Statistical Physics for Communications, Signal Processing, and Computer Science

EPFL

Nicolas Macris and Rüdiger Urbanke

Contents

Foreword

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

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

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

Second, except for extremely simple models, the "calculations" which are necessary are often long and daunting and not seldomly 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 does 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. Butt it is important to cleanly separate the two worlds.

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

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

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 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 studying. Using this familiarity, we can then explain basic statistical physics concepts and techniques. 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 1-5, 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 7, we then introduce the main protagonist, a generic message-passing algorithm which is also know as belief-propagation algorithm. The remaining chapters of the second part, Chapters 8-15, 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 19-21, 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.

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 and its analysis is quite long. We will give an outline of the whole story, but we will not discuss every 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.

Clearly the hardest problem is 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

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

1.1 Introduction

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 similar 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.

1.2 Coding

Basic definitions

Codes are used in order to reliably transmit information across a noisy channel. Let us start with a basic definitions.

DEFINITION 1.1 (Binary Block Code) A binary block code C of length n is a collection of binary n-tuples, $C = \{\underline{x}_1, \ldots, \underline{x}_\kappa\}$, where $\underline{x}_i$, $1 \leq i \leq \kappa$, is called a codeword, and where the components of the codeword are elements of $\mathbb{F}_2$, the binary field.

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.

DEFINITION 1.2 (Binary Linear Block Code) A binary linear binary block code is a subspace of $\mathbb{F}_s^n$. Equivalently, a binary block code C is linear iff for any two codewords $codewd_i$ and $\underline{x}_j$, $\underline{x}_i - \underline{x}_j \in C$. In particular $\underline{x}_i - \underline{x}_i = \mathbf{0} \in C$. Since C is a subspace it has a dimension, call it k, $0 \leq k \leq n$. Hence, $\kappa = |C| = 2^k$.

All codes which we consider in this course are binary and linear. Therefore, in the sequel we sometimes omit these qualifiers.

It will sometimes be convenient to represent C as the kernel 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

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

The factor graph associated to the parity-check matrix H (of a code C)

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

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

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

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

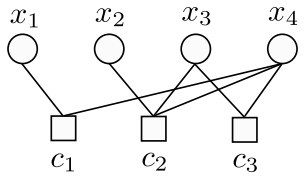


Figure 1.1 The factor graph corresponding to the parity-check matrix of Example 1.

The three main tasks related to the coding problme are *encoding*, *transmission*, and the *decoding*. Let us briefly discuss each of them.

- **Encoding:** Given 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, $g : \mathbb{F}_2^k \to C$, which maps each information word into a codeword.

  Although this function is of crucial immportance for real systems, it only plays a minor role for our purpose. This is true since, as we will discuss in more detail later one, for "typcial" channels, by symmetry the performance of the system is independent of the transmitted codeword. We therefore typically assume that the all-zero codeword was transmitted. Also, in terms of complexity, the encoding process is not a difficult task.
- **Transmission and channel model:** We assume that we pick a codeword $\underline{x}$ uniformly at random from the code 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 very complicated process we make a mathematical model of the end-to-end effect of all these physical processes. This is the "channel model" which we consider.

Channel Model: Formally, the channel has the input alphabet $\mathcal{X} = \{0, 1\}$ and an output alphabet $\mathcal{Y}$. We assume that the channel is *memoryless*, i.e., we assume 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 a product form since we assume 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). We will describe each of them in more detail later one.

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 quantity which we are intereste in and then to pick a decoding function which optimizes the quantity. The most common such criteria are:
  - minimize the block error probability: $\mathbb{P}\left[\underline{\hat{x}}(y) \neq \underline{x}\right]$, and
  - minimize the bit error probability: $\frac{1}{n}\sum \mathbb{P}\left[\underline{\hat{x}}(y)_i \neq x_i\right]$.

  In practice, due to complexity constraints, it is not always possible to implement an optimal decoding function. Rather one often implements a low-complexity algorithm. Of course, the closer we can pick it to optimal the better.

Gallager's (l, r)-regular ensemble and the configuration model

A common theme of 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, and then we study the "average" behavior of this ensemble. Once the average is determined, we know that there must be at least one 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 (l, r)-regular *Gallager* ensemble. It was introduced by Gallager in 1961, [**?**]. Rather than specifying the codes directly we specify their factor graphs.

The ensemble is characterized by the triple (n, l, r), where $n, l, r \in \mathbb{N}$, and also $n\frac{l}{r} \in \mathbb{N}$. The parameter n is the length of the code, l is the variable node degree, and r is the check node degree.

To sample from the ensemble we proceed as follows. Pick n variable nodes and $n\frac{l}{r}$ check nodes. Each variable node has l *sockets* and each check node has r *sockets*. Number the ln variable sockets in an arbitrary but fixed way from 1 till nl. Do the same with the nl check node sockets. Pick a permutation π uniformly at random from the set Π of permutations on nl letters. For i from 1 till nl, insert an edge which connect variable node socket i to check node socket $\pi(i)$. If, after construction, we delete sockets then we get a bipartite graph, which we call the factor graph of the parity check matrix H. Note that in this model there can be multiple edges between nodes. In practice this is not desirable and more sophisticated graph generation algorithms are employed. But for our purpose this will not play a role and we will ignore this issue in the sequel.

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. I.e., 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.

Shannon Capacity

Give a very short overview.

Questions

We have defined codes and introduced the decoding problem. In this context, there are some questions that we would like 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 (see Fig. 1.2)? 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?

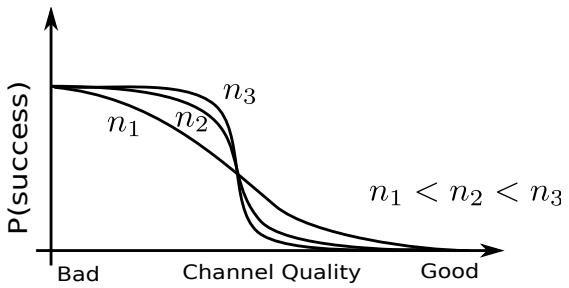


Figure 1.2 The probability of sucess of decoding the transmitted message versus the channel quality.

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

1.3 Compressive sensing

Here is the perhaps simplest version of compressive sensing. Let A be an $r \times n$ matrix with real entries, $1 \leq r < n$. Let $\underline{x} \in \mathbb{R}^n$, with $\|\underline{x}\|_0 = s < n$. All vectors are column vectors. Let $\underline{y} \in \mathbb{R}^r$ be given by $\underline{y} = A\underline{x}$. We think of $\underline{y}$ as the result of r linear measurements, one corresponding to each row of A. Our aim is to reconstruct $\underline{x}$ from $\underline{y}$.

Since $r < n$, and in fact r is typically *much smaller*, we cannot simply solve a linear system of equations since there will be many such solutions. But we

know in addition that $\underline{x}$ is s-sparse, i.e., only s entries, $s < n$, of x are non-zero. Therefore, we should look for

$$\{\hat{\underline{x}} : A\hat{\underline{x}} = \underline{y} \text{ and } \|\hat{\underline{x}}\|_0 = s\}. \tag{1.2}$$

If this set has cardinality one then we have found our solution x.

A sligthly more realistic version of compressive sensing is the model $\underline{y} = A\underline{x}+\underline{z}$, where $\underline{z}$ denotes a noise vector, typically assumed to consist of n iid zero-mean Gaussian random variables with a variance of σ^2.

If we ignore the sparsity constraint then it is natural to pick that $\hat{\underline{x}}$ which solves the least-squares problem,

$$\min_{\hat{\underline{x}}} \|A\hat{\underline{x}} - \underline{y}\|_2^2. \tag{1.3}$$

This problem is easily solved and the solution is given by $\hat{\underline{x}} = (A^T A)^{-1} A^T \underline{y}$.

But in general this solution will not be s-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,

$$\min_{\hat{\underline{x}}} \|A\hat{\underline{x}} - \underline{y}\|_2^2 + \lambda\|\hat{\underline{x}}\|_0,$$

for a properly defined constant λ. Unfortunately this minimization problem is hard. A mathematically easier version is to consider the following minimization problem

$$\min_{\underline{x}} \|A\underline{x} - \underline{y}\|_2^2 + \lambda\|\underline{x}\|_1.$$

This is called the LASSO. This problem can be solved by standard convex optimizaton techniques but in general we have lost something by this reformulation.

The objective of compressive sensing is to minimize the number of measurements while being able to recover the solution with low complexity and high probability. Our aim will be to analyse the trade-offs which are inherent in this problem. As for the previous two problems we will investigate the regime where the dimension n of the problem tends to infinity.

Graphical Representation

Ensembles

As for the coding and the K-SAT problem it is often convenient to define an ensemble of such problems. One common assumption is that the matrix A has iid Gaussian entries of zero mean and variance $\frac{1}{\sqrt{n}}$ so that each row of A has an expected L_2 norm of 1. Further, we will assume that $\underline{x}$ is chosen in the following way. Given s, pick s out of the n positions uniformly at random. In these positions the entries of $\underline{x}$ are s iid zero-mean Gaussian random variables of variance 1. The remaining positions are set to 0. Finally, the noise vector $\underline{z}$ consists of n iid zero-mean Gaussian random variables of variance σ^2.

Questions

Consider the regime where n tends to infinity and s/n is constant.

- How many measurements do we need so that with high probability we can recover $\underline{x}$ 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?

1.4 Satisfiability

Suppose that we are given a set of n Boolean variables $\{x_1, \ldots, x_n\}$. Each variable x_i can take on the values 0 and 1, where 0 means "false" and 1 means "true". We define a *literal* to be either a variable x_i or its negation $\bar{x}_i$. A *clause* is a disjunction of literals, e.g., $C = x_1 \vee x_2 \vee \bar{x}_3$ where the operator "$\vee$" denotes the Boolean "or" operator. 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 *satisfy* or *not satisfy*. 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 which is satisfied. A SAT formula is a conjunction of a set of clauses. For example, F which is defined as $F = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_2 \vee \bar{x}_4) \wedge x_3$ is a SAT formula.

DEFINITION 1.3 (SAT Problem) Given a SAT formula F on the variables $\{x_1, \ldots, x_n\}$ determine the satisfiability of F, i.e., determine if there exists an assignment on $\{x_1, \ldots, x_n\}$ so that F is satisfied. If such an assignment exists we might also want to find an explicit instance.

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 problem is NP-complete.

We say that a formula F is K-SAT, $K \in \mathbb{N}$, 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. Then the following facts are known:

- 2-SAT formulas are easy to check for satisfiability. Problem!1.1 discussed a simple algorithm called unit-clause propagation. It solves a 2-SAT formula in at most $2n$ steps.
- The K-SAT problem is NP-complete for $K \geq 3$.

Graphical representation of SAT formulas (using factor graphs)

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

EXAMPLE 2 (Factor Graph of SAT Formula) As an example, the graphical presentation of $F = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_2 \vee x_3 \vee \bar{x}_4)$ is shown in Fig. 1.3. □

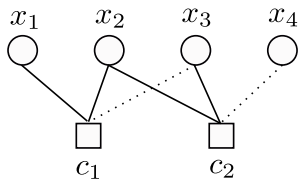


Figure 1.3 The factor graph corresponding to the SAT formula of Example 2.

Random K-SAT Formulas

We will be interested in the behavior of random K-SAT formulas. So let us define an *ensemble* of such formulas. The ensemble $\mathcal{F}(n, K, M)$ 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.

How to sample from $\mathcal{F}(n, K, M)$

We define $\mathcal{F}(n, K, M)$ by showing how to sample from it. To this end, pick M clauses 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.

Now let us consider the following experiment. Fix $K \geq 3$ (e.g., $K = 3$) and sample from the $\mathcal{F}(n, K, M)$ ensemble. Is such a formula satisfiable with high probability? It turns out that the most important parameter that effects the answer is $\alpha = \frac{M}{n}$.

Fig. 1.4 show the probability of satisfiability 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 α_K such that

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

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?

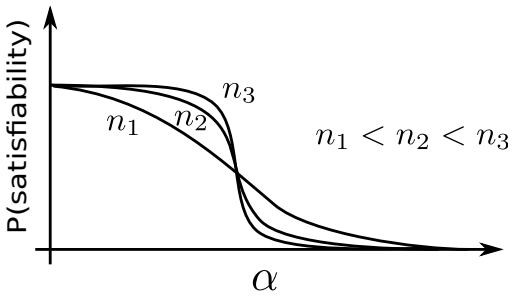


Figure 1.4 The probability that a formula generated from the random K-SAT ensemble is satisfied versus the clause density α.

1.5 Notes

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

Problems

1.1 The aim of this weeks homework is to write programs which can sample a random bipartite graph and then to test this program for random 3-SAT instances by running a very simple routine called unit clause propagation. You can use any languages you like.

1.Configuration Model Let l be the *variable* node degree and r be the *check* node degree. Pick n variable nodes and $m = nl/r$ check nodes. Each variable node has l *sockets* and each check node has r sockets. Number these sockets in a fixed but arbitrary order from 1 to nl on both sides. Pick a random permutation from the set of permutations on nl letters uniformly at random. Construct a bipartite graph by connecting the variable node socket i to check node socket $\pi(i)$. This is called the *configuration* model.

Your program should take as input the parameters n, m, l and r. It should check that the input is valid and return a bipartite graph according to this configuration model. Think about the data structure. Later on we will run algorithms on such a graph. It will then be 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.Poisson Model Pick two integers, n and m. As before, there are n variable nodes and m check nodes. Further, let r be the degree of a check node. For each check node pick r 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 since the node degree distribution on the variable nodes converges to a Poisson distribution for large n.

Again, think of the data structure. We will use this model right away to run some simple algorithm on it.

