and bring the complexity down to order $\Theta(n)$. The key to the simplification is that the coupling constants are weak. Indeed, recall that we have

$J_{ij} = \tilde{J}_{ij}/sqrtn$ (with fluctuations of $\tilde{J}_{ij} = O(1)$). So we assume in general that the coupling constants J_{ij} are small when $n \to +\infty$, and perform an expansion of the message passing equations. This has be done with care however and typically one must go beyond the lowest order term in order to obtain correct results. Interestingly, these simplifications of message-passing equations lead to the Thouless-Anderson-Palmer (TAP) equations. The TAP equations (in their iter- ative form) have a complexity of $\Theta(n)$ at each iteration. Thus they provide a linear complexity algorithm to compute an algorithmic "TAP-estimate" of the magnetization.

Interestingly, these simplifications of message-passing equations lead to the Thouless-Anderson-Palmer (TAP) equations. The TAP equations (in their iterative form) have a complexity of $\Theta(n)$ at each iteration. Thus they provide a linear complexity algorithm to compute an algorithmic "TAP-estimate" of the magnetization.

Consider the BP iteration (7.10) at step t. Using the local field

$$\eta_i \equiv h_i + \sum_{k \in \partial i} \hat{h}_{k \to i} \tag{7.12}$$

we can rewrite (7.10) (see also (**??**)) as

$$h^t_{i\to j} = \frac{1}{\beta} \operatorname{atanh}\Big\{ \tanh(\beta J_{ij}) \tanh(\beta \eta_i^{t-1} - \beta h^{t-1}_{j\to i}) \Big\}.$$

Now, since J_{ij} is of order $1/\sqrt{n}$ we Taylor expand both tanh and atanh This yields

$$h^t_{i\to j} = J_{ij} \tanh(\beta \eta_i^{t-1} - \beta h^{t-1}_{j\to i}) + O(\beta^2 J^3_{ij}). \tag{7.13}$$

This equation shows that each cavity field is $O(J_{ij})$. On the other hand η_i^{t-1} is the sum of h_i and $n-1$ such cavity fields. Therefore we expect $h^{t-1}_{j\to i}$ to be much smaller than η_i^{t-1} and we further expand the tanh in (7.13) in powers of the cavity field,

$$h^t_{i\to j} = J_{ij} \tanh(\beta \eta_i^{t-1}) - \beta J_{ij} h^{t-1}_{j\to i} (1 - (\tanh(\beta \eta_i^{t-1}))^2) + O(\beta^2 J^3_{ij}). \tag{7.14}$$

Recalling (7.11) we can rewrite the cavity field as,

$$h^t_{i\to j} = J_{ij} m_i^{t-1} - \beta J_{ij} h^{t-1}_{j\to i} \big(1 - (m_i^{t-1})^2\big) + O(\beta^2 J^3_{ij}) \tag{7.15}$$

Now we seek to express $h^{t-1}_{j\to i}$ on the right hand side of this equation, in terms of the magnetization. This will allow to approximate cavity fields entirely in terms of the magnetization. We note that if we interchange the roles of i and j in (7.15) (note $J_{ij} = J_{ji}$) and use $\hat{h}^{t-1}_{j\to i} = O(J)$, we get

$$h^t_{j\to i} = J_{ij} m_j^{t-1} + O(\beta J^2_{ij}). \tag{7.16}$$

Replacing (7.16) in (7.15) we obtain

$$h^t_{i\to j} = J_{ij} m_i^{t-1} - \beta J^2_{ij} m_j^{t-1} \big(1 - (m_i^{t-1})^2\big) + O(\beta^2 J^3_{ij}). \tag{7.17}$$

Replacing 7.17 in (7.11) for m_j^t we arrive at

$$m_j^t = \tanh\left\{\beta\left(h_j + \sum_{i\in\partial j} J_{ij} m_i^{t-1} - \beta m_j^{t-1} \sum_{i\in\partial j} J_{ij}^2\left(1-(m_i^{t-1})^2\right)\right)\right\} + O(\beta^3 J_{ij}^3). \tag{7.18}$$

Finally, dropping teh error terms $O(\beta^3 J_{ij}^3)$ we arrive at teh TAP equations. It hould be said that in the statistical mechanics literature the TAP equations correspond to the fixed point form, and their original derivation is through a heuristic "mean field" argument (see the notes for references and history).

We note that dropping $O(\beta^3 J_{ij}^3)$ terms may not be harmless because at each iteration these errors accumulate. Some thought shows that the accumulated error is $O(t\beta^3/n^{3/2})$, so at least for $t << n^{3/2}/\beta^{3/2}$ the error remains small.

Discussion of TAP equations for the SK model

With the scaling of the coupling constant made explicit the TAP equations are

$$m_j^t = \tanh\left\{\beta\left(h_j + \frac{1}{\sqrt{n}}\sum_{i\in\partial j} \tilde{J}_{ij} m_i^{t-1} - \frac{\beta}{n} m_j^{t-1} \sum_{i\in\partial j} \tilde{J}_{ij}^2\left(1-(m_i^{t-1})^2\right)\right)\right\} + O(\beta^3 J_{ij}^3). \tag{7.19}$$

nly estimates of the magnetization are involved and there no messages flowing on edges anymore. At each iteration step magnetisation estimates at vertices of the graph are updated and there are n such updates, so the complexity is now $\Theta(n)$ times the number of iterations.

The local field in (7.22) is given by the external field h_i plus a cavity field (an approximation of the original cavity field discussed in Sect. 7.1)

$$h_{i,\text{cav}}^t = \frac{1}{\sqrt{n}}\sum_{i\in\partial j} \tilde{J}_{ij} m_i^{t-1} - \frac{\beta}{n} m_j^{t-1} \sum_{i\in\partial j} \tilde{J}_{ij}^2\left(1-(m_i^{t-1})^2\right. \tag{7.20}$$

This is the field produced by the rest of the spins when node i is removed from the graph. Each contribution has an interpretation. The first term is the usual "Curie-Weiss mean field" already discussed in Chapter 4 (when $J_{ij} = J/n$ in the CW model). This term is the average field exerted by the system on spin i, but this average "overcounts" the influence of i itself on the system. This "back reaction" should be subtracted, and this is exactly what the second term in (7.20) does. This term is called an *Onsager reaction term.*

The two sums in (7.20) have the same order of magnitude. This also explains why it was needed to Taylor expand to higher orders when we derived the TAP equations from BP. A naive (but wrong) justification of this statement would run as follows. Assuming that the terms in the sums of (7.20) are independent or even weakly correlated, the central limit theorem and the law of large numbers implie that both sums are of order one with respect to n. This argument is much too naive but still reasonably suggests that the CW mean field and Onsager reaction term both contribute equally to the total cavity field.

The natural initial condition for (7.22) follows from the one in (7.10), which means $m_i^0 = \tanh(\beta h_i)$. Obviously as we iterate, the magnetizations m^{t1} will acquire a complicated dependence on $\tilde{J}_{ij}$'s. Thus it is far from true that the terms in the sums (7.20) are independent or even weakly correlated. It turns out that the central limit theorem never applies to the first sum which therefore does not tend to a Gaussian r.v as $n \to +\infty$. The law of large number applies to the second sum only in a "high temperature" regime. In this regime the sum concentrates on its average. This is the regime that we will study discuss in the next section. There is a "low temperature" regime where the sum does not concentrate which we briefly discuss in Section 7.4. The reader may already appreciate how intricate and subtle the SK model is.

It is worth to point out that the TAP equations take their simplest form when $\tilde{J}_{ij} \sim \pm J$ with Ber(1/2) signs. indeed then $\tilde{J}_{ij}^2 = J^2$, and setting

$$q_{t-1} \equiv \frac{1}{n}\sum_{i=1}^{n}(m_i^{t-1})^2 \tag{7.21}$$

we get

$$m_j^t = \tanh\left\{\beta\left(h_j + \frac{1}{\sqrt{n}}\sum_{i\in\partial j}\tilde{J}_{ij}m_i^{t-1} - \frac{\beta}{n}m_j^{t-1}\sum_{i\in\partial j}J^2(1-q_{t-1})\right)\right\} + O(\beta^3 J_{ij}^3). \tag{7.22}$$

he quantity q_{t1} is called the Edwards-Anderson parameter and we will shortly see that it plays a fundamental role. Note that here we ahve an algorithmic estimate of the theromodynamic equilibrium EA parameter $q_{EA} = \sum_{i=1}^{n}\langle si\rangle^2$.

A parenthesis: the CW model revisited

Recall that the exact solution of the CW model in Chapter 4 led us to the fixed point equation

$$m = \tanh(\beta(h + Jm)) \tag{7.23}$$

Here the local field is just the sum of the external field h and the CW mean field Jm.

Why is it that there is no Onsager reaction term is not needed here? One can repeat the same theory developed in this chapter for (non random) coupling constants $J_{ij} = J/n$. Starting from BP equations and then approximating them to leading orders in coupling constants we obviously find again (7.18). At this point, setting $J_{ij} = J/n$ one easily sees that the cavity field becomes $O(1/n)$ and only the usual CW mean field remains. We are lead to the iterative equations

$$m_j^t = \tanh(\beta(h + \frac{J}{n}\sum_{i=1}^{n}m_i^{t-1})) \tag{7.24}$$

Since the right hand side does not depend on j we can seek a uniform solution $m_j^t = m^t$. Equation (7.23) becomes

$$m^t = \tanh(\beta(h + Jm^{t-1})). \tag{7.25}$$

To summarize, for the CW model the TAP equation reduces to the CW equation because teh Onsager term is negligible.

This remark teaches us an important lesson. Recall that the exact solution for the magnetization is found by selecting the fixed point of (7.23) which minimises the free energy. The BP approach leads to the iterative form (7.25) which we should solve with the initial condition $m^0 = 0$. Whether the estimate m^t converges to the (true) magnetisation depends on the region of the phase diagram in the (β, h) plane. When the fixed point is unique the algorithmic and thermodynamic solutions obviously match. Such intimate connections between algorithmic and thermodynamic solutions will be discussed in more depth in Part III, so we close this parenthesis here.

7.3 Evolution equations for TAP iterations - replica symmetric equation

The goal of density evolution is to write down an iterative equation that tracks the evolution of the probability density of the "state" of the system. We review basic results for the SK model that are valid in the high temperature phase where the TAP equations themselves are valid. A rigorous justification is beyond the scope of this chapter. But in the homeworks we propose a numerical justification.

We take the Bernoulli model for which the discussion is slightly simpler. The results are independent of the precise distribution of $\tilde{J}_{ij}$ for a wide class of distibutions. Recall expression (7.11) for the BP-magnetization,

$$m_i^{(t)} = \tanh\big\{\beta(h + \sum_{j\neq i} \hat{h}_{j\to i}^{(t)})\big\}$$

The TAP approximation consists in replacing the exact cavity field $\hat{h}_{i\to j}^{(t)}$ by (see equ. (7.17))

$$\hat{h}_{i\to j}^{(t)} \approx \frac{1}{\sqrt{n}}\left\{\tilde{J}_{ij} m_i^{(t-1)} - \frac{\beta}{\sqrt{n}} m_j^{(t-1)}(1 - (m_i^{(t-1)})^2)\right\}$$

The main assumption of density evolution here is that *these cavity fields are sufficiently weakly correlated* so that the sum

$$\sum_{j\neq i} \hat{h}_{i\to j}^{(t)} \tag{7.26}$$

is a Gaussian r.v with zero mean and variance

$$\mathbb{E}\left[\left(\tilde{J}_{ij}m_i^{(t-1)} - \frac{\beta}{\sqrt{n}}\hat{m}_j^{(t-1)}(1-(m_i^{(t-1)})^2)\right)^2\right] \tag{7.27}$$

$$\approx \mathbb{E}\left[(m_i^{(t-1)})^2\right] + O(n^{-1/2}) \tag{7.28}$$

The assumption of weak correlation of the cavity fields is non-trivial, and amounts to say that the Onsager reaction term corrects for the *non-Gaussian* nature of the pure Curie-Weiss contribution

$$\frac{1}{\sqrt{n}}\sum_{j\neq i}\tilde{J}_{ij}m_i^{(t-1)}.$$

When the Onsager reaction term is included the local fiel becomes Gaussian.[1] It is the goal of the homework to check this assumption numerically. Let us discuss one heuristic argument to gain some further intuition. Consider the SK model on a random regular graph of vertex degree d. This is a sparse graph so it is quite natural to consider the BP algorithm in exactly the same way as we did in chapter 6. For a fixed number of iterations t and n large enough the neiborhood of a vertex is a tree with probability $1 - O(d^l/n)$, so that the messages $\hat{h}_{i\to j}^{(t)}$ are independent. Now consider the limit $d \to +\infty$. In this limit the meaningful scaling is $J_{ij} = \tilde{J}_{ij}/\sqrt{d}$. Of course it is not necessarily legitimate to interchange the limits $d \to +\infty$ and $n \to +\infty$ but, assuming this is possible then the sum (7.26) behaves as a Gaussian.

Let us now set

$$m^{(t)} = \mathbb{E}[(m_i^{(t)})^2], \qquad q^{(t)} = \mathbb{E}[(m_i^{(t)})^2]$$

Averaging the TAP equation (**??**) we get

$$m^{(t)} = \int_{-\infty}^{+\infty} dz \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} \tanh\{\beta(h + z\sqrt{q^{(t-1)}})\} \tag{7.29}$$

Squaring and then averaging the TAP equation (**??**) we get

$$q^{(t)} = \int_{-\infty}^{+\infty} dz \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} \tanh^2\{\beta(h + z\sqrt{q^{(t-1)}})\}. \tag{7.30}$$

These density evolution equations allow to compute the average magnetization and Edwards-Anderson parameter.

The statistical mechanics solution of the SK model (i.e. the calculation of the free energy, magnetization, etc) proceeds by the replica method (a purely algebraic method) or by the cavity method (which has probabilistic flavor). Quite remarkably there is an exactly known high-temperature region depicted on figure **??** where they both predict that the average magnetization $\mathbb{E}[\langle s_i\rangle]$ and Edwards-Anderson parameter $\mathbb{E}[\langle s_i\rangle^2]$ satisfy the fixed-point form of the density evolution

[1] Rigorous proof of this statement appears in recent works of E. Bolthausen (2009) and S. Chaterjee (2010).

equations (7.29), (7.30). In the low temperature region the theory is much more subtle: let us just mention here that the Edwards-Anderson parameter does not concentrate on its mean but has a non-trivial distribution.

7.4 Exact solution of the SK model

STILL TO DO HERE

The derivations in this chapter follow an algorithmic approach and may be seen as a natural analog of those made in coding in Chapter **??**. However in statistical mechanics the perpective is usually different. One wants to compute the exact free energy, as we did for the Curie-Weiss model, or a very good approximation of it. Then one hopes to be able to analyze thermodynamic phase transitions. Natural questions are whether or not the message passing algorithmic solution allows to recover the free energy, the thermodynamic phase transition, and what is the relation, if any, between the algorithmic and thermodynamic phase transitions. Despite the simplicity of its Hamiltonian the SK model is not the simplest model in which such questions can be answered. We briefly discuss these issues in the last section of this chapter and will come back to them in the third part of the course.

The expectation of (7.1) over quenched variables equals $-h\sum_{i\in V} s_i$ which is $O(n)$. Because of the scale factor $1/\sqrt{n}$ the variance equals nJ^2 . So the scale factor is adjusted so that the relative fluctuations of the Hamiltonian are of order $O(1/\sqrt{n})$. The ultimate justification for this normalization is that this leads to a well defined free energy per spin in the thermodynamic limit, $\lim_{n\to+\infty} -\frac{1}{n}\ln Z$. It can shown that this limit exists with probability one and equals limiting average free energy per spin $-\lim_{n\to+\infty}\frac{1}{n}\mathbb{E}[\ln Z]$. The proof of existence of the theromodynamic limit is non-trivial and has been an open problem for more than 30 years. The method of proof has come to be known as the "interpolation method" which is one of the subjects in Part III.

The (practical) computation of the average free energy has been a facinating problem since the 70's which in the end led to much recent progress well beyond the SK model. For example, it has led to the *cavity method* - a sophisticated extension of BP - that we will study in the context of K-SAT in part III. A few remarks on the history of this computation might be in order here. More extensive informtaion and references are given in the notes. Parisi first proposed a formula for the average free energy in the late 70's, and a somewhat magical method of derivation with an algebraic flavor, called the "replica method". The Parisi formula was re-derived later by another approach called the *cavity method* which has a probabilistic flavor. The mathematical breakthrough came with Talagrand who was able to use and extend the interpolation method to prove that the Parisi formula is indeed correct and to shed some light on the cavity method. Despite this progress the older replica method and its success has remained a little bit mysterious to date. But (for the best or the worse) some of the

associated terminology has remained and is often used to designate results which are obtained by other methods.

In this chapter we will arrive at an expression for the free energy that is only valid in a "high temperature - large magnetic field" regime, and is called the *replica symmetric formula*. In compressed sensing this is the only "type of formula" we will need. We will not discuss the full Parisi formula here which is valid in the whole temperature-magnetic field plane, because this would distract us too much from our goals.

7.5 Notes

In 1936 Onsager was concerned with the dielectric properties of molecular liquids where the so-called "Onsager reaction terms" are important and correct the earlier 1912 theory of Debye. The term "cavity field" was also coined by him. Bethe had similiar insights for magnetism. In 1977 Thouless, Anderson and Palmer (TAP) where the first to point out the importance of the Onsager term in random spin systems. The TAP paper includes a non-algorithmic derivation of the Onsager term through a diagrammatic expansion in the high temperature regime. The SK model has played a very important role in the development of methods and concepts of spin glass theory. These were developed through the 70's and 80's by many physicists and it remained an open mathematical problem for more than 25 years to prove their validity. This was accomplished a decade ago in break through works of Guerra and Talagrand.

Problems

7.1 Distribution of cavity fields in the TAP theory. The goal of this exercise is to numerically justify some of the heuristic arguments of this chapter. When we discuss state evolution for compressive sensing we will encounter similar arguments and hopefully these will seem familiar. Consider the SK model with i.i.d Bernoulli(1/2) coupling constants $\tilde{J}_{ij} = \pm 1$ or $\tilde{J}_{ij}$ Gaussian with zero mean and unit variance. The TAP approximation to the BP equations reads

$$m_j^{(t)} = \tanh\{\beta(h + \sum_{i \neq j} \hat{h}_{i\to j}^{(t)})\}$$

where the update of the cavity fields is

$$\hat{h}_{i\to j}^{(t)} = \frac{1}{\sqrt{n}} \tilde{J}_{ij} m_i^{(t-1)} - \frac{\beta}{n} m_j^{(t-1)} (1 - (m_i^{(t-1)})^2)$$

and the initialization $\hat{h}_{i\to j}^{(0)} = 0$.

Take a number $N = 50$ of realizations (coupling constants) of the system of size $n = 500$ or 1000 and an iteration number say $t = 10$. Try values of $(h, T = \beta^{-1})$ in the high temperature regime. The following should be suitable $(h = 0.5, T = 1.2)$ and $(h = 1, T = 0.8)$.

(i) Plot the histogram of the total cavity field

$$\hat{h}_{\text{cav}}^{(t)} = \sum_{i \neq j} \hat{h}_{i \to j}^{(t)}.$$

This field is equal to a "Curie-Weiss" field to which the "Onsager reaction term" is subtracted. Plot the histogram of the total Curie-Weiss contribution

$$h_{\text{CW}}^{(t)} = \sum_{i \neq j} \frac{1}{\sqrt{n}} \tilde{J}_{ij} m_i^{(t-1)}.$$

(ii) Check that the Edwards-Anderson parameter

$$q^{(t)} = \frac{1}{n} \sum_{i=1}^{n} (m_i^{(t)})^2.$$

is concentrated on its empirical mean over the N realizations.

(iii) Compare both histograms with the Gaussian distribution of zero mean and variance equal to the Edwards-Anderson parameter. You should observe that the histogram of the cavity field agrees with this Gaussian.

8 Compressive Sensing: Approximate Message Passing and State Evolution

Recall that a meaningful estimator for the compressive sensing problem is the Least Absolute Shrinkage Selection Operator (LASSO), given by

$$\underline{\hat{x}}_1(\underline{y},\lambda) = \text{argmin}_{\underline{x}}\Big\{\frac{1}{2}\|\underline{y}-A\underline{x}\|_2^2+\lambda\|\underline{x}\|_1\Big\}. \tag{8.1}$$

where the parameter λ has to be adjusted to the best possible value that minimizes the "risk".

The use of this estimator can be justified from several points of views as discussed in Chapters 1 and 3. For example one can settle for this estimator because, in the noiseless limit and for a certain range of parameters, the ℓ_1 and ℓ_0 minimization problems are equivalent. Another point of view is that the "zero temperature" version of the MMSE estimator for a Laplacian prior yields the LASSO estimator, and the Laplacian prior is a simple and tractable model for sparse signals with unknown distribution. A "justification" for using this estimator can also be given in hindsight. We will see that this estimator works well in a fairly general setting. Moreover, together with the right structure for the measuring matrix we can even, in some cases, get optimal performance in terms of its asymptotic (in the size) behavior if we look at the required number of measurements compared to the sparsity of the signal. However it is a long road until we can arrive at this conclusion in Chapter 14, so for the moment we will not worry about this, and we simply want to implement the LASSO in an algorithmically efficient manner.

The basic idea to implement LASSO is straightforward. We first set up a factor graph corresponding to (8.1). Given the factor graph we can mechanically write down the message-passing rules following the general framework about factor graphs set out in Chapter 5, no thinking required. Since the LASSO asks for the best global minimizer $\underline{\hat{x}}(\underline{y},\lambda)$ our starting point is the min-sum algorithm. This is to some degree a matter of convenience and alternative derivations exist which start with the BP algorithm. Quite surprisingly this works although the graph is dense and not at all sparse.

In principle this program only takes a few lines and we could stop at this point. But there are a few issues. We will see that for the straightforward message-passing algorithm the number of messages which need to exchanged in each iteration is of quadratic oder in the graph size. This is true since the graph is dense. The second problem is that the messages are functions and not numbers as

was the case for coding. This increases the complexity even further. Fortunately, as we will see, one can approximate the original message-passing algorithm to (i) first simplify the messages to numbers, and (ii) bring down the number of messages which need to be exchanged in each iteration to linear order. The final algorithm we derive is called AMP, where AMP stand for *approximate message-passing*.

Besides the pratical motivation to reduce complexity there is also another, perhaps more important, reason for going through these simplifications. The performance of the resulting AMP algorithm can be (rigorously) analyzed in detail. This would be out of the question for the original min-sum algorithm. Finally, even though the AMP algorithm is an approximation, it works very well, and moreover its performance can be caracterised precisely. In the context of coding we were able to assess the performance of the BP algorithm thanks to DE. Recall that in the large-size limit the state of the BP algorithm is given in terms of a distribution (density). DE then allows us to track this state as a function of the iteration. It is possible to develop a similar formalism for the AMP algorithm. In the context of compressive sensing, this formalism is called *state evolution* (SE). As we will see, one can derive recursive equations for the MSE whose average behavior is tracked by SE.

An important application of SE is a principled way to compute an optimal threshold parameter λ. We will aslo discuss a related application which consists of determining an "algorithmic phase transition line" in the phase diagram of compressive sensing. Remarquably this line is independent of the noise level and determines the region of equivalence of the ℓ_1 and ℓ_0 problems. It was first derived by Donoho and Tanner by completely independent means.

8.1 LASSO for the Scalar Case

We begin with the analysis of a toy problem, namely the estimation of a scalar signal corrupted by noise. This turns out to be not only an interesting non-trivial problem, but also an important ingredient for the solution of the estimation of vector signals. Let then

$$y = x + z,$$

where $z \sim \mathcal{N}(0, \sigma^2)$. We asssume that the scalar signal x is "sparse" in the sense that it is a random variable with mass of weight $1 - \epsilon$ at $x = 0$ and mass of weight ϵ distributed (in an unknown way) for $x \neq 0$. More formally, this is the class $\mathcal{F}_\epsilon$ of distributions of the form

$$p_0(x) = (1 - \epsilon)\delta(x) + \epsilon\phi_0(x).$$

where $\phi_0(x)$ is non-negative continuous distribution function normalized to one. The LASSO estimator

$$\hat{x}_1(y,\lambda) = \text{argmin}_x\Big\{\frac{1}{2}(y-x)^2 + \lambda|x|\Big\}.$$

corresponds to the Hamiltonian

$$\mathcal{H}(x|y) = \frac{1}{2}(y-x)^2 + \lambda|x|.$$

Let us check where this Hamiltonian takes on its minimum. For $x > 0$ its derivative with respect to x equals $-(y-x)+\lambda$. Setting this derivative to 0 we get the solution $\hat{x} = y - \lambda$, which is valid if $y > \lambda$. On the other hand for $x < 0$ the derivative is $-(y-x)-\lambda$. Setting this derivative to 0 we get the condition $\hat{x} = y+\lambda$, which is valid if $y < -\lambda$. For the remaining case $-\lambda < y < \lambda$ one checks the inequality $\frac{1}{2}y^2 \le \frac{1}{2}(y-x)^2 + \lambda|x|$ which means that $\hat{x} = 0$. Summarizing, we get the estimator

$$\hat{x}_1(y,\lambda) = \eta(y\,;\,\lambda) \equiv \begin{cases} y-\lambda, & \text{if } y > \lambda, \\ 0, & \text{if } -\lambda < y < \lambda, \\ y+\lambda, & \text{if } y < -\lambda. \end{cases}$$

This is called the "soft thresholding estimator" and $\eta(y\,;\,\lambda)$ is called the "soft thresholding function". The corresponding graph is shown in Figure 8.1.

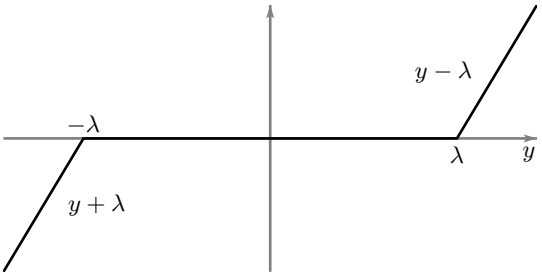


Figure 8.1 Graph of the soft-threshold function $\eta(y\,;\,\lambda)$.

In the above estimator we need to choose the threshold λ. How shall we choose this value? One possible criterion is to solve the following minimax problem: "choose the best λ for the worst prior $p_0(x)$." In mathematical terms we compute the minimax-MSE

$$\inf_{\lambda} \sup_{p_0(\cdot)\in\mathcal{F}_\epsilon} \mathbb{E}[|\hat{x}_1(y,\lambda) - x|^2]. \tag{8.2}$$

Writing it explicitly and making teh change of variables $y \to x+z$ teh minimax-MSE equals

$$\inf_{\lambda} \sup_{p_0(\cdot)\in\mathcal{F}_\epsilon} \int dxdz\; p_0(x) \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}z^2} [\eta(x+z\,,\,\lambda) - x]^2. \tag{8.3}$$

It is natural to set $\lambda = \alpha\sigma$ and to determine α instead of λ. Mathematically this is of course equivalent, but the interpretation is that it is natural to choose the threshold on the scale of the noise. Performing the change of variables $x \to \sigma x$, $z \to \sigma z$, and noting that

$$\sigma p_0(\sigma x) = (1-\epsilon)\delta(x) + \epsilon\phi_0^{(\sigma)}(x), \qquad \phi_0^{(\sigma)} = \sigma\phi_0(\sigma x)$$

is a normalized distribution belonging to $\mathcal{F}_\epsilon$ (in other words the ensemble $\mathcal{F}_\epsilon$ is scale invariant) we see that (8.3) equals

$$\sigma^2 \inf_{\alpha\geq 0} \sup_{p_0\in\mathcal{F}_\epsilon} \int dx p_0(x) \int dz \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} [\eta(x+z\,;\,\alpha) - x]^2 \tag{8.4}$$

This shows that the solution of the minimax problem is essentially independent of the noise level. The only thing that really depends on the noise level is the overall scale of the minimax-MSE. It should be clear that this is so because since $\mathcal{F}_\epsilon$ is scale invariant, σ^2 is the only scale or "dimensionful quantity" in the problem, so dimensional analysis tells us that the minimax-MSE is proportional to σ^2. This is generally not true for the usual MMSE estimator used when the prior is known and introduces another scale besides σ^2 in the problem.

It turns out that one can compute the worst case distribution and best possible α exactly. Let us set

$$M_{\text{scalar}}(\epsilon,\alpha,p_0) \equiv \int dx p_0(x) \int dz \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} [\eta(x+z\,;\,\alpha) - x]^2 \tag{8.5}$$

and

$$M_{\text{scalar}}(\epsilon,\alpha) \equiv \sup_{p_0\in\mathcal{F}_\epsilon} M_{\text{scalar}}(\epsilon,\alpha,p_0), \qquad M_{\text{scalar}}(\epsilon) \equiv \inf_\alpha M_{\text{scalar}}(\epsilon,\alpha) \tag{8.6}$$

For fixed α the worst case distribution turns out to be (Donoho and Johnson 1994/ make an exercise)

$$p_{0,\text{worst}}(x) = (1-\epsilon)\delta(x) + \frac{\epsilon}{2}\delta_{+\infty}(x) + \frac{\epsilon}{2}\delta_{-\infty}(x).$$

Using this expression one easily deduces that

$$M_{\text{scalar}}(\epsilon,\alpha) = \epsilon(1+\alpha^2) + (1-\epsilon)\left[2(1+\alpha^2)\Phi(-\alpha) - 2\alpha\frac{e^{-\frac{\alpha^2}{2}}}{\sqrt{2\pi}}\right], \tag{8.7}$$

where $\Phi(\alpha) = \int_{-\infty}^{\alpha} \frac{e^{-\frac{u^2}{2}}}{\sqrt{2\pi}} du$ the cdf of a standardized Gaussian. To find the best possible α we now minimize (8.7) over α. Setting its derivative to zero we obtain

$$\epsilon = \frac{2\big(\frac{e^{-\frac{\alpha^2}{2}}}{\sqrt{2\pi}} - \alpha\Phi(-\alpha)\big)}{\alpha + 2\big(\frac{e^{-\frac{\alpha^2}{2}}}{\sqrt{2\pi}} - \alpha\Phi(-\alpha)\big)} \tag{8.8}$$

The right hand side is a monotone decreasing function of α, thus given ϵ there

exist a unique optimal $\alpha_{\text{best}}(\epsilon)$ found by inverting (8.8). Finally, the minimax-MSE for the scalar problem is

$$M_{\text{scalar}}(\epsilon) = M_{\text{scalar}}(\epsilon, \alpha_{\text{best}}(\epsilon)). \tag{8.9}$$

8.2 The vector case: preliminaries

From the point of view of statistical physics (8.1) is equivalent to minimizing the Hamiltonian (or cost function)

$$\mathcal{H}(\underline{x}|\underline{y}, A) = \sum_{a=1}^{m} \frac{1}{2}(y_a - (A\underline{x})_a)^2 + \lambda \sum_{i=1}^{n} |x_i| \tag{8.10}$$

We explained in Chapter 3 that this cost function can be interpreted as a spin-glass Hamiltonian. The matrix A and the observation $\underline{y}$ are random, but once we have a realization they are considered fixed. These are the *quenched* variables. The degrees of freedom reside in the signal components x_i. These are "continuous spins" since $x_i \in \mathbb{R}$ rather than the usual binary variable $s_i \in \{\pm 1\}$.

In the formulation above we are looking for the global minimum of the Hamiltonian. For the scalar case this could be done analytically, but now in the vector case this is not possible and we have to settle for an algorithmic solution. According to the factor graph framework developed in Chapter **??** we use the min-sum algorithm. The underlying factor graph is the complete bipartite graph with variable nodes corresponding to the signal components x_i, and two types of factor nodes corresponding to the factors

$$\frac{1}{2}(y_a - (A\underline{x})_a)^2, \qquad \text{and} \qquad \lambda|x_i|.$$

A straightforward application of the message opassing rules leads to the following equations involving two ypes of messages, call them $\hat{E}_{a\to i}(x_i)$ and $E_{i\to a}(x_i)$, $i = 1, \cdots, n$ and $a = 1, \cdots, m$,

$$\begin{cases} E^{t+1}_{i\to a}(x_i) = \lambda|x_i| + \sum_{b\in\partial i\setminus a} \hat{E}^t_{b\to i}(x_i), \\ \hat{E}^{t+1}_{a\to i}(x_i) = \min_{\underline{x}\setminus x_i} \left\{ \frac{1}{2}(y_a - (A\underline{x})_a)^2 + \sum_{j\in\partial a\setminus i} E^{t+1}_{j\to a}(x_j) \right\}. \end{cases} \tag{8.11}$$

In addition we have the initialization

$$\begin{cases} E^0_{i\to a}(x_i) = \lambda|x_i|, \\ \hat{E}^0_{a\to i}(x_i) = \min_{\underline{x}\setminus x_i} \{ \frac{1}{2}(y_a - (A\underline{x})_a)^2 + \sum_{j\in\partial a\setminus i} \lambda|x_j| \}. \end{cases} \tag{8.12}$$

The min-sum estimate at time t, call it $\hat{x}^t_i(\lambda)$, is computed from

$$\hat{x}^t_i = \text{argmin}_{x_i} E^t_i(x_i), \tag{8.13}$$

where

$$E^t_i(x_i) = \lambda|x_i| + \sum_{b\in\partial i} \hat{E}^t_{b\to i}(x_i). \tag{8.14}$$

Recall that in chapter 5 we discussed the BP equations for compressive sensing. As explained there, the min-sum equations (8.11) can be obtained by taking the $\beta \to +\infty$ limit of BP equations. Alternatively one can derive them by a direct application of the distributive law to the min and + operations (see problems in chapter 5.

We stress here that $\hat{\underline{x}}^t$ in (8.13) is the *min-sum estimate* (an algorithmic quantity) and although one might hope that as $t \to +\infty$ it converges to the LASSO estimator $\hat{\underline{x}}_1(\underline{y}, \lambda)$ this is far from obvious a priori. We will have the occasion to introduce two other related estimates in this chapter and we come back to the issue of their comparison in Section 8.8.

Running min-sum on a complete bi-partite graph with a bi-partition of size n and m respectively, requires to transmit $\Theta(mn)$ messages at each iteration. For large instances this complexity is prohibitive. We will now show that we can get away with linear complexity. To be sure, the algorithm which we now develop is no longer exact, but it is a good approximation. Further, recall that we are not operating on a tree and so even a full fledged BP is not necessarily optimal. There is therefore no reason to insist on an exact implementation of the BP algorithm.

How can we derive such an approximation? The model and the situation is analogous to that of the SK model. Therefore, it should not come as a surprise that the methodology which we follow for the analysis is also similar. We have seen in the previous chapter that for the SK model we can go from the BP equations to the TAP equations by exploiting the fact that the interaction coefficients are small, explicitly by exploiting that $J_{ij} \sim \mathcal{N}(0, \frac{1}{n})$ or $J_{ij} \sim \mathrm{Ber}(1/2)$ in $\{+\frac{1}{\sqrt{n}}, -\frac{1}{\sqrt{n}}\}$. In the present case we can also exploit the fact that $A_{ai} \sim \mathcal{N}(0, \frac{1}{m})$, so that each entry is $O(1/\sqrt{m})$. As shown in section 8.4 this leads to significant simplifications and linear complexity. Note that these simplifications will appear even more clearly with the Bernoulli(1/2) ensemble $A_{ai} \in \{+\frac{1}{\sqrt{m}}, -\frac{1}{\sqrt{m}}\}$.

Before we takle this derivation there is one complication we first have to deal with. Contrary to the SK or coding models the "spin variables" (here the signal components) are not binary and therefore the min-sum messages cannot be exactly parametrized by numbers (the log-likelihood variables in the binary case). However it turns out that a "quadratic approximation" of the messages is possible, which approximates each message by a set of two real numbers. This is the subject of the next section.

8.3 Quadratic Approximation

The following is a fairly long somewhat mechanical and technical calculation. In a first reading the reader may just look at formulas (8.15) and (8.17) that define the parametrization, and then skip forward directly to the message passing equations

(8.18) and (8.19). These equations are all that is needed for the derivation of the AMP algorithm in Section 8.4.

A simple but crucial observation is that in the message passing expression (8.11) for $\hat{E}^{t-1}_{a\to i}(x_i)$ the x_i dependence only enters as $A_{ai}x_i$ in $(A\underline{x})_a$. Now since $A_{ai}x_i \sim \frac{1}{\sqrt{m}}$ this contribution is small as m tends to infinity. We can therefore consider the Taylor expansion of $\hat{E}^{t+1}_{a\to i}(x_i)$ in powers of $A_{ai}x_i$ and keep only the lowest order terms,

$$\hat{E}^{t+1}_{a\to i}(x_i) = \hat{E}^{t+1}_{a\to i}(0) - \alpha^{t+1}_{a\to i}(A_{ai}x_i) + \frac{1}{2}\beta^{t+1}_{a\to i}(A_{ai}x_i)^2 + O((A_{ai}x_i)^3), \quad (8.15)$$

where the Taylor coefficients $\alpha^{t+1}_{a\to i}$ and $\beta^{t+1}_{a\to i}$ are real numbers that we will determine later. These are two real valued messages that approximate $\hat{E}^{t+1}_{a\to i}(x_i)$. Equation (8.15) constitutes the quadratic approximation for $\hat{E}^{t+1}_{a\to i}(x_i)$: in terms of two real valued messages $\alpha^{t+1}_{a\to i}$ and $\beta^{t+1}_{a\to i}$. Replacing (8.15) in the message passing equation (8.11) for $E^{t+1}_{i\to a}(x_i)$ we get

$$\begin{aligned} E^{t+1}_{i\to a}(x_i) &\approx E^{t+1}_{i\to a}(0) + \lambda|x_i| - x_i \sum_{b\in\partial i\setminus a} A_{bi}\alpha^t_{b\to i} + \frac{x_i^2}{2}\sum_{b\in\partial i\setminus a} A^2_{bi}\beta^t_{b\to i} \\ &= E^{t+1}_{i\to a}(0) - \frac{\lambda(a_1^t)^2}{2a_2^t} + \frac{\lambda}{a_2^t}\left\{a_2^t|x_i| + \frac{1}{2}(x_i - a_1^t)^2\right\} \end{aligned} \quad (8.16)$$

where

$$a_1^t = \frac{\sum_{b\in\partial i\setminus a} A_{bi}\alpha^t_{b\to i}}{\sum_{b\imath\partial i\setminus a} A^2_{bi}\beta^t_{b\to i}}, \qquad a_2^t = \frac{\lambda}{\sum_{b\in\partial i\setminus a} A^2_{bi}\beta^t_{b\to i}}.$$

The second equality in (8.16) has been obtained by completing the square. A calculation similar to the one presented for the scalar LASSO case shows that the minimum of the term in brackets in (8.16) is equal to $\eta(a_1^t\,;\,a_2^t) \equiv x^{t+1}_{i\to a}$. Thus, when the right hand side of (8.16) is expanded around its minimum one finds (up to an irrelevant constant)

$$E^{t+1}_{i\to a}(x_i) = \text{Constant} + \frac{1}{2\gamma^{t+1}_{i\to a}}(x_i - x^{t+1}_{i\to a})^2 + O((x_i - x_{i\to a})^3) \quad (8.17)$$

where

$$x^{t+1}_{i\to a} = \eta(a_1^t\,;\,a_2^t), \qquad \gamma^{t+1}_{i\to a} = \frac{a_2^t}{\lambda}\eta'(a_1^t\,;\,a_2^t) \quad (8.18)$$

Equation (8.17) constitutes the quadratic approximation for $E^{t+1}_{i\to a}(x_i)$. In these formulas $\eta(y\,;\,\lambda)$ is the same soft thresholding function that was used in the scalar case. The expansion would be exact and the cubic remainder absent for $\lambda = 0$ in which case $\eta(y\,;\,0) = y$. For $\lambda \neq 0$ the absolute value is not differentiable at the origin so the derivation involves a few technical subtleties that are worth discussing. Why can one hope that it is a good approximation to expand $E^{t+1}_{i\to a}(x_i)$ near its minimum? One way to understand this is to recall the connection between min-sum and BP. For $\beta \to +\infty$ the BP messages are proportional to $e^{-\beta E^{t+1}_{i\to a}(x_i)}$, a weight that is dominated by x_i close to the

minimum of the exponent. Once this is accepted, it remains to find this minimum and write down the Taylor expansion around it. From the scalar minimisation problem we learn that the minimum of (8.16) over x_i is attained at $x^t_{i\to a} = \eta(a^t_1\,;\,a^t_2)$. The expansion is best performed by first assuming that $x^t_{i\to a} > 0$, i.e. $x^t_{i\to a} = \eta(a^t_1\,;\,a^t_2) = a^t_1 - a^t_2$. In this case we can set $|x_i| = x_i$ and the first derivative of (8.16) is $\frac{\lambda}{a^t_2}(a^t_2 + (x_i - a^t_1))$ which vanishes at $x^t_{i\to a}$. The second derivative is equal to $\lambda/a^t_2 = \lambda/(a^t_2\eta'(a^t_1\,;\,a^t_2)) = 1/\gamma^t_{i\to a}$. Therefore (8.17) holds when $x^t_{i\to a} > 0$. The reader can work out the case $x^t_{i\to a} < 0$ is a similar way. Finally we consider the singular case $x^t_{i\to a} = 0$, i.e. $\eta(a^t_1\,;\,a^t_2) = \eta'(a^t_1\,;\,a^t_2) = 0$. At the origin the first derivative of $|x_i|$ has a jump, and the second derivative is formally infinite. Therefore we have to take $\gamma^t_{i\to a} = 0$ which is consistent with $\gamma^t_{i\to a} = \frac{a^t_2}{\lambda}\eta'(a^t_1\,;\,a^t_2)$.

The final step is to determine $\alpha^t_{b\to i}$ and $\beta^t_{b\to i}$. For this we replace (8.17) in the second min-sum equation (8.11). Then we compare with the expansion (8.15). After some long but *exact* algebraic calculations this yields

$$\alpha^t_{a\to i} = \frac{y_a - \sum_{j\in\partial a\setminus i} A_{aj}x^t_{j\to a}}{1 + \sum_{j\in\partial a\setminus i} A^2_{aj}\gamma^t_{j\to a}}, \qquad \beta^t_{a\to i} = \frac{1}{1 + \sum_{j\in\partial a\setminus i} A^2_{aj}\gamma^t_{j\to a}}. \tag{8.19}$$

Let us summarize these calculations. The quadratic approximation assumes that the expansions (8.15) and (8.17) to second order are good approximations and neglects cubic and higher order terms. The min-sum equations (8.11) then reduce to the set of message passing equations (8.18), (8.19) for real valued messages $x^t_{i\to a}$, $\gamma^t_{i\to a}$, $\alpha^t_{a\to i}$, $\beta^t_{a\to i}$.

