Statistical Physics for Communications, Signal Processing, and Computer Science

EPFL

Nicolas Macris and Rüdiger Urbanke

Contents

Foreword *page* 1

Part I Models and their Statistical Physics Formulations 5

1 Models and Questions: Coding, Compressive Sensing, and Satisfiability 7
1.1 Coding 7
1.2 Compressive sensing 13
1.3 Satisfiability 18
1.4 Overview of coming attractions 22
1.5 Notes 23

2 Basic Notions of Statistical Mechanics 25
2.1 Lattice gas and Ising models 26
2.2 Gibbs distribution from maximum entropy 29
2.3 Free energy and variational principle 31
2.4 Marginals, correlation functions and magnetization 33
2.5 Thermodynamic limit and notion of phase transition 35
2.6 Spin glass models 37
2.7 Gibbs distribution from Boltzmann's principle 39
2.8 Notes 43

3 Formulation of Problems as Spin Glass Models 46
3.1 Coding as a spin glass model 47
3.2 Channel symmetry and gauge transformations 51
3.3 Conditional entropy and free energy in coding 52
3.4 Compressive Sensing as a spin glass model 54
3.5 Free energy and conditional entropy in compressive sensing 57
3.6 K-SAT as a spin glass model 58
3.7 Notes 60

4 Curie-Weiss Model 62
4.1 Curie-Weiss model 63
4.2 Variational expression of the free energy 64
4.3 Average magnetization 65

4.4 Phase diagram and phase transitions 67
4.5 Analysis of the fixed point equation 70
4.6 Ising model on a tree 73
4.7 Phase transitions in the Ising model on $\mathbb{Z}^d$ 73
4.8 Notes 74

Part II Analysis of Message Passing Algorithms 77

5 Marginalization and Belief Propagation 79
5.1 Factor graph representation of Gibbs distributions 80
5.2 Marginalization on trees 81
5.3 Marginalization via Message Passing 85
5.4 Decoding via Message Passing 89
5.5 Message Passing in Compressed Sensing 91
5.6 Message passing in K-SAT 94

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 [**?**, **?**].

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-**??**, 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 **??**-**??**, 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, $\mathcal{C} = \{\underline{x}^{(1)}, \dots, \underline{x}^{(\mathcal{M})}\}$, 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 $V = \{x_1, \dots, x_n\}$ is the set of n *variable* nodes corresponding to the n bits (and hence to the n columns of H), and where $C = \{c_1, \dots, c_{n-k}\}$ 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

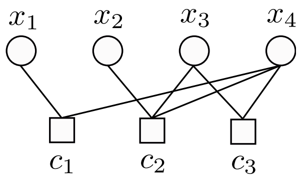


Figure 1.1 The factor graph corresponding to the parity-check matrix of Example 1.

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, [**?**, **?**]. 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 [**?**]. 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).

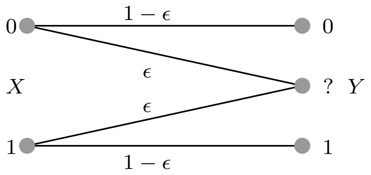


Figure 1.2 Binary erasure and symmteric channels with parameter ϵ.

<u>BEC</u>. 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 $p(y|x) = (1 - \epsilon)\delta(y - x) + \epsilon\delta(y - ?)$.

<u>BSC</u>. 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* $\mathbb{P}\left[\hat{\underline{x}}(y) \neq \underline{x}\right]$, and the *bit error probability* $\frac{1}{n} \sum_{i=1}^{n} \mathbb{P}\left[\hat{\underline{x}}(y)_i \neq x_i\right]$. 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

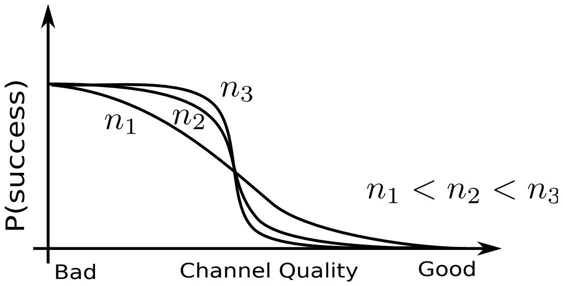


Figure 1.3 The probability of decoding error for a transmitted message versus the channel quality. As the blocklength of the code gets larger, we expect to see a sharper and sharper transition between range of the channel parameters where the system "works" and where it "breaks down."

number of non-zero components $\|\underline{x}^{\text{in}}\|_0 = |\{i|x_i^{\text{in}} \neq 0, i = 1, \dots, n\}| = k$ 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 $\text{RIP}(2k, \delta)$ 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 $\text{RIP}(2k, \delta)$

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 $\text{RIP}(2k,\delta')$ 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 $V = \{x_1, \dots, x_n\}$ is the set of *variable* nodes corresponding to the n signal components and $C = \{c_1, \dots, c_m\}$ 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 $\{x_1, \dots, x_n\}$. 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 $\{x_1, \dots, x_n\}$ 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 $V = \{x_1, \dots, x_n\}$ are the Boolean variables and $C = \{c_1, \dots, c_m\}$ 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. □

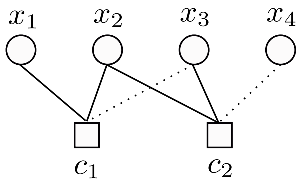


Figure 1.6 The factor graph corresponding to the SAT formula of Example 2.

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 $x_1, \dots, x_n$ 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 $c_1, \dots, c_m$ 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 $\binom{m}{\binom{n}{K}2^K}$.

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 $\mathbb{P}[-]$ 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 **??**, **??**.

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 $x_1, \dots, x_n \in \{0,1\}^n$. 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 $c_1, \dots, c_m$. 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 **??**,

??. 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?

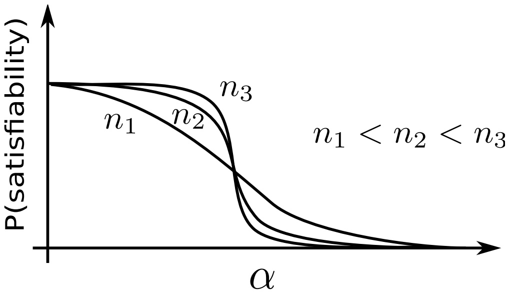


Figure 1.7 The probability that a formula generated from the random K-SAT ensemble is satisfied versus the clause density α.