1.Unit Clause Propagation for Random 3-SAT Instances Generate random instances of the Poisson model. Pick $n = 10^5$ and let $r = 3$. 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 a bipartite graph as a random instance 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. The binary variable associated to each edge indicates whether in this clause we have the variable itself or its negation.

For each instance you generate, try now 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 now the 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? Hand in this plot.

2 Big Picture

The objective of these notes is to introduce common and powerful tools from statistical mechanics by showing how they can help us to answer the questions we introduced in the preceding chapter. But before we can delve into the details it might help to consider the bigger picture. Statistical mechanics is an old subject and as such it has developed its own mysterious language which can be daunting on a first encounter. It is like in the age old story of a prison where people have been telling the same set of jokes for such a long time that there is no need to recount any joke. Some says 11, and everyone laughs – except if you have just joined the club!

The following pages give a high-level summary of some of the techniques and how they can be applied. They are not meant to be read and uderstood in a single reading. Rather, skim over them to familiarize yourself with some important notions. From time to time reread them. Hopefully – after many paints – the language and notions will start to become clear and you will have joined the club – welcome!

In the first chapter we have introduced three problems: *codes* on sparse graphs, random **K-SAT**, and **compressive sensing**. We will see throughout the course that there is a coherent set of theoretical tools and concepts that can be used to analyze these models.

The mathematical formalism that we will develop has its origins in a variety of intersecting subjects such as statistical mechanics, probability theory, theoretical computer science, discrete structures, and information theory. At times one may loose track of the underlying unity of the subject, and it is therefore important to have a high level view. The goal of this chapter is to give a "road map" across the tools, methods and concepts that will be encountered through the lectures. The description given here is necessarily informal and short, the idea being that you can refer back to this chapter as the material builds up in the next chapters.

TO BE DONE: Rewrite old version of this chapter.

3 Principles of Statistical Mechanics

Gibbs distributions, and methods to study them, play a fundamental role when we want to study our three models. On the one hand these distributions can be viewed as purely mathematical objects. On the other hand, much insight can be gained by understanding why they are such natural objects. It is the goal of this chapter to explain some of this insight.

Statistical mechanics describes the behavior of *macroscopic systems* that are composed of a large number of degrees of freedom. For example condensed matter systems are composed of $\approx 10^{23}$ atoms, molecules, magnetic moments (or spins) etc. Similarly, we are interested in the system behavior of our models when the size of the system becomes large.

A precise knowledge and description of the deterministic microscopic motion of each molecule in a macroscopic system would be impossible and is in fact not required for the understanding of the macroscopic properties of the system. The general approach of statistical mechanics is to replace the full deterministic microscopic description in terms of laws of motion, by a probabilistic description based on appropriate probability distributions.[1]

For most physical systems, the correct probabilistic description is known only once the so-called *thermodynamic equilibrium* is reached. A system is said to be in thermodynamic equilibrium if its temperature is homogeneous so that there are no heat currents, its pressure is homogeneous so that there are no mechanical stresses, its chemical potential is homogeneous so that there are no particle currents and chemical reactions[2]. The second law of thermodynamics states that an isolated system will, after a long enough time reach the state of thermodynamic equilibrium. In this state the macroscopic laws of usual thermodynamics apply.

The probability distributions that are relevant to systems in thermodynamic equilibrium are called Maxwell-Boltzmann or more generally Gibbs distributions. In this chapter we will derive Gibbs distributions from two principles which one may consider as our definition of thermodynamic equilibrium.

Systems that are not in thermodynamic equilibrium are said to be *out of equilibrium*. Their fundamental probabilistic description(s) (if there is one) is not

[1] At the turn of the 19th to 20th century this statistical description constituted an important shift of paradigm, which emerged through the works of Maxwell, Boltzmann, Planck, Gibbs, Einstein and others.

[2] It is not possible to give a completely logical a priori definition of thermodynamic equilibrium that is not void, or circular, or makes no use of other abstract concepts.

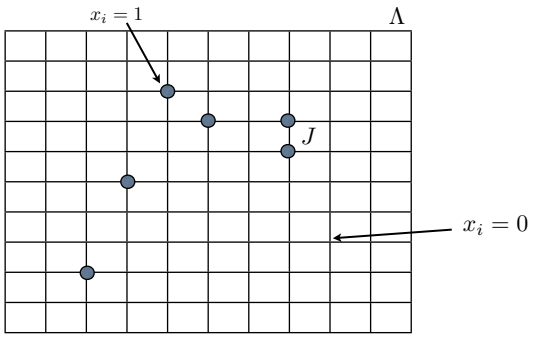


Figure 3.1 The lattice gas model. At most one particle occupies a lattice site. There is an energy cost for neighboring particles.

yet elucidated. Such systems range all the way from stationary heat or electric flows up to the more fancy living systems.

The Gibbs measures that we seek do not depend on the detailed form of the microscopic dynamics for the degrees of freedom (the equations of motion of Newton for classical particles or of Heisenberg for a quantum system) but only on the fact that there exist conserved quantities. In fact even if the dynamics is unknown, or unspecified, or random, we can write down the Gibbs measures simply in terms of the conserved quantities.

The prime example of a conserved quantity is the energy. We will stick to the simple case where there is only one conserved quantity, namely the energy. It will also be useful to have a concrete working example in mind. The following is a toy model which turns out to be one of the most important and most studied models of classical statistical mechanics. We will have more to say about it at a later point.

EXAMPLE 3 (Lattice gas model) Let us replace the continuum space by a discrete d-dimensional grid (see Fig. 3.1; 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 will 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. We suppose that only neighboring atoms interact and that the interaction "energy" is $-J$ ($J < 0$ corresponds to repulsion and $J > 0$ to attraction). 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

$$\mathcal{H}(\underline{x}) = -J \sum_{\langle i,j\rangle \in E} x_i x_j - \mu \sum_{i \in V} x_i. \tag{3.1}$$

Each edge $\langle i, j\rangle$ is counted once in the sum.

The real number μ is a cost associated to the presence or absence of a particle (this might be a chemical affinity or a chemical potential). The detailed dynamics $x_i(t)$, $i \in V$, as a function of time t is not specified here. But we *assume* in this model that the dynamics is such that the total energy, call it E, of the system is conserved. This means that we assume that at any time t we have

$$\mathcal{H}(\underline{x}(t)) = E. \tag{3.2}$$

The set of configurations satisfying this equation is called the energy surface and is denoted Γ_E. Note that $\Gamma_E \subset \{0,1\}^{|V|}$.

EXAMPLE 4 (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 *spins* $s_i = \pm$, $i \in V$, which are binary variables taking values in $\{+1, -1\}$. More precisely, the Hamiltonian is

$$\mathcal{H}(\underline{s}) = -J \sum_{\langle i,j\rangle \in E} s_i s_j - h \sum_{i \in V} s_i. \tag{3.3}$$

where $\underline{s} = (s_1, \ldots, s_{|V|})$. 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 $x_i = \frac{1-s_i}{2}$ or $(-1)^{x_i} = s_i$ and redefining the interaction constants.

Remark 3.1 Real world systems have continuous degrees of freedom. In classical particle systems, $\{0,1\}^{|V|}$ is replaced by the phase space which is the set of all positions and velocities, and Γ_E really is a hyper-surface in phase space. For magnetic systems, the spins $s_i = \pm$ are replaced by vector-like quantities. For quantum systems, the degrees of freedoms such as positions, velocities and spins are non commuting operators (matrices). Remarkably, despite these significant differences with the simple systems introduced above, the basic concepts of statistical mechanics are the same as in the discrete setting.

3.1 Two principles

We will derive the Maxwell-Boltzmann or Gibbs distributions from two basic principles. In this section we state the principles and the Gibbs distribution is derived in the next section. There is no unique or canonical way of introducing

a new physical law: ultimately it has to be guessed and validated by countless experiments. This is the case for the Gibbs distribution. This is also why we do not attempt nor desire to make the discussion in this section more formal. The two principles of this section should be viewed as way to guess the Gibbs distribution.

The microcanonical measure

Let $[0,T]$ be the time interval over which we measure an observable quantity $\phi(\underline{x}(t))$ and let τ be a characteristic microscopic time scale, for example the time scale on which a single spin flips or an atom 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

$$\frac{1}{T}\int_0^T dt\, \phi(\underline{x}(t)), \tag{3.4}$$

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 (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 $\Gamma_E \subset \{0,1\}^{|V|}$. Let $t(\underline{x})$ be the total time it spends in state $\underline{x}$. We assume that when thermodynamic equilibrium is reached, the ratio $t(\underline{x})/T$ is the probability to find the system in state $\underline{x}$ when it is observed at a random time in the interval $[0,T]$ (here $T \gg \tau$). What is this probability distribution ?

Our first principle states that *for an isolated system* this probability distribution is the uniform measure on the energy surface $\Gamma_E = \{\underline{x} | \mathcal{H}(\underline{x}) = E\}$. In other words for $t(x)/T$ we take,

$$\mu_{\text{micro}}(\underline{x}) = \frac{\mathbb{I}(\underline{x} \in \Gamma_E)}{\sum_{\underline{x}\in\{0,1\}^{|V|}} \mathbb{I}(\underline{x} \in \Gamma_E)}. \tag{3.5}$$

This measure is called the *microcanonical measure*. 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 (3.4) by a configurational average,

$$\frac{1}{T}\int_0^T dt\, \phi(\underline{x}(t)) = \frac{\sum_{\underline{x}\in\{0,1\}^{|V|}} \mathbb{I}(\underline{x} \in \Gamma_E)\phi(\underline{x})}{\sum_{\underline{x}\in\{0,1\}^{|V|}} \mathbb{I}(\underline{x} \in \Gamma_E)}, \qquad T \gg \tau \tag{3.6}$$

Remark 3.2 (Ergodic hypothesis.) Often equ. (3.6) is formalized and called the *ergodic hypothesis*. The ergodic hypothesis states that the above equation is an exact consequence of the microscopic deterministic dynamics, 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 $\phi(\underline{x})$. This hypothesis

has led to a deep branch of mathematics called "ergodic theory". The ergodic hypothesis has only been proved for simple systems with a few particles (a finite number of them) and very simple dynamical laws. The first proof goes back to Sinai (around 1970) for one particle in a billiard shaped region, whose dynamics is given by straight lines reflecting at the billiard walls, and has since then been extended to a finite fixed number of hard spheres in the billiard. Such systems are not macroscopic and are do not display thermodynamic behavior.

Remark 3.3 (Criticism.) In the context of statistical mechanics which deals with macroscopic systems, the ergodic hypothesis has never been proven and its validity has been much debated. The modern point of view is that this hypothesis is neither sufficient, nor necessary for the foundations of statistical mechanics. It is not sufficient because we know it is true for model systems with a few degrees of freedom (e.g Sinai billiards) which have no thermodynamic behavior. It is not necessary because it is too strong an hypothesis. It would be enough that it is only true for a reasonable class of observables (e.g sum functions) and for a sufficiently large number of degrees of freedom (see Kintchine). One obvious objection to the relevance of the ergodic hypothesis and the microcanonical measure is that real systems in thermal equilibrium are never isolated, but in contact with a thermal bath. There are approaches to the foundations of statistical mechanics that altogether avoid the ergodic hypothesis and the microcanonical measure (see for example Landau and Lifshitz) but directly "derive" the Gibbs distribution from other arguments. For one thing, we all know of systems that are not ergodic, but are still described by the Gibbs distribution that will be derived in the next section. For example in a ferromagnet that is magnetized, the typical configurations of spins mostly point in one direction, and do not uniformly explore the whole configuration space.

Boltzmann principle

Consider the normalization term of the microcanonical measure (3.5). Set

$$W(E) = \sum_{\underline{x} \in \{0,1\}^{|V|}} \mathbb{I}(\underline{x} \in \Gamma_E). \tag{3.7}$$

For the systems of interest the above expression has an exponential behavior as the size of the system grows.

$$\sum_{\underline{x} \in \{0,1\}^{|V|}} \mathbb{I}(\underline{x} \in \Gamma_E) \simeq \exp(\mathcal{S}(E)), \tag{3.8}$$

with $\mathcal{S}(E) = O(|V|)$. Define the *Boltzmann entropy* as

$$\mathcal{S}_{\text{Boltz}}(E) = \ln W(E). \tag{3.9}$$

A priori, this is a purely mathematical combinatorial quantity.

EXAMPLE 5 Let us consider the lattice gas model introduced in the previous

example for the simple case $J = 0$. Pick E/μ lattice nodes among $|V|$ nodes with the state $+1$ and the rest 0. Hence,

$$W(E) = \binom{|V|}{E/\mu} \simeq \exp\left(|V| h_2(\frac{E}{\mu|V|})\right), \tag{3.10}$$

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

$$s(e) = \lim_{\substack{|V|\to\infty \\ E/|V|=e}} \frac{\mathcal{S}(E)}{|V|} = h_2\left(\frac{E}{\mu|V|}\right) = h_2\left(\frac{e}{\mu}\right), \tag{3.11}$$

where $e = E/|V|$. Note that this is a concave function (for physically sensible Hamiltonians the Boltzmann entropy is a concave function of e; this is not always the case in computer science and coding problems with hard constraints).

There is a purely thermodynamic 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 in their work on heat engines. This is an experimentally measurable quantity. Here we cannot enter into this discussion. We will just say that for a system at thermodynamic equilibrium with homogeneous temperature T and pressure p the thermodynamic entropy $\mathcal{S}_{\text{thermo}}(E, V)$ is a function of the total energy E and volume V satisfying

$$\frac{\partial \mathcal{S}_{\text{thermo}}}{\partial E} = \frac{1}{T}, \qquad \frac{\partial \mathcal{S}_{\text{thermo}}}{\partial V} = \frac{p}{T}\,. \tag{3.12}$$

From T and p one can in principle recover $\mathcal{S}_{\text{thermo}}$.