8.4 Derivation of the AMP Algorithm

First simplications of (8.18) and (8.19)

Our derivation rests on the assumption that the term in the denominator of (8.19)

$$1 + \sum_{j\in\partial a\setminus i} A^2_{aj}\gamma^t_{j\to a}$$

can be treated as independent of a and i. Why might this be true? Note that $A^2_{aj} \sim \frac{1}{m}$ and that we sum over $m-1$ terms. This sum is therefore up to a negligible term equal to the empirical mean of $\gamma^t_{j\to a}$ over all edges of the graph, and we therefore expect this to concentrate on a value independent of i and a. In the sequel we set

$$1 + \sum_{j\in\partial a\setminus i} A^2_{aj}\gamma^t_{j\to a} \equiv \frac{\theta_t}{\lambda} \tag{8.20}$$

and we treat θ_t as independent of a and i. The determination of θ_t is discussed later on. We also set

$$r^t_{a\to i} = y_a - \sum_{j\in\partial a\setminus i} A_{aj} x^t_{j\to a}, \tag{8.21}$$

so that (8.19) become

$$\alpha^t_{a\to i} = \frac{\lambda}{\theta_t} r^t_{a\to i}, \qquad \beta^t_{a\to i} = \frac{\lambda}{\theta_t}. \tag{8.22}$$

Let us now look at a^t_1 and a^t_2. From $\beta^t_{b\to i} = \lambda/\theta_t$ we deduce that the denominator of a^t_1 and a^t_2 is equal to

$$\frac{\lambda}{\theta_t} \sum_{b\in\partial i\setminus a} A^2_{bi}$$

Furthermore we note that $\sum_{b\in\partial i\setminus a} A^2_{bi} \approx 1$ since the A_{bi} are iid $\sim \mathcal{N}(0, \frac{1}{m})$ (for the Bernoulli model this sum is exactly equal to $(m-1)/m$ which tends to 1 in the large system size limit). With these remarks we obtain

$$a^t_1 = \sum_{b\in\partial i\setminus a} A_{bi} r_{b\to i}, \qquad a^t_2 = \theta_t. \tag{8.23}$$

Replacing in the first message passing equation (8.18) one finds

$$x^{t+1}_{i\to a} = \eta\big(\sum_{b\in\partial i\setminus a} A_{bi} r^t_{b\to i}\,;\,\theta_t\big). \tag{8.24}$$

The current form of the message-passing rules (8.21) and (8.24). We still need an equation for the updates of θ_t. This is now easily obtained by multiplying the second equation in (8.18) by A^2_{ai} and summing over i. We get

$$1 + \sum_{i\in\partial a} A^2_{ai}\gamma^{t+1}_{i\to a} = 1 + \sum_{i\in\partial a} A^2_{ai}\frac{a^t_2}{\lambda}\eta'(a^t_1\,;\,a^t_2) \tag{8.25}$$

which, in the large size limit, becomes equivalent to (using (8.20), $A^2_{ai} \sim 1/m$, and (8.23))

$$\theta_{t+1} = \lambda + \frac{\theta_t}{m}\sum_{i\in\partial a}\eta'\big(\sum_{b\in\partial i\setminus a} A_{bi} r^t_{b\to i}\,;\,\theta_t\big) \tag{8.26}$$

Notice a nice property of the tresholding function: the derivative $\eta' = 0$ when $\eta = 0$ and $\eta' = 1$ when $\eta \neq 0$. This prompts us to introduce a notation for the "0-absolute value of a real number",

$$|z|_0 = \begin{cases} 1, & \text{if } z \neq 0, \\ 0, & \text{if } z = 0. \end{cases}$$

The update equation for θ_t can now be written in the nice form

$$\theta_{t+1} = \lambda + \frac{\theta_t}{m}\sum_{i\in\partial a} |x^{t+1}_{i\to a}|_0. \tag{8.27}$$

We have simplified (8.18), (8.19) down to (8.21),(8.24) and (8.27) but at this point we still have $\Theta(nm)$ messages to update at each iteration. A further simplification bringing this complexity down to linear order is the subject of the next subsection.

But before we address this issue it is useful to first consider the estimate obtained by minimizing $E_i(x_i)$ (see Equs. (8.13), (8.14)). Without going into similar calculations as above in detail, the reader should not be surprized that within the quadratic approximation

$$E_i^t(x_i) \approx \frac{1}{2\gamma_i^t}(x_i - \hat{x}_i^t)^2 + O((x_i - \hat{x}_i^t)^3), \tag{8.28}$$

where

$$\hat{x}_i^t = \eta(\tilde{a}_1^t\,;\,\tilde{a}_2^t), \tag{8.29}$$

and

$$\tilde{a}_1^t = \frac{\sum_{b\in\partial i} A_{bi}\alpha_{b\to i}^t}{\sum_{b\in\partial i} A_{bi}^2\beta_{b\to i}^2}, \qquad \tilde{a}_2^t = \frac{\lambda}{\sum_{b\in\partial i} A_{bi}^2\beta_{b\to i}^t}. \tag{8.30}$$

This leads to the estimate at time t of the form

$$\hat{x}_i^t = \eta(\sum_{b\in\partial i} A_{bi} r_{b\to i}^t\,;\,\theta_t). \tag{8.31}$$

In (8.31) all messages $r_{b\to i}^t$ entering nodes i are involved, wheras in (**??**) the message $r_{a\to i}^t$ is ommitted. This is a usual feature of mesage passing.

Finals steps

We are now ready to proceed to the final steps leading to the AMP algorithm. From (8.24) we have

$$\begin{aligned} x_{i\to a}^{t+1} &= \eta(\sum_{b\in\partial i} A_{bi} r_{b\to i}^t - A_{ai} r_{a\to i}^t\,;\,\theta_t) \\ &\approx \eta(\sum_{b\in\partial i} A_{bi} r_{b\to i}^t\,;\,\theta_t) - A_{ai} r_{a\to i}^t \eta'(\sum_{b\in\partial i} A_{bi} r_{b\to i}^t\,;\,\theta_t) \qquad (8.32) \\ &= \hat{x}_i^t - A_{ai} r_{a\to i}^t |\hat{x}_i^t|_0, \qquad (8.33) \end{aligned}$$

The second approximate equality above is obtained by a Taylor expansion to first order in $A_{ai} r_{a\to i}^t \sim 1/\sqrt{m}$. If you go back to chapter 7 you will see that a similar step was performed when we derived the TAP equations from teh BP equations for the SK model. This step is crucial and will lead to an "Onsager reaction term". The last equality follows by remarking again that that $\eta' = 1$ (resp. $\eta' = 0$) whenever $\eta \neq 0$ (resp. $\eta = 0$) and using (8.31). Replacing this

result in (8.21),

$$r^t_{a\to i} = y_a - \sum_{j\in\partial a\setminus i} A_{aj}\hat{x}^{t-1}_j + \sum_{j\in\partial a\setminus i} A^2_{aj} r^{t-1}_{a\to j}|\hat{x}^{t-1}_j|_0$$
$$= (y_a - \sum_{j\in\partial a} A_{aj}\hat{x}^{t-1}_j) + \sum_{j\in\partial a} A^2_{aj} r^{t-1}_{a\to j}|\hat{x}^{t-1}_j|_0 + A_{ai}\hat{x}^{t-1}_i - A^2_{ai} r^{t-1}_{a\to i}|\hat{x}^{t-1}_i|_0.$$

We see that $r^t_{a\to i}$ consists of a main term which is of order one and which is independent of i and the last two terms which do depend on i but which are of order $1/\sqrt{m}$ and $1/m$. So let us write

$$r^t_{a\to i} = r^t_a + \delta r^t_{a\to i}.$$

Up to leading order this yields for the main term

$$r^t_a \approx y_a - \sum_{j\in\partial a} A_{aj}\hat{x}^{t-1}_j + r^{t-1}_a \sum_{j\in\partial a} A^2_{aj}|\hat{x}^{t-1}_j|_0.$$

and for the next order term

$$\delta r^t_{a\to i} \approx A_{ai}\hat{x}^{t-1}_i$$

Using again $A^2_{ai} \sim \frac{1}{m}$ (note again for the Bernoulli model this is exact) the last two equations are summarized as

$$r^t_a = y_a - \sum_{j\in\partial a} A_{aj}\hat{x}^{t-1}_j + r^{t-1}_a \frac{\|\hat{\underline{x}}^{t-1}\|_0}{m}, \qquad \delta r^t_{a\to i} = A_{ai}\hat{x}^{t-1}_i. \tag{8.34}$$

Moreover, replacing $r^t_{b\to i} = r^t_b + \delta r^t_{b\to i} = r^t_b + A_{bi}\hat{x}^{t-1}_i$ in the LASSO estimate (8.31) we find

$$\hat{x}^t_i = \eta(\sum_{b\in\partial i} A_{bi} r^t_a + \sum_{b\in\partial i} A^2_{bi}\hat{x}^{t-1}_i\,;\,\theta_t)$$
$$= \eta(\sum_{b\in\partial i} A_{bi} r^t_a + \hat{x}^{t-1}_i\,;\,\theta_t). \tag{8.35}$$

Finally using the leading term in (8.33) the update equation (8.27) for θ_t becomes

$$\theta_{t+1} = \lambda + \theta_t \frac{\|\hat{\underline{x}}^t\|_0}{m}. \tag{8.36}$$

The first equation in (8.35) and (8.34), (8.36) form the AMP algorithm.

8.5 AMP algorithm for the LASSO

Let us now collect the fruits of our efforts and discuss the AMP algorithm. Recall that we are in the framework of an unknown prior signal distribution. With only minor extra effort we can derive a variant of AMP adapted to the case of a known (sparse) prior signal distribution when the MMSE estimator is used instead. This is discussed in Section 8.9.

The final AMP iterative equations (8.35), (8.34) we have arrived at can be written in a somewhat more compact notation

$$\begin{cases} \hat{x}_i^t = \eta(\hat{x}_i^{t-1} + (A^T \underline{r}^t)_i \,;\, \theta_t), \\ r_a^t = y_a - (A\underline{\hat{x}}^{t-1})_a + r_a^{t-1} \frac{\|\underline{\hat{x}}^t\|_0}{m}. \end{cases} \tag{8.37}$$

These have to be supplemented with the update equation (8.36) for the threshold,

$$\theta_{t+1} = \lambda + \theta_t \frac{\|\underline{\hat{x}}^t\|_0}{m} \tag{8.38}$$

Note the estimate $\underline{\hat{x}}^t$ obtained by this algorithm is an approximation of the min-sum estimate (8.13). For conceptual clarity, and also because we will shortly introduce another slightly more convenient variant, we will sometimes call it the *λ-AMP estimate* and the corresponding algorithm the *λ-AMP algorithm*.

Clearly, in (8.37) the "messages" now do not flow on edges but are emitted "isotropically" by vertices. In other words there are $\Theta(n)$ messages to update at each iteration; we have gained one order of complexity with respect to the initial message passing equations. This is similar to the situation we encountered with the TAP and BP equations for the SK model.

There are two reasons to somehow state the updates for θ_t as a separate equation. First, this equation does not depend on the vertices of the graph and as such is not really a message passing equation. Second there are other ways to update θ_t which are somewhat less costly in terms of computation and in practice lead to similar algorithmic performance. Here we discuss a variant of AMP with a simpler update of θ_t and whose performance can be precisely assessed as shown in the next two sections.

In the scalar case we saw in Section 8.1 that the threshold in η is naturally set on the scale of the noise, i.e. $\lambda = \alpha\sigma$ (and then the best possible α is determined by solving a minimax problem). In that case, σ was the standard variation of $y - x$. By analogy, for the vector case it is natural to take θ_t on the scale of the standard devistion of $(A^T \underline{r}^t)_i$ which is the term added to the estimate x_i^{t-1} in the first AMP equation (8.37). A rough guess for this standard deviation is

$$\sqrt{\underline{r}^T \mathbb{E}[AA^T] \underline{r}^t} = \frac{1}{m} \|\underline{r}^t\|_2^2.$$

Therefore we may take the heuristic value for the soft threshold at time t

$$\theta_t = \frac{\alpha}{\sqrt{m}} \|\underline{r}^t\|_2 \tag{8.39}$$

and replace it directly in the first AMP equation (8.37). This completely defines a useful variant of the AMP algorithm whose performance we will assess in Section 8.6. The corresponding algorithm and estimate $\underline{\hat{x}}^t$ will be called *α-AMP algorithm* and *α-AMP estimate*.

The AMP algorithm is almost the same than the much older Iterative Soft

Thresholding (IST) algorithm

$$\begin{cases} \hat{x}_i^t & = \eta(\hat{x}_i^t + (A^T \underline{r}^t)_i \,;\, \frac{\alpha}{\sqrt{m}}\|r^t\|_2), \\ {r_a}^t & = y_a - (A\underline{\hat{x}}^{t-1})_a. \end{cases}$$

The fundamental difference between IST and AMP lies in the "Onsager recation term", namely $r_a^{t-1}\frac{\|\hat{x}^{t-1}\|_0}{m}$. One can run experiments and check that this term is responsible for the improved performance of AMP over IST. One typically obtains a much smaller empirical MSE with much lesser iterations.

From the law of large numbers, one could perhaps hope that, when the IST algorithm is tested numerically for a given signal, the unthresholded estimate $\hat{x}_i^t + \frac{1}{\sqrt{m}}\sum_{b=1}^m \tilde{A}_{bi} r_b^t$ has a Gaussian histogram (here we set $A = \frac{1}{\sqrt{m}}\tilde{A}$). It is the subject of an exercise to show that *this is not so*. Correlations between the terms in the sum develop along the trajectory of the IST algorithm and the law of large numbers does not hold. Remarkably, it turns out that when the extra Onsager correction term is added to r_b^t (so the AMP algorithm is used) the histogram of the unthresholded estimate *becomes Gaussian*! The Onsager term has the effect of cancelling the correlations between the terms in the sum.

Again, the situation is exactly analogous to the one in the SK model. We saw that the naive Curie-Weiss local field, namely $\frac{1}{\sqrt{n}}\sum_{i=1;i\neq j}^n \tilde{J}_{ij} m_i^{t-1}$, does not have a Gaussian histogram; whereas when the Onsager correction is added the TAP local field, namely $\frac{1}{\sqrt{n}}\sum_{i=1;i\neq j}^n \tilde{J}_{ij} m_i^{t-1} - \beta m_j^{(t-1)}(1-q^{t-1})$, has a Gaussian histrogram.

8.6 Heuristic Derivation of State Evolution

In coding theory we derived DE equations that track the state of the BP algorithm, i.e. the probability distributions of messages. Density evolution then allows to compute the probability of a decoding error and more generally to assess the performance of a coding ensemble. There exist a similar formalism called State Evolution (SE) that tracks the state of the α-AMP algorithm and allows to calculate its performance. For the "state" at time t we take $\|\underline{\hat{x}}^t - \underline{x}\|^2$ the quadratic error of the estimator $\underline{\hat{x}}^t$ with the input signal $\underline{x}$. State Evolution tracks the average behavior of the state in the large size limit. In other words we seek an update equation for

$$\tau_t = \lim_{n\to+\infty} \frac{1}{n}\mathbb{E}[\|\underline{\hat{x}}^t - \underline{x}\|^2] \tag{8.40}$$

where the expectation is over the ensemble of measurement matrices, input signals and noise. The large system size limit (or "thermodynamic limit") is defined as $n, m \to +\infty$ with fixed undersampling rate $\delta = m/n$ and fixed sparsity parameter $\rho = k/m = \epsilon\delta$. We interchangeably take take (δ, ρ) or (δ, ϵ) as our free parameters.

The key feature that allows us to derive a closed form equation relating τ_{t+1}

to τ_t is the Gaussianity of the untresholded estimate (given the input signal). As explained in the previous section numerical experiments show with the Onsager term present the sum $\frac{1}{\sqrt{m}}\sum_{b\in\partial i}\tilde{A}_{bi}r_b^t$ behaves as if the law of large numbers applied (from now on we set $\tilde{A}=\frac{1}{\sqrt{m}}A$). Effectively, one could therefore remove the Onsager term from the algorithm if one sampled afresh the measurement matrix at each time step so that the law of large numbers applies. This observation is at the basis of the "conditioning technique" (originally developped by E. Bolthauzen for the derivation of the RS equation from the TAP equations in the SK model) which allows for a rigorous derivation of SE. The rigorous proofs would lead us too far here, and we will simply accept based on numerical observations, that the Onsager term $r_a^{t-1}\frac{\|\underline{\hat{x}}^t\|_0}{m}$ can be removed if simultaneously we replace the quenched measurement matrix elements $\tilde{A}_{bi}$ by new iid realizations $\tilde{A}_{bi}^t$ sampled afresh from $\mathcal{N}(0,1)$ or uniformly from $\{-1,+1\}$.

In other words we are analyzing the following set of equations

$$\begin{cases}\hat{x}_i^t &= \eta(\hat{x}_i^t+\frac{1}{\sqrt{m}}(\tilde{A}^{t\,T}\underline{r}^t)_i\,;\,\frac{\alpha}{\sqrt{m}}\|\underline{r}^t\|_2),\\ r_a{}^t &= (\frac{1}{\sqrt{m}}(\tilde{A}^t\underline{x})_a+z_a)-\frac{1}{\sqrt{m}}(\tilde{A}^t\underline{\hat{x}}^{t-1})_a.\end{cases}\tag{8.41}$$

where to be consistent we have also replaced the measurements $\underline{y}=\frac{1}{\sqrt{m}}\tilde{A}\underline{x}+\underline{z}$ by "new measurements" at each time step $\underline{y}^t=\frac{1}{\sqrt{m}}\tilde{A}^t\underline{x}+\underline{z}$, $z_a\sim\mathcal{N}(0,1)$.

We will show that in thermodynamic limit: (i) the first argument of the thresholding function in (8.41) tends to a Gaussian with mean $\underline{x}$ and variance $(\sigma^2+\frac{\tau_t^2}{\delta})^{1/2}$; (ii) the second argument $\frac{\alpha}{\sqrt{m}}\|\underline{r}^t\|_2)$ tends to $\alpha(\sigma^2+\frac{\tau_t^2}{\delta})^{1/2}$. Thus from (8.41) each component of the α-AMP estimate at time $t+1$ is distributed as the *random variable*

$$\hat{x}^{t+1}=\eta(x+\sqrt{\sigma^2+\frac{\tau_t^2}{\delta}}u\,;\,\alpha\sqrt{\sigma^2+\frac{\tau_t^2}{\delta}})\tag{8.42}$$

where $u\sim\mathcal{N}(0,1)$ and $x\sim p_0(\cdot)$. Using definition (8.40) (and the symmetry with respect to permutation of vertices) the corresponding normalized average MSE satisfies the SE equation

$$\begin{aligned}\tau_{t+1}^2&=\mathbb{E}[(\hat{x}^{t+1}-x)^2]\\&=\int dx\,p_0(x)\int du\,\frac{e^{-\frac{u^2}{2}}}{\sqrt{2\pi}}\left[\eta(x+\sqrt{\sigma^2+\frac{\tau_t^2}{\delta}}u\,;\,\alpha\sqrt{\sigma^2+\frac{\tau_t^2}{\delta}})-x\right]^2.\end{aligned}\tag{8.43}$$

The consequences of SE for the phase diagram of the phase diagram of the AMP algorithm are discussed in the next section. For completeness we give a somewhat informal proof of this update equation.

Technical details leading to (8.43)

Let us first show point (i) above. Merging the two equations together in (8.41) the first argument of teh thresholding function is

$$x_i + \frac{1}{\sqrt{m}} \sum_{b=1}^{m} \tilde{A}_{bi}^{(t)} z_b + \sum_{j=1}^{n} (\delta_{ij} - \frac{1}{m} (\tilde{A}^{(t)\top} \tilde{A}^{(t)})_{ij}) (\hat{x}_j^{(t-1)} - x_j) \tag{8.44}$$

We discuss the behavior of each sum in (8.44), in the thermodynamic limit. Clearly, given $\underline{z}$, from the central limit theorem

$$\frac{1}{\sqrt{m}} \sum_{b=1}^{m} \tilde{A}_{bi}^{(t)} z_b \tag{8.45}$$

tends to a Gaussian with zero mean and variance $\frac{1}{m} \sum_{b=1}^{m} z_b^2 \to \sigma^2$. Next, again by the central limit theorem, one shows that the matrix entries $(\delta_{ij} - \frac{1}{m} (\tilde{A}^{(t)\top} \tilde{A}^{(t)})_{ij})$ tend to a zero mean Gaussian with variance $1/m$. By looking at the covariance of these entries we see that they are independent to leading order. Thus the term

$$\sum_{j=1}^{n} (\delta_{ij} - \frac{1}{m} (\tilde{A}^{(t)\top} \tilde{A}^{(t)})_{ij}) (\hat{x}_j^{(t)} - x_j) \tag{8.46}$$

is also a Gaussian, with zero mean and variance

$$\frac{1}{m} \sum_{j=1}^{n} (\hat{x}_j^{(t)} - x_j)^2 = \frac{1}{\delta} \frac{1}{n} \|\underline{\hat{x}}^{(t)} - \underline{x}\|_2^2 \to \frac{\tau_t^2}{\delta}$$

In the very last step we use conecntration of the Euclidean norm on its average. Finally, one can look at the covariance of the two approximate Gaussian variables in (8.45) and (8.46) and show that they are approximately independent. Let us summarize: we have obtained that in the thermodynamic limit (8.45) is $\mathcal{N}(0, \sigma^2)$, that (8.46) is $\mathcal{N}(0, \frac{1}{\delta}(\tau^{(t)})^2)$, and that they are independent. Thus their sum is $\mathcal{N}(0, \sigma^2 + \frac{1}{\delta}(\tau^{(t)})^2)$ and the first argument of the thresholding function (8.44) tends to the random variable

$$x + (\sigma^2 + \frac{1}{\delta}(\tau^{(t)})^2)^{1/2} u \tag{8.47}$$

where $u \sim \mathcal{N}(0,1)$ and $x \sim p_0(\cdot)$ as announced.

It remains to show point (ii). Using the second equation in (8.41) and expanding the Euclidean norm,

$$\begin{aligned} \|\underline{r}\|_2^2 &= \sum_{b=1}^{m} \left(z_b + \frac{1}{\sqrt{m}} \sum_{i=1}^{n} A_{bi}^{(t)} (x_i - \hat{x}_i^{(t)}) \right)^2 \\ &= \sum_{b=1}^{m} z_b^2 + \frac{2}{\sqrt{m}} \sum_{b=1}^{m} \sum_{i=1}^{n} z_b \tilde{A}_{bi}^{(t)} (x_i - \hat{x}_i^{(t)}) \\ &+ \sum_{b=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} \tilde{A}_{bi}^{(t)} \tilde{A}_{bj}^{(t)} (x_i - \hat{x}_i^{(t)}) (x_j - \hat{x}_j^{(t)}) \end{aligned}$$

Clearly the first term tends to $\alpha^2\sigma^2$. By similar arguments as in point (i) the second term can be shown to tend to zero and the third term to $\frac{\alpha^2}{\delta}(\tau^{(t)})^2$. Thus in the thermodynamic limit

$$\frac{\alpha}{\sqrt{m}}\|\underline{r}^t\|_2 \to \alpha\sqrt{\sigma^2 + \frac{1}{\delta}(\tau^{(t)})^2}. \tag{8.48}$$

as announced.

8.7 Performance of AMP

In this section we derive the phase diagram of AMP in the plane of parameter (ϵ, δ) where we recall that $\epsilon = k/n$ is the fraction of non-zero components in the signal and $\delta = m/n$ is the fraction of measurements (also called undersampling rate). It is also common in the literature, but somehow less natural, to parametrize the phase diagram in terms of (ρ, δ) where $\rho = k/m = \epsilon/\delta$.

The phase diagram of the algorithm is deduced from a study of SE updates (8.43) and the first question we should address is to determine the multiplicity of solutions of the corresponding fixed point equation

$$\tau^2 = \int dx p_0(x) \int du \frac{e^{-\frac{u^2}{2}}}{\sqrt{2\pi}} \left[\eta\big(x + \sqrt{\sigma^2 + \frac{\tau^2}{\delta}}u\,;\, \alpha\sqrt{\sigma^2 + \frac{\tau^2}{\delta}}\big) - x\right]^2. \tag{8.49}$$

It is the subject of an exercise to show that this equation has a *unique solution* $\tau_*^2(\epsilon, \delta, \alpha, p_0, \sigma)$ in the extended real line $[\sigma^2, +\infty]$. Therefore the SE iterations will tend this fixed point solution.

It is useful to note for further use the following property of the fixed point solution,

$$\tau_*^2(\epsilon, \delta, \alpha, p_0, \sigma) = \sigma^2\tau_*^2(\epsilon, \delta, \alpha, p_0^\sigma, 1), \tag{8.50}$$

where $p_0^\sigma(x) = \sigma p_0(\sigma x) = (1-\epsilon)\delta(x) + \sigma p_0(\sigma x)$. To prove (8.50) we set $\tau = \sigma\tau_1$ and notice that τ_1 satisfies the fixed point equation (8.49) with σ and p_0 replaced by 1 and p_0^σ respectively. This last point is seen by making the change of variables $x \to \sigma x$ and using the property $\eta(\sigma y\,;\, \sigma\lambda) = \sigma\eta(y\,;\,\lambda)$.

We also remark that $p_0^\sigma \in \mathcal{F}_\epsilon$ if $p_0 \in \mathcal{F}_\epsilon$, in other word the class of distributions $\mathcal{F}_\epsilon$ is scale invariant. This scale invariance property will play a crucial role as we will shortly see.

Minimax Criterion and noise sensitivity phase transition

We have to make a suitable choice for the parameter α in the α-*AMP algorithm*. Recall, since p_0 is unknown, we must choose the best possible α given the worst possible $p_0 \in \mathcal{F}_\epsilon$. Formally we have to compute the minimax-MSE,

$$\inf_{\alpha\geq 0}\sup_{p_0\in\mathcal{F}_\epsilon} {\tau_*}^2(\epsilon, \delta, \alpha, p_0, \sigma), \tag{8.51}$$

Using (8.50) and the scale invariance of $\mathcal{F}_\epsilon$ we find

$$\begin{aligned}\inf_{\alpha\geq 0}\sup_{p_0\in\mathcal{F}_\epsilon}{\tau_*}^2(\epsilon,\delta,\alpha,p_0,\sigma) &= \sigma^2\inf_{\alpha\geq 0}\sup_{p_0\in\mathcal{F}_\epsilon}{\tau_*}^2(\delta,\rho,\alpha,p_0^\sigma,1)\\ &= \sigma^2\inf_{\alpha\geq 0}\sup_{p_0\in\mathcal{F}_\epsilon}{\tau_*}^2(\delta,\rho,\alpha,p_0,1)\\ &\equiv \sigma^2 M(\epsilon,\delta). \end{aligned} \tag{8.52}$$

The quantity $M(\epsilon,\delta)$ is the rate of change (or the "response") of the minimax-MSE under variations of the noise. It is often called the *noise sensitivity*.

Remarkably, the noise sensitivity is independent of the level of noise. A look at the derivation above shows that this is due to the scale invariance of the class of sparse distributions. It turns out that scale invariance has more consequences: it allows to easily derive an explicit formula for the noise sensitivity

$$M(\epsilon,\delta)=\begin{cases}\frac{M_{\text{scalar}}(\epsilon)}{1-\frac{1}{\delta}M_{\text{scalar}}(\epsilon)} & \delta>\delta_c(\epsilon)\\ +\infty & \delta<\delta_c(\epsilon),\end{cases} \tag{8.53}$$

where $\delta_c(\epsilon)=M_{\text{scalar}}(\epsilon)$. Moreover the minimax point $p_{0,\text{worst}}$, α_{best} is the same as the one for the scalar problem in Section 8.1.

Figure 8.2 shows the phase diagram of the AMP algorithm. The curve $\delta_c(\epsilon)$

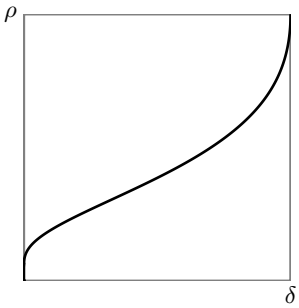


Figure 8.2 Left: the algorithmic noise sensitivity phase transition line in the (ϵ,δ) plane. Right: the same line in the (δ,ρ) plane.

is an algorithmic phase transition line, which separates the (ϵ,δ) plane in two regions. Below the curve the undersampling rate is too small and noise sensitivity (as well as minimax-MSE) is infinite, and there is no hope to recover the sparse signal with the AMP estimate. Above the curve, the sampling rate is large enough so that we can recover the signal with finite error.

We point out that the noise sensitivity phase transition line has a rather explicit parametrized form whose derivation is the subject of an exercise.

$$\begin{cases}\delta=\dfrac{2\frac{e^{-\frac{\alpha^2}{2}}}{\sqrt{2\pi}}}{\alpha+2\left(\frac{e^{-\frac{\alpha^2}{2}}}{\sqrt{2\pi}}-\alpha\Phi(-\alpha)\right)}\\ \epsilon=\dfrac{2\left(\frac{e^{-\frac{\alpha^2}{2}}}{\sqrt{2\pi}}-\alpha\Phi(-\alpha)\right)}{\alpha+2\left(\frac{e^{-\frac{\alpha^2}{2}}}{\sqrt{2\pi}}-\alpha\Phi(-\alpha)\right)}.\end{cases} \tag{8.54}$$

Derivation of (8.53)

The starting point is again a scaling argument applied to the fixed point equation (8.49). With the change of variables $x \to \sqrt{\sigma^2 + \frac{\tau^2}{\delta}}x$ we obtain

$$\tau^2 = (\sigma^2 + \frac{\tau^2}{\delta})M_{\text{scalar}}(\epsilon, \alpha, p_0^\tau) \tag{8.55}$$

with

$$M_{\text{scalar}}(\epsilon, \alpha, p_0^\tau) = \int dx p_0^\tau(x) \int du \frac{e^{-\frac{u^2}{2}}}{\sqrt{2\pi}} [\eta(x+u\,,\,\alpha) - x]^2 \tag{8.56}$$

and $p_0^\tau(x) = \sqrt{\sigma^2 + \frac{\tau^2}{\delta}} p_0(\sqrt{\sigma^2 + \frac{\tau^2}{\delta}}x)$. Looking back at the solution of the LASSO for the scalar problem we see that $M_{\text{scalar}}(\epsilon, \alpha, p_0^\tau)$ is noting else than the *scalar* MSE for a scaled signal distribution p_0^τ and noise level $\sigma^2 = 1$. Remark also that scale invariance of $\mathcal{F}_\epsilon$ implies

$$\sup_{p_0 \in \mathcal{F}_\epsilon} M_{\text{scalar}}(\epsilon, \alpha, p_0^\tau) = \sup_{p_0 \in \mathcal{F}_\epsilon} M_{\text{scalar}}(\epsilon, \alpha, p_0) = M_{\text{scalar}}(\epsilon, \alpha) \tag{8.57}$$

where the supremum is attained for $p_{0\,\text{worst}}$.

Suppose the parameters are such that $M_{\text{scalar}}(\epsilon, \alpha) > \delta$. Then replacing p_0 by $p_{0,\text{worst}}$ in (8.55) we find that the only solution is $\tau_*(\epsilon, \delta, \alpha, p_{0,\text{worst}}, \sigma) = +\infty$. Therefore we necessarily have $\sup_{p_0 \in \mathcal{F}_\epsilon} \tau_* = +\infty$ when $M_{\text{scalar}}(\epsilon, \alpha) > \delta$. On the other hand if $M_{\text{scalar}}(\epsilon, \alpha) < \delta$ we also have $M_{\text{scalar}}(\epsilon, \alpha, p_0^\tau) < \delta$ and Equ. (8.55) has a finite solution,

$$\tau_*^2 = \sigma^2 \frac{M_{\text{scalar}}(\epsilon, \alpha, p_0^{\tau_*})}{1 - \frac{1}{\delta} M_{\text{scalar}}(\epsilon, \alpha, p_0^{\tau_*})} \tag{8.58}$$

This ratio is an increasing function of $M_{\text{scalar}}(\epsilon, \alpha, p_0^{\tau_*})$ so it also follows that for $M_{\text{scalar}}(\epsilon, \alpha) < \delta$

$$\sup_{p_0 \in \mathcal{F}_\epsilon} \tau_*^2 = \sigma^2 \frac{M_{\text{scalar}}(\epsilon, \alpha)}{1 - \frac{1}{\delta} M_{\text{scalar}}(\epsilon, \alpha)}. \tag{8.59}$$

Now it remains to minimise over α. Recall $\inf_\alpha M_{\text{scalar}}(\epsilon, \alpha) = M_{\text{scalar}}(\epsilon)$. So when α varies over the positive real line, $M_{\text{scalar}}(\epsilon, \alpha)$ varies over $[M_{\text{scalar}}(\epsilon), +\infty[$. Since the ratio in (8.59) is an increasing function of $M_{\text{scalar}}(\epsilon, \alpha)$ which diverges at $M_{\text{scalar}}(\epsilon, \alpha) = \delta$ (and remains infinite thereafter), its minimum is attained at $M_{\text{scalar}}(\epsilon)$ when $M_{\text{scalar}}(\epsilon) < \delta$ and at $+\infty$ when $M_{\text{scalar}}(\epsilon) > \delta$. This is precisely the statement of (8.53).

8.8 Relations between λ-AMP, α-AMP and LASSO

We wish to revisit here a few issues that have perhaps been swept under the rug. We started by formulating a minimization problem (8.1) which yields the LASSO. We cannot a priori solve this problem analytically (except for the scalar

case) so we settled for a min-sum approach. After several natural approximations of the min-sum equations we where led to the λ-AMP algorithm, which gives an estimate of the signal parametrized by λ. We switched to a variant of this algorithm, the α-AMP algorithm, which gives an estimate parametrized by α instead. The reason for introducing this variant is that its performance can be neatly analyzed thanks to SE.

This approach raises two natural questions. First, what is the relation between the λ-AMP and α-AMP algorithms and how do their performance compare? In particular, do they have the same phase transition line? Second, what is the relation between the true LASSO and AMP estimates? The first question is a purely algorithmic one, whereas the second really belongs to the third part of the course where we discuss the relations between message passing algorithms and optimal solutions. In the present case we can quite simply obtain at least partial answers which are worth stating immediately.

The Hamiltonian (8.10) is a convex function of $\underline{x} \in \mathbb{R}^n$, so the minima are solutions of the stationnarity condition

$$A^T(\underline{y} - A\underline{x}) = \lambda \underline{v} \tag{8.60}$$

where $v_i = \mathrm{sign}(x_i)$ for $x_i \neq 0$ and $v_i \in [-1, +1]$ for $x_i = 0$.

Take the λ-AMP equations (8.37), (8.38) and consider a fixed point $(\underline{\hat{x}}^*, \underline{r}^*, \theta_*)$. One can see that $\underline{x}^*$ satisfies (8.60) provided we take in that equation $\lambda = \theta_*(1 - \frac{\|\underline{\hat{x_*}}\|_0}{m})$ (this is also a condition that the fixed point of λ-AMP must satisfy, see (8.38)). We conclude that, for any fixed λ, when the λ-AMP updates converge to a fixed point, this fixed point is also a solution of the LASSO minimization problem (8.1).

Consider now the α-AMP update equations (8.37), (8.39) and a corresponding fixed point $(\underline{\hat{x}}^*, \underline{r}^*)$. This time $\underline{x}^*$ satisfies (8.60) provided we take $\lambda = \alpha \frac{\|\underline{r}\|_2}{\sqrt{m}}(1 - \frac{\|\underline{\hat{x_*}}\|_0}{m})$. Using the analysis of Section 8.6 (specifically (8.47) and (8.48)) this relation becomes in thermodynamic limit

$$\lambda(\alpha) = \alpha\sqrt{\sigma^2 + \tau_*^2}\left(1 - \frac{1}{\delta}\int dx\, p_0(x) \int du\, \frac{e^{-\frac{u^2}{2}}}{\sqrt{2\pi}}\left[\eta'(x + \sqrt{\sigma^2 + \tau_*^2}u\,;\, \alpha\sqrt{\sigma^2 + \tau_*^2})\right]\right). \tag{8.61}$$

The α-AMP and $\lambda(\alpha)$-AMP algorithms converge to the same fixed point (and this fixed point is a solution of teh LASSO minimization problem).

These remarks also show that the two variants of AMP are equivalent in terms of performance in the large size limit. In particular the noise sensitivity phase transition line is the same.

8.9 A variant of AMP for the MMSE estimator

Even if this perhaps a less realistic situation, it is instructive to consider the case of a signal with *known* prior distribution from the class $\mathcal{F}_\epsilon$. In other words

$p_0(x) = (1-\epsilon)\delta_0(x) + \epsilon\phi_0(x)$ for a known $\phi_0(x)$. A good example to keep in mind is a Gaussian distribution $\phi_0(x) = (1/\sqrt{2\pi})e^{-x^2/2}$; one then refers to $p_0(x)$ as the Bernoulli-Gaussian model.

As explained in Chapter 3, in this setting the optimal estimator is the MMSE estimator (3.33). Since we cannot a priori hope to compute it exactly we resort to a message passing calculation. In Chapter 5 we went through the BP equations in Example 16, and this approach can be systematically developed in order to recursively compute the BP-estimate for the signal. The complexity of the message passing step is again quadratic because the factor graph is bipartite complete; but following the same route as in Sections 8.3, 8.4, the message-passing equations can be simplified in order to arrive at an AMP algorithm (that we will call mmse-AMP) that is very similar to (8.37). Instead of taking this lengthy route, by skimming through the previous results one can make an educated guess of the form of the new algorithm.

In Section 8.5 the AMP algorithm uses the soft thresholding function $\eta(y\,;\,\lambda)$ found by solving the scalar LASSO problem. The reader should not be too surprised that now the AMP updates will involve a *thresholding function given by the MMSE estimator of the scalar case*. Consider a scalar measurement $y = x+z$ of "signal" x affected by Gaussian noise with variance ν^2. The thresholding function is

$$\eta_0(y\,;\,\nu) = \mathbb{E}[X|X+Z=y] = \frac{\int dx\, x\, p_0(x) e^{-\frac{(y-x)^2}{2\nu^2}}}{\int dx\, p_0(x) e^{-\frac{(y-x)^2}{2\nu^2}}}. \tag{8.62}$$

We stress that, contrary to the case of LASSO, $\eta_0(y\,;\,\nu)$ is not universal and depends on the prior. Here ν plays the role of a threshold level analogous to λ. The mean square error for this optimal estimator (of the scalar problem) is the MMSE function (by convention the argument of the MMSE function is a signal-to-noise-ratio, here ν^{-2})

$$\begin{aligned}\mathrm{mmse}(\nu^{-2}) &= \mathbb{E}\big[(X - \mathbb{E}[X|X+Z])^2\big] \\ &= \int dx\, p_0(x) \int dz\, \frac{e^{-\frac{z^2}{2\nu^2}}}{\sqrt{2\pi\nu^2}} [\eta_0(x+z\,;\,\nu) - x]^2.\end{aligned} \tag{8.63}$$

The mmse-AMP updates (for the vector case) are the similar to (8.37)

$$\begin{cases} \hat{x}_i^{t+1} = \eta_0(x_i^t + (A^T \underline{r}^t)_j\,;\,\nu_t), \\ r_a^t = y_a - (A\underline{\hat{x}}^t)_a^{(t-1)} + b_t r_a^{t-1}. \end{cases} \tag{8.64}$$

with a number of differences that we now discuss. As already pointed out, naturally η_0 replaces η. The Onsager term is also different. In the derivations of Section 8.4 it can be traced back to a derivative of the soft thresholding function. We can therefore guess that now

$$b_t = \frac{1}{m}\sum_{i=1}^{n} \eta_0'\big(x_i^{t-1} + (A^T \underline{r}^t)_a\,;\,\nu_t\big). \tag{8.65}$$

Finally recall that for the α-AMP algorithm we expressed in Section 8.6 the threshold level thanks to the MSE through the relation $\theta_t = \alpha\sqrt{\sigma^2 + \frac{\tau_t^2}{\delta}}$. Here the analysis leads to a similar conclusion, namely

$$\nu_t^2 = \sigma^2 + \frac{\tau_t^2}{\delta}, \tag{8.66}$$

(where by definition $\tau_t^2 = \lim_{n\to+\infty} \frac{1}{n}\mathbb{E}\|\hat{\underline{x}}^{(t)} - \underline{x}\|_2$). Note that the MMSE problem does not involve any parameter λ or α over which one should optimise. Note also that to run the mmse-AMP updates (8.64) one has to precompute τ_t. To do this one has to write down the corresponding SE equations.

The performance analysis follows the same steps than in Section **??**. The result is a SE recursion with η_0 replacing η

$$\begin{aligned}\tau_{t+1}^2 &= \mathrm{mmse}((\sigma^2 + \frac{\tau_t^2}{\delta})^{-1}) \\ &= \int dx\, p_0(x) \int du\, \frac{e^{-\frac{u^2}{2}}}{\sqrt{2\pi}} \left[\eta_0\left(x + u\sqrt{\sigma^2+\tau_t^2}\,;\, \sqrt{\sigma^2+\tau_t^2}\right) - x\right]^2. \end{aligned} \tag{8.67}$$

This equation has a nice interpretation: at time $t+1$ the total quadratic error τ_{t+1}^2 for the mmse-AMP estimate is given by the MMSE of a scalar signal with effective noise variance $\sigma^2 + \frac{\tau_t^2}{\delta}$ at time t.

Let us summarize. Equations (8.67) and (8.66) give the evolution of the MSE and the threshold level. These quantities can be precomputed. Equations (8.64), (8.65), (8.66) define the mmse-AMP algorithm, and allow to compute the estimates for the signal.

We now turn our attention towards the phase diagram of the mmse-AMP. As usual we must get a hold on the solutions of the fixed point equation corresponding to (**??**). Contrary to the LASSO case where only one solution exists, here the situation is more complicated and multiple solutions can appear. Moreover for the LASSO the solution could be determined rather because of scale invariance. In the present case there is no such scale invariance since $p_0(x)$ is a fixed distribution but it is still possible to make qualitative statements that are valid for a fairly wide class of distributions. Moreover the phase transition line can precisely characterised in a simple manner. For the Bernoulli-Gauss model $\eta_0(y\,;\,s)$ can be explicitly be computed and all statements fairly explicitly checked; this is the subject of an exercise.

Define

$$\tilde{\delta}(p_0) \equiv \sup_{\nu}\{\nu^{-2}\mathrm{mmse}(\nu^{-2})\} \tag{8.68}$$

From $\lim_{\nu\to 0}\nu^{-2}\mathrm{mmse}(\nu^{-2}) = \epsilon$ we immediately deduce the general inequality $\tilde{\delta}(p_0) > \epsilon$. For a sampling rate $\delta > \tilde{\delta}(p_0)$ there exists only one fixed point solution called $\tau^2_{*,\mathrm{good}}$ such that the "noise sensitivity" $\lim_{\sigma\to 0}(\tau^2_{*,\mathrm{good}}/\sigma^2)$ remains finite. Thus for $\delta > \tilde{\delta}(p_0)$ the algorithm yields a correct reconstruction in the small noise limit $\sigma \to 0$ (and more generally a finite error for finite noise). Now, decrease

the sampling rate in the range $\epsilon < \delta < \tilde{\delta}(p_0)$. One finds two or more stable fixed points (as well as unstable ones) for all $\sigma^2 > 0$. Besides the "good" fixed point which satisfies $\tau^2_{*,\text{good}} = O(\sigma^2)$ there is a "bad" one, i.e. $\tau^2_{*,\text{bad}} = \Theta(1)$ as $\sigma \to 0$. Clearly, under the (natural) initial condition $\tau_0^2 = +\infty$ one always tends to the largest stable fixed point i.e $\tau^2_{*,\text{bad}}$. This means that the noise sensitivity $\lim_{\sigma\to 0}(\tau^2_{*,\text{bad}}/\sigma^2)$ diverges, and exact reconstruction is not possible even for very small noise.

We can therefore conclude that $\tilde{\delta}(p_0)$ is the algorithmic phase transition threshold of mmse-AMP, a remarkably neat result! This threshold is lower than the LASSO threshold derived in Section 8.6. This is not too surprising since the later concerns the worst case distribution for $p_0 \in \mathcal{F}_\epsilon$. Note also that the inequality $\tilde{\delta}(p_0) > \epsilon$ now appears as trivial; it just syas that the algorithmic threshold is higher than the "optimal" one. It is instructive to compute the phase diagram of mmse-AMP in the (ϵ, δ) plane for the Bernoulli-Gauss model and compare with the LASSO and optimal phase transition lines (see exercises). The result is illustrated on figure **??**.

Problems

8.1 A generalization of IST and its connection to LASSO. The standard Iterative Soft Thresholding algorithm has the form

$$\begin{cases} x_i^{t+1} = \eta(x_i^t + (A^T \underline{r}^t)_i \,;\, \lambda) \\ \underline{r}^t = \underline{y} - A\underline{x}^t \end{cases}$$

starting from the initial condition $x_i^0 = 0$. Consider the following generalization. Let θ_t and b_t be two sequences of scalars (called respectively "thresholds" and "reaction terms") that converge to fixed numbers θ and b. Construct the sequence of estimates according to the iterations

$$\begin{cases} x_i^{t+1} = \eta(x_i^t + (A^T \underline{r}^t)_i \,;\, \theta_t) \\ \underline{r}^t = \underline{y} - A\underline{x}^t + b_t r^{t-1} \end{cases}$$

The goal of the exercise is to prove that if x^*, r^* is a fixed point of these iterations, then x^* is a stationary point of the LASSO cost function $\mathcal{H}(\underline{x}|\underline{y}, A) = \frac{1}{2}||\underline{y} - A\underline{x}||_2^2 + \lambda||x||_1$ for $\lambda = \theta(1-b)$.