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 $\|\underline{x}\|_0 = \sharp(\text{non zero } x_1, \dots, x_n)$ 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 $\hat{\underline{x}}(\underline{y}) = (A^T A)^{-1} A^T \underline{y}$.

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.

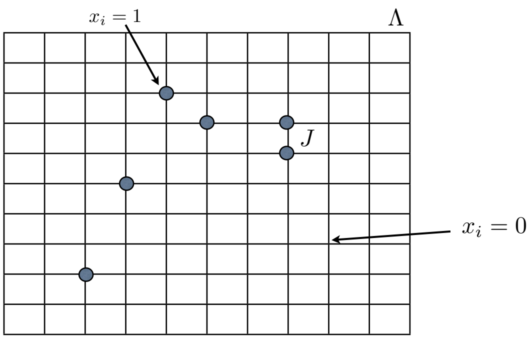


Figure 2.1 Left: a particle configuration in the lattice gas model. Full circles represent occupied sites $x_i = 1$ and empty circles unocupied sites $x_i = 0$. At most one particle occupies a lattice site. Right: a magnetic configuration in the Ising model. Positive signs indicate “up spins” $s_i =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

Ising model

The 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 spin systems

It is quite clear that one can generalize such models to other regular grids or lattices, eg. a triangular lattice. Usually the nature of the grid depends on the physical system. It may represent an underlying crystalline structure or a mathematical approximation of continuous space. One can also go beyond the hypothesis of nearest neighbor interactions which means that there are terms $-J_{ij}x_ix_j$ or $-J_{ij}s_is_j$ in the cost function with associated to sites i, j separated by more than one edge. More generally one may consider multispin interactions, for example on a square grid the four spins of elementary plaquettes may interact through terms of the form $-\sum_{(i,j,k,l)\in P} J_{ijkl}s_is_js_ks_l$ where P is the set of all elementary plaquettes of the square grid and J_{ijkl} is the "plaquette interaction strength".

The most general Ising 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 canonical Ising or lattice gas models then corresponds to the choice $J_A = h$ for $A = \{i\}$, $i \in V$; $J_A = J$ for all $A = \{i, j\} \in E$ and $J_A = 0$ otherwise. If we add plaquette interaction we also have $J_A = J_{ijkl}$ for all $A = \{i, j, k, l\} \in P$ the set of all plaquettes.

The factor graph representation is a convenient representation for such systems. Here the factor graph is a bipartite graph with variable nodes associated

FIGURE

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

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.

The reader can already see that the LDPC codes and K-SAT models have cost functions that are of the Ising type. For compressed sensing the "spins' are real numbers and one talks about "continuous spins". All that will be described in more depth in Chapter 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, Brillouin 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))

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

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,

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

[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.

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

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

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 $j = 1, \dots i-1, i+1, \dots n$. 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 $\ln Z(\underline{\lambda})$, $\underline{\lambda} = (\lambda_1, \dots, \lambda_n)$. 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

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 spin systems, such as (2.1), (2.2), (2.3), with *random interaction constants*. These models where introduced by Anderson and Edwards in the 1970’s in an attempt to capture the irregular arrangements of degrees of freedom or their irregular interactions. The *Edwards-Anderson model* is simply given by the Hamiltonian (2.2) with $J_{ij} = \pm J$ where the sign on each edge is iid Bernoulli (probability 1/2 for each sign), and $h_i = h$ is constant. Another widely studied model is the *random field Ising model* with Hamiltonian

(2.2) with $J_{ij} = J$ constant and $h_i = \pm h$ with iid Bernoulli signs. Variants of these models use other distributions for teh 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 such simple models has turned out be very non-trivial and is a source of many fundamental concepts in statistical mechanics of so-called *disordered systems*. We point out that even after fourty years the Edwards-Anderson and random field Ising models are still not well understood and many open questions remain. 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 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 3 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

$$W(E) = \binom{n}{E/\mu} \simeq \exp\left(nh_2\Big(\frac{E}{\mu n}\Big)\right), \tag{2.40}$$

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).

[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 [**?**]

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

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

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 $(x_1, \ldots, x_m)$ and those outside Σ by $(x_{m+1}, \ldots, x_n)$. 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, $A(x_1, \ldots, x_m)$. Of course we can compute them with the microcanonical distribution

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

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

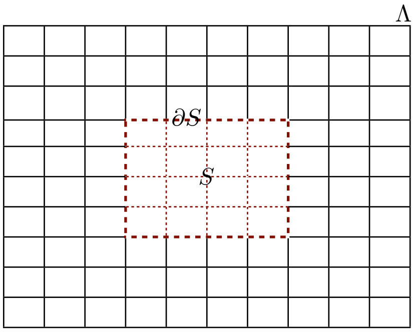


Figure 2.3 The system S is embedded in a thermal bath V. The total system V is considered as an isolated system and its total energy E is conserved. We compute the induced measure on S.

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 $x_1, \ldots, x_m$ reads

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

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,

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

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 $(x_1, \ldots, x_n)$ belongs to the energy surface Γ_E then $(x_{m+1}, \ldots, x_n)$ belongs to the energy surface $\Gamma_{E-\mathcal{H}_\Sigma(x_1, \ldots, x_m)}$. With these remarks we obtain

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

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

$$\begin{aligned}\frac{\partial^3}{\partial\lambda_i\lambda_j\lambda_k}\ln Z(\underline{\lambda})|_{\underline{\lambda}=0} =&\langle x_ix_jx_k\rangle - \langle x_ix_j\rangle\langle x_k\rangle - \langle x_jx_k\rangle\langle x_i\rangle \\ &- \langle x_ix_k\rangle\langle x_j\rangle + 2\langle x_i\rangle\langle x_j\rangle\langle x_k\rangle\end{aligned}$$

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

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

(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 $p(\underline{y}|\underline{s}) \times \frac{\mathbb{1}(\in\mathcal{C})}{|\mathcal{C}|}$. 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* $\underline{\hat{s}}(\underline{y}) = \text{argmax}_{\underline{s}}\ p(\underline{s} \mid \underline{y})$ and the associated the block probability of error $\mathbb{P}_{\text{B}}[\text{error}] = \frac{1}{|\mathcal{C}|} \sum_{\underline{s}^{\text{in}} \in \mathcal{C}} \mathbb{P}_B[\underline{\hat{s}}(\underline{Y}) \neq \underline{s}^{\text{in}}]$. 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 $i = 1, \dots, n$ and checks[1] $a = 1, \dots, m$; 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

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

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 $i, j, k, \dots$ and checks by $a, b, c, \dots$

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

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

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 $s_i^{\mathrm{in}} = 1$, $i = 1, \dots, n$, or $\underline{x} = 0$ the "all-zero codeword". To simplify the notations we set $c(\underline{h}|\underline{1}) = c(\underline{h})$ and $\mathbb{E}_{\underline{h}|\underline{1}^{\mathrm{in}}} = \mathbb{E}_{\underline{h}}$. For symmetric channels the average bit error probability is given by

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

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

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

[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 4 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

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

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 **??**. 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 $p(\underline{s}|\underline{y})$,

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

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 $\underline{\tau} \in \mathcal{C}$,

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

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, (**??**)

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

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,

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

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, $a = 1, \dots, m$ 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

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

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 **??**.