We are now ready to state our second principle, which is due to Boltzmann. If you visit Boltzmann's grave in Vienna you will see the inscription $S = k \ln W$. This formula[3] states that the thermodynamic entropy (left hand side) is equal to the combinatorial entropy (right hand side). The left hand side is a physical quantity that can in principle be measured and the right hand side is a mathematical quantity that can in principle be calculated.

More formally, the Boltzmann principle states that

$$\mathcal{S}_{\text{thermo}} = k_B \mathcal{S}_{\text{Boltz}}, \tag{3.13}$$

Here k_B is Boltzmann's constant that relates temperature units of the thermometer with energy units (one can always measure temperature in units of energy and set $k_B = 1$, this is not usual practice though).

This principle makes the connection between statistical mechanics and thermodynamics. It allows to compute thermodynamic quantities from combinatorial/statistical considerations. Besides, as we will see, it is a crucial ingredient in the derivation of the Gibbs distribution.

[3] which was explicitly stated in this form by Planck

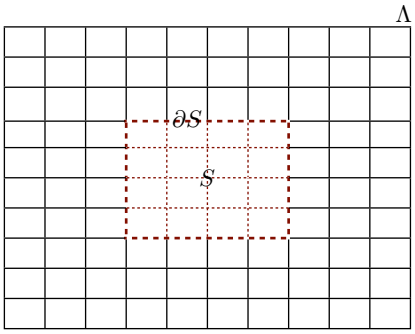


Figure 3.2 The system S is embedded in a thermal bath V. The total system V is considered as an isolated system and its total energy E is conserved. We compute the induced measure on S.

3.2 The Gibbs measure

The microcanonical measure 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. The measure that is appropriate for such a setting is the Gibbs measure.

For simplicity, let us again take our lattice gas model where we have a *large isolated system* denoted by the graph $G = (V, E)$. We assume that this large isolated system has reached thermal equilibrium with temperature T. Therefore, we know that it is described by

$$\mu_{\text{micro}}(\underline{x}) = \frac{\mathbb{I}(\underline{x} \in \Gamma_E)}{\sum_{\underline{x} \in \{0,1\}^{|V|}} \mathbb{I}(\underline{x} \in \Gamma_E)}. \tag{3.14}$$

Now, let us consider *a much smaller but still macroscopic system* $S \subset V$ (see Figure 3.2). The main question we answer in this section is: what is the induced measure on S? The probability that the configuration of this smaller systems is $x_1, \ldots, x_{|S|}$ reads

$$\mu_{\text{ind}}(x_1, \ldots, x_{|S|}) = \sum_{x_{|S|+1}, \ldots, x_{|V|}} \mu_{\text{micro}}(x_1, \ldots, x_{|V|}) = \frac{\sum_{x_{|S|+1}, \ldots, x_{|V|}} \mathbb{I}(\underline{x} \in \Gamma_E)}{\sum_{x_1, \ldots, x_{|V|}} \mathbb{I}(\underline{x} \in \Gamma_E)}. \tag{3.15}$$

The total energy can be written as,

$$\begin{aligned} E &= \mathcal{H}(x_1, \dots, x_{|V|}) \\ &= \mathcal{H}_S(x_1, \dots, x_{|S|}) + \mathcal{H}_{V\setminus S}(x_{S+1}, \dots, x_{|V|}) + \mathcal{H}_{int}, \end{aligned}$$

where $\mathcal{H}_{int}$ is the term capturing the interactions between particles in the sets S and V. Note that in general we have $\mathcal{H}_S = O(|S|)$, $\mathcal{H}_{V\setminus S} = O(|V \setminus S|)$ and $\mathcal{H}_{int} = O(|\partial S|)$. Since $O(|V \setminus S|) \gg O(|S|) \gg O(|\partial S|)$, the term $\mathcal{H}_{int}$ can be neglected from the above expression for energy. Note however that this is the term that allowed S to reach thermal equilibrium through the interactions with the bath. For fixed $x_1, \dots, x_{|S|}$ we get

$$\begin{aligned} \mu_{\text{ind}}(x_1, \dots, x_{|S|}) &= \frac{\sum_{x_{|S|+1},\dots,x_{|V|}} \mathbb{I}((x_{|S|+1}, \dots, x_{|V|}) \in \Gamma_{E-\mathcal{H}_S(x_1,\dots,x_{|S|})})}{\sum_{x_1,\dots,x_{|S|}} \sum_{x_{|S|+1},\dots,x_{|V|}} \mathbb{I}((x_{|S|+1}, \dots, x_{|V|}) \in \Gamma_{E-\mathcal{H}_S(x_1,\dots,x_{|S|})})} \\ &= \frac{\exp(\mathcal{S}(E - \mathcal{H}_S(x_1, \dots, x_{|S|}))}{\sum_{x_1,\dots,x_{|S|}} \exp(\mathcal{S}(E - \mathcal{H}_S(x_1, \dots, x_{|S|}))} \\ &\overset{(a)}{=} \frac{\exp(\mathcal{S}(E) - \mathcal{H}_S(x_1, \dots, x_{|S|})\frac{\partial \mathcal{S}}{\partial E} + \dots)}{\sum_{x_1,\dots,x_{|S|}} \exp(\mathcal{S}(E) - \mathcal{H}_S(x_1, \dots, x_{|S|})\frac{\partial \mathcal{S}}{\partial E} + \dots)} \\ &\overset{(b)}{=} \frac{\exp\left(-\frac{\mathcal{H}_S(x_1,\dots,x_{|S|})}{k_B T}\right)}{\sum_{x_1,\dots,x_{|S|}} \exp\left(-\frac{\mathcal{H}_S(x_1,\dots,x_{|S|})}{k_B T}\right)}, \end{aligned}$$

where in (a) we used the Taylor expansion and in (b) Boltzmann's principle (18.9) together with (3.12). The resulting measure is nothing else than the Gibbs measure.

DEFINITION 3.1 (Gibbs measure) We define the Gibbs measure of the system S at thermal equilibrium with a bath of temperature T as

$$\mu_{\text{Gibbs}}(x_1, \dots, x_{|S|}) = \frac{1}{Z} \exp\left(-\frac{\mathcal{H}_S(x_1, \dots, x_{|S|})}{k_B T}\right), \tag{3.16}$$

where the normalizing factor Z is called the *partition function*

$$Z = \sum_{x_1,\dots,x_{|S|}} \exp\left(-\frac{\mathcal{H}_S(x_1, \dots, x_{|S|})}{k_B T}\right)$$

Remark 3.4 In this derivation an important assumption was that $\mathcal{H}_{int}$ between the system and its complement can be neglected. For finite dimensional systems with local (i.e. finite range, or fast decaying with distance) interactions between particles this is always true. However if one deals with infinite dimensional systems (meaning here that $d \to +\infty$ or that the graph G cannot be metrically embedded in a finite dimensional space) or if the interactions are very long ranged this assumption may be problematic. This is the case for gravitational interactions for example. Such systems are not described by standard statistical mechanics and thermodynamics.

3.3 Free energy, entropy and equivalence of ensembles

The formulation in terms of the Gibbs measure above is also called *canonical ensemble* formulation. In practice which ensemble should one choose for the theoretical description of a large system: the microcanonical or the canonical ? No system is really isolated and conceptually the canonical description is more natural. However for large systems the energy fluctuations are negligible (of the order of the surface to be compared to the volume) and the microcanonical can also be used. It is a matter of convenience which one to choose[4] and there are rules that allow to pass from one ensemble to another.

In the microcanonical ensemble one computes the *entropy* per unit volume

$$s(e) = \lim_{\substack{|V|\to\infty \\ E/|V|=e}} \frac{1}{|V|} \ln W(E). \tag{3.17}$$

In the canonical ensemble the relevant quantity is the *free energy* per unit volume

$$f(T) = -k_B T \lim_{|S|\to\infty} \frac{1}{|S|} \ln Z. \tag{3.18}$$

One can show that free energy and entropy are related by a *Legendre transformation*,

$$f(T) = \min_e (e - k_B T s(e)). \tag{3.19}$$

Note that $f(T)$ is a concave function of T. If $s(e)$ is also concave, the Legendre transform can be inverted, and the entropy recovered from the free energy. This is what is meant by equivalence of ensembles. We stress that the equivalence of ensembles does not hold when $s(e)$ is not concave.

Let us sketch the derivation of the last relation. The partition function can be written as

$$\begin{aligned} \sum_{x_1,\ldots,x_{|S|}} \exp\left(-\frac{\mathcal{H}_S(x_1,\ldots,x_{|S|})}{k_B T}\right) &= \sum_E W(E) \exp(-\frac{E}{k_B T}) \\ &\approx |S| \int de\, e^{-|S|(\frac{e}{k_B T} - s(e))} \end{aligned}$$

Taking the logarithm on both sides and going to the infinite size limit yields

$$\lim_{|S|\to+\infty} \frac{1}{|S|} \ln Z = -\min_e (\frac{e}{k_B T} - s(e)) \tag{3.20}$$

which is equivalent to the relationship between $f(T)$ and $s(e)$.

Remark 3.5 According to the physical situation, other measures or ensembles may be more convenient or relevant. When there are many conserved quantities

[4] In principle. An important condition is locality of interactions.

besides energy, call them $I_j(\underline{x})$, $j = 1, ..., g$, one can take for the statistical mechanics description of the system the measure (or ensemble),

$$\mu(\underline{x}) = \frac{1}{Z} \exp(-\sum_{j=1}^{g} \mu_j I_j(\underline{x})) \tag{3.21}$$

where the multipliers μ_j have thermodynamic interpretations. The multiplier associated to conserved energy is the inverse temperature; the one associated to conserved particle number is the chemical potential; the one associated to conserved volume is pressure, etc.... All the Legendre transformations between relevant thermodynamic quantities can be derived similarly than above.

3.4 Marginals and the thermodynamic limit

Usually the Gibbs measure contains too much information. It is often enough to calculate the first two marginals. More precisely,

$$\mu_i(x_i) = \sum_{\sim x_i} \mu_{Gibbs}(x_1, \ldots, x_{|S|}), \tag{3.22}$$

and

$$\mu_{i,j}(x_i, x_j) = \sum_{\substack{\sim x_i \\ \sim x_j}} \mu_{Gibbs}(x_1, \ldots, x_{|S|}). \tag{3.23}$$

It is usually enough to know the averages[5]

$$\langle x_i \rangle = \sum_{x_i} x_i \mu_i(x_i) = \sum_{x_1, \ldots, x_{|S|}} x_i \, \mu_{Gibbs}(x_1, \ldots, x_{|S|}), \tag{3.24}$$

and

$$\langle x_i x_j \rangle = \sum_{x_i, x_j} x_i x_j \mu_{i,j}(x_i, x_j) = \sum_{x_1, \ldots, x_{|S|}} x_i x_j \, \mu_{Gibbs}(x_1, \ldots, x_{|S|}). \tag{3.25}$$

Note that for binary variables $x_i = 0, 1$ (or ± 1) these averages suffice to reconstruct the marginals μ_i and $\mu_{i,j}$. The following covariance is usually called a *correlation function*

$$C_{i,j} = \langle x_i x_j \rangle - \langle x_i \rangle \langle x_j \rangle. \tag{3.26}$$

A simple but fundamental fact, is that these quantities can all be computed once the free energy is known. Let us modify slightly the Gibbs measure[6] by introducing extra "source" factors (the λ_i),

$$\mu^{\underline{\lambda}}_{Gibbs}(x_1, \ldots, x_{|S|}) = \frac{\exp\left(-\beta \mathcal{H}_S(x_1, \ldots, x_{|S|} + \sum_{i=1}^{|S|} \lambda_i x_i)\right)}{Z_{\underline{\lambda}}}, \tag{3.27}$$

[5] The bracket $\langle - \rangle$ is the standard notation for expectations with respect to Gibbs distributions.

[6] Here we use the standard notation $\beta = \frac{1}{k_B T}$.

where $Z_{\underline{\lambda}}$ is the normalization factor. The reader should check the very important identities

$$\langle x_i \rangle_{\underline{\lambda}} = \frac{\partial}{\partial \lambda_i} \ln Z_{\underline{\lambda}}, \tag{3.28}$$

and

$$\langle x_i x_j \rangle_{\underline{\lambda}} - \langle x_i \rangle_{\underline{\lambda}} \langle x_j \rangle_{\underline{\lambda}} = \frac{\partial^2}{\partial \lambda_i \partial \lambda_j} \ln Z_{\underline{\lambda}}. \tag{3.29}$$

To calculate the original quantities namely, $\langle x_i \rangle$ and $\langle x_i x_j \rangle$, we only need to compute $\ln Z_\lambda$ near $\underline{\lambda} = \underline{0}$. Let us warn the reader that although it sometimes happens that $\ln Z$ is known at $\underline{\lambda} = 0$, for $\underline{\lambda} \neq 0$ small the problem is orders of magnitude harder.

Statistical mechanics describes macroscopic systems. This regime is captured by computing the free energy and marginals in the infinite size limit,

$$\lim_{|S| \to +\infty} \frac{1}{|S|} \ln Z, \qquad \lim_{|S| \to +\infty} \langle x_i \rangle, \qquad \lim_{|S| \to +\infty} \langle x_i x_j \rangle. \tag{3.30}$$

This limit is called the *thermodynamic limit*.