Note that this theorem does not say how to specify suitable sequences b_t and θ_t. The point of AMP is that it specifies unambiguously that one should take $b_t = ||x||_0/m$ (for θ_t there is more flexibility).

The proof proceeds in two steps. First, show that the stationarity condition for the LASSO cost function is

$$A^T(\underline{y} - A\underline{x}^*) = \lambda \underline{v},$$

where $v_i = \text{sign}(x_i^*)$ for $x_i^* \neq 0$ and $v_i \in [-1, +1]$ for $x_i^* = 0$. Second, show that

the fixed point equations corresponding to the iterations above are

$$x_i^* + \theta v_i = x_i^* + (A^T \underline{r}^*)_i$$
$$(1-b)\underline{r}^* = \underline{y} - A\underline{x}^*$$

Third, remark that these two steps implie $\lambda = \theta(1-b)$.

8.2 Statistics of AMP and IST un-thresholded estimates. Consider a sparse signal $\underline{x}_0$ with n iid components distributed as $(1-\epsilon)\delta(x_0) + \frac{\epsilon}{2}\delta(x-1) + \frac{\epsilon}{2}\delta(x+1)$. Generate m noisy measurements $\underline{y} = \frac{1}{\sqrt{m}}\tilde{A}\underline{x} + \underline{z}$ where $\tilde{A}_{ai}$ are iid uniform in $\{+1, -1\}$ and z_a are iid Gaussian zero mean and variance σ^2.

Consider the AMP iterations (8.37) with the choice $\theta^{(t)} = \alpha\|\underline{r}^{(t)}\|_2/\sqrt{m}$. The derivation of state evolution rests on the assumption that the i-th component, given $\underline{x}_0$, of the un-thresholded estimate

$$\hat{x}_i^{(t)} + \frac{1}{\sqrt{m}}\sum_{b=1}^{m} \tilde{A}_{bi} r_b^{(t)},$$

has Gaussian statistics. The mean is x_{0i} and the variance $\sigma^2 + (\tilde{\tau})^{(2)}$ where $(\tilde{\tau})^{(2)} = \|\underline{x}^{(t)} - \underline{x}_0\|_2^2/n$.

Perform an experiment to check this numerically. Compute also the statistics of the un-thresholded estimate for the IST iterations, i.e. when the Onsager term $r_a^{t-1}\frac{\|\hat{x}^{(t)}\|_0}{m}$ is removed. Compare the two histograms.

Indications: Fix a signal realization $\underline{x}_0$. Try $n = 4000$, $m = 2000$, $\epsilon = 0.125$ and 40 instances for A and $\underline{z}$. Try various values for σ and α. Look at the i-th components of the un-thresholded estimate for components such that say $x_{0i} = +1$ (or -1, or 0).

8.3 Unicity of solution of SE fixed point equation. Consider the SE fixed point equation (8.54). Show that there is a unique fixed point solution in $[\sigma^2, +\infty]$ (the value $+\infty$ included). Hint: write the fixed point equation for the new variable $\tilde{\tau}^2 = \sigma^2 + \tau^2/\delta$ in the form $\tilde{\tau}^2 = F(\tilde{\tau})$ and show that F is a concave function of $\tilde{\tau}$. Proceed graphically.

8.4 Noise sensitivity phase transition. Derive the parametrised from (8.54) of the noise sensitivity phase transition line.

8.5 mmse-AMP algorithm. Give the details of the derivation of the mmse-AMP algorithm (8.64), (8.65), (8.66) and those of the corresponding state evolution equation (8.67).

8.6 Bernoulli-Gauss model. Consider the prior $p_0(x) = (1-\epsilon)\delta(x) + \epsilon\frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}$. Show that the soft thresholding function (8.62) is

$$\eta_0(y\,;\,\nu) = \frac{y}{1+\nu^2}\,\frac{\epsilon\frac{e^{-\frac{y^2}{2(1+\nu^2)}}}{\sqrt{2\pi(1+\nu^2)}}}{\epsilon\frac{e^{-\frac{y^2}{2(1+\nu^2)}}}{\sqrt{1+\nu^2}} + (1-\epsilon)\frac{e^{-\frac{y^2}{2\nu^2}}}{\sqrt{\nu^2}}}$$

and the mmse function (8.62)

$$\mathrm{mmse}(\nu^{-2}) = \epsilon - \frac{\epsilon}{1+\tau^2}\int_{-\infty}^{+\infty} dy\, y^2 \frac{\frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}}}{1+\frac{1-\epsilon}{\epsilon}\sqrt{\frac{1+\tau^2}{\tau^2}}e^{-\frac{y^2}{2\tau^2}}}$$

Finally analyse the solutions of the mmse-AMP fixed point equation when the undersampling rate satisfies $\delta > \tilde{\delta}(p_0)$ and $\epsilon < \delta < \tilde{\delta}(p_0)$. Plot the phase transition line $\tilde{\delta}(p_0)$ and compare with the LASSO phase transition line.

9 Random K-SAT: a first approach

The satisfiability problem is considerably more difficult to analyze than either coding or compressive sensing. One reason for this difficulty is that random K-SAT is not an inference problem. Indeed, in the regime where a random formula is SAT with high probability (i.e., in the regime where the number of clauses per Boolean function is sufficiently small) there are exponetially many solutions contrary to coding or compressive sensing where we typically only have one valid solution. At first we might guess that this makes the problem easy: We are not asking for a *particular* solution – *any* solution will do! But in fact it is exactly this lack of uniqueness which makes the problem hard.

Why does this non-uniqueness cause trouble? Pick a specific Boolean variable. From the perspective of this variable this means that there are typically solutions for which this variable takes on the value 0 but also solutions for which it takes on the value 1. In fact, of the exponentially many solutions there are typically roughly equally many of either type. So even if the message-passing algorithm succeeded in computing the marginals of all bits correctly (here we assume that we put a uniform measure on all solutions and compute the marginal wrt this measure) all these marginals would we uniform and we cannot extract from them a globally valid solution. Therefore a straighforward application of a message-passing algorithm does not work. A new ingredient is needed.

One approach is quite natural given the above description. Assume for a second that message-passing is capable of accurately computing marginals. Then we can proceed as follows. Compute the marginal for one variable. As long as this marginal does not put all mass on either 0 or 1 it means that there are solutions which take on the value 0 as well as solutions which take on the value 1 for this variable. So in this case choose any value for this variable and reduce the formula, by eliminating this variable and all clauses which are now satisfied. This reduction is called the *decimation* step. If the marginal puts all its mass on 0, then pick the value 0, and if it puts all its mass on 1 then choose 1. Again, decimate. It is clear that this procedure will succeed in finding a satisfiable formula if one exists.

The above description assumed that message-passing is capable of exactly computing the marginals. Since this might not be the case we proceed slightly differently. Compute the marginals of all variables. Then pick a variable with maximal bias and decimate according to this bias. The hope is that by picking

variable with maximal bias we minimize the chance of making a mistake. This will be true as long as the message-passing algorithm predicts the marginals with reasonable accuracy. The above idea is what is used in *BP-guided* decimation. We will talk in more detail about this algorithm in the next chapter. Unfortunately, currently there does not exist a rigorous analysis for this algorithm. We consider a simpler algorithm in this chapter and show how to analyse it rigorously.

As we mentiond before, the K-SAT problem is the most difficult of our three running examples. Even very basic questions, like the *existence* of a SAT/UNSAT threshold, are currently not settled rigorously. We therefore will not be able to give a complete "solution" to this problem. The literature on this problem splits into two categories. On the one hand there are rigorous results typically concerning lower and upper bounds on the threshold, thresholds for some simple algorithms, as well as some basic structural properties of the problem. On the other hand, there are statistical physics calculations which make much more precise predictions and suggests sophisticated algorithms but which are not rigorous.

The aim of the current chapter is to introduce and rigorously analyse a very simple algorithm, called the *unit-clause propagation* algorithm. This algorithm has a somewhat mediocre performance, i.e., the threshold up to which it works is much below the actual SAT/UNSAT threshold as predicted by statistical physics. But it is relatively easy to analyze and it will give us the excuse of introducing a very powerful general machinery of analyzing such types of processes, called the *Wormald* method. In the next chapter we will then introduce a much more powerful message-passing algorithm based on belief-propagation and decimation. This algorithm has significantly better performance but currently no rigorous analysis exists.

Before we start with our analysis we give a quick tour of what is known about the problem. Readers, who are mostly interested in techniques, and not so much in the problem itself, can skip the next section.

9.1 A Brief Overview

As we mentioned earlier, satisfiability was the first problem proved to be NP-complete. Practically speaking, this means that there is no known algorithm which can efficiently decide for all SAT formulas if a satisfiable assignment exists or not and it is doubtful that such an algorithm can be found. Here, by efficient algorithm we mean an algorithm whose running time is polynomial in the number of Boolean variables.

The preceding paragraph concerns the *worst case*, i.e., algorithms that must succeed *always*. An alternative approach is to look at suitably defined *random* instances and to ask that the algorithm succeeds with high probability. For instance, suppose we construct a K-SAT formula by choosing each of the clauses uniformly at random from the set of all possible K-clauses. Hence, rather than considering deviously designed opponents (formulas), we are given an *ensemble*

of formulas, i.e., a set of formulas endowed with a probabilistic structure. We can now ask how hard it is to decide for a *typical* formula. In the following, we introduce the most famous of such probabilistic ensembles, namely the K-SAT ensemble.

Consider N Boolean variables and $M = \lfloor \alpha N \rfloor$ clauses of length K. The number α is positive and real and is called the *clause density*. To choose an instance from the K-SAT ensemble, we proceed as follows. Each of the M clauses picks uniformly at random a subset of length K of the variables and flips a fair coin to decide whether or not to negate each variable. Each of the above steps are taken independently of each other. The above procedure puts a uniform distribution on the set of all K-SAT formulas. In the following, we use $\mathrm{SAT}(N, K, \alpha)$ to denote the ensemble of random K-SAT formulas with size N and density α.

Due to its simple probabilistic structure and the importance of the satisfiability problem, the K-SAT ensemble has become a central topic of collaborations between computer scientists, mathematicians and statistical physicists. As we will see later, random K-SAT formulas enjoy a number of intriguing properties, some of which have been proven rigorously, but many of which are still awaiting a mathematical proof.

Most of the ideas and intuitions about this ensemble have been extended to other constraint satisfaction problems such as graph coloring (COL). One can argue whether or not random ensembles are good models for the highly structured SAT formulas which one finds in engineering and in the "real world." However, it is worth mentioning that random K-SAT instances are computationally hard for a certain range of densities, and this makes them a popular benchmark for testing and tuning SAT algorithms. In fact, some of the better practical ideas in use today come from insights gained by studying the performance of algorithms on random K-SAT instances [**?**].

We proceed by a brief detour of the current state of the art for the K-SAT problem.

The Threshold Conjecture

Pick a random formula from the K-SAT ensemble. What is the probability that such a formula is satisfiable? A moment of thought shows that this probability is a non-increasing function of α. Also, for small α we expect that most of the formulas are satisfiable whereas for α tending to infinity we expect most of the formulas to be un-satisfiable. What more can we say? In particular, what happens when the size of these formulas grows unbounded, i.e., when $N \to \infty$? Numerical experiments, physical arguments (as we will see later) as well as the experience from simpler constraint satisfaction problems sugggest that when the density crosses a critical threshold, these formulas undergo a *phase transition*. More precisely, as we increase α from zero to infinity the probability transitions from being almost certainly satisfiable to almost certain unsatisfiable and it does so in a jump at one *critical* value of α. Despite all evidence and effort, the conjecture

in this strong form is yet unproved for $K \geq 3$, and hence has remained as a conjecture known as the *satisfiability conjecture*.

Conjecture 9.1.1 (The Satisfiability Conjecture) For $K \geq 2$, there exists a constant $\alpha_s(K)$ such that the following holds

$$\lim_{N\to\infty} \Pr\{\text{SAT}(N, K, \alpha) \text{ is satisfiable}\} = \begin{cases} 1 & \text{if } \alpha < \alpha_s(K), \\ 0 & \text{if } \alpha > \alpha_s(K). \end{cases} \tag{9.1}$$

For $K = 2$, the satisfiability conjecture is known to be true and we have $\alpha_s(2) = 1$ [**?**]. The following theorem is the closest we know regarding the existence of such a threshold.

THEOREM 9.1 (Friedgut [**?**]) *For $K \geq 2$ there exists a sequence of numbers $\alpha_s(K, N)$ such that for all $\epsilon > 0$*

$$\lim_{N\to\infty} \mathbb{P}\{F(N, N(\alpha_s(N, K) - \epsilon)) \textit{ is SAT}\} = 1,$$
$$\lim_{N\to\infty} \mathbb{P}\{F(N, N(\alpha_s(N, K) + \epsilon)) \textit{ is SAT}\} = 0.$$

Theorem 9.1 comes very close to proving the satisfiability conjecture except that the sequence $\alpha_s(K, N)$ is not known to converge to a well-defined limit. In particular, there remains the possibility that such a sequence oscillates in a small window and hence may not converge. From now on, we let $\alpha_s(K)$ denote both the satisfiability threshold from Conjecture 9.1.1 and also the threshold sequence of Theorem 9.1, and leave the corresponding interpretation to the interest of the reader.

The consequences of Theorem 9.1 are not confined merely to the satisfiability conjecture. Another main application of this theorem is in providing bounds on $\alpha_s(K)$ in the following way. Suppose there exists a method that proves for some density $\alpha_{\text{method}}(K)$,

$$\lim_{N\to\infty} \Pr\{\text{SAT}(N, K, \alpha_{\text{method}}(K)) \text{ is satisfiable}\} \geq C, \tag{9.2}$$

where C is a positive constant. Then, from Theorem 9.1 we conclude that for any $\alpha \leq \alpha_{\text{method}}(K)$ we have

$$\lim_{N\to\infty} \Pr\{\text{SAT}(N, K, \alpha) \text{ is satisfiable}\} = 1.$$

In particular, this would show that $\alpha_s(K) \geq \alpha_{\text{method}}(K)$. Similarly, if $\alpha_{\text{method}}(K)$ is such that the inequality (9.2) holds in the opposite direction, then the probability that a random formula is satisfiable at densities above $\alpha_{\text{method}}(K)$ tends to 0 and we obtain that $\alpha_s(K) \leq \alpha_{\text{method}}(K)$.

This consequence of Theorem 9.1 has been the main venue for providing lower bounds on $\alpha_s(K)$. We now proceed by reviewing various methods and bounds on the threshold.

Various Bounds and the Asymptotic Behavior of the Threshold

Let us begin by a simple, but important, upper bound. For a random K-SAT formula F we denote by $X(F)$ its number of satisfying assignments (if $X(F)$ is zero then the formula is un-satisfiable). It is an easy exercise to show that

$$\mathbb{E}[X] = 2^N(1 - \frac{1}{2^K})^M.$$

As a result, by noticing $M = N\alpha$, if we choose

$$\alpha > \frac{-\ln 2}{\ln(1 - \frac{1}{2^K})},$$

then the value of $\mathbb{E}[X]$ is exponentially small in N and hence by an application of the Markov inequality we deduce that the probability of satisfiability is exponentially small. We thus have

$$\alpha_s(K) \leq \frac{-\ln 2}{\ln(1 - \frac{1}{2^K})} \leq 2^K \ln 2 - \frac{\ln 2}{2} - O(2^{-K}). \tag{9.3}$$

The above method, which is based on the first moment of X is called the *first moment* method. In fact, this simple upper bound can be made slightly sharper [**?**, **?**]

$$\alpha_s(K) \leq 2^K \ln 2 - \frac{1 + \ln 2}{2} - o(1), \tag{9.4}$$

where the $o(1)$ term is asymptotic in K. To obtain a lower bound, the *second moment* can be used [**?**, **?**]. The idea is that by an application of the Cauchy-Schwarz inequality we can show that

$$\Pr(X > 0) \geq \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}. \tag{9.5}$$

Now, if we find densities α for which the value $\frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}$ is bounded by a constant, it is immediate that such a value of α is a lower bound for $\alpha_s(K)$. However, on the negative side, for the choice of $X = X(F)$ to be the number of solutions, it can be shown that for any value of α, the quantity $\frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}$ decays to 0 by N. In other words, the number of solutions does not concentrate around its average. On the positive side, one can choose other candidates for X, rather than the number of solutions, to plug into (9.5). For instance, instead of giving an equal weight to all solutions of a formula F (as done in counting the number of solution), one can assign different weights to different solutions. This is called the *weighted second order method*. Using this method, it can be shown [**?**] that

$$\alpha_s(K) \geq 2^K \ln 2 - (K+1)\frac{\ln 2}{2} - 1 - o(1). \tag{9.6}$$

Very recently, by a new version of the weighted second order method, a new lower bound has been obtained in [**?**]

$$\alpha_s(K) \geq 2^K \ln 2 - \frac{3\ln 2}{2} - o(1). \tag{9.7}$$

K	3	4	5	7	10
Upper bound from (9.3)	5.19	10.74	21.83	88.37	709.44
Best upper bound [**?**]	4.51	10.23	21.33	87.88	708.94
Lower bound from [**?**]	2.68	7.91	18.79	84.82	704.94
Best algorithmic bound	3.52	5.54	9.63	33.23	172.65

Table 9.1 Best known rigorous bounds for the location of the satisfiability threshold $\alpha_s(K)$ for some small values of K. The last row gives the largest density for which a polynomial-time algorithm has been proven to find satisfying assignments.

To summarize: for large K we have

$$2^K \ln 2 - \frac{3 \ln 2}{2} - o(1) \leq \alpha_s(K) \leq 2^K \ln 2 - \frac{1 + \ln 2}{2} - o(1), \tag{9.8}$$

where the $o(1)$ term is asymptotic in K. These bounds indicate that for large values of K, the value of $\alpha_s(K)$ is just a small constant away from $2^K \ln 2$. For smaller values of K, the bounds derived from these methods are given in Table 9.1.

A different venue to find lower bounds is to provide algorithms capable of solving a random formula with a positive probability. We will have more to say about these algorithms and the methods used to analyze them later. In a nutshell, most of these algorithms act in the following way. Given a random formula, they set the variables one at a time using heuristics that use very little, and completely local, information about the variable-clause interactions. Of course, such a confinement is also what enables their analysis. Table 9.1 contains the best such algorithmic lower bounds from [**?**] and [**?**].

Outline

We will see later on that the "real" 3-SAT threshold is around $\alpha = 4.26$. This threshold is currently not provable but only "computable" by statistical physics calculations. If we use BP-guided decimation, we will find an algorithmic threshold of $\alpha_{\mathrm{BP}} = 3.86$. Even this threshold can currently only be asserted by large-scale simulations or by statistical physics calculations.

The aim of this lecture is to derive a lower bound which can be asserted rigorously. We will do so by analyzing a very simple algorithm, called unit-clause propagation (UCP). As we will see, it has a threshold of $\alpha = \frac{8}{3}$. This is not the best known lower bound. More sophisticated algorithms have been analyzed and yield a threshold of $\alpha = 3.52$. But these algorithms are considerably harder to analyze.

9.2 The Unit-Clause Propagation Algorithm

Let us now come back to the main object of study for the current chapter. We will introduce and anlyse a simple algorithm to solve K-SAT formulas. The algorithm does not have record-shattering performance. But it is natural can can be analysed by a standard and important method, called the *Wormald* method.

The Unit Clause propagation algorithm, or UC for short, is a (randomized) algorithm which sets one variable at a time. Compared to the DPLL algorithm, the UC algorithm never backtracks. Once a variable is fixed, the value stays fixed and is never changed. In brief, the algorithm works as follows: Represent a K-SAT formula in the usual way by a bipartite graph G consisting of N literals, or variable nodes, and $M = N\alpha$ clauses, or check nodes. The algorithm starts with G and in each step removes some nodes from the graph. In more detail, the UC algorithm consists of two main steps:

- *Free step:* If there does not exist a check node (clause) in the graph of degree one we perform a *free* step: Choose a variable uniformly at random and set its value uniformly at random to either 0 or 1. Remove the chosen variable node as well as any check node corresponding to a clause which is now fulfilled through the choice of the value, as well as all edges emanating from any of the removed nodes.
- *Forced Step:* If there exists a check node (clause) of degree one we perform a *forced* step: Choose a check node of degree one uniformly at random from all such check nodes. Set the value of the adjacent variable node to the unique value which fulfills the clause (hence the name "forced"). Remove from the graph the check node, the variable, all further check nodes which in addition might now be fulfilled, as well as all edges emanating from any of the removed nodes.

It is easy to see that the UC algorithm fails in finding a solution if only it generates a clause of degree 0 at some point in the course of the algorithm. In fact, once a 0-clause appears, the algorithm can halt and return a message "unable to find a solution."

The progress of UC algorithm can be predicted in terms of the solution of a set of differential equations. This method, called the Wormald method, is broadly applicable. Therefore, in the next section we describe this technique in general, before coming back to the analysis of the UC algorithm in the subsequent section.

9.3 The Wormald Method

A Simple Example

Let us start with a very simple example to illustrate the idea. Consider N particles in a box of volume V. Assume that time is discrete and takes integer values.

Assume that at each time instant and for each pair of particles (i, j) present, the probability that these two particles annihilate each other is equal to

$$\frac{1}{V^2} = \frac{N^2}{V^2}\frac{1}{N^2} = \frac{\rho^2}{N^2},$$

where ρ is the initial density of particles. Let $N(t)$ denote the number of particles which are left at time t, with $N(0) = N$. How will the number of particles evolve? We have the relationship

$$N(t+1) = N(t) - 2\sum_{(i,j)} \mathbb{1}_{\{(i,j) \text{ is annihilated between } t \text{ and } t+1\}}.$$

The evolution of this process is of course stochastic, but it is easy to write down the expected progress in one time step given the current state. We have

$$\begin{aligned}\mathbb{E}[N(t+1) \mid N(t)] &= N(t) - 2\frac{N(t)(N(t)-1))}{2}\frac{\rho^2}{N^2} \\ &= N(t) - \rho^2\frac{N(t)(N(t)-1))}{N^2}.\end{aligned}$$

This means that

$$\mathbb{E}[N(t+1) - N(t) \mid N(t)] = -\rho^2\frac{N(t)(N(t)-1))}{N^2}.$$

Assume that the process evolves exactly according to its expected progress. This means that we drop the expectations and the conditioning. This gives us

$$N(t+1) - N(t) = -\rho^2\frac{N(t)(N(t)-1))}{N^2} \approx -\rho^2\frac{N(t)^2}{N^2}.$$

Now set $t = \tau N$, where $\tau \in \mathbb{R}^+$ so that $N(t) = N(\tau N)$. Further, scale $N(t)$ by the initial number of particles, i.e., write $N(N\tau) = Nn(\tau)$. We can then write

$$Nn(\tau + 1/N) - Nn(\tau) \approx -\rho^2\frac{n(\tau)^2}{n(0)^2}.$$

With $N = \frac{1}{d\tau}$ this leads us the consider the differential equation

$$\frac{dn(\tau)}{d\tau} = -\rho^2 n(\tau)^2,\ n(0) = 1.$$

This differential equation has the solution

$$n(\tau) = \frac{1}{\rho^2(\tau + \frac{1}{\rho^2})},$$

which is best seen by direct verification. If we go back to $N(t)$ then we see that according to this model we have

$$N(t) = \frac{1}{\frac{t}{V^2} + \frac{1}{N}}.$$

In the above derivation we have waved our hands like a drunken sailor. In particular, we have replace what by its very nature was a stochstic process by a

deterministic description. Clearly, this cannot be strictly correct. But one might hope that the behavior of specific instances of $N(t)$ are "close" to this deterministic solution. Indeed, this is correct, as we will see in the next section.

The Wormald Theorem

There are myriads of versions of increasing sophistication. We will be content with stating and applying one particular incarnation. In the computer science literature the basic approach is typically referred to as the *Wormald* method. In the economics literature it is sometimes called the *Kurtz* method. Although perhaps phrased less formally, the physics community has applied this techniques for an even longer time.

THEOREM 9.2 (Wormald) *Let $Y_i^{(n)}(t)$ be a sequence (indexed by n) of real valued random processes, $1 \le i \le k$, where k is fixed, so that for all $1 \le i \le k$, all $0 \le tm(n)$, and all $n \in \mathbb{N}$*

$$|Y_i^{(n)}(t)| \le Bn, \text{ for some constant } B.$$

- *Let $H(t)$ denote the history up to time t, i.e., $H(t) = \{\underline{Y}^{(n)}(0), \ldots, \underline{Y}^{(n)}(t)\}$.*
- *Let $I = \{(y_1, \ldots, y_k) : \mathbb{P}\{\underline{Y}^{(n)}(0) = (y_1 n, \ldots y_k n)\} > 0, \text{for some } n\}$.*
- *Let D be some open connected bounded set containing the closure of $\{(0, y_1, \cdots, y_k) : (y_1, \cdots, y_k) \in I\}$.*
- *Let $f_i : \mathbb{R}^{k+1} \to \mathbb{R}$, $1 \le i \le k$:*
  1. *(Trend) For all i and uniformly for all $t < m$*

$$\mathbb{E}[Y_i(t+1) - Y_i(t) \mid H(t)] = f_i(\frac{t}{n}, \frac{Y_1^{(n)}(t)}{n}, \cdots, \frac{Y_k^{(n)}(t)}{n}) + o(1).$$

  2. *(Tail) For all i and uniformly for all $t < m$*

$$\Pr(|Y_i^{(n)}(t+1) - Y_i^{(n)}(t)| > n^{\frac{1}{5}} \mid H(t)) = o(n^{-3}).$$

  3. *(Lipschitz) For each i, the function f_i is a Lipschitz continuous on D. That is, there exists a constant L such that for any pair $x, y \in D$,*

$$|f_i(x) - f_i(y)| \le L||x - y||_1 = L \sum_{i=1}^{k} |x_i - y_i|.$$

Then we have:

(a)(Differential equation)For $(0, \hat{z}_0, \cdots, \hat{z}_k) \in D$ the system of differential equations

$$\frac{dz_i}{d\tau} = f_i(\tau, z_1, \cdots, z_k), \;\; 1 \le i \le k,$$

has a unique solution in D for $z_i : \mathbb{R} \to \mathbb{R}$ passing through $z_i(0) = \hat{z}_i$, $1 \le i \le k$, and which extends to points arbitrarily close to the boundary of D.

(b)(Concentration) Almost surely

$$Y_i^{(n)}(t) = z_i(\frac{t}{n})n + o(n),$$

uniformly for $0 \leq t \leq \min\{m(n), n\tau_{max}\}$ *and for each* i*, where* $z_i(\tau)$ *is the solution in (a) with* $\hat{z}_i(0) = \frac{Y_i^{(n)}(0)}{n}$ *and where* τ_{max} *is the maximum time until the solution can be extended before reaching in* L_1*-distance* ϵ*-close to the boundary of* Dm *where* ϵ *is arbitrary but strictly positive.*

9.4 Analysis of the UC Algorithm

Let us begin by introducing the necessary notation for the analysis:

- We let t denote current "time" of the algorithm. The term "time" means the total number of variables fixed so far.
- We let $C_i(t)$, $i \in \{1, \cdots, K\}$, denote the the number of clauses of degree i that the remaining formula at time t contains.

One important fact for the analysis of such algorithms is the so called *uniform randomness property*. In brief, this property means that at any time t, each clause of length i in the remaining formula is uniformly distributed among all the possible clauses of length i. In other words, conditioned on the number of variables and clauses of different length, the formula is uniformly random. An intuitive justification for the randomness property in our case stems from the fact that at any step (free or forced) in the UC algorithm, no information, what-so-ever, can be deduced about the structure of the remaining formula. The exact proof of the uniform randomness property in our case can be easily deduced from [**?**, Lemma 3].

We are now ready to write the set of differential equations for C_i's. Let us for simplicity assume $K = 3$ and bear in mind that for general K the analysis follows along the same path. Recall that we start with N Boolean variables. In the process we consider, at each step in the process we remove exactly one variable node. Let time t be discrete and increasing, starting at $t = 0$. Let $N(t)$ be the number of variables which are left at time t. Then we have $N(t) = N - t$.

We start with $C_3(t)$. At any time t, a variable is chosen among the $N - t$ remaining ones and is given a permanent value. This variable can either be chosen due to a forced step or due to a free step. Note that in both cases the degree distribution of the chosen variable is essentially the same. In more detail. At time t there are $N - t$ variables left and $C_1(t) + 2C_2(t) + 3C_3(t)$ edges left. Further, each edge is connected uniformly at random to each variable node. So the distribution of the number of edges for a randomly chosen variable node is equal to $C_1(t) + 2C_2(t) + 3C_3(t)$ independent Bernoulli trials with success probability $1/(N - t)$. In particular, in expectation, a randomly chosen variable node has $\frac{C_1(t)+2C_2(t)+3C_3(t)}{N-t}$ edges connected to it. Even more, in a forced step,

when we consider a random variable which is connected to a clause of degree 1, the expected number of *additional edges* is $\frac{C_1(t)+2C_2(t)+3C_3(t)-1}{N-t}$, which for large N is essentially the same number. For this reason we can treat both cases, namely free and forced step in the same way.

Now consider what happens when we fix the value of the variable. This variable is connecected in expectation to

$$\frac{C_1(t)+2C_2(t)+3C_3(t)-1}{N-t}\frac{3C_3(t)}{C_1(t)+2C_2(t)+3C_3(t)-1}=\frac{3C_3(t)}{N-t}.$$

clauses of degree 3. Therefore

$$\mathbb{E}[C_3(t+1)-C_3(t)|C_3(t)]=-\frac{3C_3(t)}{N-t}. \tag{9.9}$$

Among the 3-clauses that contain the chosen variable, half of them (in expectation) are satisfied and hence removed from the formula and the other half are shortened to two 2-clauses. We claim that

$$\mathbb{E}[C_2(t+1)-C_2(t)|C_3(t),C_2(t)]=\frac{3C_3(t)}{2(N-t)}-\frac{2C_2(t)}{(N-t)}. \tag{9.10}$$

We have already seen where the first term on the right comes from. The second term has a similar interpretation. Each variable node is connected in expectation to

$$\frac{C_1(t)+2C_2(t)+3C_3(t)-1}{N-t}\frac{2C_2(t)}{C_1(t)+2C_2(t)+3C_3(t)-1}=\frac{2C_2(t)}{N-t}$$

clauses of degree 2. In expectation, half of them will be fulfilled through the choice of the value of the variable node, and the other half will become 1-clauses.

Finally, look at the evolution of degree-1 clauses. We claim that we have

$$\mathbb{E}[C_1(t+1)-C_1(t)|C_3(t),C_2(t),C_1(t)]=\frac{C_2(t)}{N-t}-\mathbb{1}_{\{C_1(t)>1\}}. \tag{9.11}$$

This equation is somewhat more subtle. If at time t there are degree-1 clauses then we will eliminate one for sure. In this case we will also add in expectation $\frac{C_2(t)}{N-t}$ new ones. If on the other hand we do not have a degree-1 node then we only add in expectation $\frac{C_2(t)}{N-t}$ such clauses.

Note that in order to predict the evolution of $C_3(t)$ and $C_2(t)$ we only need to know $(C_3(t),C_2(t))$ but not $C_1(t)$. Therefore, let us just solve the differential equation for these two higher degrees.

At this step we need to check that all the conditions of the Wormald theorem are fulfilled. We leave this task to the reader. Most conditions are trivially fulfilled. E.g., the process starts in a bounded state and all quantities decrease and stay non-negative. Hence the process is trivially bounded. Also the initial condition is determinstic. Further, steps are small with high probability, so the tail condition is also easy to check. Further, the trend condition is also trivially fulfilled. The only condition which needs checking is that the function which gives the trend is Lipschitz. A quick check shows that this is true until almost

towards the end of the algorithm. At the very end, the denominator $1-\tau$ tends to zero, which causes problems. So according to the Wormald theorem, actual instances will behave close to the prediction given by the solution of the differential equation up to any fixed time strictly bounded away from $\tau = 1$.

As a next step let us write down the differential equations corresponding to this evolution. We get, using $\tau \equiv \frac{t}{N}, c_2(\tau) \equiv \frac{C_2(t)}{N}, c_3(\tau) \equiv \frac{C_3(t)}{N}$,

$$\frac{\mathrm{d}c_3(\tau)}{\mathrm{d}\tau} = -3\frac{c_3(\tau)}{1-\tau}, \tag{9.12}$$

$$\frac{\mathrm{d}c_2(\tau)}{\mathrm{d}\tau} = \frac{3}{2}\frac{c_3(\tau)}{1-\tau} - 2\frac{c_2(\tau)}{1-\tau}, \tag{9.13}$$

with initial conditions

$$c_3(0) = \alpha, \tag{9.14}$$

$$c_2(0) = 0. \tag{9.15}$$

The solution to the above set of equations can easily be found to be

$$c_3(\tau) = \alpha(1-\tau)^3,$$
$$c_2(\tau) = \frac{3}{2}\alpha\tau(1-\tau)^2.$$

Figure **??** compares the solutions of the differential equations with their counterpart in performing the UC algorithm over an actual random K-SAT formula.

Now let us see what this differential equation tell us about the threshold of this algorithm. We claim that the threshold is $\alpha^* = \frac{8}{3}$. Let us first show that it is at most $\frac{8}{3}$. Assume that we are operating with a higher value of α. Note that

$$\frac{C_2(t)}{N-t} = \frac{3}{2}\alpha\frac{t}{N}(1-\frac{t}{N}) + o(1), \tag{9.16}$$

Note that at time $t = \frac{1}{2}$ we have according to this prediction $\left.\frac{C_2(t)}{N-t}\right|_{t=\frac{N}{2}} = \frac{3}{8}\alpha + o(1)$. But note that $\frac{C_2(t)}{N-t}$ is the density of 2-clauses at time t. In other words, if we choose α greater than $\frac{8}{3}$ then at this point in time the density of 2-clauses is above 1. Using the uniform randomness property we see that what we would have at this point is a random 2-SAT formula with density larger than 1 with some additional 3-clauses. But such a formula is unsatisfiable with high probability. So in particular the UCP cannot possible succeed.

Now let us prove that if we pick α strictly smaller than $\frac{8}{3}$ then with high probability the algorithm succeeds. Recall that the algorithm succeeds if and only if no degree-0 clause is prodcued at any point in time. Consider the equation for the evolution of the degree-1 clauses. Note that $\frac{2C_2(t)}{N-t}$ is not only the 2-clause density, but it is also the expected number of new degree-1 clauses which are generated at time t. If this number is strictly less than 1 over the whole time interval then with high probability $C_1(t)$ is at any point in time at most a constant but never becomes linear in N. This means that the chance that when

we set a variable that this variable is connected to two such degree-1 clauses of opposite sign vanishes. Hence, we never create a degree-0 clause.

Some care has to be taken to make this argument completely rigorous. In particular, as we discussed we can only guarantee the accuracy of the prediction up to a time very close to $\tau = 1$. So we need in addition an argument which guarantees that the remaining formula is satisfiable with high probability. This is somewhat analogous to decoding, where we sometimes need an argument which guarantees that we can decode all bits assuming that we have decoded most of them. The argument for the present case goes as follows. If we look at the solution of the differential equation, we see that if we run the algorithm long enough for $\alpha < \frac{8}{3}$ then there is a time strictly before $\tau = 1$ where the sum of the 2-density plus the 3-density is strictly less than 1. We can now argue as follows. Drop a random variable from each 3-clause. Then the resulting formula is satisfiable with high probability.

9.5 K-SAT: BP-Guided Decimation

In the preceding sections of this chapter we have introduced and analysed a very simple algorithm, called unit clause propagation. This analysis established a non-trivial lower bound for the SAT/UNSAT threshold and this threshold is in particular algorithmic, i.e., we have an efficient algorithm which works up to this threshold. On the downside, the UCP algorithm is not very powerful and so the threshold is quite low.

The aim of the current chapter is to introduce and to discuss a more powerful algorithm, called BP-guided decimation. The basic idea is similar to what that of the UCP algorithm. At each step we pick a variable node (or several of them) and fix its value, i.e., we decimate the formula. The difference lies in how we choose the variable we decimate. In the UCP algorithm, the choice was either forced upon us or we chose randomly. In the BP-guided decimation algorithm we use the BP algorithm to guide the selection.

We first introduce a version of the algorithm which is guaranteed to succeed if the formula is satisfiable and if the factor graph corresponding to the formula is a tree. As we did for coding, we then introduce a more convenient parametrization of the messages. Finally, we apply the algorithm to formulas in the ensemble, even though of course in this case the factor graphs are far from trees. As we discussed previously, the K-SAT problem is considerably harder than either coding or compressive sensing. Many of the basic questions are still open from a mathematical point of view. E.g., currently there exists no rigorous analysis of BP-guided decimation. We will therefore have to be content with a somewhat more heuristic approach.

Simple Example

Let us start with a very simple example. Suppose we are given the formula

$$F = x_1 \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3). \tag{9.17}$$

The corresponding factor graph is shown in Figure 9.1. Dashed lines means the variable appears negated in the corresponding clause.

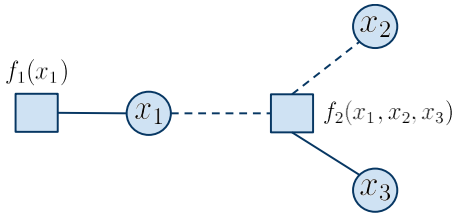


Figure 9.1 Factor graph of the equation $F = x_1 \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$

F is a Boolean function. However, we can slightly modify F and model it as a binary function that can take either of 0 or 1 values. In this case, we can write F as the product of two other binary functions: $f_1 = x_1$ and $f_2 = \text{sign}(\overline{x_1}+\overline{x_2}+x_3)$, where sign is the normal sign function with $\text{sign}(0) = 0$.

Note that in order to see if F is satisfiable, we can compute the "partition function"

$$\sum_{x_1,x_2,x_3} f_1(x_1) f_2(x_1, x_2, x_3).$$

This is true since the partition function counts the number of satisfying configurations. Hence, if the partition function is non-zero then the formula is satisfiable. For the current case there are 3 SAT solutions. Table 9.2 illustrate the satisfiability of F for all possible combination of x_1, x_2 and x_3.

We can also look at marginals with respect to different variables, for instance

$$\sum_{\sim x_1} f_1(x_1) f_2(x_1, x_2, x_3).$$

This marginal counts the number of satisfying clauses given that x_1 has a particular fixed value. From Table 9.3 we see that $\mu(x_1 = 0) = 0$ and $\mu(x_1 = 1) = 3$; $\mu(x_2 = 0) = 2$ and $\mu(x_2 = 1) = 1$; $\mu(x_3 = 0) = 1$ and $\mu(x_3 = 1) = 2$. Note that the factor graph is a tree. Therefore BP can compute the partition function as well as the marginals *exaclty*. Table 9.3 summarizes the messages exchanged in each iteration of the belief propagation in order to compute the marginal with respect to x_1, denoted by $\mu(x_1)$, for the factor graph given in Fig. 9.1. Let us illustrate the use of message passing rules for the derivation of $\mu(x_1)$. The first

x_1	x_2	x_3	$f_1(x_1)$	$f_1(x_1,x_2,x_3)$	F
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	1	1	1

Table 9.2 Satisfiability of F, given by equation (9.17), for all possible combination of x_1, x_2 and x_3.

Iteration number	Messages
1	$\mu_{f_1\to x_1} = f_1,\ \mu_{x_2\to f_2} = 1,\ \mu_{x_2\to f_2} = 1$
2	$\mu_{f_2\to x_1} = \sum_{x_2,x_3} f_2(x_1,x_2,x_3)$

Table 9.3 Messages exchanged in each iteration of the belief propagation performed over the factor graph given in Fig. 9.1.

line of the table gives the initial messages at the leaf nodes. First we compute

$$\mu_{2\to 1} = \sum_{\sim x_1} f_2(x_1,x_2,x_3).\underbrace{\mu_{2\to 2}(x_2)}_{1}\underbrace{\mu_{3\to 2}(x_3)}_{1} = 4 \text{ if } x_1 = 0 \text{ and } 3 \text{ if } x_1 = 1 \tag{9.18}$$

and

$$\mu_{1\to 1} = f(x_1) = 0 \text{ if } x_1 = 0 \text{ and } 1 \text{ if } x_1 = 1 \tag{9.19}$$

Finally,

$$\mu(x_1) = \mu_{2\to 1}(x_1)\mu_{1\to 1}(x_1) = 0 \text{ if } x_1 = 0 \text{ and } 3 \text{ if } x_1 = 1 \tag{9.20}$$

which agrees with Table 9.3.

Let us summarize. The marginals count the number of satisfying solutions with a particular assignment for the given Boolean variable. If we are interested in the *fraction* of satisfying solutions with a particular assignment for the given Boolean variable we can just normalize the messages. Also, we will see shortly, that if we can accurately compute marginals, we can also find SAT assignments.

Notation: We denote clauses by $a, b, c, \ldots$ and variables by $i, j, k, \ldots$. Furthermore, we denote the neighborhood of a node x by ∂x. The same neighborhood excluding a particular node y is indicated by $\partial x \setminus y$.

Having these notations in mind, we start by modifying the message-passing

Figure 9.2 BP Guided Decimation over Trees

1. Run belief propagation on F and compute the all marginals $\mu(x_i)$ for all of the variables.
2. Pick a variable i. If $\mu(x_i = 0) > 0$ (there exists an assignement with $x_i = 0$), then:
   1Set $x_i = 0$ in all clauses.
   2Eliminate all those clauses that x_i appears negated in them.
   3Remove x_i from the other clause.
   If on the other hand $\mu(x_i = 0) = 0$ (there doesnt exist an assignmenet with $x_i = 0$), then:
   1Set $x_i = 1$ in all clauses.
   2Eliminate all those clauses that x_i appears unnegated in them.
   3Remove x_i from the other clause.
3. Repeat the process until no variables are left.

rules. In the original message passing scheme, the message from variable i to clause a is given by equation (9.21).

$$\mu_{i\to a}(x_i) = \prod_{b\in\partial i\setminus a} \mu_{b\to i}(x_i) \tag{9.21}$$

However, since we are interested in the *fraction* of the solutions with $x_i = 0$ and $x_i = 1$, we require the new messages $\widetilde{\mu}_{i\to a}(x_i)$ to satisfy the following equation.

$$\widetilde{\mu}_{i\to a}(x_i = 0) + \widetilde{\mu}_{i\to a}(x_i = 1) = 1$$

Therefore, it is sufficient to set $\widetilde{\mu}_{i\to a}(x_i)$ according to equation (9.22).

$$\widetilde{\mu}_{i\to a}(x_i) = \frac{\mu_{i\to a}(x_i)}{\mu_{i\to a}(x_i = 0) + \mu_{i\to a}(x_i = 1)} \tag{9.22}$$

At this point, it seems as if we have to once calculate $\mu_{i\to a}(x_i)$ for $x_i = 0, 1$ and then normalize the messages. However, it is easy to show that we can directly calculate $\widetilde{\mu}_{i\to a}(x_i)$. To simplify the notations, we omit the normalization factor and write the messages as

$$\widetilde{\mu}_{i\to a}(x_i) \propto \prod_{b\in\partial i\setminus a} \widetilde{\mu}_{b\to i}(x_i). \tag{9.23}$$

9.6 From Counting the Number of Solutions to Finding a Solution

Given a SAT problem F, assume that the factor graph of F is a tree and F has a satisfying solution. Then algorithm 1 will find a solution that satisfies F.

Note that in each step of the above algorithm we must run BP. So in total we might need to run BP n times.

Terminology: Since we use belief propagation and eliminate a variable in each iteration, the algorithm is called **BP-guided decimation**.

Algorithm 1 is only guaranteed to give accurate marginals if we have a tree.

Figure 9.3 BP Guided Decimation over General Graphs
1. Run BP and calculate all marginals.
2. Pick a node i such that $|\mu(x_i = 0) - \mu(x_i = 1)|$ is maximized.
3. Set x_i to the most likely value, i.e. $x_i = 0$ if $\mu(x_i = 0) > \mu(x_i = 1)$ and to 1 otherwise.
4. Eliminate all clauses that the particular choice of x_i make them satisfied. Remove x_i from the other clause.
5. Recurse until all variables are eliminated.

But what about the more general cases? We will introduce a modified version of the above algorithm in the next section to deal with general factor graphs.

Applying BP Guided Decimation to General Factor Graphs

In this section, we apply a modified version of the BP guided decimation algorithm to general factor graphs. However, note that the graph in this section should be sparse as before.