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

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

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

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

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 **??** 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

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

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

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

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

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

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,

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

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

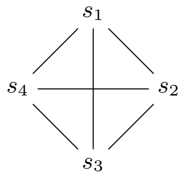


Figure 4.1 A complete graph with 4 nodes.

quite important in more advanced topics such as the cavity method in Chapter **??**, 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

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

Thus

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

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 $m_n = \{\frac{j}{n} | j = -n, -n+1, \dots, n-1, n\}$. We get

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

where $\mathcal{N}(m_n)$ is the cardinality of the set $\{\underline{s} : \sum_{i=1}^n s_i = nm_n\}$. 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 $n_+ - n_- = nm_n$ we have $n_+ = \frac{1+m_n}{2}n$ and therefore

$$\mathcal{N}(m_n) = \binom{n}{\frac{1+m_n}{2}n} \approx e^{nh_2(\frac{1+m_n}{2})}, \tag{4.9}$$

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

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

Recall that $m_n = \{\frac{j}{n} | j = -n, -n+1, \dots, n-1, n\}$. 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

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

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,

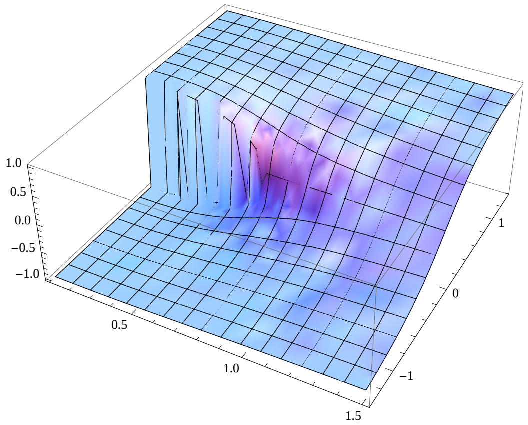


Figure 4.2 The behavior of $\overline{m}(\beta J, \beta h)$ as a function of $(1/(\beta J), \beta h)$, where $1/(\beta J) \in [0, 1.5]$ and $\beta h \in [-1.5, 1.5]$.

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

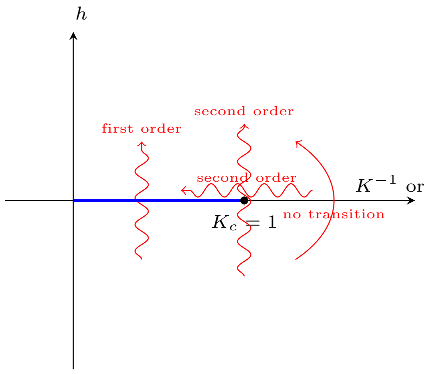
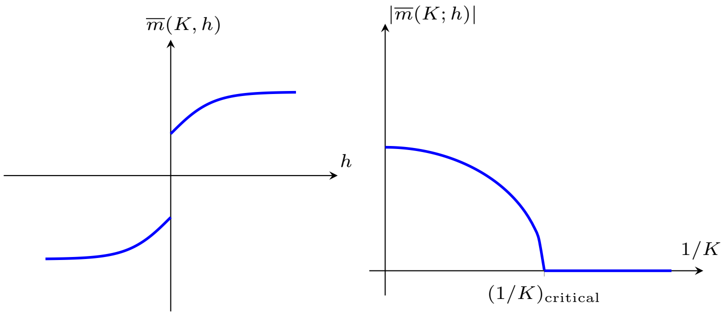


Figure 4.3 The blue line is called *coexistence line* because two thermodynamic phases (e.g. water/ice) coexist for parameters on it. Crossing the thick line is a first order phase transition. This line is terminated by the *critical point*. Crossing the critical point is a second order phase transition.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

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

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"

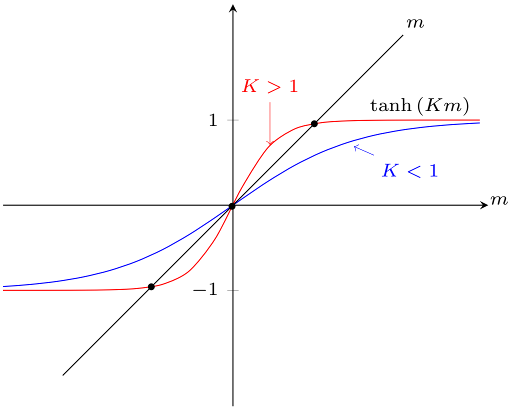


Figure 4.5 Curie-Weiss fixed points, $h = 0$

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".

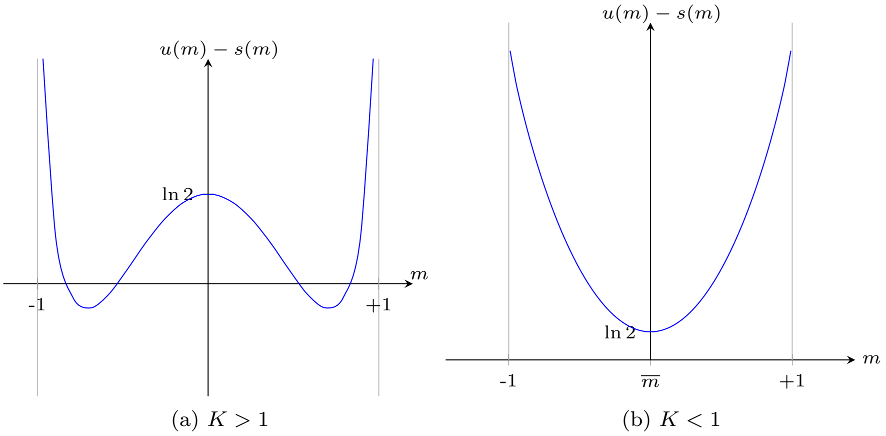
Let us look more closely at the behaviour of the magnetization for $h = 0$ as a function of $1/(\beta J)$ is shown in Figure 4.4. For βJ close to $\beta J = 1$ we can expand the Curie-Weiss equation around $\overline{m} = 0$,