One of the ambitions mathematical statistical mechanics is to make sense of the thermodynamic limit for the Gibbs distribution itself. The reader can appreciate that this is not an obvious problem simply by the fact that an infinite number of variables will be involved and that the limits of the numerator and denominator (taken separately) do not make sense. The idea is to reconstruct the full measure from the limiting marginals. It turns out that the limits of marginals depend on boundary conditions or added infinitesimal perturbations (such as the $\lambda \to 0$ terms) and as a result the limiting Gibbs measures are not necessarily unique. This is the case precisely when phase transitions are present: a unique microscopic Hamiltonian can lead to many possible phases of matter (water-ice-gas) each being described by one of the limiting Gibbs measures. This fundamental feature of Gibbs distributions gained recognition only in the 1940-50's through the works of Bethe, Peierls, Onsager. The mathematical theory of phase transitions developed in the late 1960's and is still a very active subject.

Problems

3.1 In the following problems you will solve the Ising model in one dimension: this is the simplest model for the interaction of magnetic moments of atoms in a crystal. We assume N even for simplicity and let $i \in \{-\frac{N}{2}, \cdots, \frac{N}{2}\}$ label the $N+1$ vertices of a one dimensional chain of atoms. We attach spin variables $s_i \in \{-1, +1\}$ to each site of the chain (these are the magnetic moments of atoms sitting at positions i). The Hamiltonian (or energy function, or cost function) of the one-dimensional Ising model is

$$\mathcal{H}_N = -J \sum_{i=-\frac{N}{2}}^{\frac{N}{2}-1} s_i s_{i+1} - H \sum_{i=-\frac{N}{2}}^{\frac{N}{2}} s_i \tag{3.31}$$

Here $J > 0$ is the interaction constant between spins (ferromagnetic case) and $H \in \mathbb{R}$ an external magnetic field. When the system is at thermal equilibrium at temperature T, the probability of a configuration $\{s_i\}$ is given by the Gibbs distribution (k is Boltzmann's constant defined such that kT has units of energy)

$$\mu(\{s_i\}) = \frac{1}{Z_N} e^{-\frac{\mathcal{H}}{kT}}, \qquad \text{where} \qquad Z_N = \sum_{\{s_i = \pm 1\}} e^{-\frac{\mathcal{H}_N}{kT}} \tag{3.32}$$

is the partition function (in German "Zustandssumme" which means "sum over states"). The following notation is standard: $\frac{1}{kT} = \beta$, $\beta J = K$, $\beta H = h$.

The first problem introduces the transfer matrix method, which is a general way of solving one-dimensional models. The second problem is concerned with boundary conditions. In the third one you will solve the same model thanks to the message passing approach which we will develop further in the course.

3.2 Transfer matrix method In this problem we take a periodic boundary condition which leads to simpler calculations. This means that the sites $i \in \{-\frac{N}{2}, \cdots, \frac{N}{2}\}$ are arranged on a circle, and that there is an extra interaction term in (5.27), namely $-J s_{-\frac{N}{2}} s_{\frac{N}{2}}$ (since the two extremities of the chain have been brought next to each other). Consider the *transfer matrix*

$$T = \begin{pmatrix} e^{K+h} & e^{-K} \\ e^{-K} & e^{K-h} \end{pmatrix} \tag{3.33}$$

A. Show that the partition function can be expressed as

$$Z_N = tr\,(T^N). \tag{3.34}$$

where tr is the sum over eigenvalues.

B. Find the eigenvalues of T and show that the free energy per spin is in the thermodynamic limit

$$f(h) \equiv -\lim_{N \to +\infty} \frac{1}{\beta N} \ln Z_N = -\beta^{-1} \ln[e^K \cosh h + (e^{2K} \sinh^2 h + e^{-2K})^{1/2}]. \tag{3.35}$$

C. Compute the magnetization from the *thermodynamic definition*: $m = -\frac{\partial}{\partial H} f(h)$ 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 $T > 0$[7].

D. Now we want to compute the local magnetization at a fixed site i, and the

[7] In his 1925 PhD thesis, under Lenz's guidance, Ising mistakenly concluded from this calculation that the model would not exhibit any phase transition even when formulated on two or three dimensional square grids. It was only in 1936 that Peierls proved the existence of a phase transition at a finite temperature for dimensions greater or equal to 2.

correlation between two spins at sites i and j, namely

$$\langle s_i \rangle = \frac{\sum_{\{s_k=\pm 1\}} s_i e^{-\frac{\mathcal{H}}{kT}}}{Z_N}, \qquad \langle s_i s_j \rangle = \frac{\sum_{\{s_k=\pm 1\}} s_i s_j e^{-\frac{\mathcal{H}}{kT}}}{Z_N} \tag{3.36}$$

Introduce a matrix $S = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and express these two quantities in terms of traces involving S and T. Noting that T can be diagonalized by an orthogonal rotation of angle ϕ, deduce that

$$\lim_{N\to+\infty} \langle s_i \rangle = \cos 2\phi, \qquad \lim_{N\to+\infty} \langle s_i s_j \rangle = \cos^2 2\phi + \sin^2 2\phi \left(\frac{\lambda_-}{\lambda_+}\right)^{|j-i|} \tag{3.37}$$

were $\lambda_- < \lambda_+$ are the eigenvalues of T. Check that the first formula above agrees with m found in 1.c. Interpret the second formula.

3.3 Message passing method Consider the model on the open chain with free boundary conditions (no constraint on the end spins). we want to compute $\langle s_i \rangle$ for a fixed i, in the infinite size limit $N \to +\infty$, by an iterative method. For simplicity consider the middle spin $\langle s_0 \rangle$. You can convince yourself that the method works for any fixed i.

A. In the expression for $\langle s_i \rangle$ perform the sums over the end spins $s_{-\frac{N}{2}}$ and $s_{\frac{N}{2}}$. Show that this leads to a spin system with the new hamiltonian

$$\beta \mathcal{H}_N^{(1)} = -K \sum_{i=-\frac{N}{2}+1}^{\frac{N}{2}-2} s_i s_{i+1} - h \sum_{i=-\frac{N}{2}+2}^{\frac{N}{2}-2} s_i - (h + \tanh^{-1}(\tanh K \tanh h))(s_{-\frac{N}{2}+1} + s_{-\frac{N}{2}-1}) \tag{3.38}$$

B. Iterate to show that

$$\lim_{N\to+\infty} \langle s_0 \rangle = \tanh(h + 2\tanh^{-1}(\tanh K \tanh u)) \tag{3.39}$$

where u is the solution of the fixed point equation

$$u = h + \tanh^{-1}(\tanh K \tanh u) \tag{3.40}$$

Incidentally, show that the solution of this fixed point equation is unique so that there is no ambiguity in this result. For this point it is useful to note that if a mapping is a contraction i.e, $\sup_u |g'(u)| < 1$, then the sequence $u_{t+1} = g(u_t)$ is Cauchy.

C. Check that the result agrees with the expression for m found in the first problem. Calculations are maybe simpler if you use the identity

$$\tanh(x + y) = \frac{\tanh x + \tanh y}{1 + \tanh x \tanh y} \tag{3.41}$$

4 Formulation of Problems as Spin Glass Models

Let us now reformulate all three problems in a statistical physics language. We will work out the reformulation for coding and compressive sensing in detail. The reformulation for the K-SAT problem is the topic of the homework this week.

Note that both the coding as well as the compressive sensing problem are inference problems. Therefore in principle both reformulations are straightforward. Start by writing down the a postior distribution. By taking the logarithm we can rewrite this posterior in an exponential form, i.e., in the form of a Gibbs measure. This in itself is not surprising and also not terribly helpful. But we will see that in both cases the rewriting results in a quite natural formulation.

4.1 Coding as a spin glass model

Let $\mathcal{C}$ be a code from the Gallager Ensemble LDPC(l, r, n). Recall that l is the degree of variable nodes, and that r is the degree of check nodes. Further, n is the length of the codewords, and $ln = rm$ 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 always assume that the codeword is selected uniformly at random. We will use the spin variable notation for the codebits. This means that we write $s_i = (-1)^{x_i}$. The channel is memoryless and described by transition probabilities

$$q(\underline{y}|\underline{s}) = \prod_{i=1}^{n} q(y_i|s_i) \tag{4.1}$$

The two examples which we will refer most often are the binary symmetric channel (BSC) and the binary additive white Gaussian noise channel (BAWGNC).

MAP decoding

Let $p(\underline{s}|\underline{y})$ be the posterior probability distribution of $\underline{s}$ given the received word $\underline{y}$. The bit-MAP estimator is defined as

$$\hat{s}_i(\underline{y}) = \text{argmax}_{s_i}\ p(s_i|\underline{y}) \tag{4.2}$$

where $p(s_i|\underline{y})$ is the marginal of the posterior $p(\underline{s}|\underline{y})$.[1] This estimator is optimal in the sense that it minimizes the probability of error. Suppose that the transmitted word s_i^{in} is picked uniformly at random from the code $\mathcal{C}$. Denote by $\mathbb{E}_{\underline{Y}|\underline{s}^{\text{in}}}$ the expectation over the channel outputs when $\underline{s}^{\text{in}}$ is sent. The average over all n bits of the bit-probability of error is

$$\begin{aligned}\mathbb{P}[\text{error}] &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|\mathcal{C}|}\sum_{\underline{s}^{\text{in}}\in\mathcal{C}}\mathbb{P}[\hat{s}_i(\underline{Y})\neq s_i^{\text{in}}] \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|\mathcal{C}|}\sum_{\underline{s}^{\text{in}}\in\mathcal{C}}\mathbb{E}_{\underline{Y}|\underline{s}^{\text{in}}}[\mathbb{1}(\hat{s}_i(\underline{Y})\neq s_i^{\text{in}})] \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|\mathcal{C}|}\sum_{\underline{s}^{\text{in}}\in\mathcal{C}}\frac{1}{2}\Big(1-\mathbb{E}_{\underline{Y}|\underline{s}^{\text{in}}}\big[s_i^{\text{in}}\hat{s}_i(\underline{Y})\big]\Big)\end{aligned} \tag{4.3}$$

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

The MAP estimate of a whole block is $\hat{\underline{s}}_B(\underline{y}) = \operatorname{argmax}_{\underline{s}}\, p(\underline{s}|\underline{y})$, and associated the block probability of error is $\mathbb{P}_B[\text{error}] = \frac{1}{|\mathcal{C}|}\sum_{\underline{s}^{\text{in}}\in\mathcal{C}}\mathbb{P}[\hat{\underline{s}}_B(\underline{Y})\neq\underline{s}^{\text{in}}]$. We will see below that block-MAP decoding is equivalent to finding the minimum energy states of a Hamiltonian.

The MAP decoder as a spin glass model

We now show that the posterior distribution $p(\underline{s}|\underline{y})$ is a *random* Gibbs measure. The *randomness* comes from both the channel as well as the choice of code. Let us start with a few preliminary observations.

A code word $\underline{x}$ has to satisfy all parity check constraints $\sum_{i\in\partial a}x_i = 0$, which in spin language is equivalent to $\prod_{i\in\partial a}s_i = 1$. Thus the prior distribution over codewords can be written as

$$p_0(\underline{s}) = \frac{1}{|\mathcal{C}|}\prod_{a=1}^{m}\mathbb{1}(\underline{s}\text{ satisfies }a) = \frac{1}{|\mathcal{C}|}\prod_{a=1}^{m}\frac{1}{2}(1+\prod_{i\in\partial a}s_i). \tag{4.4}$$

Channel outputs can be expressed in terms of their half-loglikelihoods

$$h_i = \frac{1}{2}\ln\frac{q(y_i|+1)}{q(y_i|-1)}. \tag{4.5}$$

It equivalent to describe the channel outputs by $\underline{h}$ or $\underline{y}$, therefore we will sometimes freely interchange them in our notations.

[1] MAP stands for "maximum a posteriori".

Using the Bayes law, the channel law (4.1), and (4.4), (4.5) we obtain

$$\begin{aligned} p(\underline{s}|\underline{y}) &= \frac{q(\underline{y}|\underline{s})p_0(\underline{s})}{p(\underline{y})} \\ &= \frac{p_0(\underline{s})\prod_{i=1}^n q(y_i|s_i)p(\underline{s})}{\sum_{\underline{s}} p_0(\underline{s})\prod_{i=1}^n q(y_i|s_i)} \\ &= \frac{1}{Z}\prod_{a=1}^m \frac{1}{2}(1+\prod_{i\in\partial a} s_i)\prod_{i=1}^n e^{h_i s_i}. \end{aligned} \tag{4.6}$$

where the denominator is

$$Z = \sum_{\underline{s}}\prod_{a=1}^m \frac{1}{2}(1+\prod_{i\in\partial a} s_i)\prod_{i=1}^n e^{h_i s_i}. \tag{4.7}$$

To get the last equality in (4.6) $p(y_i|s_i) = p(y_i|s_i = +1)e^{-h_i}e^{h_i s_i}$ and that the terms $p(y_i|+1)e^{-h_i}$ in the numerator and denominator cancel.

The posterior $p(\underline{s}|\underline{y})$ is a *random Gibbs distribution*. By this we mean that for each channel realization $\underline{h}$ and each code $\mathcal{C}$ picked from the Gallager ensemble we have a measure over the spins $\underline{s} \in \{-1,+1\}^n$. An important feature of the Gibbs distribution is the factorization into a product of "local" terms, i.e., terms which depend on a finite number of spins.