Over a tree, the previous algorithm yields exact marginals and we can pick anyone of them in each iteration. However, in general graphs it is not the case any more. As a result and in order to deal with the inherent uncertainty in marginals, in each iteration we pick a node i such that the difference $|\mu(x_i = 0) - \mu(x_i = 1)|$ is **maximized**. This way, we hope that this node has such a clear bias that its marginals are are quite exact despite the graph not being a tree.

The rest of the algorithm is the same, summarized below:

Some remarks about running BP on general graphs are in order:

- *Initialization* The typical way of initializing messages is to set all of them equal to 1/2.
- *Scheduling* In contrast to BP guided decimation over a tree, the choice of node i affect the solution and the whole algorithm. Therefore, scheduling matters. We usually use flooding as a means of scheduling. In other words, in each iteration every node sends its messages over its outgoing links.

Figure 9.4 illustrates two kinds of probabilities as a function of α (ratio of nb of clauses to variables). One can run pure BP over many instances and compute the empirical probability that it converges. This yields the upper curves in figure 9.4. For $K = 3$ we get a convergence threshold $\alpha_{\mathrm{BP}\approx 3.86}$ and for $K = 4$ we get $\alpha_{\mathrm{BP}} \approx 10.3$. Now, one can run BP guided decimation (algorithm 2) over many instances and derive the empirical probability of success. The corresponding threshold must in general be lower than α_{BP} since BP must at least converge after each decimation step. This empirivcal probability is given by the lower curve in figure 9.4. For $K = 3$ the threshold is approximately identical to α_{BP} but for $K = 4$ it is smaller and approximately equal to 9.3.

The actual SAT-UNSAT threshold is for $K = 3$, $\alpha_{\mathrm{sat-unsat}} \approx 4.26$ and for

$K = 4$, $\alpha_{\text{sat}-\text{unsat}} \approx 9.93$. We will see in future lectures how to obtain these thresholds by *survey propagation* algorithms.

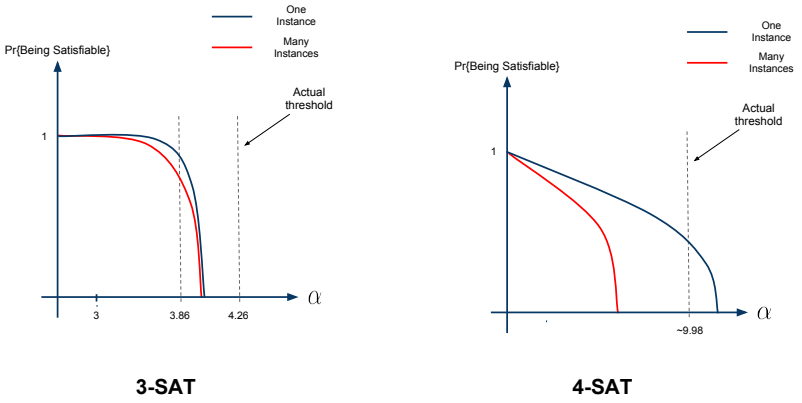


Figure 9.4 Probability of $3-SAT$ and $4-SAT$ being satisfied by BP guided decimation.

9.7 Convenient Re-parametrization

To write down the BP equations in simple form it is convenient to use the reformulation in terms of spin variables exposed in Chapter 3. Recall that a weight $J_{ia} = +1$ (resp. -1) is associated to full (resp. dashed) edges for which x_i appears un-negated (negated) in clause a. Recall also that $s_i = (-1)^{x_i}$. With these definitions $s_i = J_{ia}$ means that the assignment s_i does not satisfy a, and $s_i = -J_{ia}$ means that it satisfies a.

We parametrize the messages as follows

$$\mu_{i\to a}(s_i = \pm J_{ia}) = \frac{1 \mp \tanh h_{i\to a}}{2}, \quad \hat{\mu}_{a\to i}(s_i = \pm J_{ia}) = \frac{1 \mp \tanh \hat{h}_{i\to a}}{2}. \quad (9.24)$$

The interpretation of this notation is that $(1 - \tanh h_{i\to a})/2$ is the probability that x_i/s_i has a value which *does not satisfy* the clause corresponding to node a. Similarly, $(1 - \tanh \hat{h}_{i\to a})/2$ represents the probability that x_i/s_i is *not free* to be chosen arbitrarily since the clause a is not satisfied yet.

We need one more bit of notation. Consider a fixed edge ia with some edge weight J_{ia}. Let S_{ia} be the subset of variable nodes in ∂a that have the same edge type (weight) J_{ia}. Likewise, let U_{ia} be the subset of variable nodes in ∂a with a different edge type i.e., $-J_{ia}$.

The original message passing equations for messages from variable to check

nodes is given by:

$$\mu_{i\to a}(s_i = \pm J_{ia}) \propto \prod_{b\in\partial i\setminus a} \hat{\mu}_{b\to i}(s_i = \pm J_{ia}) \\ \propto \prod_{b\in S_{ia}} \hat{\mu}_{b\to i}(s_i = \pm J_{ib}) \prod_{b\in U_{ia}} \hat{\mu}_{b\to i}(s_i = \mp J_{ib}) \tag{9.25}$$

Hence,

$$\frac{1 \pm \tanh h_{i\to a}}{2} \propto \left(\prod_{b\in S_{ia}} \frac{1 \pm \tanh \hat{h}_{b\to i}}{2}\right)\left(\prod_{b\in U_{ia}} \frac{1 \mp \tanh \hat{h}_{b\to i}}{2}\right) \tag{9.26}$$

Taking the ratio of these two equations we find

$$h_{i\to a} = \sum_{b\in S_{ia}} \hat{h}_{b\to i} - \sum_{b\in U_{ia}} \hat{h}_{b\to i} \tag{9.27}$$

The original message passing rules for messages from constraint to variable nodes yield

$$\hat{\mu}_{a\to i}(s_i = \pm J_{ia}) \propto \sum_{\sim s_i = \pm J_{ia}} f_a(s_{\partial a}) \prod_{j\in\partial a\setminus i} \mu_{j\to i}(s_j) \tag{9.28}$$

As noted at the beginning of this section for $s_i = -J_{ia}$ the clause a is satisfied irrespective of other variables, i.e. $\psi(s_{\partial a}) = 1$. As a result, the sum of some products in (9.28) factorizes into

$$\hat{\mu}_{a\to i}(s_i = -J_{ia}) \propto \prod_{j\in\partial a\setminus i} \sum_{s_j} \mu_{j\to a}(s_j) \propto 1. \tag{9.29}$$

In other words $(1 + \tanh \hat{h}_{a\to i})/2 \propto 1$. Now we calculate (9.28) for $s_i = J_{ia}$. For this assignment the variable s_i can be eliminated from the kernel function f_a since this variable does not satisfy a. In (9.28) we have to sum over all assignments of remaining variables $j \in \{\partial a \setminus i\}$ such that at least one of them has value $s_j = -J_{ja}$. It is easy to see that this yields

$$\hat{\mu}_{a\to i}(s_i = J_{ia}) \propto 1 - \prod_{j\in\partial a\setminus i} \mu(s_j = J_{ja}). \tag{9.30}$$

Dividing out relations (9.28) and (**??**) allows to eliminate the normalization factors and one finds

$$\hat{h}_{a\to i} = -\frac{1}{2} \ln\left\{1 - \prod_{j\in\partial a\setminus i} \frac{1 - \tanh_{j\to a}}{2}\right\} \tag{9.31}$$

Equations (9.27)-(9.31) are the BP equations for K-SAT. The reader will appreciate the similarity with coding.

Problems

9.1 (Peferential Attachment). The purpose of this homework is to use the Wormald method to study a model for "preferential attachment." Consider n nodes. Initially all nodes have degree 0. Assume that we allow a maximum degree of $d_{\max}$. We proceed as follows. At every step pick two nodes from the set of all nodes which have degree at most $d_{\max}-1$. Rather than picking them with uniform probability pick them proportional to their current degree. More precisely, assume that at time t you have $D_i(t)$ nodes of degree i. Then pick a node of degree i with probability

$$\begin{cases} \frac{D_i(t)}{\sum_{j=0}^{d_{\max}-1} d_j(t)}, & 0 \le i < d_{\max}, \\ 0, & i = d_{\max}. \end{cases}$$

Initially, we have $D_0(t=0) = n$ and $D_i(t=0) = 0$ for $i = 1, \cdots, d_{\max}$. Note that at time $t = nd_{\max}/2$ all nodes will have maximum degree. Pick $d_{\max} = 4$.

(i). Write down the set of differential equations for this problem. Are the conditions fulfilled?

(ii). Plot the evolution of the degree distribution as a function of the normalized time for $\tau = t/n \in [0, d_{\max}/2]$

HINT: In general one cannot expect to solve the system of differential equations analytically. But it is typically easy to solve them numerically. Here is how you do it in Mathematica. The following lines set up the differential equation we discussed in class and plots the solution.

```
(* initial conditions      *)
cnds = {n[0] == 1};
(* set of diff equations   *)
eqns = {n'[u] == - rho n[u]^2};
(* put the two together    *)
eqnspluscnds = Flatten[Join[eqns, cnds]];
(* solve up to this point  *)
umax=10;
(* solve the diff equation *)
sol=Flatten[NDSolve[eqnspluscnds, {n}, {u, 0, umax} ]]
(* plot the solution       *)
Plot[Evaluate[{n[u]} /. sol], {u, 0.0, umax}]
```

If you have more than one variable then it is convenient to call them

```
d[0][u], d[1][u], d[2][u], ...
```

In this case you might have something like

```
cnds = {d[0][0] == ..., d[1][0]==..., ...};
eqns = {d[0]'[u] == ..., d[1]'[u]==..., ...};
eqnspluscnds = Flatten[Join[eqns, cnds]];
```

```
umax=...;
sol =
Flatten[NDSolve[eqnspluscnds, {d[0], d[1], ...}, {u, 0, umax} ]]
Plot[Evaluate[{d[0][u], d[1][u], ...} /. sol], {u, 0.0, umax}]
```

9.2 You will implement Belief Propagation (BP) for K-SAT (say $K = 3$ and $K = 4$) The first one is to find a convenient parametrization of the BP messages. This was done in class. The second is to investigate numerically the convergence of BP as a function of α (the clause density). The third is to implement a decimation algorithm that finds satisfying assignments for α not too large.

9.Belief Propagation Equations for K-SAT Go through the derivation, especially if this was not done in detail during class.

9.3 Implementation of BP] You will implement BP according to the flooding (or parallel) schedule. initialize the messages uniformly randomly in $[0, 1]$. One iteration means that you send messages from nodes to clauses and back from clauses to variables. Define the following "convergence criterion": declare that the messages have "converged" if there is an iteration number (time) $t_{\rm conv}(\delta)$ such that no messages changes by more than δ at $t_{\rm conv}(\delta)$ (take the smallest such time).

Perform the following experiment. Take 100 K-SAT instances of length say $N = 5000$ and 10000 variables and for each instance implement BP as explained above with $\delta = 10^{-2}$. If the iterations do not converge stop them at a large time say $t_{\max} \approx 1000$. When they converge, they should do so in a shorter time $t_{\rm conv}(\delta) < t_{\max}$ that does not change much with N.

Plot as a function of α the empirical probability that the iterations converge. You should see that this probability is large for $\alpha < \alpha_{\rm BP}$ and drops abruptly around some threshold $\alpha_{\rm BP}$. For $K = 3$, $\alpha_{\rm BP} \approx 3.85$ and $K = 4$, $\alpha_{\rm BP} \approx 10.3$.

9.4 BP guided decimation] Now you will implement the following algorithm for finding SAT assignments. It uses the above BP procedure as a guide to take decisions on how to fix values for the variables. Once a variable has been fixed the K-SAT formula is suitably reduced - this step is called "decimation" - and BP is run again.

- Initialize with a K-SAT formula $\mathcal{F}$ of length N.
- For n= 1, ..., N do:
  - Run BP on an instance, as in the previous exercise (with the same convergence criterion).
  - If BP does not converge, return "assignment not found" and exit.
  - If BP converges, for each variable j compute its bias (express it in terms of $\hat{z}eta$ variables!)

$$\pi_j = \mu_j(1) - \mu_j(0) = \frac{\prod_{a\in\partial j}\mu_{a\to j}(1) - \prod_{a\in\partial j}\mu_{a\to j}(0)}{\prod_{a\in\partial j}\mu_{a\to j}(1) + \prod_{a\in\partial j}\mu_{a\to j}(0)}$$

  - Pick a variable $j(n)$ that has the largest absolute bias $|\pi_{j(n)}|$.

  - If $\pi_{j(n)} \geq 0$ fix $x_{j(n)} = 1$. Otherwise fix $x_{j(n)} = -1$.
  - Replace $\mathcal{F}$ by the K-SAT formula obtained by decimating variable $j(n)$.
- End-For
- Return all fixed variables.

Give for several values of α, the empirical success probability of this algorithm when tested over 100 instances. Compare this empirical success probability with the empirical convergence probability of the previous exercise. You should observe that $K = 3$ and $K = 4$ do not behave on the same way. Try to find an approximate threshold α_t beyond which the algorithm does not find SAT assignments.

10 Maxwell Construction

The Maxwell construction is a paradigm to guess the "true" (optimal/physical) behavior of a system from a simple model. For us the "simple model" is the description in terms of message-passing quantities and this setting is well-suited for this construction. Once the Maxwell construction has given us a guess, this guess can then often be converted into a rigorous statement. The important point here is that typically the proof uses the guess as an essential input. I.e., the Maxwell construction is typically a crucial first step in the proof.

We will discuss several instances of this paradigm in this chapter. Note that whenever this program works, then this means that the message-passing algorithm is not just a convenient low-complexity algorithm but plays a fundamental role in characterizing the problem.

10.1 The Original Maxwell Construction

The original Maxwell construction goes back to the 19th century struggle of trying to understand the liquid-vapor phase transition for simple substances (say H_2O). Quite surprisingly, even though this problem seems to have little to do with our three examples, there is a very straightforward analogy between the Maxwell construction for this problem and the Maxwell construction in our case. It is therefore worth to quickly review the problem.

Assume that we have a gas consisting of N molecules in a volume of V cubic meters under a pressure of p pascals and a temperature of T Kelvins. How are these quantities related? The *ideal* gas law states that

$$pV = NkT, \tag{10.1}$$

where k is the Boltzmann constant. The left picture in Figure 10.1 shows this relationship at different temperatures T. As one can see from this picture, as we decrease the volume, the pressure increases. The derivation of this ideal gas law is based on several simplifying assumptions. In reality the molecules[1] interact via

[1] The reader should not underestimate that the atomic and molecular constitution of matter acquired the status of scientific truth, as opposed to philosophical assumption, only in the 19th century thanks to the work of numerous chemists.

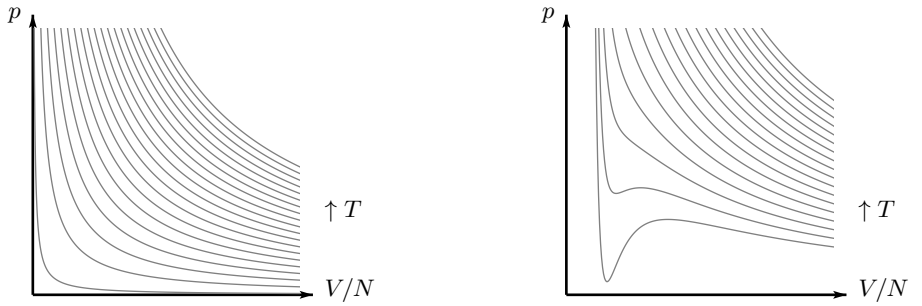


Figure 10.1 Left: Isotherms of the ideal gas equation of state. Right: Isotherms of the van der Waals equation of state. Note that below a critical temperature, the isotherms are no longer monotone.

forces of quantum mechanical origin.[2] These forces have a very short range and strong repulsive part and a weak long range attractive part. Because of the short range strong repulsion it is good model to assume that the molecules have an "effective volume". The ideal gas law simply *neglects* this effective volume as well as the attractive part of the force (so it neglects all forces hence the name ideal). The relation expressed in (10.1) is an *equation of state*, since it relates quantities that define the thermodynamic "state" of the system (namely, (p, V, T, N)).

In 1873, Johannes Diderik van der Waals derived a more accurate equation of state taking into account the non-zero effective size of the molecules as well as the weak long range attracting forces. His derivation resulted in the equation

$$(p + a\frac{N^2}{V^2})(V - bN) = NkT.$$

This equation is very similar in structure to the ideal gas law, but both the volume as well as the pressure terms are modified. The constant b takes into account the effective finite size of each molecule. Due to this finite size the *effective volume of the box* which is available to the N molecules shrinks from V to $V - bN$. The constant a takes into account attractive forces between molecules. It is assumed that these attractive forces act only between molecule of the gas but not between the wall and gas molecules. Therefore, close to a boundary, a molecule has more neighbors away from the boundary then towards the boundary and this creates an effective force "inwards," reducing the pressure of the gas. Note that the van der Waals equation is equivalent to $p = NkT/(V - bN) - a\frac{N^2}{V^2}$ so that the pressure is reduced by $a\frac{N^2}{V^2}$. The reduction is proportional to N^2 because each molecule close to the wall feels the effect of approximately N other molecules and there are of the order of N molecules close to the wall. To obtain an intensive quantity (pressure is intensive, i.e. independent of system size) we have to divide by V^2 which is the only other extensive quantity besides N. Another way to understand the form of this term is to assume that that the reduction in

[2] So it is only much later, in 1920-1930, that the true origin and proper way to model these forces was understood!

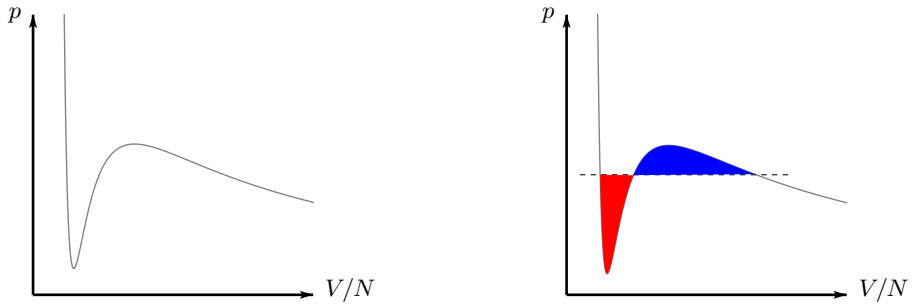


Figure 10.2 The original Maxwell construction. Left: One isotherm of the van der Waals equation of state. Right: The same isotherm, where a part of the curve is replaced by a horizontal line which is placed so that the two enclosed areas are in balance.

pressure is only a function of the density N/V close to the wall. For somewhat low densities (at least in the gas phase) one can expand this function in powers of N/V. The first order term must vanish because the attracting forces involve pairs of particles, leaving us with the second order term. Higher order terms are then neglected in the van der Waals theory.[3]

Let us write the above equation as $(p + a\frac{N^2}{V^2})(V/N - b) = kT$. Note that now all involved quantities, namely p, V/N, as well as T are *intensive* quantities, i.e., they are independent of the system size.

The right-hand side picture in Figure 10.1 shows the van der Waals isotherms for some choice of constants a and b and for various choices of T. Comparisons with measurements show that the predictions of the van der Waals equation are for the most part more accurate compared to the predictions of the ideal gas equation. But a closer look at Figure 10.1 shows a somewhat curious and non-physical behavior. Below a "critical" temperature, the isotherms are no longer relating the pressure p and the density V/N in a monotone fashion, i.e., below this critical temperate, there is a section where a decrease in density leads to a *decrease* in pressure. Clearly, the physical process is not described accurately in this range.

It was Maxwell who in 1875 suggested a modification of the van der Waals isotherms to account for this unphysical behavior. Consider Figure 10.2. The picture on the left shows one isotherm which shows a non-physical oscillating behavior. The idea of Maxwell was to modify this curve by replacing part of the curve by a horizontal line. This line is placed in such a way that the two areas (painted in red and blue in the picture) are in balance. Note that these two areas represent work since the pressure is measured in Newtons per square meters and the volume in meters cubed. So the product is Newton times meter, the units

[3] Note that such "virial expansions" in powers of density are computed in the framework of statistical mechanics once a precise model for the repulsive and attractive forces is fixed. These expansions relate coefficients like a and b to the expressions of the forces; and by experimentally measuring the equation state one extracts information about the forces.

of work. Roughly speaking, the basic thermodynamic argument to support the equality of the two areas is that the work done by compressing the gas (starting at large volumes) along the curved path and the work gained by relaxing the volume along the straight line back to its original value should be equal because the system has returned to its initial state, and no net work should have been gained or done (otherwise we would have a perpetuum mobile). The horizontal line segment corresponds to a phase in the system where the gas co-exists in two phases, namely as liquid as well as vapor. Along the line the percentage of each component changes from all vapor to all liquid. Note that as soon as all the gas is in liquid form, any further decrease in volume leads to a very large increase in pressure.

It is important to realize that for this physical system neither the ideal gas equation, nor the van der Waals equation, and not even the modified van der Waals equation with the Maxwell construction describe the system *exactly*. They are all increasingly accurate descriptions, taking into account more and more physical effects, and they agree reasonably well with experimental measurements.

For our applications we are in a somewhat easier situation. Our aim is not to find a correct theoretical description for a real physical system. Rather, we *start* with a model and this model is by definition *exact*. Therefore, in such a situation we can hope that also the Maxwell construction gives us an exact result.

10.2 Curie-Weiss Model

For the Curie-Weiss model we have in fact already "seen" the Maxwell construction, we just never mentioned it.

In Chapter 4 we computed the exact relationship between the magnetization m and the external magnetic field h for a particular interaction strength K. We saw in (**??**) that for a fixed h and K, m takes on a value which minimizes (the free energy function)

$$-(\frac{K}{2}m^2 + hm) - h_2(\frac{1+m}{2}). \tag{10.2}$$

If we take the derivative of the above expression, we see that m is a solution of the fixed-point equation

$$m = \tanh\big\{h + Km\big\}. \tag{10.3}$$

For $K < 1$, this fixed-point equation has only a single solution for each h, but for $K > 1$ it has up to three, depending on h. Note that even though there might be many solutions of m for each h, there is always exactly one solution of h for each m. The left picture in Figure 10.3 shows this relationship (which is a smooth curve) between m and h for $K = 2$. The dashed part of the curve are points (h, m) which are solutions to the fixed-point equation but where m is not the minimizer of (10.2).

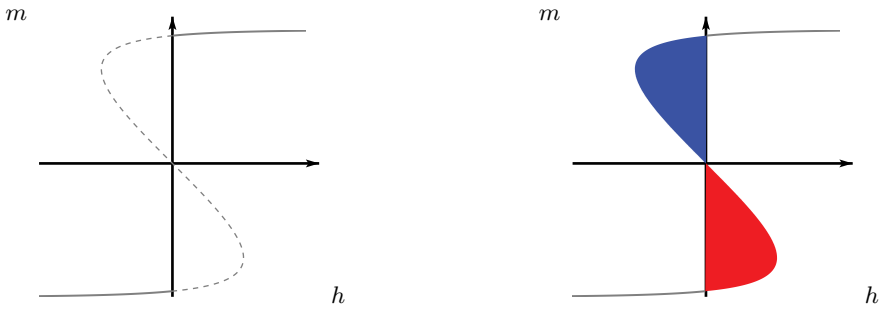


Figure 10.3 Phase transition in Curie-Weiss model when $K > 1$ as a function of h. The phase transition is at $h = 0$.

In Chapter **??** we attacked the CW (and SK) model via a message-passing approach. We first wrote down the message-passing equations. We then simplified the message-passing equations and derived the TAP equations. Note that the simplification itself was expected to be "loss-less" since it was based on the realization that only the leading terms in the message-passing equations contribute in the thermodynamic limit, the remaining terms tend to 0 with increasing system size.

But the graph corresponding to the CW model is not a tree. In fact it is as far away from a tree as one can get since it is a complete graph. It is therefore far from clear how well a message-passing analysis can capture the behavior. We saw, to our surprise, that the resulting message-passing equation, written as a fixed point equation is in fact equal to (10.3). But in the message-passing world we do not know that we "should" minimize (10.2). From the message passing perspective we start with a particular value of m and then we iterate.

Note that if we consider h as a function of m we again have in some range an unphysical behavior, namely in the branch where h decreases but m increases. It is therefore very natural to "correct" this unphysical part by a Maxwell construction, where we replace this unphysical part with a straight line which cuts the BP curve. Note that by symmetry we again have a balance of the two areas and that this Maxwell construction results in the correct phase diagram.

Let us see where we are. We have seen the Maxwell construction now for two examples, but so far it is perhaps not very convincing. For the gas model the Maxwell construction might appear like a kludge – a rough fix for an obvious problem. For the CW model, on the other hand, it might appear like a very lucky coincidence, but it did not tell us anything new.

It would be much more compelling if we could start with the BP equations and then from these equations could prove that the actual equation of state and phase transition threshold have to be of the form predicted by the Maxwell construction. In particular, this will be compelling if the actual equation of state and phase transition threshold is difficult to compute directly.

In the next section we discuss exactly such a case – namely the case of coding. Here the Maxwell construction does indeed give the correct prediction for the

MAP threshold and it is the starting point for a rigorous derivation of this quantity. More importantly, this is currently the only way of computing and proving the MAP threshold.

10.3 Coding: The Maxwell Construction for the BEC

Let us now consider coding, using elements of the (l, r)-regular LDPC ensemble, transmission over the BEC, and BP decoding. For this case we will see how we can determine the MAP threshold exactly. The Maxwell construction plays a crucial role in this determination.

As we saw in Chapter 6, the threshold for this case is determined by means of the fixed points (FP) of the equation

$$x = \epsilon f(\epsilon, x),$$

where $f(\epsilon, x) = \epsilon(1-(1-x)^{r-1})^{l-1}$. This leads us to consider the curve $(\epsilon(x), x)$ for $0 \leq x \leq 1$. Recall how from this curve we can determine the threshold – the threshold is the smallest value of ϵ which we see along this curve,

$$\epsilon^{\text{BP}} = \min_{0 \leq x \leq 1} \epsilon(x) = \min_{0 \leq x \leq 1} \frac{x}{(1-(1-x)^{r-1})^{l-1}}.$$

Instead of plotting the curve $(\epsilon(x), x)$ let us plot the curve $(\epsilon(x), (1-(1-x)^{r-1})^{l})$. Note that $(1-(1-x)^{r-1})^{l})$ is the erasure probability of the best estimate of a randomly chosen variable nodes we can make if we only use the "internal" messages but ignore the directly received observation of this bit (since we ignore the direct observation the factor ϵ is missing; on the other hand we have a power of l in the expression and not just $(l-1)$ as for the density evolution equations since we take all internal inputs into account). This is the "correct" curve to which to apply the Maxwell construction as we will see now. This curve is known as the *EXIT* curve in the literature.

LEMMA 10.1 (Graphical Characterization of Thresholds) *The left-hand side of Figure 10.4 shows the so-called BP EXIT curve associated to the* $(3, 6)$*-regular ensemble. This is the curve given by* $\{\epsilon(x), (1-(1-x)^{d_c-1})^{d_v}\}$, $0 \leq x \leq 1$. *For all regular ensembles with* $d_v \geq 3$ *this curve has a characteristic "C" shape. It starts at the point* $(1, 1)$ *for* $x = 1$ *and then moves downwards until it "leaves" the unit box at the point* $(1, x_u(1))$ *and extends to infinity.*

The right-hand side of Figure 10.4 shows the Maxwell construction for this case. The MAP threshold is constructed from the curve by inserting a vertical line. The line is inserted at that unique spot so that area of the BP EXIT curve to the left of the vertical line is equal to the area of this curve to the right.

The Maxwell conjecture only gives us a guess of the MAP threshold. To prove this conjecture needs considerably more work. We will first show that the conjectured threshold is always an upper bound on the MAP threshold. To prove

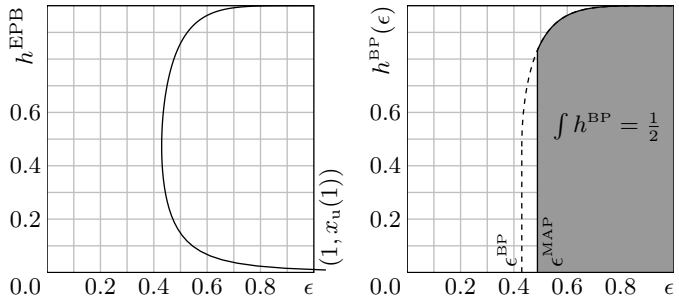


Figure 10.4 Left: The BP EXIT curve h^{BP} of the $(d_v = 3, d_c = 6)$-regular ensemble. The curve goes "outside the box" at the point $(1, x_{\mathrm{u}}(1))$ and tends to infinity. Right: The BP EXIT function $h^{\mathrm{BP}}(\epsilon)$. Both the BP as well as the MAP threshold are determined by $h^{\mathrm{BP}}(\epsilon)$.

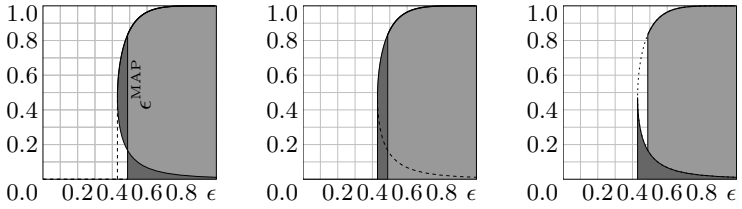


Figure 10.5 Maxwell construction.

that it is also a lower bound, and hence exact, needs different techniques, and we will discuss this later on.

Let $\mathcal{C}$ be a fixed code from the (l, r)-regular LDPC ensemble of length n. Let X denote the codeword, chosen uniformly at random from the set of all codewords and let Y be the received word, i.e., Y is the result of transmitting X over a BEC with parameter ϵ. We claim that

$$\frac{\mathrm{d}H(X|Y(\epsilon))}{n\mathrm{d}\epsilon} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{P}\{\hat{x}_i^{\mathrm{MAP}}(y_{\sim i}) = ?\} \tag{10.4}$$

To see this, assume that each bit i is transmitted over a BEC with parameter ϵ_i.

So we have

$$\begin{aligned}\frac{1}{n}\frac{\mathrm{d}H(X|Y(\epsilon_1,\cdots,\epsilon_n))}{\mathrm{d}\epsilon} &= \sum_{i=1}^{n}\frac{\partial H(X|Y(\epsilon_1,\cdots,\epsilon_n))}{\partial\epsilon_i}\Big|_{\epsilon_i=\epsilon}\\ &\overset{(a)}{=} \frac{1}{n}\sum_{i=1}^{n}\frac{\partial H(X_i|Y(\epsilon_1,\cdots,\epsilon_n))}{\partial\epsilon_i}\Big|_{\epsilon_i=\epsilon}\\ &\overset{(b)}{=} \frac{1}{n}\sum_{i=1}^{n}\mathbb{P}\{\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i})=?\}\\ &\overset{(c)}{\leq} \frac{1}{n}\sum_{i=1}^{n}\mathbb{P}\{\hat{x}_i^{\mathrm{BP}}(y_{\sim i})=?\}.\end{aligned}$$

To see (a) note that

$$H(X\mid Y)=H(X_i\mid Y)+H(X_{\sim i}|X_i,Y)=H(X_i\mid Y)+H(X_{\sim i}|X_i,Y_{\sim i}),$$

where in the last step we can drop the Y_i in $H(X_{\sim i}|X_i,Y)$ since the channel is memoryless. Now note $H(X_{\sim i}|X_i,Y_{\sim i})$ does not depend on ϵ_i so that this term drops when we take the derivative. For step (b),

$$H(X_i\mid Y)=\mathbb{P}\{Y_i=?\}\underbrace{\mathbb{P}\{\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i})=?\}}_{\text{not a function of }\epsilon_i}=\epsilon_i\mathbb{P}\{\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i})=?\}. \tag{10.5}$$

Finally, step (c) follows since the MAP decoder is optimal and hence has the lowest error probability of all decoders.

Let us now look closer at the last expression. Define

$$h^{\mathrm{BP}}(\epsilon)=\lim_{\ell\to\infty}\lim_{n\to\infty}\mathbb{E}_{\mathrm{LDPC}}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbb{P}\{\hat{x}^{\mathrm{BP},\ell}(y_{\sim i})=?\}\right]. \tag{10.6}$$

This limit exists and is given by density evolution. In fact, $h^{\mathrm{BP}}(\epsilon)$ is essentially the EXIT function which we just discussed above. This derivation makes it clearer why the EXIT function is the "right" quantity on which to apply the Maxwell construction.

Let us discuss this all in some more detail. As we discussed above, define

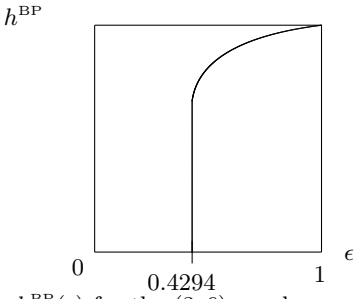


Figure 10.6 The function $h^{\mathrm{BP}}(\epsilon)$ for the $(3,6)$-regular ensemble.

$$\epsilon(x)=\frac{x}{(1-(1-x)^{r-1})^{l-1}},\quad h^{\mathrm{BP}}(x)=(1-(1-x)^{r-1})^{l},$$

and let us plot $(\epsilon(x), h^{\mathrm{BP}}(x))_{x=0}^{1}$, see Figure 10.6: Then the "envelope" of this

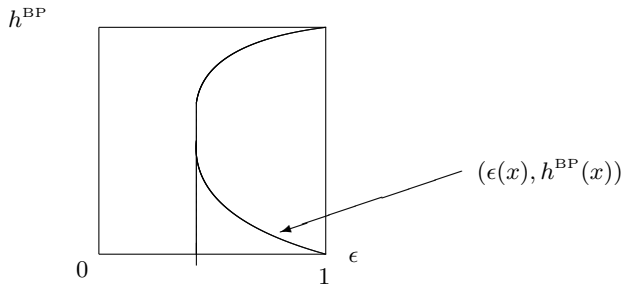


Figure 10.7 The curve $(\epsilon(x), h^{\mathrm{BP}}(x))$ and its "envelope."

curve is equal to $h^{\mathrm{BP}}(\epsilon)$ as a function of ϵ. It will be convenient to have a notation for the integral under this curve. To this end define the so called *trial entropy*:

$$P(x) = \int_0^x (1-(1-x)^{r-1})^l \epsilon'(x) dx. \tag{10.7}$$

$$= x + \frac{1}{r}(1-x)^{r-1}(l + l(r-1)x - rx) - \frac{l}{r}. \tag{10.8}$$

Note that $P(x)$ is the areas under the EXIT curve from the point $x = 0$ (this corresponds to a point at $+\infty$) until the point which is parameterized by x as indicated in Figure 10.8. Note that $P(0) = 0$. The function $P(x)$ is decreasing

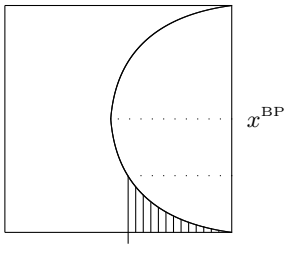


Figure 10.8 The trial entropy $P(x)$.

until $x = x^{\mathrm{BP}}$, where x^{BP} is that unique parameter so that $\epsilon^{\mathrm{BP}} = \epsilon(x^{\mathrm{BP}})$. For $x^{\mathrm{BP}} \leq x \leq 1$, $P(x)$ is increasing and $P(1) = 1 - \frac{l}{r}$, as a direct check shows.

It follows that there is a unique value of x in the region $[x^{\mathrm{BP}}, 1]$, call it x^A, so that $P(x^A) = 0$. We call $\epsilon(x^A)$ the *area threshold*, and write $\epsilon^A = \epsilon(x^A)$.

We now have the following sequence of inequalities:

$$
\begin{aligned}
&1-\frac{l}{r}-\liminf_{n\to\infty}\ \mathbb{E}_{\mathrm{LDPC}}[\frac{1}{n}H(x\mid y(\epsilon=\tilde{\epsilon}))]\\
&\overset{(a)}{=}\lim_{n\to\infty}\mathbb{E}_{\mathrm{LDPC}}[\frac{1}{n}H(x\mid y(\epsilon=1))]-\liminf_{n\to\infty}\ \mathbb{E}_{\mathrm{LDPC}}[\frac{1}{n}H(x\mid y(\epsilon=\tilde{\epsilon}))]\\
&\overset{(a)}{=}\limsup_{n\to\infty}\ \mathbb{E}_{\mathrm{LDPC}}[\frac{1}{n}\{H(x\mid y(\epsilon=1))-H(x\mid y(\epsilon=\tilde{\epsilon}))\}]\\
&\overset{(b)}{=}\limsup_{n\to\infty}\ \mathbb{E}\left[\int_{\tilde{\epsilon}}^{1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{P}\{\hat{x}_i^{\mathrm{MAP}}(y'_{\sim i})=?\}\right]d\epsilon\\
&\overset{(c)}{=}\limsup_{n\to\infty}\ \int_{\tilde{\epsilon}}^{1}\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbb{P}\{\hat{x}_i^{\mathrm{MAP}}(y'_{\sim i})=?\}\right]d\epsilon\\
&\leq\int_{\tilde{\epsilon}}^{1}\limsup_{n\to\infty}\ \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbb{P}\{\hat{x}_i^{\mathrm{MAP}}(y'_{\sim i})=?\}\right]d\epsilon\\
&\overset{(d)}{\leq}\int_{\tilde{\epsilon}}^{1}\lim_{\ell\to\infty}\lim_{n\to\infty}\ \mathbb{E}\left[\frac{1}{n}\mathbb{P}\{\hat{x}_i^{\mathrm{BP},\ell}(y'_{\sim i})=?\}\right]d\epsilon\\
&\overset{(e)}{=}P(1)-P(\tilde{\epsilon})\\
&=1-\frac{l}{r}-P(\tilde{\epsilon}).
\end{aligned}
$$

In step (a) note that $\frac{1}{n}H(x\mid y(\epsilon=1))$ is equal to the logarithm of the size of the code normalized by the length. It is intuitive that the limit of this quantity when $n\to\infty$, and averaged over the ensemble, is equal to the "design rate" of the code which is $1-\frac{d_v}{d_c}$. Even though this is intuitive, this needs some proof. Since the proof is purely combinatorial we skip the steps. But this transition is valid for all (l,r)-regular ensembles with $2\leq l\leq r$.

In step (b) we write the conditional entropy as an integral of its derivative and replace the derivative with the sum as we previously discussed. Since the integral is non-negative, we can exchange the order of the two integrals by Tonelli. This is step (c). In step (d) we apply the Fatou-Lebesgue theorem by observing that the integrand is bounded. Step (d) follows by the optimality of the MAP decoder, and in the final two steps we have used the definition of the trial entropy.

Equivalently,

$$
\liminf_{n\to\infty}\ \mathbb{E}_{\mathrm{LDPC}}\left[\frac{1}{n}H(x\mid y(\epsilon(x))\right]\geq P(x). \tag{10.9}
$$

DEFINITION 10.2 (MAP Threshold) [4] The *MAP threshold* of the (d_v,d_c)-regular ensemble for the BEC is denoted by $\epsilon^{\mathrm{MAP}}(d_v,d_c)$ and is defined by

$$
\inf\{\epsilon\in[0,1]:\liminf_{n\to\infty}\mathbb{E}[H(X_1^n\mid Y_1^n(\epsilon))/n]>0\}.
$$

□

[4] Define $\mathrm{P}_{e,i}=\Pr\{X_i\neq\hat{X}_i(Y_1^n)\}$, where $\hat{X}_i(Y_1^n)$ is the MAP estimate of bit i based on the observation Y_1^n. Note that by the Fano inequality we have $H(X_i\mid Y_1^n)\leq h_2(P_{e,i})$. Assume

We conclude that $\epsilon^{\text{MAP}} \geq \epsilon^A = \epsilon(x^A)$, the area threshold.

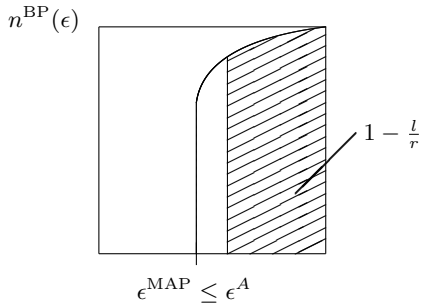


Figure 10.9 aa

So far we have seen that the threshold given by the Maxwell construction is an upper bound on the MAP threshold. There are several ways of proving the reverse inequality. For the specific case at hand, namely transmission over the BEC, one can give a purely combinatorial proof. The idea is to prove that with high probability the matrix which we get if we start with the parity-check matrix and remove all columns which correspond to non-erased bits has rank equal to the number of erased bits. This shows that with high probability the codeword can be reconstructed by solving the corresponding linear system of equations, i.e., with high probability the MAP decoder succeeds. Since this proof is very specific to the erasure channel we skip it. There is a second more conceptual

that we are transmitting above $\epsilon^{\text{MAP}}(d_v, d_c)$ so that $\mathbb{E}[H(X_1^n \mid Y_1^n)/n] \geq \delta > 0$.[5] Then

$$h_2(\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n} \mathrm{P}_{e,i}]) \geq \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n} h_2(\mathrm{P}_{e,i})] \geq \mathbb{E}[\sum_{i=1}^{n} H(X_i \mid Y_1^n)/n]$$
$$\geq \mathbb{E}[H(X_1^n \mid Y_1^n)/n] \geq \delta > 0.$$

In words, if we are transmitting *above* the MAP threshold, then the ensemble average bit-error probability is lower bounded by $h_2^{-1}(\delta)$, a strictly positive constant. This ensemble is therefore not suitable for reliable transmission above this threshold.
In general we cannot conclude from $\mathbb{E}[H(X_1^n \mid Y_1^n)/n] \leq \delta$ that the average error probability is small. This is possible if we have the slightly stronger condition $\mathbb{E}[\sum_{i=1}^n H(X_i \mid Y_1^n)/n] \leq \delta$. In this case $\delta \geq \frac{1}{n}\mathbb{E}[\sum_{i=1}^n H(X_i \mid Y_1^n)] = \frac{1}{n}\mathbb{E}[\sum_{i=1}^n \mathbb{E}_{Y_1^n}[h_2(\min_x p(x \mid Y_1^n))]] \geq \frac{1}{n}\mathbb{E}[\sum_{i=1}^n \mathbb{E}_{Y_1^n}[2\min_x p(x \mid Y_1^n)]] = \frac{1}{n}\mathbb{E}[\sum_{i=1}^n 2\mathrm{P}_{e,i}]$, so that $\frac{1}{n}\mathbb{E}[\sum_{i=1}^n \mathrm{P}_{e,i}] \leq \frac{1}{2}\delta$. The last step in the previous chain of inequalities follows since under MAP decoding the error probability conditioned that we observed y_1^n is equal to $\min_x p(x \mid y_1^n)$. An alternative way to prove this is to realize that $H(X_i \mid Y_1^n)$ represents a BMS channel with a particular entropy and to use extremes of information combining to find the worst error probability such a channel can have. The extremal channel in this case is the BEC. But for the codes we consider we will see that below ϵ^{MAP} we can indeed decode correctly with high probability, which justifies the choice of our definition.
The reader might wonder why we did not start with an operational interpretation of the MAP threshold as the channel parameter below which a MAP decoder can decode with high probability. As pointed out above, for the codes we consider the given definition is in fact equivalent to the operational one. But in addition it has the advantage that the conditional entropy connects directly to the quantities which appear in our analysis, in particular to the generalized EXIT curve.

approach using spatial coupling and the interpolation technique which applies to all such problems. We will get back to this point in the next chapter.

10.4 Compressive Sensing

Also for compressive sensing there is a Maxwell construction. As a starting point however one has to consider the compressive sensing problem for a fixed and known source distribution, rather than looking for a universal algorithm.

10.5 Random K-SAT

As always, for K-SAT the situation is the most complicated. Again it is possible to write down a Maxwell construction. However, the starting point is not the BP-guided decimation algorithm but a more sophisticated algorithm, called *survey propagation*.

10.6 Discussion

Besides the original example, we have given two explicit examples of the Maxwell construction. For the CW model, the Maxwell construction appears somewhat like a coincidence. We first computed the exact relationship between average magnetization and the external field and then we computed the same relationship from a message-passing perspective. Comparing the two expressions we see that they are related by a Maxwell construction, just like in the original construction for an ideal gas.