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

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

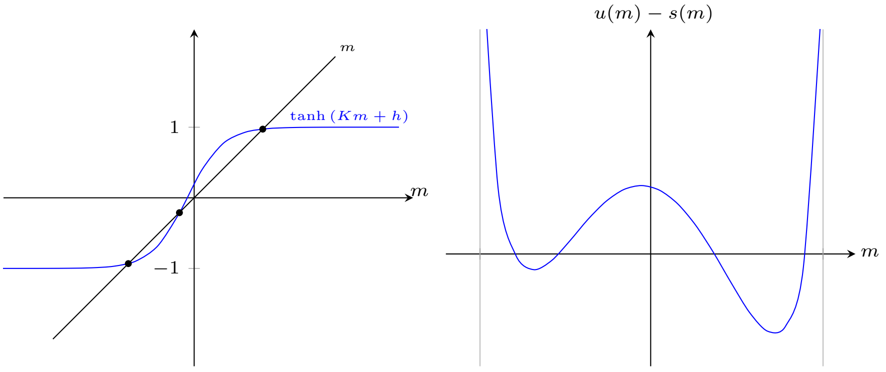
often does not depend on the details of the Hamiltonian but only on the dimensionality of the system (here $d = +\infty$), and the underlying symmetries of the Hamiltonian (here the Hamiltonian is invariant under $s_i \to -s_i$ for $h = 0$). For example in the exercises you will see that the ising model 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

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

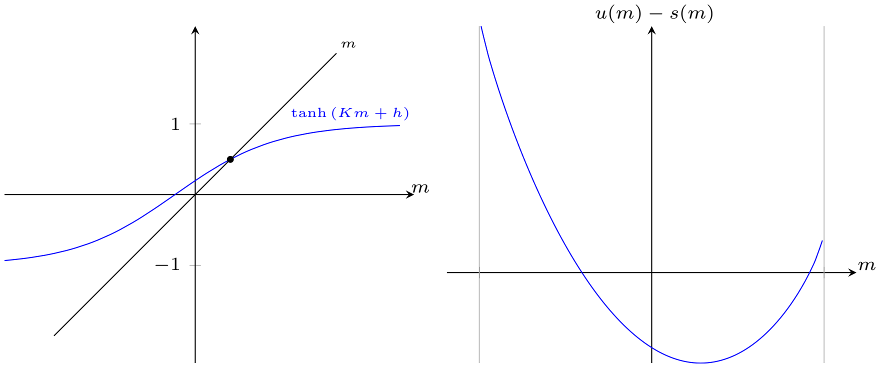
This is again an example of second order phase transition thsi time with critical exponent $\frac{1}{3}$ (exercise: show this by expanding the Curie-Weiss equation for small h when $\beta J = 1$.)



(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 $m(\beta J, \beta hh)$ 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 **??** 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 $f_a(s_{\partial a}) = \exp\{-\beta\prod_{i\in\partial a}(\frac{1+s_iJ_{ia}}{2})\}$. For coding (see (3.9)) we have two types of factors $f_i(s_i) = e^{h_is_i}$ 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 5 (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

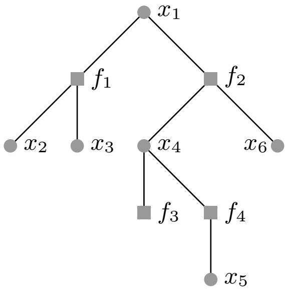


Figure 5.1 Factor graph of f given in Example 5.

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 5. 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 $\mu(z) =$

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

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

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 5 this factorization is

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

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

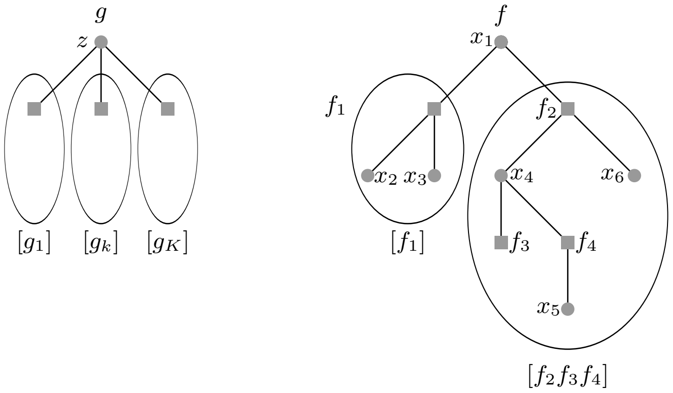


Figure 5.2 Generic factorization and the particular instance.

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

In words, the marginal $\sum_{\sim z} g(z, \ldots)$ is the product of the individual marginals $\sum_{\sim z} g_k(z, \ldots)$. 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 $g_k(z, \ldots)$.

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

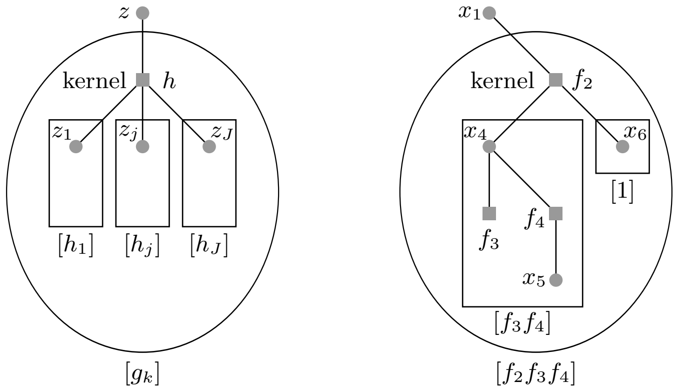


Figure 5.3 Generic factorization of g_k and the particular instance.