Another name for random Gibbs distribution over a set of spins is "spin-glass model". Let us very briefly point out where this terminology comes from. Chemical (window) glass is an amorphous material where atoms have a random spatial ordering, unlike crystals such as quartz. There are also magnetic materials with frustrated or randomly distributed ferromagnetic and antiferromagnetic interactions that induce magnetic disorder. These are called spin-glasses. While chemical glass is very important to our daily life, magnetic glasses have virtually no applications. The generic property that is believed to be common to glassy materials is the existence of microscopic degrees of freedom with vastly different time scales, the so called "annealed or dynamical" and "quenched or frozen" degrees of freedom. Spin-glass models were introduced in the 1970's to model glassy behavior. These are "toy models" where the dynamical degrees of freedoms are the spins which are distributed according to a Gibbs measure depending on random variables, the quenched degrees of freedom, whose realizations are fixed. Although the direct applicability of spin-glass models to real glassy materials is unclear, the study of these models has allowed great conceptual advances related to the glass problem in general. From our perspective here these models naturally arise in many engineering problems, coding, compressive sensing and constraint satisfaction being three paradigms. In the language of spin-glass physics, the channel outputs and the code or Tanner graph are the quenched or frozen variables. Once they are fixed we do not change them. The spins are the dynamical variables which adjust themselves in the environment of the quenched variables.

The bit-MAP decoder has a natural relation to the magnetization of the spin

system. To see this, note

$$\begin{aligned}\hat{s}_i(\underline{y}) &= \operatorname{sign}(p(s_i = 1|\underline{y}) - p(s_i = -1|\underline{y})) \\ &= \operatorname{sign}(\sum_{s_i} s_i p(s_i|\underline{y})),\end{aligned} \tag{4.8}$$

The bit-MAP estimate is given by the sign of the average of s_i with respect to the posterior $p(\underline{s}|\underline{y})$. This average is nothing else than the magnetization

$$\langle s_i \rangle = \frac{1}{Z(\underline{h})} \sum_{\underline{s}} s_i \prod_{a=1}^{m} \frac{1}{2}(1 + \prod_{i \in \partial a} s_i) \prod_{i=1}^{n} e^{h_i s_i} \tag{4.9}$$

To summarize,

$$\hat{s}_i(\underline{y}) = \operatorname{sign}(\langle s_i \rangle) \tag{4.10}$$

and the average probability of error (4.3) is directly related to the magnetization,

$$\mathbb{P}[\text{error}] = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|\mathcal{C}|} \sum_{\underline{s} \in \mathcal{C}} \frac{1}{2} \left(1 - \mathbb{E}_{\underline{Y}|\underline{s}^{\text{in}}} \left[s_i^{\text{in}} \operatorname{sign}(\langle s_i \rangle) \right] \right). \tag{4.11}$$

In the expectation the $\underline{Y}$ dependence is implicit in $\langle s_i \rangle$.[2] The magnetization is a very natural quantity to handle in the context of Gibbs measures. The presence of the sign unfortunately makes this quantity harder to deal with. However as we will see in coding theory also one can measure the performance with more tractable quantities that do not involve the awkward sign function. The important point is that these quantities have the same threshold behavior than the average probability of error.

Hamitonian interpretation

In Chapter 3 we defined the Gibbs measures in the more traditional way through Hamiltonians. What is the Hamiltonian associated to the MAP decoder? One way to identify it is to write

$$p(\underline{s}|\underline{y}) = \lim_{K_a \to +\infty} \frac{\prod_{a=1}^{m} (1 + \tanh K_a \prod_{i \in \partial a} s_i) \prod_{i=1}^{n} e^{h_i s_i}}{\sum_{\underline{s}} \prod_{a=1}^{m} (1 + \tanh K_a \prod_{i \in \partial a} s_i) \prod_{i=1}^{n} e^{h_i s_i}}. \tag{4.12}$$

Using the identity (use $\prod_{i \in \partial a} s_i = \pm 1$)

$$\begin{aligned} e^{K_a \prod_{i \in \partial a} s_i} &= \cosh K_a + \sinh K_a \prod_{i \in a} s_i \\ &= \cosh K_a (1 + \tanh K_a \prod_{i \in \partial a} s_i), \end{aligned}$$

[2] In the statistical mechanics notations one often omits such dependencies, but if emphasis is needed one writes $\langle s_i \rangle_{\underline{Y}}$ or $\langle s_i \rangle(\underline{Y})$.

we get

$$p(\underline{s}|\underline{y}) = \frac{e^{-\mathcal{H}(\underline{s}|\underline{h},\mathcal{C})}}{\sum_{\underline{s}} e^{-\mathcal{H}(\underline{s}|\underline{h},\mathcal{C},)}}, \tag{4.13}$$

for the Hamiltonian

$$\mathcal{H}(\underline{s}|\underline{y},\mathcal{C}) = \lim_{K_a \to +\infty} \Big\{ -\sum_{a=1}^{m} K_a (\prod_{i \in \partial a} s_i - 1) - \sum_{i=1}^{n} h_i s_i \Big\}. \tag{4.14}$$

The limit takes values in the extended real line $\bar{\mathbb{R}} = \mathbb{R} \cup 0$. If the spin assignement satisfies all parity checks a the limit is equal to $-\sum_{i=1}^{n} h_i s_i$, while if one parity check is violated it is equal to $+\infty$. This is the Hamiltonian of a spin system. The parity-check constraints appear as interactions or couplings between spins. Their strength is infinite $K_a \to +\infty$ because the parity checks are hard constraints. The channel output h_i biases the spin s_i in a particular direction. In statistical physics language we say that the channel realization acts like a magnetic field h_i which biases the spins. The temperature in this spin system is $k_B T = 1$.

Finite temperature decoder

It is sometimes useful to take a broader view and look at a slightly more general Gibbs measures defined for any temperature. We will adopt the standard notation $\beta = (k_B T)^{-1}$ for the *inverse temperature*. Consider

$$p_\beta(\underline{s}|\underline{y}) = \frac{e^{-\beta \mathcal{H}(\underline{s}|\underline{h},\mathcal{C})}}{\sum_{\underline{s}} e^{-\beta \mathcal{H}(\underline{s}|\underline{h},\mathcal{C},)}}. \tag{4.15}$$

One can define a "finite temperature decoder" by

$$\hat{s}_{i,\beta}(\underline{y}) = \mathrm{sign} \langle s_i \rangle_\beta \tag{4.16}$$

where $\langle - \rangle_\beta$ is the Gibbs bracket corresponding to (4.15). The average bit probability of error is given by the same formula than in (4.3) where the magnetization is replaced by the one at inverse temperature β.

The finite temperature decoder interpolates between the bit-MAP and block-MAP decoders as the inverse temperature β varies from 1 to $+\infty$ (the temperature varies from one down to to zero). Of course, setting $\beta = 1$ we get back the bit-MAP decoder. As we will see in the next section for $\beta = 1$ there are remarkable symmetry properties, that are absent for other values of β, and make the analysis of the bit-MAP decoder much easier. For $\beta \to +\infty$ the Gibbs measure (4.15) concentrates on the spin configurations that minimize the Hamiltonian. It is not hard to see that

$$\lim_{\beta \to +\infty} \langle \underline{s}_i \rangle_\beta = \mathrm{argmin}_{\underline{s}^{\mathrm{in}}} \mathcal{H}(\underline{s}|\underline{h},\mathcal{C}) = \hat{\underline{s}}_B(\underline{y}) \tag{4.17}$$

Therefore block MAP decoding is equivalent to finding the lowest energy state of the Hamiltonian. It is also equivalent to computing the magnetization at zero temperature.

4.2 Channel symmetry and gauge transformations

For "symmetric" channels remarkable simplifications apply when $\beta = 1$ (bit-MAP decoding). The BEC, the BSC, as well as the BAWNGC belong to this class.

DEFINITION 4.2.1 A binary input channel is said to be (output) *symmetric* if $p(y_i|s_1) = p(-y_i| - s_i)$.

It is useful to see how this property translates in terms of the half-loglikelihood distribution. Note that $c(h_i)dh_i = p(y_i|+1)dy_i$. It follows that 4.2.1 is equivalent to

DEFINITION 4.2.2 A binary input channel is said to be (output) *symmetric* if $c(-h) = c(h)e^{-2h}$.

For our main examples of symmetric channels this property can be explicitly checked on the expressions:

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

Let $\underline{\tau} = (\tau_1, \ldots, \tau_n)$ be a codeword in $\mathcal{C}$. It is immediate to see that $p(\underline{s}|\underline{h})$ is invariant under the transformation $s_i \mapsto \tau_i s_i, h_i \mapsto \tau_i h_i$. Note that this works because τ is a codeword so that $\prod_{i\in\partial a}\tau_i = 1$ for all a. Since codewords form a group, the set of such transformations also form a group. Moreover the transformations are local in the sense that each variable gets multiplied by a different sign. In physics, transformations with these two properties are called "gauge transformations" and when they leave a Hamiltonian invariant one says that the system has a "gauge symmetry"[3].

Channel symmetry and linearity of the code therefore induces a gauge symmetry of the spin glass model. Let us now explore the most important consequences of this symmetry.

Independence of input codeword

In this paragraph and the next one it is convenient to use the notation $\underline{v} \star \underline{u}$ for the "Hadamard product" of two vectors $(v_i u_i, i = 1, \ldots n)$, and also $\mathbb{E}_{\underline{h}|\underline{s}^{\text{in}}}$ for $\mathbb{E}_{\underline{Y}|\underline{s}^{\text{in}}}$ The invariance of the Gibbs distribution under a gauge transformation

[3] 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 the four fundamental forces that are known: elctromagnetic, weak, strong, gravitational.

implies $\langle s_i \rangle \to \tau_i \langle s_i \rangle$, where $\langle - \rangle$ is the *same* expectation on both sides. Therefore

$$\begin{aligned} \mathbb{E}_{\underline{h}|\underline{s}^{\text{in}}}[\text{sgn}\langle s_i \rangle] &= \mathbb{E}_{\underline{\tau}\star\underline{h}|\underline{s}^{\text{in}}}[\tau_i \text{sgn}\langle s_i \rangle] \\ &= \tau_i \mathbb{E}_{\underline{h}|\ldots\underline{\tau}\star\underline{s}^{\text{in}}}[\text{sgn}\langle s_i \rangle] \end{aligned} \tag{4.18}$$

Using $\underline{\tau} = \underline{s}^{\text{in}}$ for the gauge, we get

$$\mathbb{E}_{\underline{h}|\underline{s}^{\text{in}}}[\text{sgn}\langle s_i \rangle] = s_i^{\text{in}} \mathbb{E}_{\underline{h}|+1}[\text{sgn}\langle s_i \rangle] \tag{4.19}$$

and so equation (4.3) simplifies to

$$\mathbb{P}[\text{error}] = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2}(1 - \mathbb{E}_{\underline{h}|\underline{1}}[\text{sgn}\langle s_i \rangle]) \tag{4.20}$$

Thus, for symmetric channels we can assume without loss of generality that the input word is the all $+1$ word (note that in the 0/1 language this is the all 0 codeword). From now on we denote $\mathbb{E}_{\underline{h}|\underline{1}}$ or $\mathbb{E}_{\underline{Y}|\underline{1}}$ as $\mathbb{E}_{\underline{h}}$ or $\mathbb{E}_{\underline{Y}}$.

Nishimori identities

Gauge symmetry of spin glass models implies a host of useful identities between averages of Gibbs brackets. These identities are often called Nishimori identities. Here we only prove the simplest possible such identity for the Gibbs distribution associated to a symmetric channel and any linear code.

Proposition 4.2.3 (Simplest Nishimori Identity for Coding) For a binary input, output symmetric, memoryless without feedback channel, and any linear code

$$\mathbb{E}_{\underline{h}}[\langle s_i \rangle] = \mathbb{E}_{\underline{h}}[\langle s_i \rangle^2].$$

Proof Using a gauge transformation, $s_j \to \tau_j s_j$, $h_j \to \tau_j h_j$ for $\tau \in \mathcal{C}$, together with channel symmetry (definition 4.2.2), we have,

$$\begin{aligned} \mathbb{E}_{\underline{h}}[\langle s_i \rangle] &= \mathbb{E}_{\underline{\tau}\star\underline{h}}[\tau_i \langle s_i \rangle] \\ &= \mathbb{E}_{\underline{h}}\left[\tau_i \langle s_i \rangle \prod_{j=1}^{n} e^{h_j \tau_j - h_j}\right] \end{aligned} \tag{4.21}$$

Summing this identity over all the codewords $\underline{\tau} \in \mathcal{C}$

$$\begin{aligned} \mathbb{E}_{\underline{h}}[\langle s_i \rangle] &= \frac{1}{|\mathcal{C}|} \mathbb{E}_{\underline{h}}\left[Z \langle \tau_i \rangle \langle s_i \rangle \prod_{j=1}^{n} e^{-h_j}\right] \\ &= \frac{1}{|\mathcal{C}|} \sum_{\underline{\eta} \in \mathcal{C}} \mathbb{E}_{\underline{h}}\left[\langle \tau_i \rangle \langle s_i \rangle \prod_{j=1}^{n} e^{h_j \eta_j} \prod_{j=1}^{n} e^{-h_j}\right] \end{aligned} \tag{4.22}$$

The last equality has been obtained by spelling out the partition function Z as a sum over $\underline{\eta} \in \mathcal{C}$. Now, for each term in this sum, we do a gauge transformation

$s_j \to \eta_j s_j$, $\tau_j \to \eta_j \tau_j$, $h_j \to \eta_j h_j$. This sum becomes

$$\begin{aligned}
&\frac{1}{|\mathcal{C}|} \sum_{\underline{\eta} \in \mathcal{C}} \mathbb{E}_{\underline{\eta} \star \underline{h}} \left[\langle \tau_i \rangle \langle s_i \rangle \eta_i^2 \prod_{j=1}^{n} e^{h_j \eta_j^2} \prod_{j=1}^{n} e^{-h_j \eta_j} \right] \\
&\quad = \frac{1}{|\mathcal{C}|} \sum_{\underline{\eta} \in \mathcal{C}} \mathbb{E}_{\underline{h}} [\langle \tau_i \rangle \langle s_i \rangle] \\
&\quad = \mathbb{E}_{\underline{h}} [\langle s_i \rangle^2] \qquad (4.23)
\end{aligned}$$

The first equality follows from channel symmetry 4.2.2, and the second one is trivial because τ and s are dummy variables i.e., so $\langle \tau \rangle = \langle s \rangle$. □

4.3 Conditional entropy and free energy in coding

Recall from Chapter 3 that the free energy $-\ln Z/n$ plays an important role. For example differentiating it with respect to h_i yields the magnetization $\langle s_i \rangle$. For spin glass models this quantity is random, but almost always concentrates in the thermodynamic limit. This may be quite hard to prove but we have examples where such a proof exists. Therefore it is relevant to consider the *average free energy* over channel realizations: $-\mathbb{E}_{\underline{Y}}[\ln Z]/n$.