Even more interesting is the situation if we cannot in fact compute the exact free energy expression but, starting with the message-passing formulation, can construct it using a Maxwell construction. This was the case for our second example, namely coding. There is currently no classical way of computing the MAP threshold. We have seen that the Maxwell construction gives us a guess of where this phase transition appears and we have also seen how we can prove that this guess is an *upper bound* on the MAP threshold. In the third part of these notes we will see how we can further show that this guess is also a *lower bound* on the MAP threshold using the concepts of spatial coupling and the so-called interpolation method. So in this case, the Maxwell construction, together with further techniques, allows us to solve, what from a classical perspective seems to be a hard problem.

This is a general theme. But, there is no trivial recipe for how to apply the Maxwell construction and how to prove that it is indeed correct. Each case requires some slightly different tricks and techniques. In fact, it is easy to construct examples (like K-SAT with BP guided decimation) where the predictions given

by the Maxwell construction are not even correct. But with a little bit of experience the Maxwell construction is a powerful paradigm.

Problems

10.1 *Magnetization of the Ising model on a d-regular graph with large girth.* In this problem we consider the ferromagnetic Ising model on a d-regular graph with large girth. Using the probabilistic method Erdős and Sachs proved that there exist a graphs $G_{n,d}$ on n vertices, with all vertex degrees equal to d and with a girth $g_{n,d} \geq (1 - o(1)) \log_{d-1} n$ (here $o(1)$ stands for a function that goes to zero as $n \to +\infty$). We recall that the girth is the length of the shortest loop in the graph.

Consider the Gibbs distribution of the Ising model on $G_{n,d}$

$$\mu_{n,d}(\underline{s}) = \frac{1}{Z_{n,d}} \exp\left(\frac{\beta J}{d} \sum_{\{i,j\} \in \text{edges}} s_i s_j + \beta h \sum_{i=1}^{n} s_i \right)$$

The Hamiltonian is given by the contribution of all ferromagnetic interactions associated to edges $\{i, j\}$, and a contribution from a constant magnetic field. The strength of the interaction is scaled by d for later convenience. Note that $J > 0$ but h can take both signs.

Recall that the magnetization at a vertex o is defined as $\langle s_o \rangle_{n,d}$ where $\langle - \rangle_{n,d}$ is the usual Gibbs average. This quantity is non trivial to compute. On the other hand we can run BP and compute the BP estimates of the magnetization.

(i) The second Griffith-Kelly-Sherman correlation inequality states that for Ising models with all interaction coefficients and all magnetic fields positive the magnetization can only decrease when one coefficient decreases. In the present case this inequality implies that the magnetization decreases when an edge is removed from $G_{n,d}$. Now consider the neighborhood of a vertex o, namely $N = \{i \in G_{n,d} | \text{dist}(o, i) \leq g_{n,d} - 1\}$. Define $\langle - \rangle_N$ the Gibbs average for the Ising model restricted to N. Show that for $h \geq 0$

$$\langle s_o \rangle_{n,d} \geq \langle s_o \rangle_N$$

and that for $h \leq 0$

$$\langle s_o \rangle_{n,d} \leq \langle s_o \rangle_N$$

Hint: for the second inequality use symmetry properties under the operation $h \to -h$.

(ii) The average $\langle s_o \rangle_N$ can be computed exactly from the BP recursion. Why? Show that this recursion is:

$$m^{(t)} = \tanh(\beta h + d \tanh^{-1}(\tanh \beta \frac{J}{d} \tanh u^{(t)}))$$

$$u^{(t)} = \beta h + (d-1) \tanh^{-1}(\tanh \frac{\beta J}{d} \tanh u^{(t-1)}), \qquad u^{(0)} = h$$

and that $\langle s_o \rangle_N = m^{(g_{n,d}-1)}$.

Remark: go back to homework 4 and observe this is the same recursion that you had derived by "other means".

(iii) Take now a fixed sequence of graphs $G_{n,d}$ with respect to n. Observe from above that for $h > 0$ and all t,

$$\liminf_{n\to+\infty} \langle s_o \rangle_{n,d} \geq m^{(t)},$$

and for $h \geq 0$

$$\limsup_{n\to+\infty} \langle s_o \rangle_{n,d} \leq m^{(t)}.$$

We want to look at the limit $d \to +\infty$. Show that

$$\lim_{d\to+\infty} \liminf_{n\to+\infty} \langle s_o \rangle_{n,d} \geq \lim_{t\to+\infty} m_{\mathrm{CW}}^{(t)},$$

and for $h \leq 0$ and all t

$$\lim_{d\to+\infty} \limsup_{n\to+\infty} \langle s_o \rangle_{n,d} \leq \lim_{t\to+\infty} m_{\mathrm{CW}}^{(t)},$$

where $m_{\mathrm{CW}}^{(t)}$ is the BP-magnetization of the CW model and satisfies the recursion

$$m_{\mathrm{CW}}^{(t)} = \tanh(\beta(h + J m_{\mathrm{CW}}^{(t-1)}))$$

with the initial condition $m_{\mathrm{CW}}^{(o)} = \tanh \beta h$.

Remark: These inequalities suggest the conjecture

$$\lim_{d\to+\infty} \liminf_{n\to+\infty} \langle s_o \rangle_{n,d} = \lim_{d\to+\infty} \limsup_{n\to+\infty} \langle s_o \rangle_{n,d} = \langle s_o \rangle_{\mathrm{CW}}$$

where $\langle s_o \rangle_{\mathrm{CW}}$ is the true CW magnetization.

Part III

Advanced Topics: from Algorithms to Optimality

11 Variational Formulation and the Bethe Free Energy

In our previous lectures we have discussed how we can analyze the performance of various low-complexity algorithms, in particular algorithms of message-passing type. We have seen that in the limit of infinite system size, such algorithms have threshols and we were able to characterize these thresholds quantitatively. Such thresholds are often called *dynamical* thresholds since they are associated to the *dynamics* of a process (for us this is the algorithm).

But there is typically also a *static* phase transition. This corresponds to a phase transition which describes a change of the system behavior itself, independent of any algorithmic question. E.g., in coding we can ask how much noise we can add so that with high probability there is a unique codeword which is "compatible" with the received information. In communicatons jargon, this corresponds to the MAP threshold. For compressive sensing we can ask how the number of measurements has to scale with the number of unknowns so that with high probability there is a unique sparse vector which is compatible with the measurements. Finally, in K-SAT we can ask how many constraints we can have per Boolean variable so that with high probability a random formula is satisfiable. This is usally refered to as the SAT-UNSAT threshold.

Why are we interested in these quantities? Some systems are given to us and we cannot change them (e.g., K-SAT). In this case it is important to know how well a computationally unbounded system could do in order to gauge how well our algorithm is performing. But often we are actually in control of the system itself. E.g., think of the coding problem or also compressive sensing. It is typically us who designs the code or the measurement matrix. So in these cases it is important to know that the system itself is designed in such a way that at least in principle (if we had unbounded computational resources at our disposal) it has a good performance (comparable to the optimal one). E.g., in coding we can then compare the MAP threshold to the ultimate limit, namely the Shannon threshold and hopefully these two thresholds are close.

As we will see, there are two basic themes which appear. First, static thresholds are in general much harder to compute than the dynamical ones. This is why we have postponed this discusson towards the end. In a few cases we will be able to derive rigorous quantitative statements. In some other ones, we will have to be content with computations which are believed to yield the correct value but fall short of a mathematical proof. The second, perhaps more surprising theme

is that the analysis of the static threshold can often be done by looking at the behavior of the message-passing algorithm! Why message-passing, a sub-optimal algorithm, should have any bearing on the behavior of the optimal algorithm is at first glance puzzling.

As we will see, the key object which connects these two themese is the so-called Bethe free energy. It is an "approximation" to the true free energy which itself depends on the fixed points of the message-passing algorithm. In some instances the static thresholds predicted by the Bethe free energy can be shown to be indeed correct.

Let us discuss this in more detail. Computing the true free energy for general graphical models (or statistical mechanics models) is an impossible task. An important approximation philosophy is the so-called "mean-field theory." In this theory, when looking at the interactions of a "spin" with the rest of the system, we only take into account very close neighbors exactly, but model influences of the remaining system simply by a "mean field," i.e., a field which models the average influence of this part of the system. For models defined on sparse graphs that are locally tree-like, a very good form of mean filed theory was developed by Bethe and Peierls. This leads to the so-called Bethe free energy approximation. We note that this is already a "sophisticated" version of the most basic mean field theory.

As we will see the Bethe-Peierls theory involves fixed point equations that are the same as those occurring in Belief-Propagation. Their use and to some extent interpretation are however different. Note that there is a clash of initials (BP) that is solely due to an historical accident. We hope that this will not cause major confusions.

In this chapter we treat in detail the case of graphical models with a discrete alphabet $\mathcal{X}$. As a direct application we will look more closely at the cases of coding and K-SAT. For models with a continuous alphabet such as those occuring in the context of compressive sensing the ideas are conceptually the same, but the calculations have to be slightly adapted. We consider a general Gibbs measure of the form

$$\mu\left(\underline{s}\right) = \frac{1}{Z}\prod_{a} f_a\left(x_{\partial a}\right), \tag{11.1}$$

where the variables $x_i \in \mathcal{X}$, $i = 1, \ldots, n$ and f_a, $a = 1, \ldots, m$ are kernel functions associated to check nodes which depend on $x_{\partial a} = \{x_i, i \in \partial a\}$. In Chapter 5 we discussed the sum-product algorithm that computes *BP-marginals* for such measures. Recall that these are the *exact marginals* when the graph is a tree. Similarly we will see that on a tree the free energy

$$f = -\frac{1}{\beta n}\ln Z, \tag{11.2}$$

can be expressed exactly in terms of the marginals of the measure. This is the starting point of the formalism developped in this chapter.

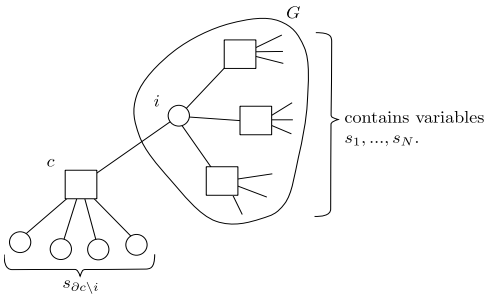


Figure 11.1 Induction procedure: G is the original tree to which we add check c connected to i such that the new graph is a tree

11.1 The Gibbs measure on trees

Consider the (exact) marginals

$$\nu_i(x_i) = \sum_{\sim x_i} \mu(x_1, ..., x_N), \qquad \nu_a(x_{\partial a}) = \sum_{\sim x_{\partial a}} \mu(x_1, ..., x_N).$$

As explained in Chapter 5 on a tree these can be computed exactly by the sum-product algorithm. More is true.

Lemma 11.1.1 The Gibbs measure on a tree can be expressed in terms of its marginals as follows,

$$\mu(\underline{x}) = \prod_a \nu_a(x_{\partial a}) \prod_i (\nu_i(x_i))^{1-d_i} \tag{11.3}$$

where d_i is the degree of node i.

Proof We prove (11.3) by induction over number the number m of check nodes. For $m = 1$ the unique clause is connected to variable nodes with $d_i = 1$. Thus (11.3) is trial in this case. Now, we assume (11.3) is true for a tree graph G with m check nodes and prove that it also holds for the new Gibbs measure

$$\mu_{\text{new}}(x_{\partial c \setminus i}, x_1, ..., x_n) = \frac{1}{Z_{\text{new}}} f_c(x_{\partial c}) \prod_a f_a(x_{\partial a}) \tag{11.4}$$

obtained when one adds one check node c connected to a variable node i in such a way that the new graph[1] is a tree. The original tree G and the new tree are depicted on figure11.1

Consider the conditional probability $\Pr(x_{\partial c \setminus i} \mid x_1, ..., x_n)$ of an assignement $x_{\partial c \setminus i}$ given $x_1, \ldots, x_n$. We observe that

$$\begin{aligned} \Pr(x_{\partial c \setminus i} \mid x_1, ..., x_n) &= \Pr(x_{\partial c \setminus i} \mid x_i) \\ &= \frac{\nu_{\text{new},c}(x_{\partial c})}{\nu_{\text{new},i}(x_i)}. \end{aligned}$$

[1] We do not discuss the somewhat trivial case where the new check is disconnected.

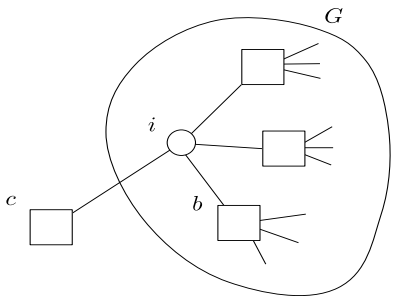


Figure 11.2 Factor graph for the marginal distribution (11.6). We select an arbitrary check $b \in \partial i \setminus c$.

Therefore, denoting by $\nu_{\text{new}}(x_1, ..., x_n)$ the marginalisation of (11.4) over the variables $x_{\partial c \setminus i}$,

$$\begin{aligned}\mu_{\text{new}}\left(x_{\partial c \setminus i}, x_1, ..., x_n\right) &= \Pr\left(x_{\partial c \setminus i} \mid x_1, ..., x_N\right) \nu_{\text{new}}(x_1, ..., x_n) \\ &= \nu_{\text{new},c}(x_{\partial c}) \left(\nu_{\text{new},i}(x_i)\right)^{-1} \nu_{\text{new}}(x_1, ..., x_n). \end{aligned} \tag{11.5}$$

Now, by definition of $\nu(x_1, ..., x_n)$ we have

$$\begin{aligned}\nu_{\text{new}}(x_1, ..., x_N) &= \frac{1}{Z_{\text{new}}} \sum_{x_{\partial c \setminus i}} f_c(x_{\partial c}) \prod_a f_a(x_{\partial a}) \\ &= \frac{1}{Z_{\text{new}}} \widetilde{f}_c(x_i) \prod_a f_a(x_{\partial a}). \end{aligned} \tag{11.6}$$

where we have set $\sum_{x_{\partial c \setminus i}} f_c(x_{\partial c}) = \widetilde{f}_c(x_i)$. This distribution has the factor graph depicted on figure 11.2 This tree still has $m+1$ check nodes. However c can be absorbed in any arbitrarily selected check $b \in \partial i \setminus c$:

$$\begin{aligned}\nu_{\text{new}}(x_1, ..., x_n) &= \frac{1}{Z_{\text{new}}} \widetilde{f}_c(x_i) \prod_a f_a(x_{\partial a}) \\ &= \frac{1}{Z_{\text{new}}} \widetilde{f}_c(x_i) f_b(x_{\partial b}) \prod_{a \neq b} f_a(x_{\partial a}) \\ &= \frac{1}{Z_{\text{new}}} \widetilde{f}_b(x_{\partial b}) \prod_{a \neq b} f_a(x_{\partial a}) \end{aligned}$$

where we have set $\widetilde{f}_c(x_i) f_b(x_{\partial b}) = \widetilde{f}_b(x_{\partial b})$. We recognize this expression as a Gibbs measure defined on a tree with m check nodes, so that we can apply the induction hypothesis

$$\nu_{\text{new}}(x_1, \ldots, x_n) = \prod_a \tilde{\nu}_{\text{new},a}(x_{\partial a}) \prod_i \left(\nu_{\text{new},i}(x_i)\right)^{1-d_i}.$$

Here $\nu_{\text{new},a}$ and $\nu_{\text{new},i}$ are the marginals of ν_{new}. But clearly, they are also the marginals of ν_{new} in (11.4). Combining this last formula with (11.5) yields the desired result. □

11.2 The free energy on trees

We begin with a general and important expression for the free energy which is universally valid, and in particular is not restricted to trees. This formula is best understood when the Gibbs measure (11.1) is expressed in its traditional physics form

$$\mu\left(\underline{x}\right)=\frac{1}{Z}\exp\left(-\beta\mathcal{H}\left(\underline{x}\right)\right). \tag{11.7}$$

The formal relation between the Hamiltonian and the kernel functions is

$$\beta\mathcal{H}\left(\underline{x}\right)=-\sum_{a}\ln f_{a}(x_{\partial a}) \tag{11.8}$$

Replacing (11.7) in the definition of the freee energy (11.2) one easily finds for the un-normalized free energy $F\equiv nf$,

$$F=\langle\mathcal{H}\rangle-\beta^{-1}S\left[\mu\right] \tag{11.9}$$

where

$$\begin{aligned}\langle\mathcal{H}\rangle&=\sum_{x_1,\ldots,x_N}\mathcal{H}\left(x_1,...,x_N\right)\mu\left(x_1,...,x_N\right)\\ S\left[\mu\right]&=-\sum_{x_1,\ldots,x_N}\mu\left(x_1,...,x_N\right)\ln\mu\left(x_1,...,x_N\right).\end{aligned}$$

Here $\langle\mathcal{H}\rangle$ is the average value of the Hamiltonian. Physically this represents the total average internal energy that the system possesses, and is commonly called the internal energy. $S[\mu]$ is called the Gibbs entropy. This is nothing else than a special form of Shannon's entropy written down for the Gibbs measure. In thermodynamics one shows that the free energy is the amount of work that a system can perform. Equ. (11.9) says that this is equal to the the total internal energy minus an unsuable part equal given by the temperature times the entropy.

We now apply formula (11.9) to the Gibbs measure on a tree graph. This leads to

PROPOSITION 11.1 On a tree graphical model the (un-normalized) free energy $F=nf$ can be expressed in terms of its marginals as

$$F=\sum_{a}\sum_{x_{\partial a}}\nu_{a}\left(x_{\partial a}\right)\ln\frac{\nu_{a}\left(x_{\partial a}\right)}{f_{a}\left(x_{\partial a}\right)}+\sum_{i}\left(1-d_{i}\right)\sum_{x_{i}}\nu_{i}\left(s_{i}\right)\ln\nu_{i}\left(x_{i}\right) \tag{11.10}$$

Proof Using (11.8) the internal energy contribution yields

$$\begin{aligned}\langle\mathcal{H}\rangle_{\mu}&=-\sum_{a}\sum_{x_1,\ldots,x_N}\mu\left(x_1,...,x_N\right)\ln f_{a}(x_{\partial a})\\ &=-\sum_{a}\sum_{x_{\partial a}}\nu\left(x_{\partial a}\right)\ln f_{a}\left(x_{\partial a}\right).\end{aligned}$$

Note that this formula is completely general and does not depend on having a tree graph.

To compute the contribution of the enetropy we use (11.3) in lemma 11.1.1. This gives

$$\begin{aligned} S[\mu] &= -\sum_a \sum_{x_1,\ldots,x_N} \mu(x_1,\ldots,x_N)\left(\ln \nu_a(x_{\partial a})\right) \\ &\quad + \sum_i (1-d_i) \sum_{x_1,\ldots,x_N} \mu(x_1,\ldots,x_N) \ln\left(\nu_i(x_i)\right) \\ &= -\sum_a \sum_{x_{\partial a}} \nu_a(x_{\partial a}) \ln \nu_a(x_{\partial a}) + \sum_i (1-d_i) \sum_{x_i} \nu_i(x_i) \ln \nu_i(x_i) \end{aligned}$$

Combining the the energetic and entropic contributions gives (11.10) □

In chapter 5 we learned how to compute the marginals in terms an exact message passing equations on the tree. Recall that we have two types of messages: those flowing from variable to check nodes $\mu_{i\to a}(x_i)$ and those flowing from check to variables node $\mu_{a\to i}(x_i)$. The exact marginals are given by

$$\begin{aligned} \nu_i(x_i) &= \frac{\prod_{a\in\partial i} \hat{\mu}_{a\to i}(x_i)}{\sum_{x_i} \prod_{a\in\partial i} \hat{\mu}_{a\to i}(x_i)} \\ \nu_a(x_{\partial a}) &= \frac{f_a(x_{\partial a}) \prod_{i\in\partial a} \mu_{i\to a}(x_i)}{\sum_{x_{\partial a}} f_a(x_{\partial a}) \prod_{i\in\partial a} \mu_{i\to a}(x_i)}. \end{aligned}$$

and the messages by the sum-product equations by

$$\begin{aligned} \mu_{i\to a}(x_i) &= \prod_{b\in\partial i\setminus a} \hat{\mu}_{b\to i}(x_i) \\ \hat{\mu}_{a\to i}(x_i) &= \sum_{\sim x_i} f_a(x_{\partial a}) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(x_j) \end{aligned}$$

Moreover the messages are uniquely defined by their "initial" values at the leaf nodes. Recall, when the leaf node is a check the outgoing message equals $f_a(x_{\partial a})$ when the leaf node is a check, and equals 1 when the leaf node is a variable.

Using these expressions in (11.10), a straightforward calculation leads to the alternative expression for the free energy

PROPOSITION 11.2 On a tree graphical model the (un-normalized) free energy $F = nf$ can be expressed in terms of the BP messages as a sum of three contributions associated to variable nodes, check nodes and edges

$$F = \sum_i F_i + \sum_a F_a - \sum_{(i,a)} F_{ia},$$

where the three contributions are

$$F_i = \ln\left\{\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}_{b\to i}(x_i)\right\}$$
$$F_a = \ln\left\{\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu_{i\to a}(x_i)\right\}$$
$$F_{ia} = \ln\left\{\sum_{x_i}\mu_{i\to a}(x_i)\,\hat{\mu}_{a\to i}(x_i)\right\}$$

We stress that in this formula the messages do not have to be normalized. Indeed they were not normalized in the first place in the sum-product equations. The anxious reader can check that F is invariant under the renormalizations $\hat{\mu}_{a\to i}\to\hat{z}_{a\to i}\hat{\mu}_{a\to i}$ and $\mu_{i\to b}\to\hat{z}_{i\to a}\mu_{i\to a}$ for any arbitrary numbers $\hat{z}_{a\to i}$ and $z_{i\to a}$.

11.3 Bethe free energy for general graphical models

We now turn our attention to general graphical models of the type (11.1) with a factor graph that is not necessarily a tree, and introduce a definition. We assign to each edge two ditributions $\mu_{i\to a}(s_i)$ and $\mu_{a\to i}(s_i)$. The set of all distributions forms two vectors denoted by μ and $\underline{\hat{\mu}}$. The notation is the same than for the BP messages for reasons that will become clear, however the reader should bear in mind that conceptually these are general distributions, not necessarily equal to the BP messages (for one thing the BP equations do not necessarily have a unique solution). The *Bethe free energy* is by definition the functional

$$F_{\text{Bethe}}\left[\underline{\mu},\underline{\hat{\mu}}\right] = \sum_i F_i\left[\{\mu_{i\to b}, b\in\partial i\}\right] + \sum_a F_a\left[\{\mu_{i\to a}, i\in\partial a\}\right] - \sum_{ai} F_{ai}\left[\{\mu_{i\to a},\hat{\mu}_{a\to i}\}\right]. \tag{11.11}$$

with the three contributions associated to variable and check nodes, and edges.

$$F_i = \ln\left\{\sum_{s_j}\prod_{b\in\partial i}\hat{\mu}_{b\to i}(s_i)\right\} \tag{11.12}$$
$$F_a = \ln\left\{\sum_{s_{\partial a}} f_a(s_{\partial a})\prod_{i\in\partial a}\mu_{i\to a}(s_i)\right\} \tag{11.13}$$
$$F_{ai} = \ln\left\{\sum_{s_i}\mu_{j\to a}(s_i)\,\hat{\mu}_{a\to j}(s_i)\right\}. \tag{11.14}$$

what is the idea behind this definition? The Bethe free energy exactly gives the true free energy for factor graphs that are trees. For a loopy factor graph it may seem a reasonable idea to propose the Bethe free energy as an ansatz (an educated guess) that hopefully approximates the true one. However there

are various problems that immediately arise. The most urgent is: how does one choose the messages? The BP equations do not necessarily have a unique solution for loopy graphs. The rule of thumb is to take the messages that minimize the Bethe functional. Were does this rule of thumb come from? In the standard physics variationnal approaches the true free energy is always lower than the ansatz. Then minimizing the ansatz over a set of open parameters is the best possible choice. This is not true for the Bethe free energy, so the usual rule of thumb has be considered with a grain of salt. We stress that there is no general inequality that states that the true free energy is always smaller than the Bethe functional. In general, quantifying the difference between the true and minimal bethe free energy is a hard problem about which we do not know much.

The discussion above suggests that a first important step is to look at stationnary points of the Bethe functional. One then discovers the follwing important result.

PROPOSITION 11.3 The stationary points of the Bethe free energy satisfy the sum-product message passing equations and conversely the solutions of the sum-product equations are stationary points of the Bethe free energy.

Proof For a finite system with a discrete alphabet the Bethe free energy functional is really a function of many variables, namely $\mu_{i\to a}(x_i)$, $\mu_{a\to i}(x_i)$ for $x_i \in \mathcal{X}$. Thus the stationnarity conditions are simply

$$\frac{\partial F_{\text{Bethe}}}{\partial \mu_{i\to a}(x_i)} = 0, \qquad \frac{\partial F_{\text{Bethe}}}{\partial \widehat{\mu}_{a\to i}(x_i)} = 0$$

For the first derivative there is a contribution from F_a and F_{ia},

$$\frac{\partial F_{\text{Bethe}}}{\partial \mu_{i\to a}(x_i)} = \frac{\hat{\nu}_{a\to i}(x_i)}{\sum_{x_i} \mu_{i\to a}(x_i)\,\hat{\mu}_{a\to i}(x_i)} - \frac{\sum_{\sim x_i} f_a(x_{\partial a}) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(x_j)}{\sum_{x_{\partial a}} f_a(x_{\partial a}) \prod_{j\in\partial a} \mu_{j\to a}(x_j)},$$

and for the second one the contribution comes from F_i and F_{ia},

$$\frac{\partial F_{\text{Bethe}}}{\partial \widehat{\mu}_{a\to i}(x_i)} = \frac{\nu_{i\to a}(x_i)}{\sum_{x_i} \mu_{i\to a}(x_i)\,\hat{\mu}_{a\to i}(x_i)} - \frac{\prod_{b\in\partial i\setminus a} \hat{\mu}_{b\to i}(x_i)}{\sum_{x_i} \prod_{b\in\partial i} \hat{\mu}_{b\to i}(x_i)}.$$

If we set the two derivatives to zero we find

$$\widehat{\mu}_{a\to i}(x_i) \propto \sum_{\sim x_i} f_a(x_{\partial a}) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(x_j)$$
$$\mu_{i\to a}(x_i) \propto \prod_{b\in\partial i\setminus a} \widehat{\mu}_{b\to i}(x_i).$$

which are equivalent to the sum-product equations. Conversely it is easy to revert these calculations and show that the sum-product equations imply the stationnarity condition. □

11.4 Application to coding

We explained in Chapter 5 that the posterior measure used for MAP decoding is

$$\frac{1}{Z(\underline{h})}\prod_a \frac{1}{2}(1+\prod_{i\in\partial a} s_i)\prod_{i=1}^n e^{h_i s_i}.$$

where $s_i \in \mathcal{X} = \{-1,+1\}$. There are two types of kernel functions

$$f_i(s_i) = e^{h_i s_i}, \quad \text{and} \quad f_a(\{s_i, i\in\partial a\}) = \frac{1}{2}(1+\prod_{i\in\partial a} s_i), \tag{11.15}$$

associated to leaf checks and usual parity checks. An example with the corresponding factor graph is shown in figure 5.6.

The messages flowing on edges connecting variable nodes and parity checks can be parametrized as

$$\mu_{i\to a}(s_i) \propto e^{h_{i\to a}s_i}, \qquad \widehat{\mu}_{a\to i}(s_i) \propto e^{\widehat{h}_{a\to i}s_i} \propto 1 + s_i \tanh \widehat{h}_{a\to i}.$$

The messages flowing on edges connecting leaf checks and variable nodes are

$$e^{h_i s_i}, \qquad \prod_{a\in\partial i} e^{\widehat{h}_{a\to i}s_i} \propto \prod_{a\in\partial i}(1+s_i\tanh\widehat{h}_{a\to i}).$$

As pointed out above the normalization factors of teh messages cancel out in The Bethe free energy. This is why our parametrization only involves proportionality relations.

Replacing these messages in expressions (11.12)-(11.14) it is possible to perform exactly all sums over the spins, and express the Bethe free energy as function of $(\underline{h},\underline{\hat{h}}) = \{h_{i\to a}, \hat{h}_{a\to i}\}$. We give the main steps of this calculation. From (11.12) the contribution of variable nodes is

$$\begin{aligned} F_i &= \ln\Big\{\sum_{s_i=\pm 1} e^{h_i s_i}\prod_{a\in\partial i}(1+s_i\tanh\hat{h}_{a\to i})e^{h_i s_i}\Big\} \\ &= \ln\Big\{e^{h_i}\prod_{a\in\partial i}(1+\tanh\hat{h}_{a\to i}) + e^{-h_i}\prod_{a\in\partial i}(1-\tanh\hat{h}_{a\to i})\Big\}. \end{aligned} \tag{11.16}$$

From (11.13), for parity checks we have

$$F_a = \ln\Big\{\sum_{s_{\partial a}}\frac{1}{2}(1+\prod_{i\in\partial a} s_i)\prod_{i\in\partial a}(1+s_i\tanh h_{i\to a})\Big\}.$$

Observe that

$$\begin{aligned} \sum_{s_{\partial a}}\prod_{i\in\partial a}(1+s_i\tanh h_{i\to a}) &= \prod_{i\in\partial a}\sum_{s_i=\pm 1}(1+s_i\tanh h_{i\to a}) \\ &= 2^{|\partial a|} \end{aligned}$$

and

$$\sum_{s_{\partial a}} \prod_{i\in\partial a} s_i \prod_{i\in\partial a}(1+s_i\tanh h_{i\to a}) = \prod_{i\in\partial a}\sum_{s_i=\pm 1}(s_i+\tanh h_{i\to a}) = 2^{|\partial a|}\prod_{i\in\partial a}\tanh h_{i\to a}.$$

Now we compute the contribution of checks. The contribution of parity checks is

$$F_a = \ln\left\{\frac{1}{2}(1+\prod_{i\in\partial a}\tanh h_{i\to a})\right\} + |\partial a|\ln 2. \tag{11.17}$$

There is also a contribution from leaf check nodes that happens to be given by (11.16), and also happens to cancel with the contribution of edges connecting variable and leaf check nodes. There remains the contribution of edges connecting variable and parity check nodes

$$\begin{aligned} F_{ai} &= \ln\left\{\sum_{s_i=\pm 1}(1+s_i\tanh h_{i\to a})(1+s_i\tanh\hat{h}_{a\to i})\right\} \\ &= \ln\left\{1+\tanh h_{i\to a}\tanh\hat{h}_{a\to i}\right\} + \ln 2. \end{aligned} \tag{11.18}$$

The Bethe free energy is given by the sum of the three types of contributions (11.16), (11.17) and (11.18)

$$\begin{aligned} F_{\text{Bethe}}(\underline{h},\underline{\hat{h}}) = &\sum_i \ln\left\{e^{h_i}\prod_{a\in\partial i}(1+\tanh\hat{h}_{a\to i}) + e^{-h_i}\prod_{a\in\partial i}(1-\tanh\hat{h}_{a\to i})\right\} \\ &+\sum_a \ln\left\{\frac{1}{2}(1+\prod_{j\in\partial a}\tanh\ h_{j\to a})\right\} \\ &+\sum_{ai}\ln\left\{1+\tanh h_{i\to a}\tanh\hat{h}_{a\to i})\right\} \end{aligned} \tag{11.19}$$

As an exercise the reader can check that the stationary points of the Bethe functional satisfy the BP equations, in other words

$$\begin{cases} h_{i\to a} = h_i + \sum_{b\in\partial i\setminus a}\hat{h}_{b\to i} \\ \hat{h}_{a\to i} = \tanh^{-1}\left\{\prod_{j\in\partial a\setminus i}\tanh\ h_{j\to a}\right\} \end{cases}$$

We will see that the average over the channel outputs and the graph ensemble of the Bethe free energy allows to derive the so-called replica-symmetric (RS) formula for the average free energy[2]. It is known that for a large class of LDPC codes and BMS channels the RS free energy is equal to the exact free energy.

[2] The adjective "replica-symmetric" is due to historical reasons. indeed these formulas were first derived thanks to the so-called replica method which we do not cover in this course. The approach of the replica method is algebraic in nature but mathematically more mysterious.

In particular it allows to correctly predict the MAP noise threshold. In the next chapters we will derive the RS formula with the specific application of the BEC in mind, and partly prove that the RS formula is exact.

11.5 Application to compressive sensing

To do.

11.6 Application to K-SAT

Recall from Chapter 3 the partition function of K-SAT (at finite temperature) which counts the number of solutions.

$$Z = \sum_{s_1,\dots,s_n \in \{-1,+1\}^n} \prod_{a=1}^{M} \Big(1 - (1 - e^{-\beta}) \prod_{i \in a} (\frac{1 + s_i J_{ia}}{2}) \Big). \tag{11.20}$$

The Bethe free energy here serves as a first ansatz for $-(\beta n)^{-1} \ln Z$. Recall that for $\beta = +\infty$, Z counts the number of solutions. Thus as long as there exist at least one solution and $\ln Z$ is well defined for $\beta = +\infty$ one can also use the Bethe formula to write down an ansatz for the entropy of the uniform measure over solutions (the Boltzman entropy!).

To compute the Bethe free energy we replace the kernel function

$$f_a(\{x_i, i \in \partial a\}) = 1 - (1 - e^{-\beta}) \prod_{i \in a} (\frac{1 + s_i J_{ia}}{2}).$$

in (11.12)-(11.14) and use the parametrization (9.24) introduced in Chapter 9.5. Let $\partial_{J_{ia}} i$ the the set of checks connected to i by an edge such that $J_{ia} = -1$ (dashed) or $J_{ia} = 1$ (full). The resulting expressions are easily found to be

$$F_{\text{Bethe}}(\underline{h}, \underline{\hat{h}}) = \sum_i F_i(\{h_{j \to a}, j \in \partial a\}) + \sum_a F_a(\{\hat{h}_{b \to i}, i \in \partial b\}) \tag{11.21}$$

$$- \sum_{ia} F_{ia}(h_{i \to a}, \hat{h}_{a \to i}) \tag{11.22}$$

with

$$F_i = \ln\Bigg\{ \prod_{a\in\partial_- i} (1-\tanh\hat{h}_{a\to i}) \prod_{a\in\partial_+ i} (1+\tanh\hat{h}_{a\to i}) + \prod_{a\in\partial_- i} (1+\tanh\hat{h}_{a\to i}) \prod_{a\in\partial_+ i} (1-\tanh\hat{h}_{a\to i}) \Bigg\} \tag{11.23}$$

$$F_a = \ln\left\{ 1-(1-e^{-\beta}) \prod_{i\in\partial a} \frac{1-\tanh h_{i\to a}}{2} \right\} \tag{11.24}$$

$$F_{ai} = \ln\left\{ 1+\tan h_{i\to a} \tanh\hat{h}_{a\to i} \right\} \tag{11.25}$$

Again, the reader can easily check that the stationnary points of $F_{\rm Bethe}(\underline{h},\underline{\hat{h}})$ satisfy the BP equations presented in Chapter 9.5 ((9.27)-(9.31) are written down for $\beta=+\infty$).

In the next chapter we discuss an important application of these formulas. When $-\beta F_{\rm Bethe}[\underline{\xi},\underline{\hat{\xi}}]/n$ is averaged over the graph ensemble one get a specific prediction for the entropy of the K-SAT ensemble. This prediction is not consistent with rigorous upper bounds on the SAT-UNSAT threshold. This means that the Bethe formulas and the corresponding BP equations are not good enough to inform us on the SAT-UNSAT transition. But this is not the end of the story. We will see that it is necessary to further develop the approach taken in this chapter and wander into the cavity method.

12 Replica Symmetric Free Energy Functionals

The main idea behind density or state evolution analysis of message passing algorithms is to track their average behaviour. This allows to analyze their performance and derive their algorithmic (or dynamic) phase transition thresholds. But we also saw that one can guess the (static) phase transition threshold through a Maxwell construction. For example for coding, at least for the BEC, we defined an EXIT curve computable from DE, on which a Maxwell construction gives the MAP threshold. However we did not provide any clear general principle for deciding what are the correct variables[1]for which the Maxwell construction works. For the CW model the guess was quite trivial, for the BEC and compressive sensing it was less so. For K-SAT we have to postpone the discussion after the cavity method is introduced.

We will see in this chapter that by carrying the variational approach one step further we will be able to provide some clues for these questions. In particular we will be able to provide certain guiding lines determining the static phase transition threshold, and the variables on which the Maxwell construction works. In fact the variational approach allows to reformulate the Maxwell construction in a less ambiguous and useful way.

We have seen that the sum-product or BP equations are the stationarity conditions for the Bethe free energy. We will see in this chapter that the density and state evolution equations are the stationarity conditions of an averaged form of an averaged form of the Bethe free energy. This averaged form is called the *replica symmetric free energy functional*. The adjective "replica symmetric" mostly comes from historical reasons but, it has a meaning which we will explain once we have gone through the cavity method. We will explain how this functional allows to predict the algorithmic as well as static phase transition thresholds. Until recently this prediction was rigorously proved only in somewhat special cases or was supported by bounds. Recent proof techniques such as the interpolation method and spatial coupling have allowed to provide relatively simple and intuitive proofs in the cases of coding and compressive sensing. Such proof techniques are the subject of chapter 13. For K-SAT we will see that the predictions of the replica symmetric free energy functional are wrong. In-

[1] In physics parlance determining the "correct variables" for the description of a phase transition is part of a more general and deep problem, called the determination of the *order parameter* (see notes).

stead of being a curse this makes the subject even more fascinating. We will see in Chapter 15 that the correct thresholds and Maxwell constructions are given by pushing the notions of Bethe and replica free energy functionals "one level up". That these predictions are correct for K-SAT and other similar constraint satisfaction problems is still an open and alive problem.

We refrain from giving a completely general definition of the replica symmetric free energy functional because this immediately leads to cumbersome notations. Rather we directly treat our three paradigms in the next paragraphs. In fact each one has its own features and going through each of them allows to cover most essential cases.

12.1 Coding

We first discuss the general definition of the replica symmetric free energy functional for the regular Gallager (l, r) ensemble over a BMS channel $p_{Y|X}$, and then specialize to the case of the BEC where the functional simply becomes a function of a real variable. Recall the notation $c(.)$ for the distribution of half-loglikehod ratios $h(y) = \frac{1}{2} \ln p_{Y|X}(y|1)/p_{Y|X}(y|-1)$.

Replica symmetric functionals for BMS channels

The main idea is to pretend that in expression (11.19) the messages $h_{i\to a}$, are iid random variables distributed according to a trial distribution x(.), and that $\hat{h}_{a\to i}$ are dependent random variables defined through the BP equation

$$\hat{h}_{a\to i} = \tanh^{-1}\Big\{ \prod_{j\in\partial a\setminus i} \tanh\ h_{j\to a} \Big\}$$

Then one averages (11.19) which yields a functional of $x(.)$.

Let us give the formal definition. Here x(.) is a fixed trial probability distribution over $\mathbb{R}$. Pick r iid copies of $H \sim$ x(.), and call them H_k, $k = 1, \dots r$. Let

$$\hat{H} = \tanh^{-1}\{ \prod_{k=1}^{r} \tanh\ H_j \} \tag{12.1}$$

Pick l iid copies $\hat{H}_k$, $k = 1, \dots l$. Let

$$\begin{aligned} f(h, \underline{H}, \underline{\hat{H}}) = &\ln\Big\{ e^{h} \prod_{k=1}^{l} (1 + \tanh \hat{H}_k) + e^{-h} \prod_{a=1}^{k} (1 - \tanh \hat{H}_k) \Big\} \\ &+ \frac{l}{r} \ln \frac{1}{2}\Big\{ 1 + \prod_{k=1}^{r} \tanh H_k \Big\} - l \ln\Big\{ 1 + \tanh H \tanh \hat{H} \Big\} \end{aligned}$$

The RS free energy functional is defined as:

$$f_{\rm RS}[\mathrm{x}(.)] = \mathbb{E}[f(h, \underline{H}, \hat{\underline{H}}]$$

where the expectation is with respect to $h \sim c(.)$ and $\underline{H} \sim \mathrm{x}(.)$ (and $\hat{\underline{H}} \sim \hat{\mathrm{x}}(.)$ the induced distribution that depends on $\mathrm{x}(.)$). For an irregular LDPC ensemble (l, r) are random and one has an extra average over their distribution. The RS entropy functional is defined as

$$h_{\rm RS}[\mathrm{x}(.)] = -f_{\rm RS}[\mathrm{x}(.)] + \mathbb{E}[h] \tag{12.2}$$

The motivation for introducing the functional $h_{\rm RS}[\mathrm{x}(.)]$ will become clear in the next paragraph (see equ. (12.4)).

How to determine the MAP threshold

Recall that the (true) average free energy is given by the thermodynamic limit $-\lim_{n\to+\infty} \mathbb{E}[\ln Z]/n$ where Z is the partition function for coding (3.10). The replica symmetric formula states that

$$-\lim_{n\to+\infty} \frac{1}{n}\mathbb{E}[\ln Z] = \inf_{\mathrm{x}\in\mathcal{S}} f_{\rm RS}[\mathrm{x}(.)] \tag{12.3}$$

In this formula $\mathcal{S}$ is the space of (Nishimori) symmetric distributions (see Chapter 3). That the infimum can be restricted to this space of distributions is a special feature coming from channel symmetry. Such formulas relating a free energy to a replica functional have been long standing conjectures since the mid 70's in the field of spin glass models (on sparse and complete graph models) but much progress have been made in the last fifteen years towards their proofs. The present one is a case where we have a partial proof that combines interpolation methods with spatial coupling. This will be sketched in the subsequent chapter. In the next sub-section we take a closer look at (12.3) for the BEC, and show that it is equivalent to the Maxwell construction.

The MAP threshold is defined as the smallest ϵ such that $\liminf_{n\to\infty} \mathbb{E}[H(\underline{X} \mid \underline{Y}(\epsilon))/n] > 0$ (see definition 10.2). Recall also the relationship (3.43)

$$\frac{1}{n}\mathbb{E}[H(\underline{X} \mid \underline{Y}(\epsilon))] = -\frac{1}{n}\mathbb{E}[\ln Z] + \mathbb{E}[h] \tag{12.4}$$

Equation (12.3) has two consequences. One can replace $\liminf$ by $\lim$ in the definition of the MAP threshold, but more importantly,

$$\lim_{n\to+\infty} \frac{1}{n}\mathbb{E}[H(\underline{X} \mid \underline{Y}(\epsilon))] = \sup_{\mathrm{x}\in\mathcal{S}} h_{\rm RS}[\mathrm{x}(.)] \tag{12.5}$$

and

$$\epsilon_{\rm MAP} = \inf\{\epsilon \in [0,1] : \sup_{\mathrm{x}\in\mathcal{S}} h_{\rm RS}[\mathrm{x}(.)] > 0\}$$

In order to concretly calculate the MAP threshold one has to solve the variational problem consisting in minimizing (or maximizing) the replica symmetric

free energy (or entropy). It is easy to write down the stationary point conditions (homework) and one finds the density evolution fixed point equations (see Equ. (7.29)-(7.30))

$$\mathrm{x} = c \otimes \hat{\mathrm{x}}^{\otimes(l-1)}, \qquad \hat{\mathrm{x}} = \mathrm{x}^{\oplus(r-1)} \tag{12.6}$$

Remark that $\hat{\mathrm{x}}(.)$ is the distribution of $\hat{H}$ in Equ. (12.1). This is not surprising: the stationary points of the Bethe free energy are given by the BP equations and the stationary points of the replica functional are given by the density evolution equations. Once stationary points, i.e. fixed points of (12.1) have been found one selects the one that yields the largest $h_{\mathrm{RS}}[\mathrm{x}(.)]$ (or smallest $f_{\mathrm{RS}}[\mathrm{x}(.)]$) and determines ϵ_{MAP}. Since in practice fixed points are found by iterative methods, it is fortunate that we only need to find *stable* fixed points. Indeed the maximum of $h_{\mathrm{RS}}[\mathrm{x}(.)]$ (or minimum of $f_{\mathrm{RS}}[\mathrm{x}(.)]$) is necessarily a stable fixed point.