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

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 6 (Message-Passing Algorithm for f of Example 5) 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. $\Diamond$

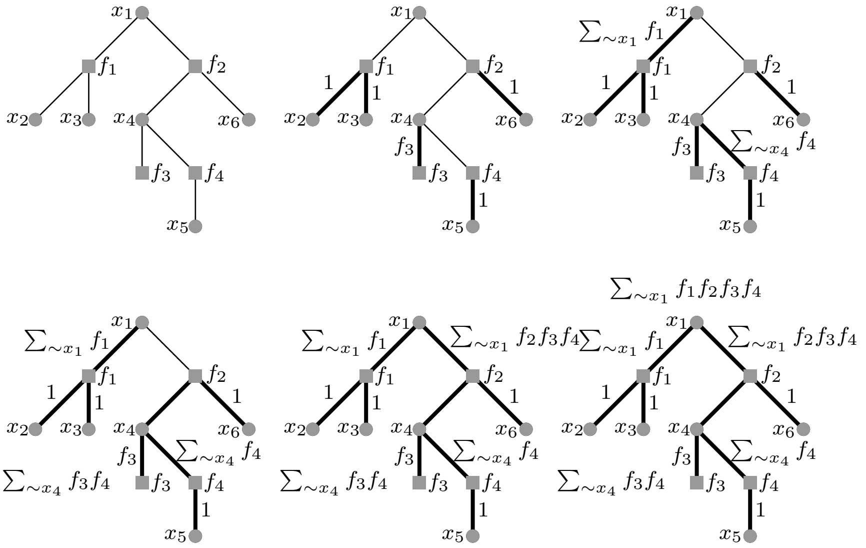


Figure 5.4 Marginalization of function f from Example 5 via message passing. Message passing starts at the leaf nodes. A node that has received messages from all its children processes the messages and forwards the result to its parent node. Bold edges indicate edges along which messages have already 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 simultaneously 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 $a, b, c, \dots$ are reserved for function nodes and $i, j, k, \dots$ 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

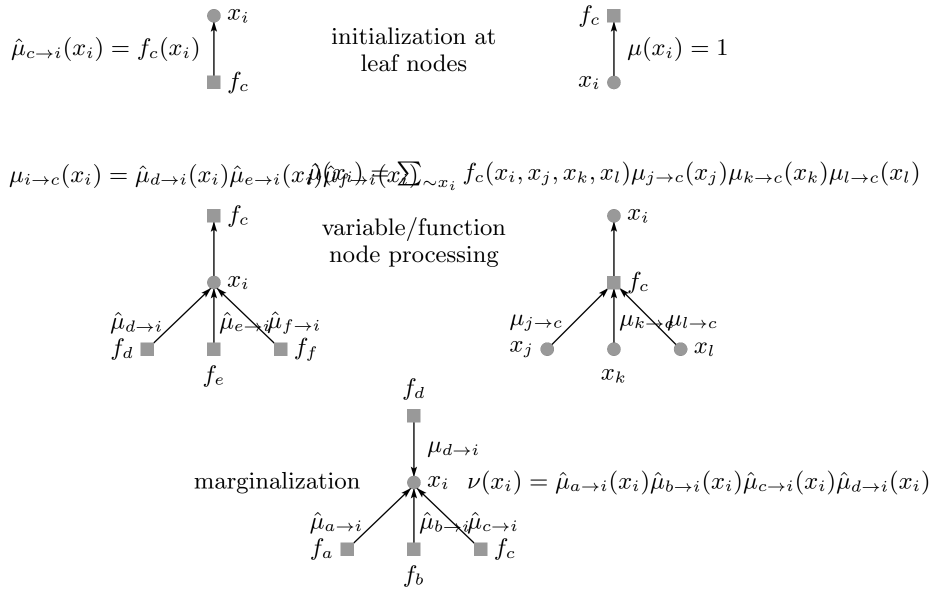

$\hat{\mu}_{c\to i}(x_i) = f_c(x_i)$
x_i
f_c
initialization at leaf nodes
f_c
$\mu(x_i) = 1$
x_i
$\mu_{i\to c}(x_i) = \hat{\mu}_{d\to i}(x_i)\hat{\mu}_{e\to i}(x_i)\hat{\mu}_{f\to i}(x_i)$
$\hat{\mu}(x_i) = \sum_{\sim x_i} f_c(x_i, x_j, x_k, x_l)\mu_{j\to c}(x_j)\mu_{k\to c}(x_k)\mu_{l\to c}(x_l)$
f_c
variable/function node processing
x_i
x_i
f_c
$\hat{\mu}_{d\to i}$
$\hat{\mu}_{e\to i}$
$\hat{\mu}_{f\to i}$
f_d
f_e
f_f
$\mu_{j\to c}$
$\mu_{k\to c}$
$\mu_{l\to c}$
x_j
x_k
x_l
f_d
$\mu_{d\to i}$
marginalization
x_i
$\nu(x_i) = \hat{\mu}_{a\to i}(x_i)\hat{\mu}_{b\to i}(x_i)\hat{\mu}_{c\to i}(x_i)\hat{\mu}_{d\to i}(x_i)$
$\hat{\mu}_{a\to i}$
$\hat{\mu}_{b\to i}$
$\hat{\mu}_{c\to i}$
f_a
f_b
f_c

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 $t = 0, \dots, T$ 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 $\hat{s}_i(\underline{h}) = \text{argmax}_{s_i \in \{\pm 1\}} p(s_i \mid h) = \text{sign}\langle s_i \rangle$ 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 7 (Bit-wise MAP Decoding) Consider the code defined by its parity-check matrix with Tanner graph shown on the left of Fig. 5.6.

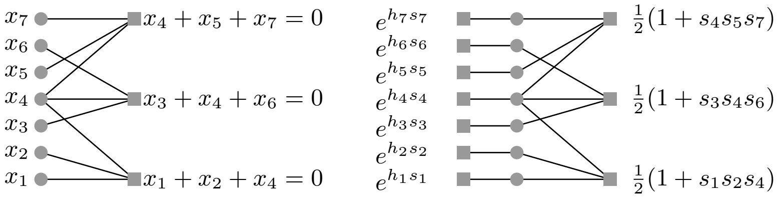


Figure 5.6 Left: graphical representation of the parity check code. Right: factor graph associated to the distribution (5.13) of our running example.