The following important result gives the connection between the free energy and the conditional entropy.

Proposition 4.3.1 For transmission over a symmetric channel and any fixed linear code we have the following relation

$$H(\underline{X}|\underline{Y}) = \mathbb{E}_{\underline{Y}}[\ln Z] - n \int dh \, c(h) h. \qquad (4.24)$$

The last term depends only on the underlying channel: thus one may say that the *average* over channel outputs of the free energy and Shannon's conditional entropy are essentially one and the same thing.

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.

Proof in the case of a Gaussian channel There are various ways to prove this relation. Here we show one that is valid for the BIAWGNC not because it is the simplest, but because it illustrates a nice use of Nishimori identities. The proof for general channels can be found in the literature.

First note that for the BAWGNC the last term is equal to σ^{-2}.

$$\begin{aligned} H(\underline{X}|\underline{Y}) &= -\mathbb{E}_{\underline{Y}}\Big[\sum_{\underline{s}} p(\underline{s}|\underline{y}) \ln p(\underline{s}|\underline{y})\Big] \\ &= \mathbb{E}_{\underline{Y}}[\ln Z(\underline{y})] - \mathbb{E}_{\underline{Y}}\Big[\sum_{\underline{s}} p(\underline{s}|\underline{y}) \ln \prod_{c\in\mathcal{C}} \frac{1}{2}(1+\prod_{i\in c} s_i)\Big] \\ &\quad - \mathbb{E}_{\underline{Y}}\Big[\sum_{\underline{s}} p(\underline{s}|\underline{y}) \sum_{i=1}^{n} h_i s_i\Big] \\ &= \mathbb{E}_{\underline{Y}}[\ln Z(\underline{y})] - \sum_{i=1}^{n} \mathbb{E}_{\underline{Y}}[h_i \langle s_i\rangle] \end{aligned} \tag{4.25}$$

It remains to show $\mathbb{E}_{\underline{Y}}[h_i\langle s_i\rangle] = \sigma^{-2}$, an identity that does not seems trivial at first sight. First we consider the average over h_i only. For a BAWGNC we check by explicit calculation that $\sigma^2 c(h)h = -\frac{\partial}{\partial h}c(h) + c(h)$ and use integration by parts to obtain

$$\begin{aligned} \sigma^2 \int dh_i\, c(h_i) h_i \langle s_i\rangle &= \int \Big(-\frac{\partial}{\partial h_i} c(h_i) + c(h_i)\Big)\langle s_i\rangle \\ &= \int dh_i\, c(h_i)\Big(\frac{\partial}{\partial h_i}\langle s_i\rangle + \langle s_i\rangle\Big) \\ &= \int dh_i\, c(h_i)\Big(\langle s_i^2\rangle - \langle s_i\rangle^2 + \langle s_i\rangle\Big) \\ &= 1 - \int dh_i\, c(h_i)\langle s_i\rangle^2 + \int dh_i\, c(h_i)\langle s_i\rangle \end{aligned}$$

Averaging over all other h_j's and using the Nishimori identity we find $\mathbb{E}_{\underline{Y}}[h_i\langle s_i\rangle] = \sigma^{-2}$ as required. □

4.4 Compressive Sensing as a spin glass model

Recall that we are considering the model

$$\underline{y} = A\underline{x} + \underline{z}, \tag{4.26}$$

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 r 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 ϵ-*sparse* class, $p_0 \in \mathcal{F}_\epsilon$, that is

$$p_0(x) = (1-\epsilon)\delta(x) + \epsilon\phi_0(x) \tag{4.27}$$

where ϕ_0 is a continuous positive and normalized density. The conditional probability of observing y given x is

$$q(\underline{y} \mid \underline{x}) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} e^{-\frac{1}{2\sigma^2}\|\underline{y}-A\underline{x}\|_2^2}, \tag{4.28}$$

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

$$p(\underline{x}, \underline{y}) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} e^{-\frac{1}{2\sigma^2}\|\underline{y}-A\underline{x}\|_2^2} \prod_{i=1}^{n} p_0(x_i). \tag{4.29}$$

One can distinguish two scenarios. In the first one ϕ_0 is known, in which case 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). In the second scenario which is more realistic one only knows that the prior belongs to $\mathcal{F}_\epsilon$. In other words ϵ is assumed to be known but not ϕ_0. As explained in Chapter **??** in this case a popular choice for the estimator is the Lasso. The justification for choosing this estimator somehow comes a posteriori. We will see in Chapter 13 that in a sense this estimator is as good as pure l_1 minimization for the noiseless problem, over the whole region of parameters were $l_0 - l_1$ equivalence holds. This is enough justification to define these two estimators.

MMSE estimator

Given an observation $\underline{y}$ the minimum mean square estimator is defined as[4]

$$\underline{\hat{x}}_\sigma(\underline{y}) = \mathbb{E}_{\underline{X}|\underline{y}}[X] = \int d^n\underline{x}\, \underline{x}\, p(\underline{x} \mid \underline{y}), \tag{4.30}$$

This estimator involves the posterior $p(\underline{x} \mid \underline{y})$, which analogously to the case of coding, we will interpret as a Gibbs distribution. This is an optimal estimator in the sense that it minimizes the mean squared error. On the downside, this estimator is in general hard to compute, an it requires the knowledge of the prior distribution which is often not a very realistic assumption.

We recall the proof that (4.30) is the unique minimizer of the MSE. The MSE is the functional over the space of estimators $\underline{\hat{x}}(\underline{y}) : \mathbb{R}^r \to \mathbb{R}^n$

$$\mathrm{MSE}[\underline{\hat{x}}] = \mathbb{E}[(\underline{\hat{x}}(\underline{Y}) - \underline{X})^2] \tag{4.31}$$

Here the expectation is with respect to the joint distribution (4.29) and the i.i.d gaussian entries of A. For any variation $\underline{\hat{x}}(\underline{y}) \to \underline{\hat{x}}(\underline{y}) + \underline{\eta}(\underline{y})$ we have

$$\begin{aligned} \mathrm{MSE}[\underline{\hat{x}} + \underline{\eta}] - \mathrm{MSE}[\underline{\hat{x}}] &= \mathbb{E}[(\underline{\hat{x}}(\underline{Y}) + \underline{\eta}(\underline{Y}) - \underline{X})^2 - (\underline{\hat{x}}(\underline{Y}) - \underline{X})^2] \\ &= 2\mathbb{E}[\underline{\eta}^T(\underline{Y})(\underline{\hat{x}}(\underline{Y}) - \underline{X})] + \mathbb{E}[\|\underline{\eta}(\underline{Y})\|_2^2] \end{aligned} \tag{4.32}$$

For the estimator (4.30) the first variation vanishes for all $\underline{\eta}(\underline{y})$, while the second

[4] We adopt the notation $d\underline{x} = \prod_{i=1}^n dx_i$.

variation is positive *for arbitrary* $\underline{\eta}(\underline{y})$. Thus estimator (4.30) indeed gives the minimum mean square error (and hence is called the MMSE).

Lasso estimator

The Lasso estimator is defined as

$$\underline{\hat{x}}_\lambda(\underline{y}) = \mathrm{argmin}_{\underline{x}} \left\{ \frac{1}{2} \|\underline{y} - A\underline{x}\|_2^2 + \lambda \|x_i\|_1 \right\}. \tag{4.33}$$

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,

$$\inf_{\lambda \in \mathbb{R}} \sup_{p_0 \in \mathcal{F}_\epsilon} \mathbb{E}[(\underline{\hat{x}}_\lambda(\underline{Y}) - \underline{X})^2] \tag{4.34}$$

In this expression the expectation is over the joint distribution (4.29) and the random matrix ensemble.

It is not easy to unambiguously justify a priori the choice of this estimator. We will be able to solve exactly this problem in Chapter 13 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. This justifies the use of Lasso a posteriori. In paragraph 4.4 we give a somewhat more phenomenological justification which does not require to develop the whole theory. We will see that the Lasso estimator can be considered as a zero temperature limit of the MMSE estimator when a Laplacian prior is assumed for the unknown distribution p_0.

MMSE estimation as a spin glass problem

Let us discuss the posterior measure that is needed in the MMSE (4.30). From 4.29

$$\begin{aligned} p(\underline{x} \mid \underline{y}) &= \frac{1}{Z} e^{-\frac{1}{2\sigma^2}\|\underline{y} - A\underline{x}\|_2^2} \prod_{i=1}^n p_0(x_i) \\ &= \frac{1}{Z} \prod_{a=1}^m e^{-\frac{1}{2\sigma^2}(y_a - A_a^T \underline{x})^2} \prod_{i=1}^n p_0(x_i), \end{aligned} \tag{4.35}$$

where A_a^T is the a-th row of the matrix A. The explicit expression of the normalisation factor is

$$Z = \int \prod_{i=1}^n dx_i p_0(x_i) \prod_{a=1}^m e^{-\frac{1}{2\sigma^2}(y_a - A_a^T \underline{x})^2} \tag{4.36}$$

This formulation is of course in the spirit of factor graphs since all components in (4.35) are factorized.

The connections with statistical mechanics of spin glasses are analogous to the case of coding. The posterior (4.35) can be though of as a random Gibbs distribution and (4.36) as a partion function. The dynamical variables $x_i \in \mathbb{R}$ belong to a continuous alphabet, and one often speaks of "continuous spins". The measure is random because the measurement matrix A and the observations $\underline{y}$ are r.v. In the language of spin glasses these are the "frozen" or "quenched" r.v.

The MMSE estimator is nothing else than a "magnetization" for the "continuous spins". In statistical mechanics notation for each component (4.30) reads

$$\hat{x}_i(\underline{y}) = \langle x_i \rangle = \int d\underline{x}\, x_i\, p(\underline{x} \mid \underline{y}) = \int dx_i x_i\, p(x_i \mid \underline{y}), \tag{4.37}$$

Hamiltonian interpretation and Lasso estimator

One may ask, what is the hamiltonian associated to the Gibbs measure (4.35)? The answer is simple. For given A and $\underline{y}$ it is

$$\begin{aligned}\mathcal{H}(\underline{x}|\underline{y}, A) &= \frac{1}{2\sigma^2}\|\underline{y} - A\underline{x}\|_2^2 - \sum_{i=1}^{n} \ln p_0(x_i) \\ &= \frac{1}{2\sigma^2}\sum_{i,j=1}^{n}(A^T A)_{ij} x_i x_j - \sum_{i=1}^{n}\{p(x_i) + \frac{1}{2\sigma^2}(\sum_{a=1}^{m} y_a A_{ai}) x_i\} + \frac{1}{\sigma^2}\|\underline{y}\|_2^2\end{aligned} \tag{4.38}$$

We have expanded the norm in order to interpret more explicitely this Hamiltonian. The first term is an interaction between pairs of "continuous spins" with random interaction strengths $(A^T A)_{ij}$. The second term is analogous to a "magnetic field" term in the sense that it does not involve interactions between continuous spins.

Notice that if we formally replaced $-\ln p_0(x_i)$ by $\lambda|x_i|$ the minimizer of the Hamiltonian would yield the Lasso estimate. But why should we make this choice for p_0 and why should we consider the minimum of the Hamiltonian? In coding we briefly discussed the finite temperature decoder. One can proceed similarly here and look at a generalized estimator based on the Gibbs measure at inverse temperature β,

$$p_\beta(\underline{x}|\underline{y}) = \frac{e^{-\beta\mathcal{H}(\underline{x}|\underline{y},A)}}{\int d^n\underline{x}\, e^{-\beta\mathcal{H}(\underline{x}|\underline{y},A)}}.$$

The generalized estimator is

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

Of course for $\beta = 1$ we get back the usual MMSE. In the zero temperature limit $\beta \to +\infty$ the configurations that dominate the integral are those that minimize the Hamiltonian. One can show that

$$\lim_{\beta\to+\infty} = \mathrm{argmin}_{\underline{x}}\mathcal{H}(\underline{x}|\underline{y}, A)$$

Now, when the prior is not known one may decide to take $p_0(x_i) = e^{-\lambda|x_i|}$ on "phenomenological grounds". In the generalized estimator $p_0(x_i)^\beta = e^{-\beta\lambda|x_i|}$ and as $\beta \to +\infty$ this choice of p_0 gives most of the weight to signal components that vanish. The parameter λ is left open, and its optimal value as a function of ϵ is determined by the minimax problem (4.34).

4.5 Free energy and conditional entropy in compressive sensing

In this paragraph we assume that the prior is known. We derive a relation between the conditionnal entropy and the average free energy that is perfectly analogous to the one for coding in section 4.3. In fact the derivation is easier than in coding and is a matter of simple algebra.