But that is not all. We already know that allow to determine the BP threshold. The BP threshold is the smallest noise for which a non-trivial fixed point is reached under iterations initialized with $\mathrm{x}(.) = c(.)$. Therefore this information is also contained in the RS functional. The BP threshold is the smallest noise such that the RS functional has a non trivial stationary point.

To summarize, the RS functional contains all the information we want. In particular it allows to deduce the DE equations. To determine the BP threshold it suffices to solve the DE equation. But, to evaluate the MAP threshold we have to solve the DE equations *and* to evaluate corresponding largest RS entropy or smallest RS free energy.

In the next paragraph we specialize this discussion to the case of the BEC. This will also allow us to derive the Maxwell construction in a more principled way.

12.2 Explicit Case of the BEC

A bit transmitted through the BEC is either perfectly transmitted with probability ϵ or earsured with probability $1-\epsilon$. This implies that $c(h) = \epsilon\delta(h) + (1-\epsilon)\delta_\infty(h)$, and that we can restrict the RS functionals to distributions parametrized as

$$\mathrm{x}(H) = x\delta(H) + (1-x)\ \delta_\infty(H)$$

where x is the erasure probabaility emanting from variables. This also implies that $\hat{\mathrm{x}}(\hat{H}) = \hat{x}\delta(\hat{H}) + (1-\hat{x})\delta_\infty(\hat{H})$ with $\hat{x} = 1-(1-x)^{r-1}$ the erasure probability emanating from checks. With this parametrization one can compute each term in the RS expression for the free energy. One easily finds the contributions of "check nodes"

$$\mathbb{E}[\ln \frac{1}{2}(1 + \prod_{k=1}^{r} \tan H_k)] = (1-x)^r \ln 2 - \ln 2$$

and "edges"

$$\mathbb{E}[\ln(1+\tan H \tan \hat{H})] = (1-x)(1-\hat{x})\ln 2$$

For the BEC, one should include the term $\mathbb{E}[h]$ in (12.2) directly in the contribution of "variable nodes" in order to avoid working with infinite quantities. One finds

$$\begin{aligned}
&\mathbb{E}[\ln(\prod_{k=1}^{l}(1+\tanh\hat{H}_k)+e^{-2h}\prod_{k=1}^{l}(1-\tanh\hat{H}_k))] \\
&= (1-\epsilon)\sum_{e=0}^{l}\binom{l}{e}\hat{x}^e(1-\hat{x})^{l-e}\ln 2^{l-e} + \epsilon\sum_{e=0}^{l-1}\binom{l}{e}\hat{x}^e(1-\hat{x})^{l-e}\ln 2^{l-e} \\
&\quad + \epsilon\binom{l}{l}\hat{x}^l(1-\hat{x})^{l-l}\ln 2 \\
&= \sum_{e=0}^{l}\binom{l}{e}\hat{x}^e(1-\hat{x})^{l-e}(l-e)\ln 2 + \epsilon\hat{x}^l\ln 2 \\
&= (1-\hat{x})\sum_{e=0}^{l}\hat{x}^e\frac{d}{dy}y^{l-e}\mid_{y=1-\hat{x}}\ln 2 + \epsilon\hat{x}^l\ln 2 \\
&= (1-\hat{x})\frac{d}{dy}(\hat{x}+y)^l\mid_{y=1-\hat{x}}\ln 2 + \epsilon\hat{x}^l\ln 2 \\
&= l(1-\hat{x})\ln 2 + \epsilon\hat{x}^l\ln 2
\end{aligned}$$

Putting these results together one finds the replica symmetric entropy function for the BEC

$$\frac{h_{\mathrm{RS}}(x;\epsilon)}{\ln 2} = (\frac{l}{r}-l)(1-x)^r + l(1-x)^{r-1} + \epsilon(1-(1-x)^{r-1})^l - \frac{l}{r}$$

According to (12.3) the conditionnal entropy is given by

$$\lim_{n\to+\infty}\frac{1}{n}\mathbb{E}[H(\underline{X}\mid\underline{Y}(\epsilon))] = \max_{0\le x\le 1} h_{\mathrm{RS}}(x;\epsilon) \tag{12.7}$$

and the MAP threshold can be calculated from $\epsilon_{\mathrm{MAP}} = \inf\{\epsilon : \max_{0\le x\le 1} h_{\mathrm{RS}}(x;\epsilon) > 0\}$. It is immediate to check that the stationnary points are given by the usual density evolution fixed point equation $x = \epsilon(1-(1-x)^{r-1})^{l-1}$.

As pointed out before, the function $-h_{\mathrm{RS}}$ contains all the information about the BP and MAP thresholds, so it is very useful to have an idea of the shape of the RS function. Figure **??** shows $-h_{\mathrm{RS}}$ as a function of x, for various values of ϵ.[2] We prefer to plot *minus* the RS entropy function[3] because this quantity is the free energy (up to an irrelevant term) and is better suited to make the physical analogies more transparent. For all ϵ there is a trivial minimum at $x = 0$, which

[2] This plot is generic only for regular ensembles with $l \ge 3$. Irregular ensembles can have a richer behavior and the corresponding discussion is more complicated. The case $l = 2$ is somewhat special because $\epsilon_{\mathrm{BP}} = \epsilon_{\mathrm{MAP}}$.

[3] To avoid any confusion let us stress that there is no reason why $h_{\mathrm{RS}}(x)$ should be non-negative. It is only $\max_{0\le x\le 1} h_{\mathrm{RS}}(x)$ that has to be non-negative.

is also the trivial stable fixed point of DE. For $\epsilon < \epsilon_{\text{BP}}$ this minimum is unique (hence global). At $\epsilon = \epsilon_{\text{BP}}$ the function develops a flat inflexion point and a second (local) minimum as well as a (local) maximum branch of. The local minimum is the stable non-trivial fixed point of density evolution, $x_{\text{st}}(\epsilon)$, and the local maximum is the unstable fixed point $x_{\text{un}}(\epsilon)$. As one increases ϵ further the local minimum at $x_{\text{st}}(\epsilon)$ decreases until it touches the horizontal axis for ϵ_{MAP}. At this threshold value there are two global minima, $h_{\text{RS}}(0;\epsilon_{\text{MAP}}) = h_{\text{RS}}(x_{\text{st}}(\epsilon_{\text{MAP}};\epsilon_{\text{MAP}})$. Finally, $\epsilon > \epsilon_{\text{MAP}}$ it is $x_{\text{st}}(\epsilon)$ that becomes the unique global minimum.

To summarize, one should retain from this discussion that the RS function contains all the information we want. The BP threshold is found by searching values of ϵ where the function develops flat inflexion points, and the MAP threshold is found by looking at values of ϵ where the two minima are at the same height. The reader should go back to the exact solution of the CW model in Chapter 4 and notice the intimate structural analogies with the present situation. The CW free energy is given by a variational problem $\min_{-1\leq m\leq 1} f(m)$ whose solutions determine both the phase transition ("MAP") threshold $h = 0$ and the spinodal ("BP") points $\pm h_{\text{sp}}$.

We conclude this paragraph by casting (12.7) in an equivalent form. For $\epsilon > \epsilon_{\text{MAP}}$ the derivative of the right hand side of $\max_{0\leq x\leq 1} h_{\text{RS}}(x;\epsilon)$ equals

$$\begin{aligned}\frac{d}{d\epsilon}h_{\text{RS}}(x_{\text{st}};\epsilon) &= \frac{\partial}{\partial\epsilon}h_{\text{RS}}(x_{\text{st}};\epsilon) + \frac{\partial}{\partial x}h_{\text{RS}}(x_{\text{st}};\epsilon)\frac{dx_{\text{st}}}{d\epsilon}\\ &= \frac{\partial}{\partial\epsilon}h_{\text{RS}}(x_{\text{st}};\epsilon)\end{aligned}$$

The second equality is valid because x_{st} is a stationnary point of h_{RS} and $\frac{dx_{\text{st}}}{d\epsilon}$ is finite for $\epsilon \in]\epsilon_{\text{MAP}}, 1]$. This last point can be checked rather explicitly for the BEC but for other channels this is much more difficult. We obtain

$$\frac{d}{d\epsilon}\lim_{n\to+\infty}\frac{1}{n}\mathbb{E}[H(\underline{X}\mid\underline{Y}(\epsilon))] = \begin{cases}0, \epsilon < \epsilon_{\text{MAP}}\\ \frac{\partial}{\partial\epsilon}h_{\text{RS}}(x_{\text{st}}(\epsilon);\epsilon) = (1-(1-x_{\text{st}}(\epsilon))^{r-1})^{l}, \epsilon > \epsilon_{\text{MAP}}\end{cases}$$

Note that for $\epsilon > \epsilon_{\text{MAP}}$ the derivative of the conditional entropy coincides with the EXIT curve introduced somewhat arbitrarily in Chapter 10.

12.3 Back to the Maxwell Construction

The Maxwell construction identifies the MAP threshold ϵ_{MAP} with the area threshold ϵ_{A} on the EXIT curve. We are now in a position to show that this identity is equivalent to the equality of the two minima of $-h_{\text{RS}}(x;\epsilon)$ when $\epsilon = \epsilon_{\text{MAP}}$. Apart from the conceptual importance of this result, this shows that for coding a proof of the Maxwell construction boils down to the one of the RS formula.

Consider $\epsilon > \epsilon_{\text{BP}}$. The non-trivial minimum and maximum of $-h_{\text{RS}}(x;\epsilon)$, namely $x_{\text{st}}(\epsilon)$ and $x_{\text{un}}(\epsilon)$, form a curve in the (ϵ, x)-plane. This curve is precisely $(\epsilon(x), x)$ where $\epsilon(x) = x/(1-(1-x)^{r-1})^{l-1}$ (since the stationary points

of $-h_{\rm RS}(x;\epsilon)$ are given by DE). Now consider the path starting from $(\epsilon_{\rm MAP},0)$ to $(+\infty,0)$ on the horizontal axis and then along the curve till $(\epsilon(x),x)$ for some x. Look at the total change in RS entropy along this path. We have

$$\begin{aligned} h_{\rm RS}(x;\epsilon(x)) - h_{\rm RS}(0;\epsilon_{\rm MAP}) &= \int_{\rm path} dh_{\rm RS} = \int_0^x dx \frac{d}{dx} h_{\rm RS}(x;\epsilon(x)) \\ &= \int_0^x dx (\frac{\partial}{\partial x} h_{\rm RS}(x;\epsilon(x)) + \epsilon'(x) \frac{\partial}{\partial \epsilon} h_{\rm RS}(x;\epsilon(x))) \\ &= \int_0^x dx \epsilon'(x) \frac{\partial}{\partial \epsilon} h_{\rm RS}(x;\epsilon(x)) \\ &= \int_0^x dx \epsilon'(x) (1-(1-x)^{r-1})^l \end{aligned}$$

The last integral is recognized as the trial entropy $P(x)$, the area under the EXIT curve $(\epsilon(x),(1-(1-x)^{r-1})^l)$ (see (10.7)).

Let us highlight the main points of this discussion. The natural definition of the EXIT curve in parametric form is,

$$(\epsilon(x), \frac{\partial}{\partial \epsilon} h_{\rm RS}(x;\epsilon(x))).$$

and satisfies

$$h_{\rm RS}(x;\epsilon(x)) - h_{\rm RS}(0;\epsilon_{\rm MAP}) = \int_0^x dx \epsilon'(x) \frac{\partial}{\partial \epsilon} h_{\rm RS}(x;\epsilon(x)).$$

The right hand side is the area under the EXIT curve and the left hand side is the corresponding change in entropy. On one hand the area threshold is by definition $\epsilon_{\rm A} = \epsilon(x_{\rm A})$ such that the area under the EXIT curve vanishes, and on the other hand the MAP threshold is $\epsilon_{\rm MAP} = \epsilon(x_{\rm MAP})$ such that the minima of $-h_{\rm RS}$ are at the same height $h_{\rm RS}(x_{\rm MAP};\epsilon(x_{\rm MAP})) - h_{\rm RS}(0;\epsilon_{\rm MAP}) = 0$. Therefore these two thresholds are identical.

12.4 Compressive Sensing

Write RS free energy (can be derived by integrating out state evolution). Illustrate thresholds it predicts. Discuss that RS is exact. Do it for Lasso or for know prior case ?

12.5 K-SAT

Recall that in Chapter 11 we gave the Bethe expression for the free energy of K-SAT. From this expression one also gets a Bethe formula for the entropy density. There is natural RS functional associated to this formula, which leads to a natural conjecture for the entropy density. We will see that, contrary to

coding and compressive sensing, the conjecture cannot be fully correct.[4] This is one of the main motivations for developping a better theory, namely the cavity method.

The construction of the natural RS functional for K-SAT proceeds like in the coding case: one takes as a starting point the Bethe expression (11.21) and treats the messages $h_{i\to a}$ as independent random variables distributed according to a trial distribution Q(.). The message passing equation (9.31),

$$\hat{h}_{a\to i} = -\frac{1}{2}\ln\Big\{1 - \prod_{j\in\partial a\setminus i}\frac{1-\tanh h_{j\to a}}{2}\Big\} \tag{12.8}$$

induces the distribution $\hat{Q}(.)$. In the coding case we discussd the case of regular Gallager (l, r) ensembles. One difference here is that while the check nodes have degree K, the variable node degrees are (asymtotically) Poisson distributed with average degree αK.

Here is the formal definition of the RS functional for the entropy. Fix a trial distribution $Q(.)$ on $\mathbb{R}$. Pick K iid copies of the random variable $H \sim Q(.)$. Call them $H_1, \ldots, H_K$. Define the random variable

$$\hat{H} = -\frac{1}{2}\ln\Big\{1 - \prod_{k=1}^{K-1}\frac{1-\tanh H_k}{2}\Big\}. \tag{12.9}$$

Pick two Poisson distributed integers p and q with average αK, and pick $p+q$ iid copies of $\hat{H}_k$, $k = 1, \ldots, p+q$. Let

$$\begin{aligned} s(\underline{H}, \underline{\hat{H}}, p, q) = \ln\Big\{ &\prod_{k=1}^{p}(1-\tanh\hat{H}_k)\prod_{k=p+1}^{p+q}(1+\tanh\hat{H}_k) \\ &+ \prod_{k=1}^{p}(1+\tanh\hat{H}_k)\prod_{k=p+1}^{p+q}(1-\tanh\hat{H}_k)\Big\} \\ &+ \ln\Big\{1 - \prod_{k=1}^{K}\frac{1-\tanh H_k}{2}\Big\} \\ &- \ln\Big\{1 + \tanh H \tanh\hat{H}\Big\} \end{aligned}$$

The RS entropy functional is defined as

$$s_{\mathrm{RS}}(Q(.)) = \mathbb{E}[s(\underline{H}, \underline{\hat{H}}, p, q)] \tag{12.10}$$

where the expectation is over all random variables $p, q, \underline{H}, \underline{\hat{H}}$.

The replica symmetric prescription for computing the entropy density is to

[4] While in coding and compressive sensing it is quite hard to prove the RS formulas are exact, in K-SAT it is relatively easier to prove that they cannot be correct or at least fully correct.

take

$$s_{\rm RS}(\alpha) \equiv \sup_{Q(.)} s_{\rm RS}(Q(.))$$

The stationnary points of (12.10) yields an integral equation for $Q(.)$. Similarly to coding, this can be split in two integral equations linking $Q(.)$ and $\hat{Q}(.)$ where $\hat{Q}(.)$ is the distribution of $\hat{H}$. These two equations can equivalently be written as (homework)

$$H \stackrel{d}{=} \sum_{k=1}^{p} \hat{H}_k - \sum_{k=p+1}^{p+q} \hat{H}_k, \quad \hat{H} \stackrel{d}{=} -\frac{1}{2} \ln\Big\{1 - \prod_{k=1}^{K-1} \frac{1-\tanh H_k}{2}\Big\}.$$

where $\stackrel{d}{=}$ means equality in distribution. The second relation is of course the same as (12.9), and you will derive the first one in the homeworks. These equations can be solved numerically (e.g. by the population dynamics method of homework). This allows to find the maximizer of the RS functional and compute $s_{\rm RS}(\alpha)$.[5] Figure **??** shows that $s_{\rm RS}(\alpha)$ for $K = 3$. the function decreases as the clause density increases, and vanishes at $\alpha \approx 4.677$. Thus the present replica symmetric analysis predicts that there exist exponentially many solutions at least until this value of α, and that in particular the SAT-UNSAT threshold should be larger. However it is known that this is wrong. For example in problem **??** we guide you through the proof of $\alpha_{\text{sat-unsat}} \leq 4.666$ for $K = 3$. In fact, as we will see in Chapter 15 the cavity method proposes that the RS formula is exact till a threshold value $\alpha_c < \alpha_{\text{sat-unsat}}$, called the "condensation threshold", and that another one called RSB formula[6] holds in the range $\alpha_c < \alpha < \alpha_{\text{sat-unsat}}$. At the condensation threshold there is a genuine phase transition: $\lim n^{-1}\mathbb{E} \ln Z$ is not analytic, in other words the same (analytic) formula cannot hold both above and below α_c. For $K = 3$ we have $\alpha_c \approx 3.86$ and $\alpha_{\text{sat-unsat}} \approx 4.26$. None of these claims have been proven so far.

12.6 Notes

A few words about the concept of order paramter. Like for many physical concepts there is no rigid definition, and finding the correct order parameter is an art validated by experiment. Depending on the problem at hand this can seem more or less obvious like in fluids (the volume per particle) or in magnetism (the magnetization), but can be much more subtle like in superconductivity (the "wave function" of Cooper pairs). The Higgs field is the order parameter associated to the electroweak phase transition that occurred at an early epoch of the universe. The recently discovered Higgs bosons are elementary excitations of this

[5] Note the global maximum necessarily corresponds to a stable fixed point and therefore iterative methods to solve the density evolution equations can find it. Similarly global minima of the free energy necessarily correspond to stable fixed point of density evolution.

[6] As we will see "B" stands for broken.

field, much like spin flips are elementary excitations associated to magnetization. As we will see K-SAT is one of these problems for which the guess of the order parameter requires a stretch of imagination: probability distributions of random probability distributions.

Problems

12.1 *RS analysis for K-SAT* Derive the density evolution equations for K-SAT. Use population dynamics (as seen in homeworks of Chapter **??**) to compute the RS prediction for $\alpha_{\text{sat-unsat}}$.

12.2 *Upper bounds on the SAT-UNSAT threshold.* Upper bounds for the SAT-UNSAT threshold, we call it α_s, are usually derived by counting arguments. The first exercise develops the simplest such argument. In the second exercise you will study a more subtle counting argument which leads to an important improvement[7]. This method can be further refined and has led to better bounds.

An assignment is a tuple $\underline{x} = (x_1, \ldots, x_n)$ where $x_i = 0, 1$ of n variables. The total number of possible clauses with k variables is equal to $2^k \binom{n}{k}$. A random formula F is constructed by picking, with replacement, uniformly at random, m clauses. Thus there are $(2^k \binom{n}{k})^m$ possible formulas.

We set $m = \alpha n$ and think of n and m as tending to ∞ with α fixed. This is the regime displaying a SAT-UNSAT threshold.

It is useful to keep in mind that $\mathbb{P}[A] = \mathbb{E}[1(A)]$ where $1(A)$ is the indicator function of event A. In what follows probabilities and expectations are with respect to the random formulas F.

12.3 Crude upper bound by counting all satisfying assignments Let $S(F)$ be the set of all assignments satisfying F and let $|S(F)|$ be its cardinality. Since F is a random formula, $|S(F)|$ is an integer valued random variable.

a) Show the Markov inequality $\mathbb{P}[F \text{ satisfiable}] \leq \mathbb{E}[|S(F)|]$.

b) Fix an assignement $\underline{x}$. Show that $\mathbb{P}[\underline{x} \,\text{satisfies}\, F] = (1 - 2^{-k})^m$. Then deduce that

$$\mathbb{E}[|S(F)|] = 2^n (1 - 2^{-k})^m.$$

c) Deduce the upper bound

$$\alpha_s < \frac{\ln 2}{|\ln(1 - 2^{-k})|}.$$

For $k = 3$ this yields $\alpha_s < 5.191$.

12.4 Bound by counting a restricted set of assignments] We define the set $S_{\text{m}}(F)$ of *maximal* satisfying assignments as follows. An assignment $\underline{x} \in S_{\text{m}}(F)$ iff:

- $\underline{x}$ satisfies F,

[7] by Kirousis, Kranakis, Krizanc and Stamatiou, *Approximating the Unsatisfiability Threshold of Random Formulas*, in Random Struct and Algorithms (1998).

- for all i such that $x_i = 0$ (in $\underline{x}$), the *single flip* $x_i \to 1$ yields an assignment - call it $\underline{x}^i$ - that *violates* F.

a) Show that if F is satisfiable then $S_m(F)$ is not empty. *Hint*: proceed by contradiction.

b) Show as in the first exercise the Markov inequality $\mathbb{P}[F \text{ satisfiable}] \leq \mathbb{E}[|S_m(F)|]$

c) Show that

$$\mathbb{E}[|S_m(F)|] = (1-2^{-k})^m \sum_{\underline{x}} \mathbb{P}[\cap_{i:x_i=0} (\underline{x}^i \text{ violates } F) \mid \underline{x} \text{ satisfies } F].$$

d) Fix $\underline{x}$. The events $E_i \equiv (\underline{x}^i \text{ violates } F)$ are negatively correlated, i.e

$$\mathbb{P}[\cap_{i:x_i=0} E_i \mid \underline{x} \text{ satisfies } F] \leq \prod_{i:x_i=0} \mathbb{P}[E_i \mid \underline{x} \text{ satisfies } F]$$

For the full proof which uses a correlation inequality (of FKG type) we refer to the reference given above. Here is a rough intuition for the inequality. First note that if $x_i = 0$ and $\underline{x}^i$ violates F, there must be some set S_i of clauses (in F) that are satisfied *only* by this variable $x_i = 0$ (this set might contain only one clause). This restricts the possible formulas contributing to the event E_i. Second note that sets S_i, S_j corresponding to different such variables $x_i = 0$, $x_j = 0$ must be *disjoint*. This "repulsion" between the sets S_i and S_j puts even more restrictions on the possible formulas, compared to a hypothetical situation where the events (and thus the sets S_i and S_j) would have been independent.

e) Now show that

$$\mathbb{P}[E_i \mid \underline{x} \text{ satisfies } F] = 1 - \left(1 - \frac{\binom{n-1}{k-1}}{(2^k-1)\binom{n}{k}}\right)^m.$$

Hint: note that in the event E_i there must be at least one clause containing $x_i = 0$ and containing other variables that do not satisfy it.

f) Deduce from the above results that $\lim_{n\to 0} \mathbb{P}[F \text{ satisfiable}] = 0$ as long as α satisfies

$$(1-2^{-k})^\alpha (2 - e^{-\frac{\alpha k}{2^k-1}}) < 1.$$

The improvement compared with the first exercise resides in the factor $e^{-\frac{\alpha k}{2^k-1}}$. A numerical evaluation for $k = 3$ yields the bound $\alpha_s < 4.667$.

13 Interpolation Method

13.1 Guerra bounds for Poissonian degree distributions

13.2 RS bound for coding

13.3 RS and RSB bounds for K sat

13.4 Application to spatially coupled models: invariance of free energy, entropy ect...

14 Spatial Coupling and Nucleation Phenomenon

So far we have seen that a variety of problems can be phrased in a natural way in terms of marginalizing a highly-factorized function. Message-passing algorithms are then the logical choice to accomplish this marginalization and we have seen how such algorithms perform in the thermodynamic limit.

Perhaps more surprisingly, we saw that the same quantities which were important for the analysis of the suboptimal message-passing algorithm reappeared when we looked at the seemingly more fundamental question of determining static thresholds, like the MAP threshold or the SAT/UNSAT threshold. The Maxwell construction is a graphical representation of this phenomenon.

We will now tie these two threads together. We will discuss a generic construction, called spatial coupling, which can be applied to a wide range of graphical models. The idea is to take many copies of a graphical model, to place them next to each other on a line and then to start connecting these models by "exchanging edges" in such a way that the local structure of the graphical model remains unchanged but that globally we create a larger graphical model which forms a one-dimensional chain. If in addition we impose suitable conditions at the boundaries of the model, this larger graphical model behaves very well under message-passing. Roughly speaking, the performance of the large spatially-coupled model under message-passing (in terms of the resulting threshold) is as good as if we had done optimal processing on the original graphical model.

For the most part we will only discuss the phenomenon but we will not give proofs. We will see how this phenomenon has again a nice physical interpretation. In fact – it is what is called the *nucleation* phenomenon in physics. Nucleation explains amongst other things how crystals grow, starting with a *seed* or *nucleus*.

We will discuss two important consequences of the nucleation phenomenon.

First, whenever we are in control of the graphical structure and the size of the graph is not very crucial, it is natural to construct the graph according to the above recipe. This results in graphs which are well suited for message-passing processing and give very good performance. E.g., for the coding problem this construction makes it possible to design codes which, under BP decoding, are not only provably capacity-achieving for a particular channel, but are in fact universally so, i.e., they are capacity-achieving for the whole class of BMS channels. A similar construction is possible for the compressive sensing problem.

There is a second, equally important application of the idea, namely to use

spatial coupling as a proof technique. Consider e.g. the case of the K-SAT problem. Also in this case we can use spatial coupling. This means we can construct spatially-coupled K-SAT formulas, and it is easier to find satisfiable solutions for such formulas than for the uncoupled ones. But what is the use of this? In coding, we were in charge of picking the code, and so we can pick coupled ones. The same thing applies for compressive sensing. We do not have the same degree of freedom for the constraint satisfaction problem where the formula is given to us. The idea is the following. If we are able to analyze the performance of a message-passing algorithm on coupled formulas then we can use the so-called *interpolation* method to show that this algorithmic threshold is also a lower bound on the SAT/UNSAT threshold of the uncoupled ensemble. So in this case we use spatial coupling only as a thought experiment. Indeed, the same method can be used in the context of coding to prove that the MAP threshold of the uncoupled formula is at least as large as the area threshold. Together with the upper bound on the MAP threshold which we derived in Chapter 10 this shows that the MAP threshold of the uncoupled ensemble is equal to the area threshold.

In the remainder of the chapter we go over our three running examples. In each case we describe the construction, the performance of the coupled system, as well as the consequences for our problem at hand.

14.1 Coding

There are many possible ways of constructing coupled graphical models from uncoupled ones. The "saturation phenomenon" is fairly robust with respect to the exact way of how we construct coupled models. So the difference lies mostly in how convenient the construction is either from a practical perspective or for the purpose of proofs. We present below two generic ways to achieve the spatial coupling. We start with the "protograph" construction. It has a very good performance and the additional structure is well suited for implementations. Our second construction is a "random" model. This model is well suited for proofs. Indeed, in the sequel we exclusively use the random model when it comes to showing plots and to formulating theorems.

Protograph Construction

To start, consider a protograph of a standard $(3, 6)$-regular ensemble (see [**?**, **?**] for the definition of protographs). It is shown in Figure 14.1. There are two variable nodes and there is one check node. Let M denote the number of variable nodes at each position. For our example, $M = 100$ means that we have 50 copies of the protograph so that we have 100 variable nodes at each position. For all future discussions we will consider the regime where M tends to infinity.

Next, consider a collection of $(2L+1)$ such protographs as shown in Figure 14.2. These protographs are non-interacting and so each component behaves just like

Figure 14.1 Protograph of a standard $(3, 6)$-regular ensemble.

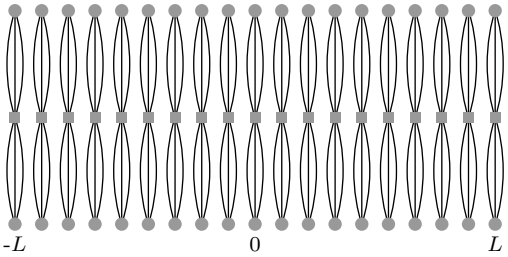


Figure 14.2 A chain of $(2L + 1)$ protographs of the standard $(3, 6)$-regular ensembles for $L = 9$. These protographs do not interact.

a standard $(3, 6)$-regular component. In particular, the belief-propagation (BP) threshold of each protograph is just the standard threshold, call it $\epsilon^{\mathrm{BP}}(d_v = 3, d_c = 6)$. Slightly more generally: start with an $(d_v, d_c = kd_v)$-regular ensemble where d_v is odd so that $\lfloor l/2 \rfloor = (d_v - 1)/2 \in \mathbb{N}$.

We will now "coupled" these copies. To achieve this coupling, connect each protograph to $\lfloor l/2 \rfloor$ protographs "to the left" and to $\lfloor l/2 \rfloor$ protographs "to the right." This is shown in Figure 14.3 for the two cases $(d_v = 3, d_c = 6)$ and $(d_v = 7, d_c = 14)$.

Note that $\lfloor l/2 \rfloor$ extra check nodes are added on each side to connect the "overhanging" edges at the boundary. This reduces the rate of this ensemble from $1 - \frac{d_v}{d_c} = \frac{k-1}{k}$ to

$$\begin{aligned} R(d_v, d_c = kd_v, L) &= \frac{(2L + 1) - (2(L + \lfloor l/2 \rfloor) + 1)/k}{2L + 1} \\ &= \frac{k - 1}{k} - \frac{2\lfloor l/2 \rfloor}{k(2L + 1)}, \end{aligned}$$

Note that this rate loss decreases with the length of the chain. Therefore, in practice we want to pick the length not too small. Of course, this increases the blocklength and so there is a natural trade-off between the block length and the rateloss due to the boundary.

In the above construction we had to assume that d_v was odd and also the "width" of the connection was linked directly to the degree d_v. In this case the construction leads to the very symmetric ensemble. It is not very hard to

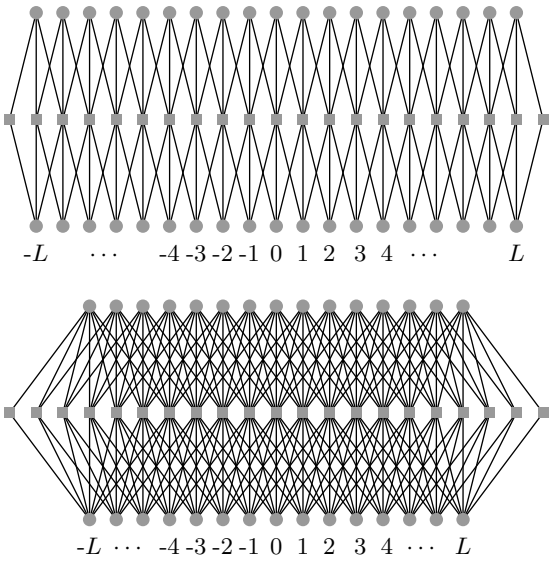


Figure 14.3 Two coupled chains of protographs with $L = 9$ and $(d_v = 3, d_c = 6)$ (top) and $L = 7$ and $(d_v = 7, d_c = 14)$ (bottom), respectively.

extend this construction to cases where d_v is even and so that "width" of the connection is no longer directly linked to d_v. But instead of following this path, let us directly go to another extreme and introduce an ensemble which includes much more randomness.

Random Construction

For the purpose of analysis, the following random ensemble is much betters suited. Let us assume that $d_c \geq d_v$, so that the ensemble has a non-trivial design rate.

We assume that the variable nodes are at positions $[-L, L]$, $L \in \mathbb{N}$. At each position there are M variable nodes, $M \in \mathbb{N}$. Conceptually we think of the check nodes to be located at all integer positions from $[-\infty, \infty]$. Only some of these positions actually interact with the variable nodes. At each position there are $\frac{d_v}{d_c}M$ check nodes. It remains to describe how the connections are chosen.

Rather than assuming that a variable at position i has exactly one connection to a check node at position $[i - \lfloor l/2 \rfloor, \ldots, i + \lfloor l/2 \rfloor]$, we assume that each of the d_v connections of a variable node at position i is uniformly and independently chosen from the range $[i, \ldots, i + w - 1]$, where w is a "smoothing" parameter. In the same way, we assume that each of the d_c connections of a check node at position i is independently chosen from the range $[i - w + 1, \ldots, i]$. We no longer require that d_v is odd.

More precisely, the ensemble is defined as follows. Consider a variable node at position i. The variable node has d_v outgoing edges. A *type* t is a w-tuple of non-

negative integers, $t = (t_0, t_1, \ldots, t_{w-1})$, so that $\sum_{j=0}^{w-1} t_j = d_v$. The operational meaning of t is that the variable node has t_j edges which connect to a check node at position $i + j$. There are $\binom{d_v+w-1}{w-1}$ types. Assume that for each variable we order its edges in an arbitrary but fixed order. A *constellation* c is an d_v-tuple, $c = (c_1, \ldots, c_{d_v})$ with elements in $[0, w-1]$. Its operational significance is that if a variable node at position i has constellation c then its k-th edge is connected to a check node at position $i + c_k$. Let $\tau(c)$ denote the type of a constellation. Since we want the position of each edge to be chosen independently we impose a uniform distribution on the set of all constellations. This imposes the following distribution on the set of all types. We assign the probability

$$p(t) = \frac{|\{c : \tau(c) = t\}|}{w^{d_v}}.$$

Pick M so that $Mp(t)$ is a natural number for all types t. For each position i pick $Mp(t)$ variables which have their edges assigned according to type t. Further, use a random permutation for each variable, uniformly chosen from the set of all permutations on d_v letters, to map a type to a constellation.

Under this assignment, and ignoring boundary effects, for each check position i, the number of edges that come from variables at position $i - j$, $j \in [0, w-1]$, is $M\frac{d_v}{w}$. In other words, it is exactly a fraction $\frac{1}{w}$ of the total number Md_v of sockets at position i. At the check nodes, distribute these edges according to a permutation chosen uniformly at random from the set of all permutations on Md_v letters, to the $M\frac{d_v}{d_c}$ check nodes at this position. It is then not very difficult to see that, under this distribution, for each check node each edge is roughly independently chosen to be connected to one of its nearest w "left" neighbors. Here, "roughly independent" means that the corresponding probability deviates at most by a term of order $1/M$ from the desired distribution. As discussed beforehand, we will always consider the limit in which M first tends to infinity and then the number of iterations tends to infinity. Therefore, for any fixed number of rounds of DE the probability model is exactly the independent model described above.

LEMMA 14.1 (Design Rate) *The design rate of the ensemble (d_v, d_c, L, w), with $w \leq 2L$, is given by*

$$R(d_v, d_c, L, w) = (1 - \frac{d_v}{d_c}) - \frac{d_v}{d_c} \frac{w + 1 - 2\sum_{i=0}^{w} \left(\frac{i}{w}\right)^{d_c}}{2L + 1}.$$

Proof Let V be the number of variable nodes and C be the number of check nodes that are connected to at least one of these variable nodes. Recall that we *define* the design rate as $1 - C/V$.

There are $V = M(2L + 1)$ variables in the graph. The check nodes that have potential connections to variable nodes in the range $[-L, L]$ are indexed from $-L$ to $L + w - 1$. Consider the $M\frac{d_v}{d_c}$ check nodes at position $-L$. Each of the d_c edges of each such check node is chosen independently from the range $[-L - w + 1, -L]$. The probability that such a check node has at least one connection in the range

$[-L, L]$ is equal to $1 - \left(\frac{w-1}{w}\right)^{d_c}$. Therefore, the expected number of check nodes at position $-L$ that are connected to the code is equal to $M\frac{d_v}{d_c}(1 - \left(\frac{w-1}{w}\right)^{d_c})$. In a similar manner, the expected number of check nodes at position $-L+i$, $i = 0, \dots, w-1$, that are connected to the code is equal to $M\frac{d_v}{d_c}(1 - \left(\frac{w-i-1}{w}\right)^{d_c})$. All check nodes at positions $-L+w, \dots, L-1$ are connected. Further, by symmetry, check nodes in the range $L, \dots, L+w-1$ have an identical contribution as check nodes in the range $-L, \dots, -L+w-1$. Summing up all these contributions, we see that the number of check nodes which are connected is equal to

$$C = M\frac{d_v}{d_c}[2L - w + 2\sum_{i=0}^{w}(1 - \left(\frac{i}{w}\right)^{d_c})].$$

□

Discussion: In the above lemma we have *defined* the design rate as the normalized difference of the number of variable nodes and the number of check nodes that are involved in the ensemble. This leads to a relatively simple expression which is suitable for our purposes. But in this ensemble there is a non-zero probability that there are two or more degree-one check nodes attached to the same variable node. In this case, some of these degree-one check nodes are redundant and do not impose constraints. This effect only happens for variable nodes close to the boundary. Since we consider the case where L tends to infinity, this slight difference between the "design rate" and the "true rate" does not play a role. We therefore opt for this simple definition. The design rate is a lower bound on the true rate.

Density Evolution

The protograph construction has a slightly better performance if we look at codes of finite length and also, due to the extra structure, it might be easier to implement. On the other hand, the random ensemble is easier to deal with when it comes to proofs. Since asymptotically they behave essentially the same, we concentrate in the sequel on the random case.

The (d_v, d_c, L, w) ensemble is just an LDPC ensemble with some additional structure. It's asymptotic performance can hence again be assessed via density evolution. Therefore, as a first step let us write down the density evolution equations. The only difference compared to the DE equations of the uncoupled ensemble is that now we have a potentially different erasure probability for *every position*. The state is therefore no longer a scalar quantity but a vector of the length of the chain.

DEFINITION 14.2 (Density Evolution of (d_v, d_c, L, w) Ensemble) Let x_i, $i \in \mathbb{Z}$, denote the average erasure probability which is emitted by variable nodes at position i. For $i \notin [-L, L]$ we set $x_i = 0$. For $i \in [-L, L]$ the FP condition

implied by DE is

$$x_i = \epsilon\Big(1 - \frac{1}{w}\sum_{j=0}^{w-1}\big(1 - \frac{1}{w}\sum_{k=0}^{w-1} x_{i+j-k}\big)^{d_c-1}\Big)^{d_v-1}. \tag{14.1}$$

If we define

$$y_i = \Big(1 - \frac{1}{w}\sum_{k=0}^{w-1} x_{i-k}\Big)^{d_c-1}, \tag{14.2}$$

then (14.1) can be rewritten as

$$x_i = \epsilon\Big(1 - \frac{1}{w}\sum_{j=0}^{w-1} y_{i+j}\Big)^{d_v-1}.$$

EXIT Curves

As for uncoupled ensembles we can draw EXIT curves for the coupled case. Recall that in the uncoupled case, the EXIT curve is a plot of the channel parameter ϵ as a function of the EXIT value $(1-(1-x)^{r-1})^l$, see e.g., Figure 10.4. In the uncoupled case we had a simple analytical formula for this curve. For the coupled case, no such formula exists, but one can compute the curves numerically.

Figure 14.4 shows the EXIT curves for the $(d_v = 3, d_c = 6, L)$ for $L = 1, 2, 4, 8, 16, 32, 64$, and 128. Note that these EXIT curves show a dramatically

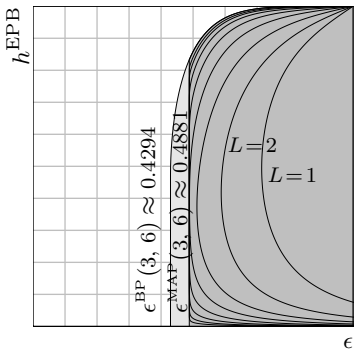


Figure 14.4 EBP EXIT curves of the ensemble $(d_v = 3, d_c = 6, L)$ for $L = 1, 2, 4, 8, 16, 32, 64$, and 128. The BP/MAP thresholds are $\epsilon^{\mathrm{BP/MAP}}(3,6,1) = 0.714309/0.820987$, $\epsilon^{\mathrm{BP/MAP}}(3,6,2) = 0.587842/0.668951$, $\epsilon^{\mathrm{BP/MAP}}(3,6,4) = 0.512034/0.574158$, $\epsilon^{\mathrm{BP/MAP}}(3,6,8) = 0.488757/0.527014$, $\epsilon^{\mathrm{BP/MAP}}(3,6,16) = 0.488151/0.505833$, $\epsilon^{\mathrm{BP/MAP}}(3,6,32) = 0.488151/0.496366$, $\epsilon^{\mathrm{BP/MAP}}(3,6,64) = 0.488151/0.492001$, $\epsilon^{\mathrm{BP/MAP}}(3,6,128) = 0.488151/0.489924$. The light/dark gray areas mark the interior of the BP/MAP EXIT function of the underlying $(3,6)$-regular ensemble, respectively.

different behavior compared to the EBP EXIT curve of the underlying ensemble. These curves appear to be "to the right" of the threshold $\epsilon^{\mathrm{MAP}}(3,6) \approx 0.48815$.

For small values of L one might be led to believe that this is true since the design rate of such an ensemble is considerably smaller than $1 - d_v/d_c$. But even for large values of L, where the rate of the ensemble is close to $1 - d_v/d_c$, this dramatic increase in the threshold is still true. Empirically we see that, for L increasing, the EBP EXIT curve approaches the MAP EXIT curve of the underlying $(d_v = 3, d_c = 6)$-regular ensemble. In particular, for $\epsilon \approx \epsilon^{\text{MAP}}(d_v, d_c)$ the EBP EXIT curve drops essentially vertically until it hits zero.

Decoding Wave

"The" key to understanding why spatially coupled ensembles perform so well is to study their FPs under density evolution. Recall that for uncoupled ensembles the FPs are scalars. For the coupled case the state of the system is no longer a scalar but a vector, where the length of the vector is equal to the length of the chain. Due to this fact, there are some very interesting FPs which appear.

Assume we are operating much above the threshold. Let us assume that we decode until we are stuck and let us plot the final erasure probability at each section along the chain. Then it is reasonably to expect that this erasure probability is equal to the erasure probability which we would observe for an uncoupled ensemble. The only exception are positions very close to the boundary where the behavior is a little bit better due to the extra information we have there. The top picture in Figure 14.6 shows this situation together with the position of the FP on the EXIT curve. Since the FP is symmetric with respect to the middle of the chain, only one half is shown. Imagine that we now slowly lower the erasure probability of the channel. Due to the improved conditions at the boundary, the "effective" erasure probability at the boundary will at some point be below the BP threshold of the uncoupled ensemble and the BP decoder will be able to decode the bits at the boundary. But once these bits are decoded this will lower the "effective" erasure probability for bits a little bit further into the chain. This effect propagates like a wave and the whole chain will get decoded. The middle and the bottom picture in Figure 14.6 show the wave in various stages.

The perhaps the most surprising aspect is that the BP threshold for the coupled chain is exactly the area threshold of the uncoupled one.

¡¡¡¡¡¡¡ .mine Figure 14.6 shows the FP for various parameters of the channel together with the position of the FP on the EXIT curve. Since the FP is symmetric with respect to the middle of the chain, only one half is shown.

Main Statement

THEOREM 14.3 (BP Threshold of the (d_v, d_c, L, w) Ensemble) *Consider transmission over the BEC(ϵ) using random elements from the ensemble (d_v, d_c, L, w). Let $\epsilon^{BP}(d_v, d_c, L, w)$ denote the BP threshold and let $R(d_v, d_c, L, w)$ denote the design rate of this ensemble.*

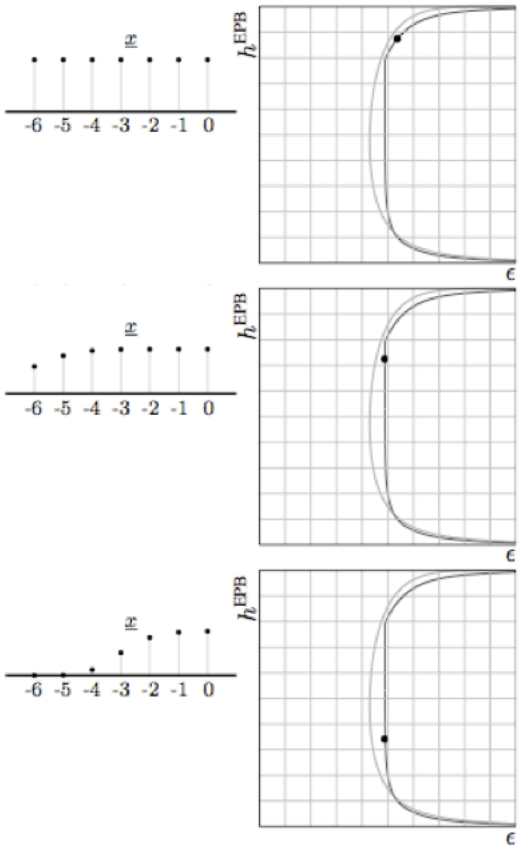


Figure 14.5 FPs for various parameters of the channel together with the position of the FP on the EXIT curve.