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 8 (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 9 (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 ◊

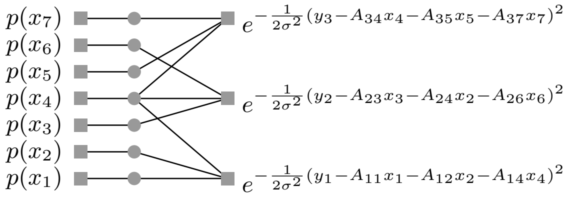

$p(x_7)$
$p(x_6)$
$p(x_5)$
$p(x_4)$
$p(x_3)$
$p(x_2)$
$p(x_1)$
$e^{-\frac{1}{2\sigma^2}(y_3 - A_{34}x_4 - A_{35}x_5 - A_{37}x_7)^2}$
$e^{-\frac{1}{2\sigma^2}(y_2 - A_{23}x_3 - A_{24}x_2 - A_{26}x_6)^2}$
$e^{-\frac{1}{2\sigma^2}(y_1 - A_{11}x_1 - A_{12}x_2 - A_{14}x_4)^2}$

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 10 (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 incoming 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 **??**. Let us for now illustrate how the marginalization proceeds on our simple tree graph example.

EXAMPLE 11 (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 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,

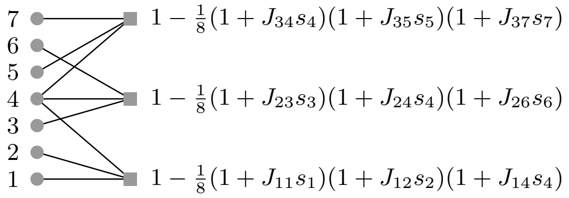

7
6
5
4
3
2
1
$1 - \frac{1}{8}(1 + J_{34}s_4)(1 + J_{35}s_5)(1 + J_{37}s_7)$
$1 - \frac{1}{8}(1 + J_{23}s_3)(1 + J_{24}s_4)(1 + J_{26}s_6)$
$1 - \frac{1}{8}(1 + J_{11}s_1)(1 + J_{12}s_2)(1 + J_{14}s_4)$

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.

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

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

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

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. ◊

Message passing at positive and zero temperatures

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

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

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

EXAMPLE 12 (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,

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

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

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

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

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

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. ◊

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 13 (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,

$$\begin{aligned}
\hat{e}_{1\to 4}(s_4) &= \min_{s_1,s_2}((\frac{1+J_{11}s_1}{2})(\frac{1+J_{12}s_2}{2})(\frac{1+J_{14}s_4}{2}) + 0 + 0),\\
\hat{e}_{2\to 4}(s_4) &= \min_{s_3,s_6}((\frac{1+J_{23}s_3}{2})(\frac{1+J_{24}s_4}{2})(\frac{1+J_{26}s_6}{2}) + 0 + 0),\\
\hat{e}_{3\to 4}(s_4) &= \min_{s_3,s_6}((\frac{1+J_{34}s_4}{2})(\frac{1+J_{35}s_5}{2})(\frac{1+J_{37}s_7}{2}) + 0 + 0).
\end{aligned}$$

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 $s_4 = \pm 1$.

◊

Problems

5.1 Min-Sum Message Passing rules. In class we discussed how to compute the marginal of a multivariate function $f(x_1, \dots, x_n)$ 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

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 $\mu_{\hat{i}\to i}(s_i) = e^{h_i s_i}$. 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,

$$\frac{\hat{\mu}_{a\to i}\,(+1)}{\hat{\mu}_{a\to i}\,(-1)} = \frac{\sum_{\sim s_i}(1+\prod_{j\in\partial a\backslash i} s_j)\prod_{j\in\partial a\setminus i}\mu_{j\to a}\,(s_j)}{\sum_{\sim s_i}(1-\prod_{j\in\partial a\backslash i} s_j)\prod_{j\in\partial a\setminus i}\mu_{j\to a}\,(s_j)}. \tag{6.6}$$

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

$$\frac{\hat{\mu}_{a\to i}\,(+1)}{\hat{\mu}_{a\to i}\,(-1)} = \frac{\sum_{\sim s_i}(1+\prod_{j\in\partial a\backslash i} s_j)\prod_{j\in\partial a\setminus i}(1+s_j\tanh l_{j\to a})}{\sum_{\sim s_i}(1-\prod_{j\in\partial a\backslash i} s_j)\prod_{j\in\partial a\setminus i}(1+s_j\tanh l_{j\to a})}. \tag{6.8}$$

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

$$
\begin{aligned}
&(1\pm \prod_{j\in\partial a\setminus i} s_j) \prod_{j\in\partial a\setminus i} (1 + s_j \tanh l_{j\to a}) \\
&\quad = (1 \pm \prod_{j\in\partial a\setminus i} s_j) \sum_{J\subset\partial a\setminus i} \prod_{j\in J} s_j \prod_{j\in J} \tanh l_{j\to a} \\
&\quad = \sum_{J\subset\partial a\setminus i} \prod_{j\in J} s_j \prod_{j\in J} \tanh l_{j\to a} \pm \sum_{J\subset\partial a\setminus i} \prod_{j\in J^c} s_j \prod_{j\in J} \tanh l_{j\to a} \qquad (6.9)
\end{aligned}
$$

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

$$
\frac{\hat\mu_{a\to i}(+1)}{\hat\mu_{a\to i}(-1)} = \frac{1 + \prod_{j\in\partial a\setminus i} \tanh l_{j\to a}}{1 - \prod_{j\in\partial a\setminus i} \tanh l_{j\to a}} \qquad (6.10)
$$

Finally taking the logarithm and using $\frac{1}{2}\ln\frac{1+x}{1-x} = \operatorname{atanh} x$ we arrive at

$$
\hat l_{a\to i} = \operatorname{atanh}\Big\{ \prod_{j\in\partial a\setminus i} \tanh l_{j\to a} \Big\} \qquad (6.11)
$$

Let us now look at the "marginals" computed from the BP equations. We will call them *BP marginals* and denote them by $\nu_i^{\rm BP}(s_i)$ 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,

$$
\nu_i^{\rm BP}(s_i) = \frac{e^{h_i s_i} \prod_{a\in i} \hat\mu_{a\to i}(s_i)}{e^{h_i} \prod_{a\in i} \hat\mu_{a\to i}(+1) + e^{-h_i} \prod_{a\in i} \hat\mu_{a\to i}(-1)} \qquad (6.12)
$$

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

$$
\begin{aligned}
\nu_i^{\rm BP}(s_i) &= \frac{e^{(h_i+\sum_{a\in\partial i}\hat l_{a\to i})(s_i+1)}}{1 + e^{2(h_i+\sum_{a\in\partial i}\hat l_{a\to i})}} \\
&= 1 + s_i \tanh(h_i + \sum_{a\in\partial i} \hat l_{a\to i}) \qquad (6.13)
\end{aligned}
$$