We expect that in the large size limit $n \to \infty$ the free energy $\frac{1}{n}\ln Z$ concentrates. It is therefore relevant to consider the *average free energy* (over A and $\underline{y}$). In fact, the following discussion holds if we just look at the average free energy over $\underline{y}$ only $-\mathbb{E}_{\underline{Y}}[\frac{1}{n}\ln Z]$ and A is fixed.

Proposition 4.5.1 For any fixed A we have the following relation

$$H(\underline{X}) - H(\underline{X}|\underline{Y}) = -\mathbb{E}_{\underline{Y}}[\ln Z(\underline{y}] - \frac{n}{2} \tag{4.39}$$

Note $H(\underline{X}) = nH(X)$. It is pleasing to see that the left hand side is the mutual information $I(\underline{X};\underline{Y})$. Thus in this context mutual information and average free energy are essentially the same.

Proof By definition (expectation with respect to the joint distribution $p(\underline{x}|,\underline{y})$)

$$H(\underline{X}|\underline{Y}) = -\mathbb{E}_{\underline{X},\underline{Y}}[\ln p(\underline{X}|\underline{Y})] \tag{4.40}$$

The logarithm of the posterior distribution is equal to

$$-\frac{1}{2\sigma^2}\|\underline{y} - A\underline{x}\|_2^2 + \sum_{i=1}^n \ln p(x_i) - \ln Z(\underline{y}) \tag{4.41}$$

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

$$\mathbb{E}_{\underline{X},\underline{Y}}\left[\sum_{i=1}^n \ln p(X_i)\right] = \sum_{i=1}^n \mathbb{E}_{\underline{X}}[\ln p(X_i)] = -nH(X) \tag{4.42}$$

For the first term it is convenient to write down explicitely the integrals

$$\begin{aligned} -\frac{1}{2\sigma^2}\int d\underline{x}\int d\underline{y}\, p(\underline{x},\underline{y})\|\underline{y} - A\underline{x}\|_2^2 & \\ = -\frac{1}{2\sigma^2}\int \prod_{i=1}^n dx_i p(x_i)\int d\underline{y}\,\|y\|_2^2 \frac{e^{-\frac{1}{2\sigma^2}\|\underline{y}\|_2^2}}{(2\pi\sigma^2)^{n/2}} & \\ = -\frac{n}{2} & \end{aligned} \tag{4.43}$$

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

4.6 K-SAT as a spin glass model

Make a brief section out of problem. Also formulate K-sat at finite temperature is useful for later on.

Problems

4.1 The goal of this homework is to discuss the statistical mechanical formulation of the random K-SAT problem. We consider the ensemble of random formulas $\mathcal{F}(n, K, M)$ defined in chapter one (in class). The clause density will be denoted $\alpha = M/n$. In the first problem you will write the Hamiltonian and the statistical mechanical measures in the spin language. In the second problem you will derive a very elementary upper bound on the sat-unsat phase transition threshold α_s. Hint: there are no big calculations in this homework.

Given a formula $F \in \mathcal{F}(n, K, M)$ consider the following cost function:

$$\mathcal{H}_F(x_1, ..., x_n) = \text{number of clauses violated by the assignment } x_1, ..., x_n. \tag{4.44}$$

This is our Hamiltonian or energy function (x_i the Boolean variables).

4.2 Hamiltonian, microcanonical measure, finite temperature Gibbs measure Introduce the "spin" variables $s_i = (-1)^{x_i}$ that take values in $\{-1, +1\}$. Furthermore if clause c_a contains x_i associate $J_{ai} = +1$, and if it contains $\bar{x}_i$ associate $J_{ai} = -1$. Thus full edges have $J_{ai} = +1$ and dashed edges have $J_{ai} = +1$, and J_{ai} are Bernoulli(1/2).

(a) Verify that each clause contributes a term

$$\prod_{i \in c_a} \left(\frac{1 + s_i J_{ia}}{2}\right) \tag{4.45}$$

and then, write down the Hamiltonian or energy function in the spin language.

(b) Explain in one sentence which are dynamical variables and which are the frozen (or equivalently quenched) random variables in the problem.

(c) Show that the following counts the number of solutions of F

$$Z = \sum_{s_1, ..., s_n \in \{-1,+1\}^n} \prod_{a=1}^{M} \left(1 - \prod_{i \in c_a} \left(\frac{1 + s_i J_{ia}}{2}\right)\right) \tag{4.46}$$

(d) Convince yourself that the microcanonical measure for the zero-energy surface is nothing else than the uniform measure over solutions of F. Also, convince

yourself that Z is the partition function (normalization factor) of the microcanonical zero-energy measure. Note that this measure is well defined only if F admits at least one solution.

(d) Now take the Hamiltonian found in question (a) and write down the Gibbs measure for inverse temperature β. Note that this measure has the advantage that it is always well defined, i.e even if F does not have a solution. Consider the free energy $f_F(\beta)$ (normalized by the number of variables) for a fixed formula F. Show that

$$\lim_{\beta\to+\infty} \beta^{-1} f_F(\beta) = \frac{1}{n} \min_{\underline{x}} \mathcal{H}(\underline{x}) \tag{4.47}$$

This formula is interesting because if we succeed in computing the free energy and if its zero temperature limit is non zero, then we can deduce that F is unsat. The catch is that computing the free energy is a difficult problem.

4.3 Crude upper bound on 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.

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

b) Show that

$$\mathbb{E}[Z] = 2^n (1 - 2^{-K})^M. \tag{4.48}$$

c) Deduce the upper bound

$$\alpha_s < \frac{\ln 2}{|\ln(1 - 2^{-K})|}. \tag{4.49}$$

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.

5 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 admitedly special. But it has two advantages. First, it has an easy 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. Analogous, but more complicated features occur in coding, K-SAT and compressive sensing. A second easily solvable model is the Ising model on a *tree*. You will solve this in the homework. 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.

We introduced the standard Ising model on a cubic grid $\mathbb{Z}^d$ in Example 4 in Chapter 3. 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. In fact, it is still the subject of fascinating mathematical investigations. For completeness we review a few basic results in section (5.6), which can be skipped in a first reading. One important concept that is discussed is the one of *pure state* or *extremal measure*. Not only this notion allow for a deeper understanding of phase transitions, but it becomes very helpful in more advanced topics such as the cavity method.

Let us also point out that the Ising model on $\mathbb{Z}^d$ with $d \to +\infty$ and the Curie-Weiss model have the same free energies and phase diagrams.

5.1 Curie-Weiss model

Let $G = (V, E)$ be a complete graph on n vertices. A complete graph on n vertices is a graph in which each of the $n(n-1)/2$ pairs of nodes is connected by an edge. The case of $n = 4$ is shown in Figure 5.1. The Hamiltonian of the

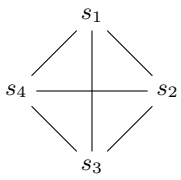


Figure 5.1 A complete graph with 4 nodes.

system is

$$\mathcal{H}_n(\underline{s}) = -\frac{J}{n} \sum_{\langle i,j \rangle \in E} s_i s_j - H \sum_{i \in V} s_i. \tag{5.1}$$

Recall that the $\langle i, j \rangle$ notation denotes an edge, i.e., this is an unordered pair. We will discuss only the case $J > 0$, i.e., the case in which equally "signed" spins *attract* each other. This is called the *ferromagnetic* case. Note that the interaction constant is scaled by n, i.e., we have the constant J/n in front of the first sum. This scaling is necessary in order to have an interesting limiting behavior: we would like both terms to scale in the same way as a function of n and in particular to scale linearly in the system size.

In the sequel we will calculate the free energy and the magnetization, and we will analyze the phase transitions for the model (5.1) in the thermodynamic limit, i.e., when $n \to +\infty$.

5.2 Computation of the free energy

Recall from Chapter 3 that in "real" physical systems the associated Gibbs measure has the form $\frac{e^{-\frac{\mathcal{H}(\underline{s})}{kT}}}{Z}$, where k is the Boltzman constant and T is the temperature. In other words, for the Gibbs measure what is important is the ratio of the energy of a configuration compared to a "background" energy kT which depends on the temperature. One is then interested in studying the behavior of this Gibbs measure as a function of T.

For our models the choice of this background energy is somewhat arbitrary and it might not seem relevant. Hence we might be tempted to simply set $kT = 1$. But, as we have seen already in previous chapters, it is often convenient to keep an extra parameter. This parameter "weighs" the importance of the various configurations with respect to each other. If we set kT to be very large then we get an almost uniform measure, whereas if we look at the case where kT tends to 0, then only configurations of minimum energy count. For notational convenience we typically write β instead of $1/(kT)$, and we call β the *inverse temperature*.

We are therefore led to study the Gibbs measure with an exponent given by

$$\frac{\beta J}{n} \sum_{\langle i,j\rangle \in E} s_i s_j + \beta H \sum_{i \in V} s_i. \tag{5.2}$$

To simplify the notation further, it is customary to set $\beta J = K$ and $\beta H = h$ so that the Gibbs measure has the form

$$\mu(\underline{s}) = \frac{e^{-\beta \mathcal{H}_n(\underline{s})}}{Z_n} = \frac{e^{\frac{K}{n} \sum_{\langle i,j\rangle \in E} s_i s_j + h \sum_{i \in V} s_i}}{Z_n} \tag{5.3}$$

The partition function can be written as

$$Z_n = \sum_{\underline{s} \in \{-1,1\}^n} e^{\frac{K}{n} \sum_{\langle i,j\rangle \in E} s_i s_j + h \sum_{i \in V} s_i}. \tag{5.4}$$

We recall the definition of the free energy $f(K,h)$ in the thermodynamic limit,

$$\beta f(K,h) = -\lim_{n \to +\infty} \frac{1}{n} \ln Z_n. \tag{5.5}$$

On a *complete graph* we have the identity,

$$\sum_{\langle i,j\rangle \in E} s_i s_j = \frac{1}{2} (\sum_{i \in V} s_i)^2 - \frac{1}{2} n. \tag{5.6}$$

Given a constellation $\underline{s}$, let us call the average of its spins the *magnetization*, denote it by $m_n(\underline{s})$,

$$m_n(\underline{s}) = \frac{1}{n} \sum_{i \in V} s_i. \tag{5.7}$$

It is then natural to express the Hamiltonian in terms of $m_n(\underline{s})$. We have

$$\beta \mathcal{H}_n(\underline{s}) = -n(\frac{K}{2} m_n(\underline{s})^2 + h m_n(\underline{s})) + \frac{K}{2}. \tag{5.8}$$

Thus

$$Z_n = e^{-\frac{K}{2}} \sum_{\underline{s} \in \{-1,1\}^n} e^{n(\frac{K}{2} m_n(\underline{s})^2 + h m_n(\underline{s}))}. \tag{5.9}$$

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

$$Z_n = e^{-\frac{K}{2}} \sum_{m_n} \#\{\underline{s} : \sum_{i=1}^n s_i = n m_n\} e^{n(\frac{K}{2} m_n^2 + h m_n)}. \tag{5.10}$$

The factor in front of the exponential is the number of spin configurations with the given magnetization m_n. Given m_n, let n_+ and n_- be the number of positive and negative spins respectively. We have

$$\begin{cases} n_+ + n_- = n, \\ n_+ - n_- = n m_n. \end{cases}$$

This implies that $n_+ = n\frac{1+m_n}{2}$. Therefore,

$$\#\{\underline{s} : \sum_{i=1}^{n} s_i = nm_n\} = \binom{n}{n\frac{1+m_n}{2}} \approx e^{nh_2(\frac{1+m_n}{2})}, \tag{5.11}$$

where the last step is valid for $n \to +\infty$ and is obtained using Stirling's formula. In the last step we have introduced the *binary entropy function* $h_2(p) = -p\ln p - (1-p)\ln(1-p)$. This leads to

$$Z_n \approx e^{-\frac{K}{2}} \sum_{m_n} e^{n(\frac{K}{2}m_n^2 + hm_n + h_2(\frac{1+m_n}{2}))}. \tag{5.12}$$

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

$$Z_n \approx e^{-\frac{K}{2}} n \int_{-1}^{+1} dm\, e^{n(\frac{K}{2}m^2 + hm + h_2(\frac{1+m}{2}))}. \tag{5.13}$$

Consider the integral. Its integrand has the form $e^{ng(m)}$, i.e., it is an exponential function. The value of such an integral can be tightly approximated by the so called Laplace method when n is large since in this case the value of the integral is dominated by the integral in a small neighborhood of that value of m where $g(m)$ takes on its maximum. Since for the free-energy computation we take the logarithm of the integral, 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

$$\begin{aligned} \beta f(K,h) &= -\max_{-1\le m\le 1}\left\{\frac{K}{2}m^2 + hm + h_2(\frac{1+m}{2})\right\} \\ &= \min_{-1\le m\le 1}\left\{-(\frac{K}{2}m^2 + hm) - h_2(\frac{1+m}{2})\right\} \\ &= \min_{-1\le m\le 1}\Big\{\beta f(m)\Big\}. \end{aligned} \tag{5.14}$$

Although our derivation has been rather casual, with a littel bit more effort, this formula can be converted into a theorem.

This formula 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. We call it the *free energy* function.