Figure 14.6 FPs for various parameters of the channel together with the position of the FP on the EXIT curve.

Then, in the limit as M tends to infinity, and for w sufficiently large

$$\epsilon^{BP}(d_v,d_c,L,w) \leq \epsilon^{MAP}(d_v,d_c,L,w) \leq \epsilon^{MAP}(d_v,d_c) + \frac{w-1}{2L(1-(1-x^{MAP}(d_v,d_c))^{d_c-1})^{d_v}}, \tag{14.3}$$

$$\epsilon^{BP}(d_v,d_c,L,w) \geq \left(\epsilon^{MAP}(d_v,d_c) - w^{-\frac{1}{8}} \frac{8d_vd_c + \frac{4d_cd_v^2}{(1-4w^{-\frac{1}{8}})^{d_c}}}{(1-2^{-\frac{1}{d_c}})^2}\right) \times \left(1-4w^{-1/8}\right)^{d_cd_v}. \tag{14.4}$$

In the limit as M, L and w (in that order) tend to infinity,

$$\lim_{w\to\infty}\lim_{L\to\infty} R(d_v, d_c, L, w) = 1 - \frac{d_v}{d_c}, \tag{14.5}$$

$$\begin{aligned}\lim_{w\to\infty}\lim_{L\to\infty} \epsilon^{BP}(d_v, d_c, L, w) &= \lim_{w\to\infty}\lim_{L\to\infty} \epsilon^{MAP}(d_v, d_c, L, w)\\ &= \epsilon^{MAP}(d_v, d_c).\end{aligned} \tag{14.6}$$

Roughly speaking, the above theorems states that the BP threshold of the coupled chain is equal to its MAP threshold and also to the MAP threshold of the uncoupled chain. The statements in the theorem are considerably weaker than what can be observed empirically. In particular, the convergence with respect to the coupling width is conjectured to be exponential in w.

A very similar statement can be shown to hold for transmission over general channels. In particular, one can show that these ensembles are good universally for the whole class of BMS channels.

14.2 Compressive Sensing

The idea of spatial coupling can also be used in compressive sensing to attain optimal performance by message passing. In a nutshell, the idea is to construct appropriate sensing matrices that correspond to a "spatially coupled" factor graph and then to apply an AMP type algorithm. The performance of the algorithm is then analyzed through a state evolution recursion tailored to the spatially coupled graph. This turns out to be a one-dimensional recursion which displays similar phenomena than those described for the BEC.

In Chapter 8 our starting point was the Lasso estimator which is a reasonable starting point to develop a universal algorithm that doe not assume a prior knowledge of the signal distribution in the class $\mathcal{F}_\epsilon$. Recall that the state evolution equation in Chapter **??** has at most one fixed point. Therefore, intuitively, one does not expect that any improvement in performance can be obtained by spatial coupling. This has indeed been corroborated by numerical simulations. We will therefore turn our attention to a setting where the prior distribution of the signal is known.

AMP when the prior is known

We assume that the signal distribution is from the class $\mathcal{F}_\epsilon$ and that it is known. In other words $p_0(x) = (1-\epsilon)\delta_0(x) + \epsilon\phi_0(x)$ for a known $\phi_0(x)$ (for example a Gaussian distribution). As explained in Chapter 3, in this setting the optimal estimator is the MMSE estimator (3.33). In Chapter 5 we went through the belief propagation equations in Example 16. This approach can be systematically developed in order to recursively compute the BP-estimate. Furthermore, following the same route as in Chapter 8, these message-passing equations can be

simplified in order to arrive at an AMP algorithm that is very similar to (8.37). By skimming through the previous chapters one can almost guess the form of the new algorithm.

In (8.37) the update of the AMP-estimate uses the soft thresholding function $\eta(y,\lambda)$ found by solving the scalar Lasso problem. The reader should not be too surprised that now the AMP updates involve a thresholding function given by the MMSE estimator of the scalar case. Consider a scalar measurement $y = x + \nu z$ of "signal" x affected by Gaussian noise with variance ν^2 (so $Z \sim N(0,1)$) the thresholding function is

$$\eta_0(y,\nu) = \mathbb{E}[X|X+\nu Z = y] = \frac{\int dx\, x\, p_0(x) e^{-\frac{(y-x)^2}{2\nu^2}}}{\int dx\, p_0(x) e^{-\frac{(y-x)^2}{2\nu^2}}}.$$

We stress that $\eta_0(y,\nu)$ is not universal and depends on the prior. Here ν plays the role of a threshold level analogous to λ in the Lasso case. It will be adjusted at each AMP iteration. The mean square error for this optimal estimator (of the scalar problem) is the MMSE function[1]

$$\begin{aligned}\mathrm{mmse}(\nu^{-2}) &= \mathbb{E}\big[(X - \mathbb{E}[X|X+\nu Z])^2\big] \\ &= \int dx\, p_0(x) \int dz\, \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} (x - \eta_0(x+\nu z,\nu))^2.\end{aligned}$$

The AMP updates are the same than in Chapter 8 except η is replaced by η_0,

$$\hat{x}_i^{(t+1)} = \eta_0\big(x_i^{(t)} + \sum_{a=1}^{m} A_{ai} r_a^{(t)}, \nu^{(t)}\big), \tag{14.7}$$

$$r_a^{(t)} = y_a - \sum_{j=1}^{n} A_{aj} \hat{x}_j^{(t-1)} + b^{(t)} r_a^{t-1}. \tag{14.8}$$

If you go back to the derivation of the Onsager term in Chapter 8 you will see that it can be traced back to a derivative of the soft thresholding function. You can guess that now

$$b^{(t)} = \frac{1}{\delta n} \sum_{i=1}^{n} \eta_0'\big(x_i^{(t-1)} + \sum_{a=1}^{m} A_{ai} r_a^{(t-1)}, \nu^{(t)}\big). \tag{14.9}$$

Similarly recall that in Chapter **??** we expressed the threshold level $\nu^{(t)}$ thanks to the MSE through (8.48). Here one arrives at the same conclusion, namely

$$(\nu^{(t)})^2 = \sigma^2 + \frac{1}{\delta}(\tau^{(t)})^2, \tag{14.10}$$

where $\tau^{(t)2}$ is the average (normalized) MSE of the AMP algorithm $(\tau^{(t)})^2 = \lim_{n\to+\infty} \frac{1}{n}\mathbb{E}\|\underline{\hat{x}}^{(t)} - \underline{x}_0\|_2$. We can track its evolution thanks to the recursion (same as (**??**) with correct η_0-function)

$$(\tau^{(t+1)})^2 = \mathrm{mmse}((\nu^{(t)})^{-2}). \tag{14.11}$$

[1] By convention the argument of the MMSE function is a signal-to-noise-ratio, here ν^{-2}.

In hindsight one can develop an interpretation for this equation: at time $t+1$ the total quadratic error $(\tau^{(t+1)})^2$ for the AMP estimate is given by the MMSE of a scalar signal with effective noise variance $\sigma^2 + \frac{1}{\delta}(\tau^{(t)})^2$ at time t.

Let us summarize. Equations (14.11) and (14.10) give the evolution of the MSE and the threshold level. These quantities can be precomputed. Equations (14.8) and (14.9) define the AMP algorithm, and allow to compute the estimates for the signal.

Construction of the measurement matrix

Let us first explain the general idea. In the standard case considered so far, the measurement matrices have iid entries $A_{ai} \sim \mathcal{N}(0, \frac{1}{\sqrt{m}})$ so that "their factor graph" is a complete bipartite graph with m checks and n variables. The ratio $\delta = m/n$ is the sampling rate. Inspired by the construction of spatially coupled codes one may try to use matrices associated to a spatial chain of L complete bipartite graphs coupled across a window of size w. This turns out to be a successful idea! The sampling rate is still equal to δ in the bulk of the chain. At the boundary one has to add extra check nodes or equivalently one has to oversample. Indeed, in order to create a seed that gets the nucleation process started one needs a good estimate of the first few components of the signal. The increase in sampling rate is negligible in the thermodynamic limit.

In practice, because the AMP algorithm updates purely local quantities (the BP messages flowing along edges have been eliminated), one can forget about the factor graph and specify directly the sensing matrix. You can convince yourself that the sensing matrix described here has a factor graph that is a chain of coupled complete bipartite graphs. There are many possible constructions and ways to optimize the finite length performance. But these issues will not concern us here, and we discus a similar construction which is similar to the one presented in the coding case.

The signal has n components in total and we make m measurements. The measurement matrix has n columns and m rows. Think of n given and m to be determined later. Partition the columns in L groups[2] $c \in \{1, \dots, L\}$ with N columns each, so $N = n/L$. Consider $L + w - 1$ groups of rows $r \in \{-(w-2), \dots, 0, 1, \dots, L\}$, each with $M = \delta N$ rows. The total number of measurements is $m = (w-1)M + ML = \delta n(1 + (w-1)/L)$. The contribution of the oversampling rate to the total rate $m/n = \delta(1 + (w-1)/L)$ vanishes for large L.

Now consider an $(L + w - 1) \times L$ matrix of variances J_{rc}. A simple choice is

$$J_{r,c} = \begin{cases} \frac{1}{2w-1} & \text{if } \ c \in \{r - w + 1, \dots, r + w - 1\} \\ 0 & \text{otherwise} \end{cases}$$

Here we use a simple square-like and symmetric shape function for $J_{r,c}$. One can generalize this to $J_{r,c} = \rho \mathcal{J}(\rho|r - c|)$ with $\rho = (2w-1)^{-1}$ and a shape function

[2] One can visualize the groups as positions along the chain.

$\mathcal{J}(z)$ that is positive, supported on $[-1,+1]$ and $\int_{-1}^{+1} dz\, \mathcal{J}(z) = 1$. Let us also note that taking larger variances for the seeding part of the matrix may lead to better performance. In the sequel all equations are valid for general choices of $J_{r,c}$.

To specify the matrix elements of A_{ai}, we introduce the notation $R(a)$ and $C(i)$ for the groups (r and c) to which row a and column i belong. A simple choice is to take iid entries

$$A_{ai} \sim (0, \frac{1}{M} J_{R(a),C(i)})$$

We notice that by construction we have the normalization $\sum_i A_{ai}^2 \approx 1$, as in the standard (uncoupled) case. This matrix has a band structure with a band of height and width $wM \times wN$. However the correct regime in which the spatially coupled model is used is $N >> L$ so effectively the matrix is "full".

Spatially coupled AMP

The starting point - the BP equations - are exactly the same except they are applied to a bigger factor graph. The derivation of the coupled AMP algorithm then proceeds in the usual way by retaining only important terms *in the regime* $N \to +\infty$ and L fixed.

It turns out that the resulting equations have a few extra complications. Namely, due to coupling, the sensing matrix elements get "renormalized" and the threshold level as well as the Onsager term get "averaged". The AMP equations now read

$$\hat{x}_i^{(t+1)} = \eta_0\big(x_i^{(t)} + \sum_{a=1}^{m} Q^{(t)}_{R(a),C(i)} A_{ai} r_a^{(t)}, \nu^{(t)}_{C(i)}\big) \tag{14.12}$$

$$r_a^{(t)} = y_a - \sum_{j=1}^{n} A_{aj}\hat{x}_j^{(t-1)} + b^{(t)}_{R(a)} r_a^{t-1} \tag{14.13}$$

where

$$b^{(t)}_{R(a)} = \frac{1}{\delta}\sum_{c=1}^{L} J_{R(a),c} Q^{t-1}_{R(a),c}\left\{\frac{1}{N}\sum_{i\,\mathrm{s.t}\,C(i)=c} \eta_0'\big(x_i^{(t)} + \sum_{b=1}^{m} Q^{(t)}_{R(b),C(i)} A_{bi} r_b^{(t)}, \nu^{(t)}_{C(i)}\big)\right\}$$

The threshold levels $\nu^{(t)}_{C(i)}$ and the weights $Q_{R(a),C(i)}$ depend only on the *local* MSE $(\tau_c^{(t)})^2 = \lim_{N\to+\infty} \frac{1}{N}\sum_{i\,\mathrm{s.t}\,C(i)=c} \mathbb{E}\|\hat{x}_i^{(t)} - x_{0,i}\|_2^2$. These quantities can all be pre-computed from state evolution. The threshold level is given by (a generalization of (14.10))

$$(\nu_c^{(t)})^{-2} = \sum_r J_{r,c}\big(\sigma^2 + \frac{1}{\delta}\sum_c J_{r,c}(\tau_c^{(t)})^2\big)^{-1}, \tag{14.14}$$

This equation says that the threshold for estimates of the signal components in group c is given by an average of the signal to noise ratios for measurements in

the groups $r \in \{c-w+1, \dots, c+w-1\}$, and the later are themselves given by an average of the local MSE in the groups $c \in \{r-w+1, \dots, r+w-1\}$. The sensing matrix gets renormalized by weights

$$Q_{r,c} = \frac{\left(\sigma^2 + \frac{1}{\delta}\sum_c J_{r,c}(\tau_c^{(t)})^2\right)^{-1}}{\sum_r J_{r,c}\left(\sigma^2 + \frac{1}{\delta}\sum_c J_{r,c}(\tau_c^{(t)})^2\right)^{-1}}.$$

Finally, the local MSE evolves as

$$(\tau_c^{(t+1)})^2 = \mathrm{mmse}((\nu_c^{(t)})^{-2}), \qquad c = 1, \dots, L \tag{14.15}$$

Equations (14.14)-(14.15) are the one dimensional state evolution recursion and can be used to derived the performance of AMP on the spatially coupled model. The reader should ponder on this recursion and realize that its structure is perfectly analogous to the DE recursion in coding for the BEC.

Analysis of Performance and Phase Diagram

The discussion in this paragraph is valid for a fairly wide class of functions $\phi_0(x)$, but a good exercise for the reader is to verify the claims for a Gaussian $\phi_0(x)$. This can be done analytically for the uncoupled case and numerically in the coupled case. Notice that in this case $\eta_0(y,s)$ can be explicitly be computed.

Consider the recursion (14.11) and look at the corresponding fixed point equation. Let

$$\tilde{\delta}(p_0) \equiv \sup_{\nu}\{\nu^{-2}\mathrm{mmse}(\nu^{-2})\} > \epsilon$$

Here the equality is definition. The inequality is a fact, which follows by remarking $\lim_{\nu\to 0}\nu^{-2}\mathrm{mmse}(\nu^{-2}) = \epsilon$. For a sampling rate $\delta > \tilde{\delta}(p_0)$ there exists only one fixed point solution $(\tau_{\mathrm{good}})^2 = O(\sigma^2)$. This corresponds to correct reconstruction in the small noise limit $\sigma \to 0$. Now, decrease the sampling rate in the range $\epsilon < \delta < \tilde{\delta}(p_0)$. One finds two or more stable fixed points (as well as unstable ones) for all $\sigma^2 > 0$. Besides the "good" fixed point satisfies $(\tau_{\mathrm{good}})^2 = O(\sigma^2)$ there is a "bad" one, i.e. $(\tau_{\mathrm{bad}})^2 = \Theta(1)$ as $\sigma \to 0$. Under the (natural) initial condition $(\tau^0)^2 = +\infty$ one always tends to $(\tau_{\mathrm{bad}})^2$. This means that the noise sensitivity $\lim_{\sigma\to 0} MSE/\sigma^2$ diverges, and exact reconstruction is not possible even for very small noise. In this context $\tilde{\delta}(p_0)$ is the algorithmic threshold of AMP. The analogous quantity in our coding model is ϵ_{BP} and it the CW model it is the spinodal point.

This threshold is lower than the Lasso (or l_1) threshold derived in Chapter **??**. This is not too surprising since the later concerns the worst case distribution for $p_0 \in \mathcal{F}_\epsilon$. It is instructive to compute the phase diagram and plot the optimal, Lasso and AMP phase transition lines in the (ϵ, δ) plane.

Let us now turn our attention to the coupled model. The performance is analyzed through the one dimensional recursion (14.14)-(14.15) which gives the evolution of the MSE profile $\tau_c^{(t)}$, as a function of time t and position along the

chain $c = 1, \dots, L$. For $\delta > \tilde{\delta}(p_0)$ the local MSE tends to $(\tau_{c,\text{good}})^2 = O(\sigma^2)$ uniformly along the chain. The advantage brought by spatial coupling appears for a sampling rate in the range $\epsilon < \delta < \tilde{\delta}(p_0)$. For $L \to +\infty$ and fixed $w \geq 2$ there is a $\tilde{\delta}(p_0, w) < \tilde{\delta}(p_0)$ such that for $\delta > \tilde{\delta}(p_0, w)$ the local MSE per position is bounded by $O(\sigma^2)$, and in particular the noise sensitivity remains finite. Because of the oversampling of the first few signal components, the MSE falls down to a level $O(\sigma^2)$ for these components, and then an estimation wave propagates along the chain. Eventually the local MSE converges to the good fixed point for all positions $\tau_{\text{good},c} = O(\sigma^2)$. Furthermore one observes that $\tilde{\delta}(p_0, w) \to \epsilon$ as $w \to +\infty$. In other words in the regime $N >> L >> w >> 1$ the dynamical AMP threshold saturates towards the optimal phase transition threshold. Figure **??** illustrates the phase diagram and the phase transition lines in teh (ϵ, δ) plane for various values of L and w.

14.3 K-SAT

For the random K-SAT problem we discussed several algorithms. The best one is BP-guided decimation. We described this algorithm and its empirical performance in Chapter 9.5. If we apply spatial coupling to this algorithm we see no boost in performance. This does not mean that spatial coupling does not help for this problem. It just means that BP-guided decimation is not the right setting for the nucleation phenomenon. The "right" setting is in fact a more sophisticated algorithm called *survey propagation*.

Rather than pursuing this avenue, let us go to a simpler algorithm, namely the UCP algorithm which we discussed in Chapter 9. We will see that spatially coupled formulas have a significantly higher threshold under UCP than uncoupled ones. Combined with the interpolation method this gives good lower bounds on the SAT/UNSAT threshold of uncoupled systems.

Construction

As for the case of coding, there are various ways of constructing coupled K-SAT formulas. E.g., Figure 14.7 shows the equivalent of a protograph ensemble for the case $K = 3$ where each clause at position i has exactly one connection to a variable at position i, $i + 1$, and $i + 2$.

For the purpose of analysis it is again more convenient to consider a random ensemble. As before, let w be a window size. Then, for each clause at position i and for each of its K connections we independently and uniformly pick a variable at a position in the range $[i, i + w - 1]$ and connect it to this variable with a uniformly chosen sign. This is the ensemble which we consider in the sequel.

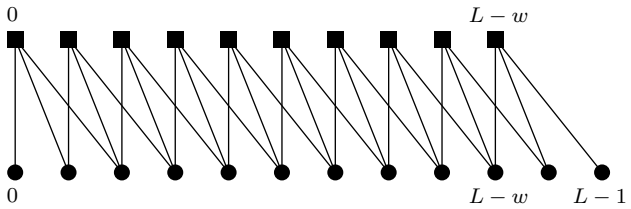


Figure 14.7 A "protograph"-like coupled K-SAT ensembles or $K = 3$.

Performance under the UCP Algorithm

Let us now focus on the UC algorithm for the coupled formulas. As for the un-coupled case, the UC algorithm consists of two main steps: free and forced. The operation of the algorithm at a forced step is clear: remove all the unit-clauses until no further unit-clause exists. However, at a free step, depending on how we might want to use the chain structure of the formula, we can have different *schedules* for choosing a free variable. For a coupled formula, the schedule within which we are choosing a variable in a free step is important

Consider for instance the following naive schedule – at a free step, pick a variable uniformly at random from all the remaining variables and fix it by flipping a coin. Computer experiments indicate that this naive schedule has no threshold gain compared to the un-coupled ensemble. This is not surprising since this schedule does not exploit the spatial (chain) structure of the formula. Hence, in order for the UC algorithm to have a threshold improvement over the coupled ensemble, we need to come up with schedules that exploit the additional spatial structure of the formula. We proceed by illustrating one such successful schedule.

In the very beginning of the algorithm, all the check nodes have degree K and there are no unit clauses. Hence, we are free to fix the variables in the first few steps of the algorithm. Let us fix the variables from the left-most position (i.e., the boundary). If we do this then we are creating in effect a seed at the boundary of the chain. Continuing this action at the free steps, we will eventually create unit clauses and at these forced steps a natural choice is just to clear all the unit clauses. However, when we are confronted with a free step, we will again try to help this seed to grow inside the chain, i.e., we always fix variables from the left-most possible position. Consequently, the schedule that we apply is as follows.

- At a *free step*, pick a variable randomly from the left-most position at which variables exists and fix it permanently by flipping a fair coin.
- At a *forced step*, remove unit clauses as long as they exist.

Computer experiments show that this schedule indeed exhibits a threshold improvement over the un-coupled ensemble. E.g., for the coupled 3-SAT problem, experiments suggest that the threshold of the UC algorithm is around 3.67. This

is a significant improvement compared to the threshold of UC for the un-coupled ensemble which is $\frac{8}{3}$.

To prove that indeed this schedule leads to this threshold we use again the Wormald method. This means, we write down a set of differential equations which describe the expected progress of the algorithm. Not surprisingly, the number of differential equations we need scales linearly in the chain length.

Phases, Types, and Rounds

For the coupled ensemble, the analysis of the evolution of UC is much more involved than the un-coupled ensemble. This is because of the fact that the schedule we have used prefers the left-most variable position in a free step. Hence, the number of variables in different positions will evolve differently. As an example, one can easily see that during the algorithm, the first position that all its variables are set is the left-most position (i.e., position 0). After the evacuation of position 0, position 1 becomes the left-most position of the graph and hence, the second position that becomes empty of variables is position 1. Continuing in this manner, the last position that is evacuated is position $L + w - 2$. With these considerations, we consider $L+w-1$ *phases* for this algorithm (see Figure 14.8). At phase $p \in \{0, 1, \cdots, L + w - 2\}$, all the variables at positions prior to p have been set permanently and as a result, at a free step we will pick a variable from position p.

This statistical asymmetry in the number of variables at each position also affects the the behavior of the number of check nodes in each position. As a result, we consider *types* for the check nodes. For instance, consider a degree two check node. It is easy to see that the probability that this degree two check node is hit (removed or shortened) is greatly dependent on the position of variables that it is connected to. This means that, dependent on the variable positions to which they are connected, we have different types of degree two check nodes. Clearly, the same statement holds for clauses of degree three, four, etc.

Let us now formally define the ingredients needed for the analysis. The notation we use here is slightly hard to swallow immediately. Thus, for the sake of maximum clarity, we try to uncover the details as smoothly as possible. We consider *rounds* for this algorithm. Each round consists of one free step followed by the forced steps that follow it. More precisely, at the beginning of each round we perform a free step and then we clear out all the unit-clauses as long as they exist (forced steps). We let time t be the number of rounds passed so far. This time variable will be called *round time*. The relation between t and the *natural time* (the total number of permanent fixes) is not linear. We also let $L_i(t)$ be the *number of literals* left in variable position $i \in \{0, 1, \cdots, L + w - 2\}$.

We now define the check types. Consider a coupled K-SAT formula to begin with. For such a formula there are L sets of check nodes placed at positions $\{0, 1, \cdots, L\}$. Let us consider a specific position $i \in \{0, 1, \cdots, L\}$ and look at the check nodes at position i. Each of these check nodes can potentially be connected

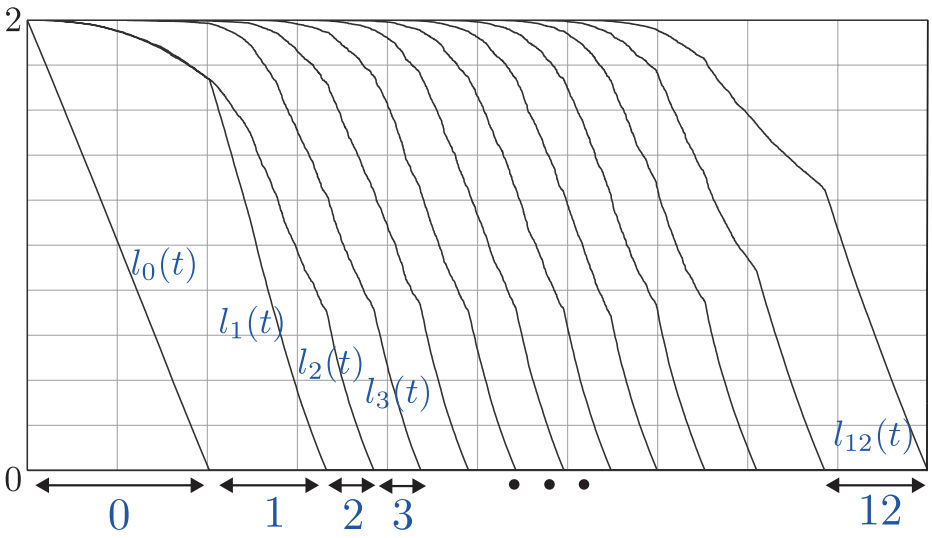


Figure 14.8 A schematic representation of how the literals at each of the positions vary in time. The horizontal axis corresponds to time t which is the number of free steps. Here we have $L = 11$ and $w = 3$. This plot corresponds to an implementation of the UC algorithm on a random coupled instance. The blue numbers below the plot are the phases of the algorithm. In the beginning of the algorithm, we are in phase 0. This phase lasts until all the literals in the first position are peeled off and as a result $\ell_0(t)$ reaches 0. We then go immediately to phase 1 and this phase lasts till $\ell_1(t)$ reaches 0 and so on. We have in total $L + w - 1 = 13$ phases.

to any set of K variables resting in variable positions $\{i, i+1, \cdots, i+w-1\}$. Some thought shows that there are various types of check nodes depending on the variable positions that they are connected to. For example, there is a type of check nodes for which all of the K edges go only into a single variable position $j \in \{i, i+1, \cdots, i+w-1\}$ or there is a type for with some of its edges go to position i and the rest go to position $i+1$ and so on. Also, as we proceed through the UC algorithm, some of these checks are shortened to create new types of checks with degrees less than K. We now explain a natural way to encode these various types.

By $C(t, i, \underline{\tau})$ we mean the number of check nodes at check position $i \in \{0, 1, \cdots, L\}$ that have type $\underline{\tau}$ at round tme t. The type $\underline{\tau} = (\tau_0, \cdots, \tau_{w-1})$ is a w-tuple and indicates that relative to position i, how many edges the check has in (variable) positions $i, i+1, \cdots, i+w-1$. The best way to explain $\underline{\tau}$ is through an example. Let us assume $w = 4$ and consider the set of check nodes at check position 20 that are only connected to variable positions $20, 22, 23$ in the following way. For each of these check nodes there are exactly two edges going to position 20, and 1 edge going to position 22 and 1 edge going to position 23 (thus each of these checks have degree 4). Figure 14.9 illustrates a generic check node of this set.

We denote the number of these checks at time t by $C(t, 20, (1, 0, 2, 1))$. In

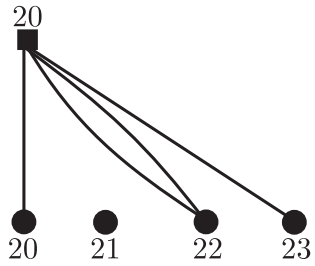


Figure 14.9 A schematic representation of checks which contribute to $C(t, 20, (1, 0, 2, 1))$. All the check nodes that contribute to $C(t, i, \underline{\tau})$, were initially (at time 0) degree K check nodes resting at check position i. However, the algorithm has evolved in a way that these check nodes have been deformed (possibly shortened or remained unchanged) to have a specific type $\underline{\tau}$.

other words, the type is computed as follows: the check position number that the check rests in is 20. This check is connected to a variable at position 20, and 2 variables at position 22, and a variable at position 23. So, relative to the check position 20, we see the edge-tuple $(1, 0, 2, 1)$. Let us now repeat and generalize: By $C(t, i, \underline{\tau})$ we mean the number of check nodes, at time t, which rest in position i, and $\underline{\tau}$ is a w-tuple that indicates relative to variable position i, the number of edges that go to positions $i, i+1, \cdots, i+w-1$, respectively. One can easily see that by summing up elements of the w-tuple $\underline{\tau} = (\tau_0, \cdots, \tau_{w-1})$, we find the degree of the corresponding check type. We denote the degree of a type $\underline{\tau}$ by $\deg(\underline{\tau})$. It is also easy to see that there are $\binom{d+w-2}{d-1}$ different types of degree d for $d \in \{2, 3, \cdots, K\}$. We are now ready to write the differential equations. Our approach is as follows. Assume the phase of the algorithm is p and we are in a round t. At a free step, we fix a variable at position p (free step). This will create a number of forced steps in each of the positions $p, p+1, \cdots, L+w-1$. We first compute the average of these forced fixes in each variable position as a function of the number of degree two check nodes. Using these averages, we then update the average number of check and variable nodes at each position. We proceed by explaining a key property for the analysis.

The Differential Equations

Now, having the vector $\underline{\beta}$ we can find how the number of variables and checks evolve. For all $i \geq 0$,

$$\Delta L_i(t) = L_i(t+1) - L_i(t) = -2\beta_i(t). \tag{14.16}$$

To see how the check types evolve, we note that for a given check type there are two kinds of flows to be considered. A negative flow going out and a positive flow coming in from the checks of higher degrees. In this regard, for a type $\underline{\tau} = (\tau_0, \cdots, \tau_{w-1})$ with $\deg(\underline{\tau}) < K$ let $\partial\underline{\tau}$ be the set of types of degree $\deg(\underline{\tau})+1$

such that by removing one edge from them we reach to the type $\underline{\tau}$. The set $\partial\underline{\tau}$ consists of w types which we denote by $\underline{\tau}^d$, $d \in \{0, 1, \cdots, w-1\}$, such that

$$\underline{\tau}^d = \underline{\tau} + (0, \cdots, \overset{d}{1}, \cdots, 0), \tag{14.17}$$

where $+$ denotes vector addition in the field of reals. Thus, if $\deg(\underline{\tau}) < K$, we obtain

$$\Delta C(t, i, \underline{\tau}) = -2 \sum_{d=0}^{w-1} \beta_{i+d}(t) \frac{\tau_d c(t, i, \underline{\tau})}{L_{i+d}(t)} + \sum_{d=0}^{w-1} (1 + \tau_d) \beta_{i+d}(t) \frac{C(t, i, \underline{\tau}^d)}{L_{i+d}(t)}. \tag{14.18}$$

The right-hand side of (14.18) has two parts. The first part corresponds to the flow that is going out of $C(t, i, \underline{\tau})$ and has negative sign. The right part is the incoming flow from the check nodes of higher degrees. In the case where $\deg(\underline{\tau}) = K$, we only have an outgoing flow since no check node with higher degrees exist. Hence, for the case $\deg(\underline{\tau}) = K$ we can write

$$\Delta C(t, i, \underline{\tau}) = -2 \sum_{d=0}^{w-1} \beta_{i+d}(t) \frac{\tau_d c(t, i, \underline{\tau})}{L_{i+d}(t)}. \tag{14.19}$$

We now write the initial conditions for the variables and check types. Firstly, note that $L_i(0) = 2N$. In the beginning of the algorithm, all checks are of degree K, thus for types $\underline{\tau}$ such that $\deg(\underline{\tau}) < K$, we have $C(0, i, \underline{\tau}) = 0$. For $\deg(\underline{\tau}) = K$ we have

$$C(0, i, \underline{\tau}) = \alpha N \frac{\binom{K}{\tau_0, \tau_1, \cdots, \tau_{w-1}}}{w^K}. \tag{14.20}$$

In order to write the differential equations, we re-scale the (round) time by N, i.e.

$$t \leftarrow \frac{t}{N}, \tag{14.21}$$

and also normalize all our other numbers by N, i.e.,

$$c(t, \cdot, \cdot) = \frac{C(Nt, \cdot, \cdot)}{N} \text{ and } \ell_i(t) = \frac{L_i(Nt)}{N}. \tag{14.22}$$

We then obtain for $i \in \{0, 1, \cdots, L + w - 2\}$,

$$\frac{d\ell_i(t)}{dt} = -2\beta_i(t). \tag{14.23}$$

For $i \in \{0, 1, \cdots, L-1\}$ and $\deg(\underline{\tau}) < K$ we have

$$\frac{dc(t, i, \underline{\tau})}{dt} = -2 \sum_{d=0}^{w-1} \beta_{i+d}(t) \frac{\tau_d c(t, i, \underline{\tau})}{\ell_{i+d}(t)} + \sum_{d=0}^{w-1} (1 + \tau_d) \beta_{i+d}(t) \frac{c(t, i, \underline{\tau}^d)}{\ell_{i+d}(t)}, \tag{14.24}$$

and otherwise if $\deg(\underline{\tau}) = K$ we have

$$\frac{dc(t, i, \underline{\tau})}{dt} = -2 \sum_{d=0}^{w-1} \beta_{i+d}(t) \frac{\tau_d c(t, i, \underline{\tau})}{\ell_{i+d}(t)}. \tag{14.25}$$

K	3	4	5
$\alpha_{\mathrm{UC}}(K)$	2.66	4.50	7.58
$\alpha_{\mathrm{UC},L=50,w=3}(K)$	3.67	7.81	15.76

Table 14.1 *First line*: The thresholds for UCP on the uncoupled ensemble. *Second line*: UCP threshold for a coupled chain with $w = 3$, $L = 50$.

The vector $\bar{\beta}$ is also found as follows. For p being the current phase, we have

$$\underline{\beta}(t) = (\beta_0(t), \cdots, \beta_{L+w-2}(t))^T = (I - A)^{-1} e_p, \tag{14.26}$$

where $A = [A_{i,j}]_{(L+w-1)(L+w-1)}$ has the form

$$A_{i,j} = \frac{1}{\ell_j(t)} \begin{cases} \sum_{k=i-w+1}^{i} 2c(t, k, \pi_{i-k,i-k}) & i = j, \\ \sum_{k=j-w+1}^{i} c(t, k, \pi_{i-k,j-k}) & 0 <\mid i - j \mid < w, \\ 0 & \text{otherwise} \end{cases} \tag{14.27}$$

Finally, the initial conditions are given by:

$$\begin{aligned} &\ell_i(0) = 2, \text{ for } 0 \le i \le L + w - 2 \\ &c(0, i, \underline{\tau}) = \begin{cases} \alpha \frac{\binom{K}{\tau_0, \tau_1, \cdots, \tau_{w-1}}}{w^K} & \text{if } \deg(\underline{\tau}) = K \text{ and } 0 \le i \le L - 1, \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{14.28}$$

Numerical Implementation

We have implemented the above set of differential equations in C. We define the threshold $\alpha_{\mathrm{UC},L,w}(K)$ as the highest density for which the spectral norm (largest eigenvalue) of the matrix A is strictly less than one throughout the whole algorithm. A practical point to notice here is that, for the sake of implementation, we assume a phase p finishes when its corresponding variable $\ell_p(t)$ goes below a (very) small threshold $\epsilon > 0$. In our implementations, we have typically taken $\epsilon = 10^{-5}$. However, it can be made arbitrarily small as long as the computational resources allow.

Table 14.1 shows the value of $\alpha_{\mathrm{UC},L,w}(K)$ with $L = 50$ and $w = 3$ for different choices of K. As we observe from Table 14.1, for the UC algorithm with the specific schedule mentioned above, there is a significant threshold improvement over the un-coupled ensemble.

For $L = 50, w = 3, K = 3$ and several values of α, we have plotted in Figure 14.10 the evolution of largest eigenvalue of A as a function of round time t.

In order to characterize analytically the ultimate threshold for the UC algorithm when L and w grow large, we proceed by further analyzing the set of differential equations.

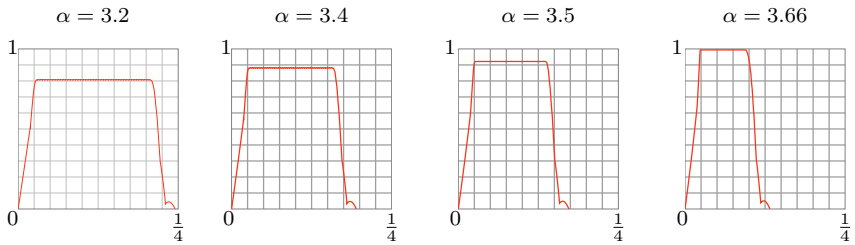


Figure 14.10 The largest eigenvalue of the matrix A, plotted versus the round time t (the number of rounds divided by the total number of variables NL). The plots correspond to an actual implementation of the UC algorithm for the 3-SAT coupled ensemble with $L = 50$ and $w = 3$. As we observe, for $\alpha < 3.67$, there is a gap between the largest eigenvalue of A and the value 1 throughout the UC algorithm. By increasing α this gap shrinks to 0. For $\alpha = 3.66$ (the right-most plot) this gap is around 0.006.

15 Cavity Method: Basic Concepts

Message passing and spatial coupling techniques have been very successful in providing efficient algorithms in the realm of coding and compressive sensing. Furthermore the variational method has allowed us to derive the phase diagram for these models, and the Maxwell construction ties the two approaches together. On the other hand these methods are not as successful for constraint satisfaction problems such as K-SAT. For example, plain BP does not allow to find solutions and had to be supplemented by a decimation process. BP guided decimation finds solutions up to some density, but it is not clear if this limitation corresponds to some sort of fundamental dynamic threshold, similar to the BP threshold say. Also (for the moment) we are not able to find the SAT-UNSAT threshold by a sort of Maxwell construction or spatial coupling technique. At the same time the RS entropy functional does not count correctly the number of solutions.

The success of message passing marginalization is related to the absence of long range correlations between dynamical variables. In constraint satisfaction problems such as K-SAT long range correlations are present and it is not possible to only take into account a tree like neighborhood of a node when its marginal is computed. The boundary conditions at the leaf nodes of the tree like neighborhood somehow matter. Often, in statistical mechanics, when long range correlations are present, the key to the analysis comes from the concept of extremal measure and convex decomposition of the Gibbs measure into extremal measures. While these notions are relatively well understood and mathematically precise for low dimensional deterministic Ising models on regular grids, the mathematical theory in the context of spin glass type models is still very much of an open challenge. As we will see the cavity method boldly pushes the idea of convex decomposition of the Gibbs distribution to its limit in the sense that we will have to deal with a convex superposition with an exponentially large number of extremal measures. Once this is accepted, the theory, although technically challenging, flows. Indeed it turns out this convex superposition defines a new factor graph model which can again be analyzed by the message passing, variational free energy and spatial coupling techniques. That we can again apply these techniques "one level up" is one of the fascinating aspects of the subject.

15.1 Notion of Pure State

The concept of extremal measure or pure state has not been introduced nor used explicitely yet, but this is the time to do so. We start by a very brief discussion in the context of the Ising model because this is the simplest best understood non-trivial paradigmatic situation. We then turn our attention to the CW model, for which this notion is somewhat special due to the absence of geometry, but allows to introduce a very useful heuristic point of view that lends itself to generalizations.

A diggression on the Ising model

The construction of infinite volume Gibbs measures is a non-trivial problem whose mathematical theory is developped mainly for Ising type models on regular grids, say $\mathbb{Z}^d$. Here we summarize very briefly and informally the main picture for the classical two dimensional Ising model with nearest neighbord ferromagnetic interactions. for which the theory is fully controlled, and the interested reader will find pointers to the litterature in the notes. The phase diagram of this model is qualitatively the same as the one of CW. The mathematical theory of the Gibbs states for infinite volume starts with the Gibbs distribution on a finite square grid $\Lambda \subset \mathbb{Z}^d$ with specified boundary conditions. The boundary conditions amount to fix the spin assignments on vertices of $\partial\Lambda$. One computes the infinite volume limit of all marginals, given the boundary conditions, and the set of these marginals defines the infinite volume Gibbs state. For any point of the (T, h) plane the set of all possible infinite volume Gibbs states is convex. Away from the coexistence line this set is trivially a point i.e, the infinite volume limits of the marginals is independent of boundary conditions. On the coexistence line the set of infinite volume limits is non-trivial. It has two extremal measures obtained by the all $+1$ and all -1 boundary conditions, and in particular $\langle s_i\rangle_\pm = \pm m \neq 0$. All other states on the coexistence line are of the form $\langle - \rangle_w = w\langle - \rangle_+ + (1-w)\langle - \rangle_-$. Extremal states have correlations that satisfy the exponential decay property; this holds when the state is unique and for the $+$ and $-$ states. For example, $|\langle s_i s_j\rangle_\pm - \langle s_i\rangle_\pm \langle s_j\rangle_\pm| \leq \text{const}\, e^{-|i-j|/\xi(T)}$ where $\xi(T)$ is a finite correlation length.[1] On the other hand mixed states with $w \neq 0, 1$ have long range order which means $\lim_{|i-j|\to+\infty}(\langle s_i s_j\rangle_w - \langle s_i\rangle_w \langle s_j\rangle_w) \neq 0$. As a good exercise one can check that the clustering property of pure states implies this limit is equal to $4w(1-w)m^2$.

The CW model revisited

On the complete graph there is no boundary so we simply start with the model on a finite graph with a fixed constant magnetic field. We saw in Chapter 4 that

[1] This length diverges when T approaches the critical temperature.

in the (T, h) plane there is a the coexistence line on which the magnetization can take two different values in the sense that $\lim_{h\to 0_\pm} \lim_{n\to+\infty} \langle s_i \rangle = \pm m \neq 1$. The magnetization is uniquely defined away from this line in the sense that it is an analytic function of h and T. It is not difficult to show that this feature is shared by any average $\langle s_{i_1} \dots s_{i_k} \rangle$, for any finite set of spins. In this sense the infinite Gibbs state is unique and "pure" away from the coexistence line, and is not unique on this line. There, one can define two "pure states" $\langle s_{i_1} \dots s_{i_k} \rangle_\pm = \lim_{h\to 0_\pm} \lim_{n\to+\infty} \langle s_{i_1} \dots s_{i_k} \rangle$, and also any convex superposition $\langle - \rangle_w = w\langle - \rangle_+ + (1-w)\langle - \rangle_-$ for $0 < w < 1$. For the CW model the "pure" states satisfy an extreme form of clustering where variables *decouple* in thermodynamic limit. For example for $k = 2$ $\langle s_i s_j \rangle_\pm - \langle s_i \rangle_\pm \langle s_j \rangle_\pm = 0$.[2] Genuine superpositions (mixed states) have correlations that do not vanish in the thermodynamic limit. For example, the decoupling property implies $\langle s_i s_j \rangle_w - \langle s_i \rangle_w \langle s_j \rangle_w = 4w(1-w)m^2$ on the coexistence line for any $i \neq j$. Remark for the Ising model the same relation is obtained for $|i - j| \to +\infty$.

For the CW model there is a one to one correspondence between "pure states" and minima of the free energy function $f(m)$ appearing in the variational expression for $-n^{-1} \ln Z$. This is an extremely simple instance of the landscape picture discussed in the next paragraph.

The landscape picture

For spin glass models the situation is not "as simple". It is not known how to define a mathematically sound notion of extremal state. For models on complete or sparse locally tree like graphs one heuristic and intuitive approach identifies the extremal states with global or quasi-global minima of the TAP or Bethe type free energy functionals[3]. Let $(\mu^{(p)}_{i\to a}, \hat{\mu}^{(p)}_{a\to i}) = (\underline{\mu}^{(p)}, \underline{\hat{\mu}}^{(p)})$ be the corresponding solutions of the sum-product equations where p indexes the minima. From these mesages one can reconstruct marginals $\nu^{(p)}(.)$ which define the "extremal measure". To distinguish this measure from the usual notion of extremal state and to avoid confusions we will call this an *extremal or pure Bethe measure*. One has to think of it as a "proxy" for an ideal notion of pure state. When message passing iterations converge one expects that there are a small number of fixed points with well defined bassins of attraction and the number of pure Bethe states is small. However when these iterations do not converge this may be due to the presence of a very large number of fixed points, and thus to a very large number of minima in the Bethe free energy. In such situations one expects a large number of pure Bethe states. This happens in the TAP approach to the SK model for the region of the phase diagram below the AT line. This

[2] The CW model is a bit special in this respect because the complete graph wipes out any trace of geometry. For finite n and any h one has $\langle s_i s_j \rangle - \langle s_i \rangle \langle s_j \rangle = O(n^{-1})$. Since there is a unit distance between any two variables, one may interpret this as a exponential decay of correlations on a length scale $O(1/\ln n)$.