From this marginal one can compute the *BP magnetization* of the i-th bit (to be distinguished from the true magnetization)

$$
m_i^{\rm BP} = \sum_{s_i=0,1} s_i \nu_i^{\rm BP}(s_i) = \tanh(h_i + \sum_{a\in\partial i} \hat l_{a\to i}) \qquad (6.14)
$$

The *BP estimate* for bit i is then

$$
\hat s_i^{\rm BP} = \operatorname{sign}(m_i^{\rm BP}) \qquad (6.15)
$$

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

$$\begin{cases} l_{i\to a} = h_i + \sum_{b\in\partial i\setminus a} \hat{l}_{b\to i} \\ \hat{l}_{a\to i} = \operatorname{atanh}\Big\{\prod_{j\in\partial a\setminus i} \tanh l_{j\to a}\Big\} \end{cases} \tag{6.17}$$

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, $l^{(t)}_{i \to a}$, and in the second step we have messages flowing from check to variable nodes, $\hat{l}^{(t)}_{a \to i}$. They satisfy

$$\begin{cases} l^{(t)}_{i \to a} = h_i + \sum_{b \in \partial i \smallsetminus a} \hat{l}^{(t-1)}_{b \to i} \\ \hat{l}^{(t)}_{a \to i} = \text{atanh}\Big\{ \prod_{j \in \partial a \setminus i} \tanh l^{(t)}_{j \to a} \Big\} \end{cases} \tag{6.19}$$

The iterative process is initialized with $l^{(0)}_{i \to a} = \hat{l}^{(0)}_{a \to i} = 0$. The total estimated likelihood ratio for bit i at time t is

$$l^{(t)}_i = h_i + \sum_{a \in \partial i} \hat{l}^{(t)}_{a \to i} \tag{6.20}$$

and the BP estimate at time t for the bit is

$$\hat{s}^{\text{BP,t}}_i = \text{sign}(\tanh\, l^{(t)}_i) \tag{6.21}$$

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

$$h_i = \log\Big(\frac{p(y_i \mid s_i = 1)}{p(y \mid s_i = -1)}\Big) = \begin{cases} \frac{1}{2}\log(\frac{1-\epsilon}{0}) = +\infty & y = 1, \\ \frac{1}{2}\log(\frac{\epsilon}{\epsilon}) = 0, & y = ?, \\ \frac{1}{2}\log(\frac{0}{1-\epsilon}) = -\infty, & y = -1. \end{cases}$$

Now, since the initial condition for the internal messages is $l^{(0)}_{i \to a} = 0, \hat{l}^{(0)}_{a \to i} = 0$ the BP equations (6.19) imply that at later times $l^{(t)}_{i \to a} = 0, \hat{l}^{(t)}_{a \to i}\{\pm\infty, 0\}$. 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 $\mathbb{E}_{\underline{h}} = \mathbb{E}_{\underline{h}|\underline{1}}$)

$$\begin{aligned} P_{\mathrm{BP,b}}(\mathcal{C}, \underline{s}^{\mathrm{in}}, \epsilon, t) &= P_{\mathrm{BP,b}}(\mathcal{C}, \epsilon, t) \\ &= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2}(1 - \mathbb{E}_{\underline{h}}[\hat{s}_i^{\mathrm{BP,(t)}}]) \end{aligned} \tag{6.24}$$

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

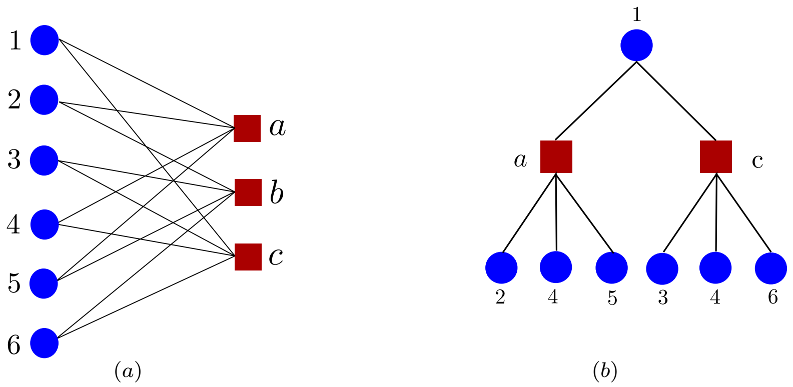


Figure 6.1 (a) The Tanner graph of a $(2,4)$-regular LDPC code with 6 variable nodes; (b) The corresponding computation graph of node 1 for the first iteration.

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

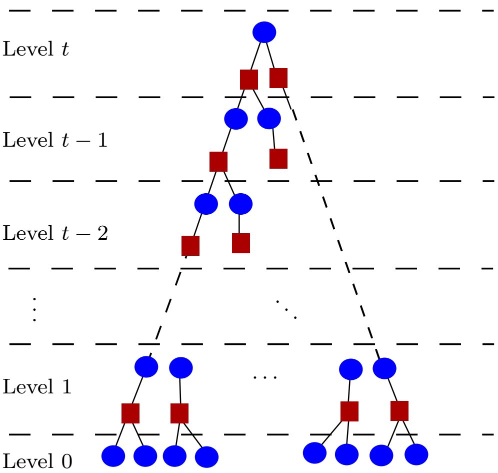


Figure 6.2 A computation graph of $(2,3)$-regular LDPC code with height t. The graph is split to $t+1$ levels.

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 $\hat{l}_1, \dots, \hat{l}_{d_v-1}$ 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

$$\hat{l} = \text{atanh}\left(\prod_{i=1}^{d_c-1} \tanh l_i)\right)$$

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 (6.35) and (6.36) 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 $\text{sign}(\tanh l) = -1$, in other words $l < 0$. Thus

$$\mathbb{P}_{\text{BP,b}}(d_v, d_c, \epsilon, t) = \int_{-\infty}^{0} dl\, (c \otimes (x_{t-1}^{\boxplus d_c - 1})^{\otimes d_v})(l) \tag{6.39}$$

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 $\epsilon_{\rm BP}$, called the BP-threshold, such that for $\epsilon < \epsilon_{\rm BP}$ the limit of $\mathbb{P}_{\rm BP,b}(d_v, d_c, \epsilon, t)$ when the number of iterations $t \to +\infty$ vanishes, while for $\epsilon > \epsilon_{\rm BP}$ this limit remains strictly positive.

In order to compute $\lim_{t\to+\infty} \mathbb{P}_{\rm BP,b}(d_v, d_c, \epsilon, t)$ 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

$$\lim_{t\to+\infty} \mathbb{P}_{\rm BP,b}(d_v, d_c, \epsilon, t) = \epsilon(1-(1-x_\infty(\epsilon))^{d_v-1})^{d_c}, \tag{6.43}$$

are increasing in ϵ as shown in Figure 6.3. Hence we can define the quantity

$$\epsilon^{\rm BP} = \sup\{\epsilon : x_\infty(\epsilon) = 0\}$$

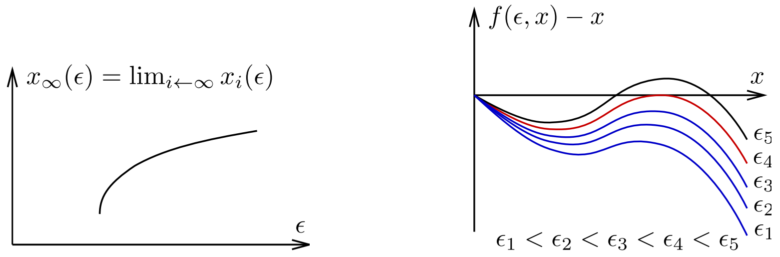


Figure 6.3 *Left*: Monotonicity of x_∞ as a function of ϵ. For $d_v \geq 3$, $d_c > d_v$, x_∞ jumps at the threshold. For $d_v = 2$, $d_c > d_v$ x_∞ changes continuously at the threshold. *Right*:The threshold ϵ_{BP} is the largest channel parameter so that $f(\epsilon, x) - x < 0$ for the whole range $x \in [0, 1]$.

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 14 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 15 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

$$H[x \otimes (a-b)] \le H[a-b], \qquad H[x \boxplus (a-b)] \le H[a-b] \tag{6.51}$$

To prove the second inequality we use the moment expansion and the fact that moments are multiplicative for the $\boxplus$ operation, $M_{2k}(a \boxplus b) = M_{2k}(a)M_{2k}(b)$,

$$\begin{aligned} H[x \boxplus (a-b)] &= -\sum_{k=1}^{+\infty} \frac{M_{2k}(x \otimes a) - M_{2k}(x \otimes b)}{2k(2k-1)} \\ &= \sum_{k=1}^{+\infty} M_{2k}(x) \frac{M_{2k}(b) - M_{2k}(a)}{2k(2k-1)} \\ &\le \sum_{k=1}^{+\infty} \frac{M_{2k}(b) - M_{2k}(a)}{2k(2k-1)} \\ &= H[a-b] \end{aligned}$$

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

$$\begin{aligned} H[x \otimes (a-b)] &= -H((x - \Delta_\infty) \otimes (a-b)] \qquad &(6.52) \\ &= \sum_{k=1}^{+\infty} M_{2k}(\Delta_\infty - x) \frac{M_{2k}(b) - M_{2k}(a)}{2k(2k-1)} \qquad &(6.53) \\ &= \sum_{k=1}^{+\infty} (M_{2k}(\Delta_\infty) - M_{2k}(x)) \frac{M_{2k}(b) - M_{2k}(a)}{2k(2k-1)} \qquad &(6.54) \\ &\le \sum_{k=1}^{+\infty} \frac{M_{2k}(b) - M_{2k}(a)}{2k(2k-1)} \qquad &(6.55) \\ &= H[a-b] \qquad &(6.56) \end{aligned}$$

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 $a_i \succ b_i$, $i = 1, \dots, n$ 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

$$\begin{aligned}
d(f(c,x_t),f(c,x_\infty)) &= H(f(c,x_t)-f(c,x_\infty)\\
&= H(c\otimes((x_t^{\boxplus d_c-1})^{\otimes d_v-1}-(x_\infty^{\boxplus d_c-1})^{\otimes d_v-1}))\\
&\leq H((x_t^{\boxplus d_c-1})^{\otimes d_v-1}-(x_\infty^{\boxplus d_c-1})^{\otimes d_v-1})\\
&= H((x_t^{\boxplus d_c-1}-x_\infty^{\boxplus d_c-1}+x_\infty^{\boxplus d_c-1})^{\otimes d_v-1}-(x_\infty^{\boxplus d_c-1})^{\otimes d_v-1})\\
&= \sum_{p=1}^{d_v-1}\binom{d_v-1}{p}H((x_t^{\boxplus d_c-1}-x_\infty^{\boxplus d_c-1})^{\otimes p}\otimes(x_\infty^{\boxplus d_c-1})^{\otimes d_v-1-p})\\
&\leq \sum_{p=1}^{d_v-1}\binom{d_v-1}{p}H(x_t^{\boxplus d_c-1}-x_\infty^{\boxplus d_c-1})\\
&= (2^{d_v-1}-1)H(x_t^{\boxplus d_c-1}-x_\infty^{\boxplus d_c-1})
\end{aligned}$$

Each term of the last sum is estimated thanks to similar tricks,

$$\begin{aligned}
H(x_t^{\boxplus d_c-1}-x_\infty^{\boxplus d_c-1}) &= H((x_t-x_\infty+x_\infty)^{\boxplus d_c-1}-x_\infty^{\boxplus d_c-1})\\
&= \sum_{q=1}^{d_c-1}\binom{d_c-1}{q}H((x_t-x_\infty)^{\boxplus q}\boxplus x_\infty^{\boxplus d_c-1-q})\\
&\leq (2^{d_c-1}-1)H((x_t-x_\infty))
\end{aligned}$$

Putting these results together we obtain the simple inequality

$$\begin{aligned}
d(f(c,x_t),f(c,x_\infty)) &\leq (2^{d_v-1}-1)(2^{d_c-1}-1)H((x_t-x_\infty))\\
&= (2^{d_v-1}-1)(2^{d_c-1}-1)d(x_t,x_\infty)
\end{aligned}$$

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 $\epsilon^{\text{BP}} = 0.084$. 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 [?]. 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 $\epsilon < \epsilon_{\text{MAP}}$ and is strictly positive for $\epsilon > \epsilon_{\text{MAP}}$. Clearly we always have $\epsilon_{\text{BP}} < \epsilon_{rmMAP}$ 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 $\epsilon \leq \epsilon_{\text{MAP}}$ and is strictly positive for $\epsilon > \epsilon_{\text{MAP}}$. 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* schedule. 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 $X_1, \dots, X_{r-1}$. 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 $X_1, \dots, X_{l-1}$. 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.