[3] It is debated wether such an approach is valid for low dimensional spin glasses e.g the Edwards-Anderson model

also happens in K-SAT for clause densities slightly above the ones found by BP guided decimation. The reason for the failure of BP guided decimation is the proliferation of minima in the Bethe free energy. Free energy functions with a proliferation of numerous minima are often called free energy landscapes. Figure **??** serves as a useful mental picture summarizing these ideas.

15.2 The Level-One Model

The convex decomposition ansatz

We formalize the heuristic landscape picture. The cavity method assumes that: (i) The Gibbs distribution is a convex sum of "pure states"; (ii) Pure states are identified with the Bethe measures corresponding to minima of the free energy; (iii) The weights of the convex superposition are determined by the Bethe free energy minima. We write

$$\mu\left(\underline{x}\right)=\sum_{p=1}^{\mathcal{N}}\frac{e^{-xF^{(p)}}}{Z(x)}\mu^{(p)}(\underline{x}),\qquad Z(x)=\sum_{p=1}^{\mathcal{N}}e^{-xF^{(p)}} \tag{15.1}$$

The sum runs over p which indexes the minima $\{\mu^{(p)}_{a\to i},\hat{\mu}^{(p)}_{a\to i}\}=(\underline{\mu}^{(p)},\underline{\hat{\mu}}^{(p)})$ of the Bethe free energy functional. The weights are determined by the free energy of these minima $F^{(p)}=F_{\text{Bethe}}\left(\underline{\mu}^{(p)},\underline{\hat{\mu}}^{(p)}\right)$. The "pure" Bethe measures $\mu^{(p)}(\underline{x})$ are defined through the collection of all their marginals, which themselves are determined from $(\underline{\mu}^{(p)},\underline{\hat{\mu}}^{(p)})$. The role of x, called the "Parisi parameter", turns out to be quite subtle.[4] For the moment one can think of it as a multiplicative "renormalization" of the temperature. In a large portion of the phase diagram the naive choice $x=1$ is correct. However we will see that there are regions of the phase diagram where values $0<x<1$ are forced upon us.

Level-one auxiliary model

In order to make technical progress with the convex decomposition ansatz we make one more assumption. One expects that at low temperatures when there are an exponential number of minima, these are exponentially more numerous than maxima and saddle points. Therefore we assume: (iv) the sum over p runs over *all* stationnary points of the Bethe free energy i.e, fixed points solutions of the sum-product equations.

The partition function (15.1) can be thought of as the one of a statistical mechanics system with dynamical variables $(\underline{\mu}^{(p)},\underline{\hat{\mu}}^{(p)})$ and effective Hamiltonian

[4] The notation x is traditional and should not be confused with the one for configurations $\underline{x}$. This parameter was first introduced by parisi in the context of the replica approach. There its role is even more mysterious an appears as an integer that is anlytically continued to values in $]0,1[$.

given by the Bethe free energy. Using assumption (iv) we are led to study the Gibbs probability distribution of an auxiliary model, called the "level-one model"

$$\mu_1(\underline{\mu},\underline{\hat{\mu}}) = \frac{1}{Z_1(x)} e^{-xF_{\text{Bethe}}\,(\underline{\mu},\underline{\hat{\mu}})}\mathbb{1}_{\text{sp}}(\underline{\mu},\underline{\hat{\mu}}) \tag{15.2}$$

and

$$Z_1(x) = \sum_{\underline{\mu},\underline{\hat{\mu}}} e^{-xF_{\text{Bethe}}\,(\underline{\mu},\underline{\hat{\mu}})}\mathbb{1}_{\text{sp}}(\underline{\mu},\underline{\hat{\mu}}) \tag{15.3}$$

The indicator function $\mathbb{1}_{\text{sp}}(\underline{\mu},\underline{\hat{\mu}})$ selects solutions of the sum product fixed point equations. Recall that in the sum-product equations and the Bethe free energy the normalization of the messages is arbitrary. In order for the sum in (15.3) to be well defined we have to fix a normalization. We will take the most natural one, namely $\sum_{x_i}\mu_{i\to a}(x_i) = \sum_{x_i}\hat{\mu}_{a\to i}(x_i) = 1$. With this normalization the sum product equations used in subsequent calculations read

$$\mu_{i\to a}(x_i) = \frac{\prod_{b\in\partial i\setminus a}\hat{\mu}_{b\to i}(x_i)}{\sum_{x_i}\prod_{b\in\partial i\setminus a}\hat{\mu}_{b\to i}(x_i)} \tag{15.4}$$

$$\hat{\mu}_{a\to i}(x_i) = \frac{\sum_{\sim x_i} f_a(x_{\partial a})\prod_{j\in\partial a\setminus i}\mu_{j\to a}(x_i)}{\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{j\in\partial a\setminus i}\mu_{j\to a}(x_i)} \tag{15.5}$$

Let us immediately give a few definitions that will be useful to us later on. Averages with respect to (15.2) are denoted by the usual bracket notation $\langle - \rangle_1$. The level-one free energy is defined as usual $f_1(x) = -\frac{1}{nx}\ln Z_1(x)$. As in Chapter 2, the free energy allows to compute numerous other quantities by differentiations with respect to the inverse temperature, here with respect to x. The level-one internal energy is $u_1(x) = \langle F_{\text{Bethe}}\rangle_1/n = \frac{\partial}{\partial x}f_1(x)$. The Shannon-Gibbs entropy associated to (15.2) is equal to $\Sigma(x) = x^2\frac{\partial}{\partial x}f_1(x) = u_1(x) - x^{-1}\Sigma(x)$.

Choice of the Parisi parameter

Small paragraph to be written. Explain briefly. Interpret $\Sigma(x)$.

15.3 Message passing, Bethe free energy and complexity one level up

Message passing

We now show how the level-one model is solved in practice. The main idea is to first recognize that the model is defined on a sparse factor graph and apply again the sum-product and Bethe formulas. If $\Gamma = (V, C, E)$ is the original factor graph, then the level-one model has the factor graph $\Gamma_1 = (V_1, C_1, E_1)$ described on Fig. 15.1. We use the shorthand notation $\mathbb{1}_i$ and $\hat{\mathbb{1}}_a$ for the indicator functions

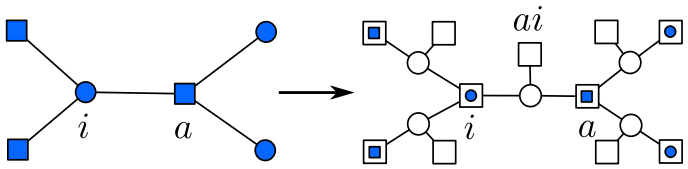


Figure 15.1 On the left, an exemple of an original graph Γ. On the right its corresponding graph Γ_1 for the level-one model.

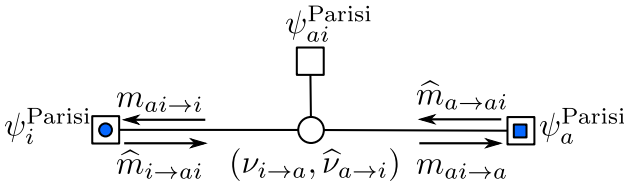


Figure 15.2 Messages are labbeled with m if they outgoing from a Parisi variable node are and with $\widehat{m}$ if they are outgoing from a Parisi function node.

forcing equations (15.4)-(15.5). Thus $\mathbb{1}(\underline{\mu}, \underline{\hat{\mu}}) = \prod_i \mathbb{1}_i \prod_a \hat{\mathbb{1}}_a$. A variable node $i \in V$, becomes a function node $i \in C_1$, with the function

$$\psi_i = e^{-xF_i} \prod_{a \in \partial i} \mathbb{1}_i. \tag{15.6}$$

A function node $a \in C$ remains a function node $a \in C_1$ with factor

$$\psi_a = e^{-xF_a} \prod_{i \in \partial a} \hat{\mathbb{1}}_a. \tag{15.7}$$

An edge $(a, i) \in E$, becomes a variable node $(a, i) \in V_1$. There is also an extra function node attached to each variable node of the new graph, or equivalently attached to each edge of the old graph. The corresponding function is

$$\psi_{ai} = e^{+xF_{ai}}. \tag{15.8}$$

With these definitions (15.2) can be written as

$$\mu_1(\underline{\mu}, \underline{\widehat{\mu}}) = \frac{1}{Z_1(x)} \prod_{i \in V} \psi_i \prod_{a \in C} \psi_a \prod_{ai \in E} \psi_{ai}. \tag{15.9}$$

The sum product equations for (15.9) involve four kind of messages shown on figure 15.2. Messages flowing from a new function node to a new variable node satisfy (the symbol $\simeq$ means equal up to a normalization factor)

$$\begin{aligned} \widehat{m}_{a\to ai} &\simeq \sum_{\sim(\mu_{i\to a}, \widehat{\mu}_{a\to i})} \psi_a \prod_{aj \in \partial a \setminus ai} m_{aj\to a} \\ &= e^{xF_{ai}} \sum_{\sim(\mu_{i\to a}, \widehat{\mu}_{a\to i})} \hat{\mathbb{1}}_a(\widehat{\mu}_{a\to i}) e^{-x(F_a - F_{ai})} \prod_{aj \in \partial a \setminus ai} m_{aj\to a} \end{aligned}$$

and

$$\begin{aligned}\widehat{m}_{i\to ai} &\simeq \sum_{\sim(\mu_{i\to a},\widehat{\mu}_{a\to i})} \psi_i \prod_{bi\in\partial i\setminus ai} \widehat{m}_{bi\to i} \\ &= e^{xF_{ai}} \sum_{\sim(\mu_{i\to a},\widehat{\mu}_{a\to i})} \mathbb{1}_i(\mu_{i\to a}) e^{-x(F_i-F_{ai})} \prod_{bi\in\partial i\setminus ai} \widehat{m}_{bi\to i}\end{aligned}$$

Messages from a new function node to a new variable node satisfy

$$m_{ai\to i} \simeq e^{xF_{ai}}\widehat{m}_{a\to ai}, \qquad m_{ai\to a} \simeq e^{xF_{ai}}\widehat{m}_{i\to ai}.$$

Notice that $m_{ai\to i}$ and $m_{ai\to a}$ are independent of $\widehat{\mu}_{a\to i}$ and $\mu_{i\to a}$ respectively; this allows us to simplify the message passing equations. To achieve the simplification define two distributions

$$Q_{i\to a}(\mu_{i\to a}) = m_{ai\to a}, \qquad \widehat{Q}_{a\to i}(\widehat{\mu}_{a\to i}) = m_{ai\to i}$$

These flow on the edges of the original factor graph $\Gamma = (V, C, E)$ and are called *cavity messages*. It is easy to see that they satisfy

$$\widehat{Q}_{a\to i}(\widehat{\mu}_{a\to i}) \simeq \sum_{\underline{\mu}} \hat{\mathbb{1}}_a(\widehat{\mu}_{a\to i}) e^{-x(F_a-F_{ai})} \prod_{j\in\partial a\setminus i} Q_{j\to a}(\mu_{j\to a}) \tag{15.10}$$

$$Q_{i\to a}(\mu_{i\to a}) \simeq \sum_{\underline{\widehat{\mu}}} \mathbb{1}_i(\mu_{i\to a}) e^{-x(F_i-F_{ai})} \prod_{b\in\partial i\setminus a} \widehat{Q}_{b\to i}(\widehat{\mu}_{b\to i}). \tag{15.11}$$

These are the *cavity equations*, an instance of sum-product equations for the level-one model. Note that the cavity equations do not make any reference to the graph Γ_1 and we can now revert to the original one. As usual, if the graph was a tree, these equations give the exact marginals of (15.2).

The x dependent exponentials are sometimes called reweighting factors. Their explicit expression will be useful later on,

$$e^{-(F_i-F_{ai})} = \sum_{x_i} \prod_{b\in\partial i\setminus a} \widehat{\mu}_{b\to i}(x_i), \qquad e^{-(F_a-F_{ai})} = \sum_{x_{\partial a}} f_a(x_{\partial a}) \prod_{\partial j\in a\setminus i} \mu_{j\to a}(x_i) \tag{15.12}$$

Note that these are in fact the normalization factors in (15.4)-(15.5).

Bethe free energy and complexity

The Bethe free energy functional of the level-one model is a functional of the cavity messages $Q_{i\to a}, \widehat{Q}_{a\to i}$. We could derive it as in Chapter **??** by first deriving the exact free energy $f_1(x)$ on a tree, and then take this expression as a defition for general graph instances. But we can also guess the fomula. It is basically given by the usual definition, but with the extra feature that it must contain the reweighting factors. Moreover its stationary points must yield (**??**). This is enough information to guess that

$$\mathcal{F}_{\text{Bethe}}(\underline{Q}, \underline{\widehat{Q}}) = \sum_{i\in V} \mathcal{F}_i + \sum_{a\in C} \mathcal{F}_a - \sum_{ai\in E} \mathcal{F}_{ai} \tag{15.13}$$

where

$$\mathcal{F}_i(\{\widehat{Q}_{b\to i}\}_{b\in\partial i}) = -\frac{1}{x}\ln\Big\{\sum_{\underline{\widehat{\mu}}} e^{-xF_i}\prod_{b\in\partial i}\widehat{Q}_{b\to i}\Big\},$$

$$\mathcal{F}_a(\{Q_{j\to a}\}_{j\in\partial a}) = -\frac{1}{x}\ln\Big\{\sum_{\underline{\mu}} e^{-xF_a}\prod_{j\in\partial a} Q_{j\to a}\Big\},$$

$$\mathcal{F}_{ai}(Q_{i\to a},\widehat{Q}_{a\to i}) = -\frac{1}{x}\ln\Big\{\sum_{\mu,\widehat{\mu}} e^{-xF_{ai}}Q_{i\to a}\widehat{Q}_{a\to i}\Big\}.$$

The complexity functional within the Bethe formalism is given by $\Sigma_{\rm Bethe} = x^2\frac{\partial}{\partial x}\mathcal{F}_{\rm Bethe}$. Explicitly,

$$\Sigma_{\rm Bethe}(\underline{Q},\underline{\widehat{Q}}) = \sum_{i\in V}\Sigma_i + \sum_{a\in C}\Sigma_a - \sum_{ai\in E}\Sigma_{ai} \tag{15.14}$$

where

$$x^{-1}\Sigma_i(\{\widehat{Q}_{b\to i}\}_{b\in\partial i}) = -\mathcal{F}_i + \frac{\sum_{\underline{\widehat{\mu}}} F_i e^{-xF_i}\prod_{b\in\partial i}\widehat{Q}_{b\to i}}{\sum_{\underline{\widehat{\mu}}} e^{-xF_i}\prod_{b\in\partial i}\widehat{Q}_{b\to i}},$$

$$x^{-1}\Sigma_a(\{Q_{j\to a}\}_{j\in\partial a}) = -\mathcal{F}_a + \frac{\sum_{\underline{\mu}} F_a e^{-xF_a}\prod_{j\in\partial a} Q_{j\to a}}{\sum_{\underline{\mu}} e^{-xF_a}\prod_{j\in\partial a} Q_{j\to a}},$$

$$x^{-1}\Sigma_{ai}(Q_{i\to a},\widehat{Q}_{a\to i}) = -\mathcal{F}_{ai} + \frac{\sum_{\mu,\widehat{\mu}} F_{ai} e^{-xF_{ai}}Q_{i\to a}\widehat{Q}_{a\to i}}{\sum_{\mu,\widehat{\mu}} e^{-xF_{ai}}Q_{i\to a}\widehat{Q}_{a\to i}}.$$

One can interpret the Bethe complexity as the difference of the Bethe free energy of the level-one model and a Bethe expression for the internal energy of the level one model,

$$x^{-1}\Sigma_{\rm Bethe} = \mathcal{F}_{\rm Bethe} - \langle F_{\rm Bethe}\rangle_{\rm cav}. \tag{15.15}$$

The bracket $\langle -\rangle_{\rm cav}$ is a natural average that can be read off from the above formulas.

Simplifications for $x = 1$

As alluded to before $x = 1$ plays a specially important role. So it is fortunate that a large portion of the formalism above can be simplified by eliminating entirely the need for reweighting factors. This makes the replica analysis much simpler and allows to make much simpler and precise numerical computations (e.g. by population dynamics) .

Let us first discuss the level-one Bethe free energy. Replacing (11.12), (11.13) and (11.14) into (15.13) one finds

$$\mathcal{F}_{\rm Bethe}(\underline{Q},\underline{\widehat{Q}})|_{x=1} = F_{\rm Bethe}(\underline{\mu}^{\rm av},\underline{\hat{\mu}}^{\rm av}) \tag{15.16}$$

which is the *usual* Bethe free energy expressed in terms of "average messages",

$$\mu^{\rm av}_{i\to a}(x_i) = \sum_{\mu_{i\to a}} \mu_{i\to a}(x_i) Q_{i\to a}(\mu_{i\to a}), \quad \hat\mu^{\rm av}_{a\to i}(x_i) = \sum_{\hat\mu_{a\to i}} \hat\mu_{a\to i}(x_i) \hat Q_{a\to i}(\hat\mu_{a\to i}).$$

Remarkably, the average messages satisfy the usual sum-product equations,

$$\mu^{\rm av}_{i\to a}(x_i) \simeq \sum_{x_i} \prod_{b\in\partial i\setminus a} \hat\mu^{\rm av}_{b\to i}(x_i), \qquad \hat\mu^{\rm av}_{i\to a}(x_i) \simeq \sum_{x_{\partial a}} f_a(x_{\partial a}) \prod_{j\in\partial a\setminus i} \mu^{\rm av}_{j\to a}(x_j).$$

One way to prove this is to notice that[5] $\delta_{Q_{i\to a}}\mathcal{F}_{\rm Bethe} = (\delta_{\mu^{\rm av}_{i\to a}} F_{\rm Bethe})\mu_{i\to a}(x_i)$ and $\delta_{\hat Q_{i\to a}}\mathcal{F}_{\rm Bethe} = (\delta_{\hat\mu^{\rm av}_{i\to a}} F_{\rm Bethe})\hat\mu_{i\to a}(x_i)$. Therefore if $(\underline{Q},\underline{\hat Q})$ is a stationary point of $\mathcal{F}_{\rm Bethe}|_{x=1}$ then $(\mu^{\rm av},\hat\mu^{\rm av})$ is a stationary point of $F_{\rm Bethe}$. Thus the cavity equations for $(\underline{Q},\underline{\hat Q})$ imply the sum-product equations for $(\mu^{\rm av},\hat\mu^{\rm av})$. This conclusion can also be reached by a direct calculation starting from the cavity equations for $x = 1$.

Conceptually $\mu^{\rm av}_{i\to a}(x_i)$ and $\hat\mu^{\rm av}_{i\to a}(x_i)$ are very natural messages to consider. Suppose for the sake of the argument that $Q(\mu_{i\to a})$ and $\hat Q(\hat\mu_{i\to a})$ are the true marginals of the level-one model. Then the average messages are the Gibbs averages of the dynamical variables of the level-one model (much like the magnetization is the Gibbs average of the spin variable). In other words if we sample among the set of solutions of the sum-product equations according to the weight $e^{-F_{\rm Bethe}}/Z_1(x=1)$ these are the expected messages that we get. From these expected messages one can reconstruct a Bethe measure which one can hope to be a good proxy for the convex superposition. However this is *not* a pure Bethe measure. As a consequence the marginals of this Bethe measure do not allow us to correctly sample from pure states $\mu^{(p)}(\underline{x})$. In particular for K-SAT they do not allow us to find solutions, and this is why BP guided decimation does not succeed above a certain density. When it does succeed this means that the the convex decomposition is essentially dominated by a unique Bethe measure (which is pure). The correct sampling procedure that suitably addresses these points is Survey Propagation guided decimation discussed in Chapter **??**.

We now turn to the Bethe complexity (15.15) for $x = 1$. For the free energy contribution we already have the simplification (15.16), so we only have to show how to eliminate the reweighting factors from the internal energy contribution.

[5] Formally $\delta_R G$ is an infinitesimal variation of G with respect to R.

Replacing (11.12) in $\langle F_i\rangle_{\rm cav}$ we find

$$\begin{aligned}\langle F_i\rangle_{\rm cav} &= \frac{\sum_{\underline{\hat{\mu}}} \ln\Big\{\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}_{b\to i}(x_i)\Big\}\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}_{b\to i}(x_i)\hat{Q}_{b\to i}}{\sum_{\underline{\hat{\mu}}}\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}_{b\to i}(x_i)\hat{Q}_{b\to i}}\\ &= \frac{\sum_{\underline{\hat{\mu}}} \ln\Big\{\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}_{b\to i}(x_i)\Big\}\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}_{b\to i}(x_i)\hat{Q}_{b\to i}}{\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}^{\rm av}_{b\to i}(x_i)}\\ &= \sum_{\underline{\hat{\mu}}}\ln\Big\{\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}_{b\to i}(x_i)\Big\}\sum_{x_i}\nu^{\rm av}_i(x_i)\prod_{b\in\partial i}\hat{R}_{b\to i}(\hat{\mu}_{b\to i}|x_i)\end{aligned}$$

In the last equality we have defined the probability distributions

$$\nu^{\rm av}_i(x_i) = \frac{\prod_{b\in\partial i}\hat{\mu}^{\rm av}_{b\to i}(x_i)}{\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}^{\rm av}_{b\to i}(x_i)}, \qquad \hat{R}_{b\to i}(\hat{\mu}_{b\to i}|x_i) = \frac{\hat{\mu}_{b\to i}(x_i)\hat{Q}_{b\to i}}{\hat{\mu}^{\rm av}_{b\to i}(x_i)}$$

Replacing (11.13) in $\langle F_a\rangle_{\rm cav}$ we find

$$\begin{aligned}\langle F_a\rangle_{\rm cav} &= \frac{\sum_{\underline{\mu}} \ln\Big\{\sum_{x_{\partial a}}\prod_{i\in\partial a}\mu_{i\to a}(x_i)\Big\}\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu_{i\to a}(x_i)Q_{i\to a}}{\sum_{\underline{\mu}}\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu_{i\to a}(x_i)\hat{Q}_{i\to a}}\\ &= \frac{\sum_{\underline{\mu}} \ln\Big\{\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu_{i\to a}(x_i)\Big\}\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu_{i\to a}(x_i)Q_{i\to a}}{\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu^{\rm av}_{i\to a}(x_i)}\\ &= \sum_{\underline{\mu}}\ln\Big\{\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu_{i\to a}(x_i)\Big\}\sum_{x_{\partial a}}\nu^{\rm av}_a(x_{\partial a})\prod_{i\in\partial a}R_{i\to a}(\mu_{i\to a}|x_i)\end{aligned}$$

with the distributions

$$\nu^{\rm av}_a(x_{\partial a}) = \frac{f_a(x_{\partial a})\prod_{i\in\partial a}\mu^{\rm av}_{i\to a}(x_i)}{\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu^{\rm av}_{i\to a}(x_i)}, \qquad R_{i\to a}(\mu_{i\to a}|x_i) = \frac{\mu_{i\to a}(x_i)Q_{i\to a}}{\mu^{\rm av}_{i\to a}(x_i)}$$

Replacing (11.14) in $\langle F_{ai}\rangle_{\rm cav}$ we find

$$\begin{aligned}\langle F_{ai}\rangle_{\rm cav} &= \frac{\sum_{\underline{\mu},\underline{\hat{\mu}}} \ln\Big\{\sum_{x_i}\hat{\mu}_{a\to i}(x_i)\mu_{i\to a}(x_i)\Big\}\sum_{x_i}\hat{\mu}_{a\to i}(x_i)\hat{Q}_{a\to i}\mu_{i\to a}(x_i)Q_{i\to a}}{\sum_{\underline{\mu},\underline{\hat{\mu}}}\sum_{x_i}\hat{\mu}_{a\to i}(x_i)\hat{Q}_{a\to i}\mu_{i\to a}(x_i)Q_{i\to a}}\\ &= \frac{\sum_{\underline{\mu},\underline{\hat{\mu}}} \ln\Big\{\sum_{x_i}\hat{\mu}_{a\to i}(x_i)\mu_{i\to a}(x_i)\Big\}\sum_{x_i}\hat{\mu}_{a\to i}(x_i)\hat{Q}_{a\to i}\mu_{i\to a}(x_i)Q_{i\to a}}{\sum_{x_i}\hat{\mu}^{\rm av}_{a\to i}(x_i)\mu^{\rm av}_{i\to a}(x_i)}\\ &= \sum_{\underline{\mu},\underline{\hat{\mu}}}\ln\Big\{\sum_{x_i}\hat{\mu}_{a\to i}(x_i)\mu_{i\to a}(x_i)\Big\}\sum_{x_i}\nu_{ai}(x_i)\hat{R}_{a\to i}(\hat{\mu}_{a\to i}|x_i)R_{i\to a}(\mu_{i\to a}|x_i)\end{aligned}$$

where

$$\nu^{\rm av}_{ai}(x_i) = \frac{\hat{\mu}^{\rm av}_{a\to i}(x_i)\mu^{\rm av}_{i\to a}(x_i)}{\sum_{x_i}\hat{\mu}^{\rm av}_{a\to i}(x_i)\mu^{\rm av}_{i\to a}(x_i)}$$

So far we have shown that the Bethe complexity can be expressed in terms of the average messages $\hat{\mu}^{\rm av}_{a\to i}$ and $\mu^{\rm av}_{i\to a}$ and the conditional distributions $\hat{R}_{a\to i}(\hat{\mu}_{a\to i}|x_i)$ and $R_{i\to a}(\mu_{i\to a}|x_i)$. We have already seen that the average messages satisfy the usual sum-product equations. We will now show that the conditional distributions satisfy similar equations.

Multiplying the cavity equations (15.10)-(15.11) by $\mu_{a\to i}(x_i)$ and $\hat{\mu}_{a\to i}(x_i)$, and using the expressions of the reweighting factor (15.12) we get for $x = 1$

$$\mu_{i\to a}(x_i)Q_{i\to a}(\mu_{i\to a}) \simeq \sum_{\underline{\widehat{\mu}}} \mathbb{1}_i(\mu_{i\to a}) \prod_{b\in\partial i\setminus a} \widehat{\mu}_{b\to i}(x_i)\widehat{Q}_{b\to i}(\widehat{\mu}_{b\to i})$$

$$\hat{\mu}_{a\to i}(x_i)\widehat{Q}_{a\to i}(\widehat{\mu}_{a\to i}) \simeq \sum_{\sim x_i} f_a(x_{\partial a}) \sum_{\underline{\mu}} \hat{\mathbb{1}}_a(\widehat{\mu}_{a\to i}) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(x_j)Q_{j\to a}(\mu_{j\to a})$$

If we normalize each member of these equalities the proportionality relations become equalities. Here normalizing means dividing by the sums of the numerators over $\mu_{i\to a}$ and $\hat{\mu}_{i\to a}$. One finds a closed set of equations linking the conditional distributions,

$$R_{i\to a}(\mu_{i\to a}|x_i) = \sum_{\underline{\widehat{\mu}}} \mathbb{1}_i(\mu_{i\to a}) \prod_{b\in\partial i\setminus a} \hat{R}_{b\to i}(\hat{\mu}_{b\to i}|x_i) \tag{15.17}$$

$$\hat{R}_{a\to i}(\hat{\mu}_{a\to i}|x_i) = \sum_{\sim x_i} \pi_{a,i}(x_{\partial a\setminus i}|x_i) \sum_{\underline{\mu}} \hat{\mathbb{1}}_a(\widehat{\mu}_{a\to i}) \prod_{j\in\partial a\setminus i} R_{j\to a}(\mu_{j\to a}|x_j) \tag{15.18}$$

where

$$\pi_{a,i}(x_{\partial a\setminus i}|x_i) = \frac{f_a(x_{\partial a})\prod_{j\in\partial a\setminus i}\mu^{\rm av}_{j\to a}(x_j)}{\sum_{\sim x_i} f_a(x_{\partial a})\prod_{j\in\partial a\setminus i}\mu^{\rm av}_{j\to a}(x_j)}$$

These equations are quite similar to standard sum-product equations and are much easier to solve than the original cavity equations.

15.4 Application to K-SAT

We work at finite temperature for reasons that will become clear below. It is straightforward to apply the general theory to K-SAT using the parametrization of messages (9.24). With this parametrization the sum-product equations become (9.27)-(9.31) (with the necessary modification for finite temperatures) so to write the cavity equations (15.10)-(15.11) we make the replacements

$$\mathbb{1}_i \to \delta\Big(h_{i\to a} - \sum_{b\in S_{ia}} \hat{h}_{b\to i} + \sum_{b\in U_{ia}} \hat{h}_{b\to i})$$

and

$$\hat{\mathbb{1}}_a \to \delta\Big(\hat{h}_{a\to i} + \frac{1}{2}\ln\Big\{1 - (1-e^{-\beta}) \prod_{j\in\partial a\setminus i} \frac{1-\tanh_{j\to a}}{2}\Big\}\Big).$$

Furthermore all sums become integrals (dropping subscripts) $\sum_\mu Q(\mu)\cdots \to \int dh Q(h)\ldots$ and $\sum_{\hat\mu} \hat Q(\hat\mu)\cdots \to \int d\hat h \hat Q(\hat h)\ldots$.

To get the general expressions for the level-one Bethe free energy and complexity (15.13), (15.14) one uses F_i, F_a and F_{ai} given in (11.23)-(11.25) and replaces sums by integrals as just indicated.

For the simplified formulas when $x = 1$ we introduce averaged messages

$$\tanh h^{\rm av}_{i\to a} = \int Q(h_{i\to a}) \tanh h_{i\to a}, \qquad \tanh \hat h^{\rm av}_{i\to a} = \int \hat Q(h_{i\to a}) \tanh \hat h_{i\to a}$$

which satisfy the finite temperature version of message passing equations (9.27)-(9.31). With these average messages the level-one Bethe free energy is the same than (11.21), i.e. it is given by the RS expression. The other set of message passing equations (15.17), (15.18) are obtained by replacing indicator functions by Dirac functions as above, $x_i \to s_i$, and (dropping subscripts) $\sum_\mu R(\mu|x_i)\cdots \to \int dh R(h|x_i)\ldots$, $\sum_\mu \hat R(\hat\mu|x_i)\cdots \to \int d\hat h \hat R(\hat h|x_i)\ldots$. With all these ingredients one also writes down the Bethe complexity for $x = 1$. This is left as an exercise.

15.5 Replica Symmetry Broken Analysis for K-SAT

General analysis

The phase diagram of K-SAT is derived from the cavity equations and the Bethe formulas through a "density evolution type" analysis done at the level of the cavity messages $Q_{i\to a}(.)$, $\hat Q_{i\to a}(.)$. One can write down formal equations linking probability distributions of the cavity messages $\mathcal{Q}(Q(.))$ and $\hat{\mathcal{Q}}(\hat Q(.))$ which are often called replica symmetry broken (1-RSB) equations. The associated average level-one free energy functional is the 1-RSB free energy.[6] Let us illustrate the 1RSB replica formula for the free energy in more detail.

Fix a trial distribution $\mathcal{Q}(Q(.))$. Take $K-1$ iid copies of the random distribution $Q(.)$ and define the random variable $\hat Q(.)$ [compute reweighting factor in here]

$$\hat Q(\hat\xi) \stackrel{\rm distr}{=} \int \prod_{k=1}^{K-1} d\xi_k Q(h_k) \left(2 - \prod_{k=1}^{K-1} \frac{1-\tanh h_k}{2}\right)^x \tag{15.19}$$

$$\times \frac{\delta\left(\hat h + \frac{1}{2}\ln\left\{1 - (1-e^{-\beta})\prod_{k=1}^{K-1}\frac{1-\tanh h_k}{2}\right\}\right)}{\int \prod_{k=1}^{K-1} d\xi_k Q(h_k)\left(2 - \prod_{k=1}^{K-1}\frac{1-\tanh h_k}{2}\right)^x} \tag{15.20}$$

This random distribution is distributed according to $\hat{\mathcal{Q}}(\hat Q(.))$. Pick two Poisson integers p and q of mean $\alpha K/2$ and $p+q$ iid copies of the random distribution

[6] Historically these equations were first derived in the context of the replica method and involve breaking the symmetry between replicas of the original system, hence the name.

$\hat{Q}(.)$. Let

$$
\begin{aligned}
& f(Q(.),\hat{Q}(.),p,q) \\
& = x^{-1}\ln\Bigg\{\int\prod_{k=1}^{p+q}d\hat{h}_k\hat{Q}_k(\hat{h}_k)\left(\prod_{k=1}^{p}(1-\tanh\hat{h}_k)\prod_{k=p+1}^{p+q}(1+\tanh\hat{h}_k)\right. \\
& \qquad \left.+\prod_{k=1}^{p}(1+\tanh\hat{h}_k)\prod_{k=p+1}^{p+q}(1-\tanh\hat{h}_k)\right)^{x}\Bigg\} \\
& + x^{-1}\ln\Bigg\{\int\prod_{k=1}^{K}dh_kQ_k(h_k)\left(1-(1-e^{-beta})\prod_{k=1}^{K}\frac{1-\tanh h_k}{2}\right)^{x}\Bigg\} \\
& - x^{-1}\ln\Bigg\{\int dhQ(h)d\hat{h}\hat{Q}(\hat{h})\left(1+\tanh h\tanh\hat{h}\right)^{x}\Bigg\}
\end{aligned}
$$

The 1-RSB free energy functional is defined as

$$f_{\text{1RSB}}(\mathcal{Q}(.);x)=\mathbb{E}[f(Q(.),\hat{Q}(.),p,q)]$$

where the expectation is with respect to Q, $\hat{Q}$, p, q. The stationnary point equation of the 1RSB functional yield the 1RSB fixed point equations for the distributions $\mathcal{Q}(.),\hat{\mathcal{Q}}(.)$. These are the DE equations corresponding to the cavity message passing equations: one of them is precisely (15.19). The derivation of the second one is left as an exercise to the reader.

The interpolation method allows to prove the following theorem,

THEOREM 15.1 *For any trial distribution $\mathcal{Q}(.)$ and any $0<x<1$, the thermodynamic limit of the free energy of SAT exists, and moreover is lower bounded by the 1RSB formula*

$$\lim_{n\to+\infty}\frac{1}{n}\mathbb{E}[\ln Z]\leq f_{\text{1RSB}}(\mathcal{Q}(.);x)$$

The 1RSB conjecture states that taking the supremum over $\mathcal{Q}(.)$ and x on the right hand side yields an equality. We point out that this conjecture is surprizing from the standpoint of deterministic mean field models because for such models the variational expression for the free energy always involves a minimization (e.g. in the CW model). Here the free energy of K-SAT is given by a variational principle involving a maximization over trial parameters, rather than a minimization. This feature is in fact generic for replica formulas was already encountered in the early days of of the replica method. Note that it has nothing to do with the fact that the solution is RS or RSB. Now, for coding the RS variational expression for the free energy involves a minimization: this is surprizing from the standpoint of replica formulas! A look at the derivation of the bounds in the interpolation method (Chapter 13) shows that this can be traced to the channel or Nishimori symmetry.

Accepting the 1RSB conjecture teaches us something about the correct choice of the Parisi parameter x. Indeed recall that the complexity is the Gibbs-Shannon

K	α_d	$\alpha_{d,80,3}$	α_c	$\alpha_{c,80,3}$	α_s	$\alpha_{s,80,3}$
3	3.86	3.86	3.86	3.86	4.267	4.268
4	9.38	9.55	9.55	9.56	9.93	10.06

Table 15.1 Thresholds of individual and coupled K-SAT model for $L = 80$ and $w = 3$. Note that for 3-SAT teh dynamical and condensation thresholds are the same. The condensation and SAT-UNSAT thresholds correspond to non analyticities of the entropy and ground state energy and remain unchanged (for $L \to +\infty$). Already for $w = 3$ the dynamical threshold saturates very close to α_c and α_s.

entropy of the level-one model $\Sigma(x) = x^2 \frac{\partial}{\partial x^2} f_1(x)$. In place of $f_1(x)$ we use the 1RSB free energy formula (for the optimal $\mathcal{Q}(.)$), a function of x that can be computed by population dynamics. As long as $\Sigma(x) \geq 0$ for $0 < x < 1$ the optimal x is given by $x = 1$. We will see that this happens as long as $\alpha < \alpha_c$, where α_c is called the condensation threshold. When $\alpha > \alpha_c$ we get $\Sigma(x) \geq 0, 0 < x < x_*(\alpha)$, and $\Sigma(x) \leq 0$, $x_*(\alpha) < x < 1$, so that the optimal value of the parisi parameter is $x = x_*(\alpha)$. As we will see in the next chapter at the SAT-UNSAT density we have $x_*(\alpha_s) = 0$; for this value of the Parisi parameter the 1RSB formulas also simplify and yield the *survey propagation formulas*. This discussion shows that the condensation threshold can be obtained from the 1RSB complexity computed for $x = 1$. The same quantity will also give us the dynamical threshold $\alpha_d = \inf\{\alpha | \Sigma(x = 1) > 0\}$. This is sufficient motivation for giving the simplified 1RSB formulas for $x = 1$.

Analysis for $x = 1$

explain that free energy is RS free energy. Give the complexity and the fixed point equations without reweighting factor. Give population dynamic pseudo code.

15.6 Dynamical and Condensation Thresholds

The most important feature of the convex decomposition ansatz is the number of pure Bethe states involved. The RSB analysis of the level-one model predicts the existence of two sharply defined thresholds α_d and α_c at which the nature of the convex decomposition (15.1) changes drastically. The values of these thresholds are given in Table 15.1 and compared to the SAT-UNSAT threshold for a few values of K. Note that $K = 3$ is not generic because $\alpha_d = \alpha_c$. Figure 15.3 gives a pictorial view of the transitions associated with the decomposition (15.1). The goal of this paragraph is to explain this picture.

As already explained for $\alpha < \alpha_c$ we have $\Sigma(x) \geq 0$ for all $x \in [0, 1]$ and the correct value of the Parisi paramter is $x = 1$. The entropy is given by the RS

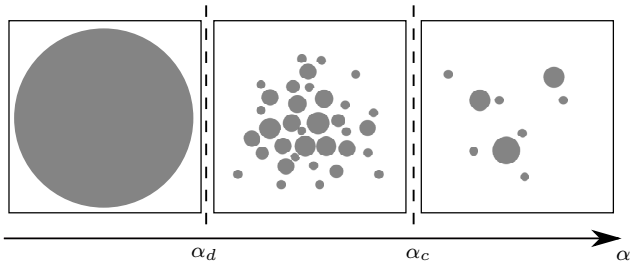


Figure 15.3 Pictorial representation of the decomposition of the Gibbs distribution into a convex superposition of extremal states. Balls represent extremal states (their size represents their internal entropy). For $\alpha < \alpha_d$ there is one extremal state. For $\alpha_d < \alpha < \alpha_c$ there are exponentially many extremal states (with the same internal free enetropy) that dominate to the convex superposition. For $\alpha > \alpha_c$ there is a finite number of extremal states that dominate the convex superposition.

formula. In particular this function is analytic for $\alpha < \alpha_c$ and therefore there is no thermodynamic static phase transition in this range. Above the condensation threshold the correct choice of the Parisi parameter $x = x_*(\alpha)$ forces the complexity to vanish. The Gibbs measure is supported by a finite number of pure Bethe states. Because of the change in x the entropy is not given by the same analytic function below and above α_c. therefore the condensation threshold is a thermodynamic static phase transition.

The complexity $\Sigma(x = 1)$ has a non trivial behavior below the condensation threshold. It vanishes for $\alpha < \alpha_d$, jumps to a positive value at α_d and is concave decreasing with increasing α till it becomes negative just above α_c. What is the interpretation of this result? Recall that the complexity is the growth rate for the number of pure Bethe states in the convex decomposition of the Gibbs measure, and the weights of this decomposition are given by the entropies of the pure states. For densities below the dynamical threshold the Gibbs measure is supported by one pure Bethe state. It is not excluded that there exist other ones of exponentially smaller weights. For densities between the dynamical and condensation thresholds an exponential number of pure Bethe states of identical entropy contribute to the convex sum. On the other hand beyond teh condensation threshold the measure is supported by only a finite number of pure Bethe states with equal entropy. All other states have exponentially smaller weights (the cavity method also predicts that the statistics of these weights is a Poisson-Dirichlet process). As already stressed the entropy is insensitive to the dynamical threshold, and thsi is not a static phase transition threshold. Rather, as its name indicates one expects that the proliferation of pure states affects the dynamics of algorithms local algorithms. In this course we have seen indications that this indeed occurs for BP guided decimation. In fact BP decimation fails slightly below α_d. This is not believed to be an inconsistency of the theory, but rather a consequence of the fcat that during teh decimation process the graph ensemble changes and therefore the threshold for BP guided decimation is set

by a different graph ensemble. It is believd that for Markov Chain Monte Carlo algorithms such as Glauber dynamics the equilibration time diverges exactly at α_d. This has been checked in simpler models.

It is interesting to consider the spatially coupled version of the K-SAT model. The same cavity theory can be applied and the RSB equations solved with the appropriate boundary conditions. this allows to determine the dynamical and condensation thresholds of the spatially coupled model (see table 15.1). The numerical observations suggest that the condensation threshold remains invariant in the limit of an infinite chain. This is consistent with its interpretation as a singularity of the entropy. In fact one can prove by the interpolation method that the entropy of the infinite coupled chain and underlying uncoupled model are the same, and therefore α_c is the same for both models, namely $\lim_{L\to+\infty} \alpha_c(w, L) = \alpha_c$. On the other hand it is observed that the dynamical threshold saturates towards the condensation threshold in the limit of an infinite chain and a large coupling range, namely $\lim_{w\to+\infty} \lim_{L\to+\infty} \alpha_d(w, L) = \alpha_c$. These results are conssistent with the interpretation of the dynamical threshold as an algorithmic barrier and the condensation threshold as a static phase transition threshold.

In section **??** we indicated that in Ising models there is an intimate connection between the decay of correlations and the extremality of the Gibbs measure. This is also true for constarint satisfaction models defined on random graph ensembles. However the correct correlation functions have to be used. In the present context two type of correlation functions have been discovered. Point-to-set correlations defined as

$$C(i, B) = \sum_{\underline{x}_{\partial B}} \nu(\underline{x}_{\partial B}(\nu(x_i|\underline{x}_{\partial B}) - \nu(x_i))^2$$

where B is the set $\{x_j|\{\text{dist}(x_i, x_j) \geq d\}$. Within the cavity method one can compute $\lim_{d\to+\infty} \lim_{n\to+\infty} C(i, B$ and finds that the limit vanishes $\alpha < \alpha_d$, while it remains strictly positive for $\alpha > \alpha_d$. Moreover for all $\alpha < \alpha_c$ and all randomly chosen bounded set of variables

$$\mathbb{E}[(\nu(x_{i_1}, \dots x_{i_k}) - \nu(x_{i_1}) \dots \nu(x_{i_k}))^2] = O(\frac{1}{n})$$

This is similar to the decoupling property we discussed for the CW model. At α_c this decoupling property breaks down.

16 Cavity Method: Survey Propagation

We have seen BP guided decimation does not find solutions beyond α_d. This chapter is an application of the cavity theory to find solutions of K-sat for densities beyond dynamical threshold. With level one model we learned about α_d and α_c. But have not yet computed α_s. We will aplly level one model with $x = 0$. RSB analysis with $x = 0$ leads to SP equations. Allows to compute α_s. Older point of view this was called "energetic cavity method". With decimation process we find solutions up densities close to α_s.

16.1 Survey propagation equations

Simplifify equations of previous chapter for $x = 0$. Derive equations.

16.2 Connection with the energetic cavity method

Briefly explain min sum point of view. Different level one model. Notion of SP complexity.

16.3 RSB analysis and sat-unsat threshold

Compute internal entropy and SP complexity. They both yield the sat-unsat threshold.

16.4 Survey propagation guided decimation

Algorithm. Experiments.

Notes

References

[1] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge Univ. Press, 2003.

[2] M. Mézard and A. Montanari, *Information, Physics, and Computation*. Oxford University Press, 2009.

[3] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. 8, pp. 21–28, Jan. 1962.

[4] ——, *Low-Density Parity-Check Codes*. Cambridge, MA, USA: MIT Press, 1963.

[5] B. Bollabás, *Modern Graph Theory*. New York, NY, USA: Springer Verlag, 1998.

[6] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. of the 29th annual ACM Symposium on Theory of Computing*, 1997, pp. 150–159.