5.3 Phase diagram

Consider the *free energy function* $f(m)$,

$$\beta f(m) = -(\frac{K}{2}m^2 + hm) - h_2(\frac{1+m}{2}). \tag{5.15}$$

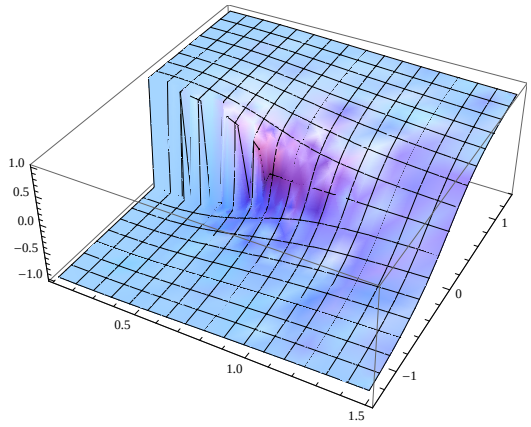


Figure 5.2 The behavior of $\overline{m}(K,h)$ as a function of $(1/K,h)$, where $1/K \in [0,1.5]$ and $h \in [-1.5,1.5]$.

For each pair of (K,h), let $\overline{m}(K,h)$ denote the value of m which minimizes $f(m)$. Our language reflects the fact that, as we will see, for most pairs (K,h), this minimizing value is unique. (In fact, it is unique for all $h \neq 0$.) What we want to study is $\overline{m}(K,h)$ as a function of K and h. Note that h can take on both positive and negative values, whereas K only takes on positive values. Instead of plotting $\overline{m}(K,h)$ as a function of K (on the x-axis) and h (on the y-axis), we will plot $\overline{m}(K,h)$ as a function of $1/K$ (on the x-axis) and h (on the y-axis).

Figure 5.2 shows the result. Why are we interested in this figure? As we will discuss in more detail shortly, the quantity $\overline{m}(K,h)$ 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 (K,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.

We will have much more to say about these phase transitions, but a quick look at the figure already reveals a few different forms of behavior. For parameters of the form $(0 \leq 1/K < 1, h = 0)$, when we move along the h-axis, the quantity $\overline{m}(K,h)$ jumps. For the point $(1/K = 1, h = 0)$, when we move along the h-axis $\overline{m}(K,h)$ changes in a continuous fashion, but its derivative (wrt to h) jumps. Finally, for all other points, $\overline{m}(K,h)$ changes smoothly (it 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 a phase transition of *second order*. The terms "first" and "second" here refers

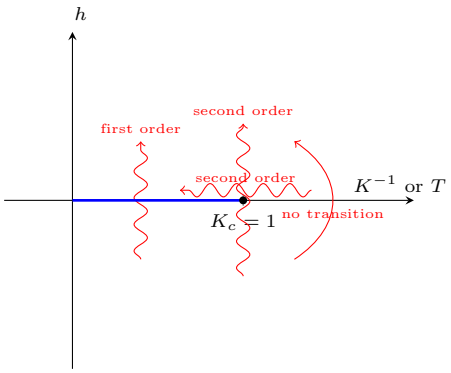


Figure 5.3 The blue line is called *coexistence line* because two thermodynamic phases (e.g. water/ice) coexist for parameters on it. Crossing the thick line is a first order phase transition. This line is terminated by the *critical point*. Crossing the critical point is a second order phase transition. There are many ways to cross it.

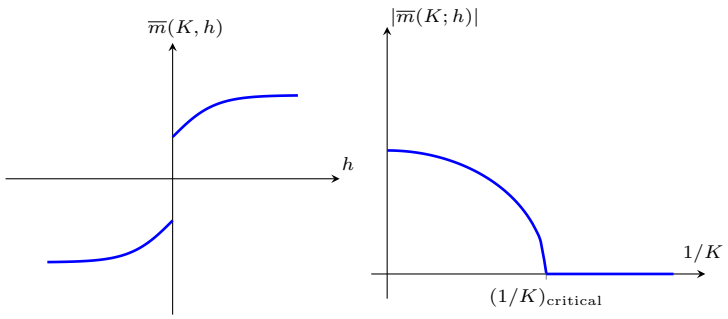
to which derivative "jumps." Perhaps sligthly confusingly, but for good reason, we do not refer here to the derivates of $\overline{m}(K,h)$ with respect to h, but to the derivates of the integral of $\overline{m}(K,h)$ (wrt to h)! Never boring, these physicists! Indeed we will see that the integral of $\overline{m}(K,h)$ is $-\beta$ times the free energy, our most primitive quantity, see (**??**).

For a sligthly different perspective, let us replot Figure 5.2 but this time let us consider the picture "from the top," i.e., we only show the K and h axis. This is shown in Figure 5.3. The different cases (movements) leading to the various phase transitions are indicated. The line indicated in blue, given by $(0 \leq 1/K < 1, h = 0)$ is typically called the co-existence line. This name is easily explained. If we approach this line from the top, i.e., we consider the limit $h \to 0_+$, then we get one value for $\overline{m}$, but if we consider the limit $h \to 0_-$, then the sign of $\overline{m}$ is flipped. So "on" the line we can think of having two possible "co-existing" phases.

Going down one further dimension by fixing a value of $1/K$ and only varying h, Figure 5.4 explicitly shows a phase transition of first and second order.

Let us point out a few more important features that are apparent from the variational expression (**??**) of the free energy $f(K,h)$. This is a continuous and concave function of K and h. 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.

The continuity and concavity of $f(K,h)$ is a general requirement in thermodynamics, and a general property of statistical mechanical models. Phase tran-



(a) First order (b) Second order

Figure 5.4 A phase transition of first and second order.

sitions[1] are defined as singularities of the free energy. If the n-th derivative is discontinuous one speaks of a phase transition of order n. There exist phase transitions of infinite order where the free energy is non-analytic but all its derivatives are continuous. This classification of phase transitions is due to Ehrenfest but is slighlty arbitrary for various reasons not discussed here. The more modern view point, which we do not discuss in this course, is to distinguish between continuous and discontinuous transitions and to classify them according to the type of symmetry change.

In this course we will essentially encounter discontinuous phase transitions. However we the reader should be aware that the study of continuous phase transitions is one of the most important chapters of statistical mechanics developped in the 70's.

5.4 Average magnetization

Before we discuss these phase transitions in more detail, let us first convince ourselves that $\overline{m}(K,h)$ in fact represents the average magnetization.

We claim that

$$\overline{m}(K,h) = \lim_{n\to+\infty} \langle m_n(\underline{s})\rangle = \lim_{n\to+\infty} \langle \frac{1}{n}\sum_{i\in V} s_i\rangle = \lim_{n\to+\infty} \langle s_i\rangle. \tag{5.16}$$

The second equality is just (5.7). And the third equality is an immediate consequence of the linearity of the Gibbs average and the symmetry of the model. But the first equality has a non-trivial content. It says that $\overline{m}(K,h)$ is the *average magnetization* in the thermodynamic limit. One usually just calls it "the magnetization."

[1] Here we discuss only *static* or *thermodynamic* phase transitions. We will see in later chapter that there is a notion of *dynamical* phase transition related to changes in behavior of the dynamics or of the algorithms. Such transitions are not related to singularities of thermodynamic quantities like the free energy.

One way to show (5.16) is to compute directly the Gibbs average

$$\langle s_i \rangle = \frac{1}{Z_n} \sum_{\underline{s} \in \{-1,+1\}} s_i \, e^{-\beta \mathcal{H}_n(\underline{s})}. \tag{5.17}$$

We have already computed the denominator. For the numerator one notes that s_i can be replaced by $\frac{1}{n}\sum_{i \in V} s_i$. Then one proceeds as before, a calculation that we leave to the reader. The essential point is now this. Recall the point in the calculation where we have converted the sum into an integral and where we now evaluate the integral. Recall that this integral is dominated by that value of m which maximizes the exponent. But this is exaclty the value $\overline{m}(K,h)$ and so for large values of n this average is determined by spin constellations of "type" $\overline{m}(K,h)$, i.e., by spin configurations which have magnetization $\overline{m}(K,h)$. It is therefore not surprising that the average magnetization is exactly $\overline{m}(K,h)$ in the thermoydynamic limit.[2]

There is a very useful and perhaps at first surprising relationship between the free energy $f(K,h)$ and the average magnetization $\overline{m}(K,h)$. As we have mentioned previously, Gibbs averages can be obtained by differentiating the free energy, i.e., we have

$$\frac{1}{n}\sum_{i=1}^{n} \langle s_i \rangle = \frac{\partial}{\partial h} \frac{1}{n} \ln Z_n. \tag{5.19}$$

Taking the limit $n \to +\infty$ one finds the important relation[3]

$$\overline{m}(K,h) = -\frac{\partial}{\partial h} \beta f(K,h). \tag{5.20}$$

The suspicious reader will notice that we have interchanged the limit $n \to +\infty$ and the h derivative. We do not prove it here, but this is permitted except at phase transition points (a set of measure zero)!

[2] As we have already seen, for $h = 0$ and $0 \leq 1/K \leq 1$, the situation is special. The free energy function $f(m)$ is even and may have two opposite global minimizers (this indeed happens for $K > 1$, see plot (5.6)). So if h is set to zero *before* taking the limit $n \to +\infty$ one finds that the contributions of the two minima cancel, which yields a zero magnetization. This is not the physically correct way to proceed. In reality there is always an infinitesimal magnetic field $h = \pm 0$ present in the environment. For $h = 0_\pm$ one defines the *spontaneous magnetization* as

$$\lim_{h \to 0_\pm} \lim_{n \to +\infty} \langle s_i \rangle = m_\pm(K) \tag{5.18}$$

For $K < 1$ the limit is unique and the spontaneous magnetization vanishes: nothing interesting happens. But for $K > 1$ the two limits differ: there is a phase transition. One says that for $h = 0_\pm$ there is a *spontaneous symmetry breaking*. This symmetry breaking is called "spontaneous" because physically we do not get to choose the orientation of the magnetization: the infinitesimal perturbations in the environment select an orientation.

[3] This relation was known in thermodynamics well before the invention of statistical mechanics.

5.5 Computing the phase diagram – the fixed point equation

We have already seen the three-dimensional picture of $\overline{m}(K,h)$ 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.

Let us therefore sole the variational problem by differentiating the free energy function

$$f(m) \equiv -(\frac{K}{2}m^2 + hm) - h_2(\frac{1+m}{2}). \tag{5.21}$$

Explicitly,

$$Km + h + \frac{1}{2}\ln\frac{1-m}{1+m} = 0. \tag{5.22}$$

Using the identity

$$\tanh(\frac{1}{2}\ln\frac{1+m}{1-m}) = m, \tag{5.23}$$

we obtain the *Curie-Weiss equation*

$$m = \tanh(Km + h). \tag{5.24}$$

Of course this equation may have many solutions. One has to select the one 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 $\beta = +\infty$ or $T = 0$.

Equ. (5.24) is often called the mean field equation. This is because it arises in the *approximate* mean field theory of the Ising model on $\mathbb{Z}^d$. For $d \to +\infty$ the mean field approximation becomes exact.

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 (5.24) 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 are shown in Figure 5.5 and the corresponding free energy function $f(m)$ are shown in Figure 5.6.

$K < 1$: unique solution, $\overline{m}(K,0) = 0$, $\quad \beta f(K,0) = \ln 2$. This is called the high temperature phase (because $K = \beta J < 1$ corresponds to high T). In this phase the magnetization vanishes.

$K > 1$: three solutions $\{\overline{m}_-, 0, \overline{m}_+\}$, the two opposite extremes are global minimizers of $f(m)$. This is the low temperature phase where there is a "spontaneous" magnetization. Here the word spontaneous refers to the fact that the magnetization does not vanish although $h = 0$. A real system will choose from $\overline{m}_-$ or

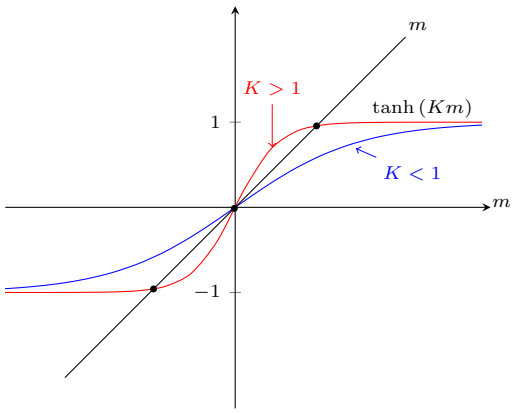
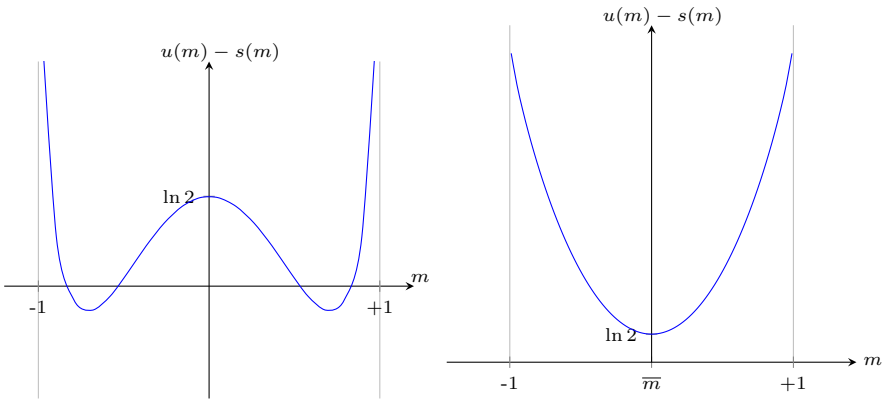


Figure 5.5 Curie-Weiss fixed points, $h = 0$



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

Figure 5.6 Free energy functional

$\overline{m}_+$ because there is always an infinitesimal $h = 0_\pm$ in the environnement. This is called "spontaneous symmetry breaking".

Phase transition as a function of K: There is a continuous phase transition at $K_c = 1$. The behaviour of the magnetization for $h = 0$ as a function of $K^{-1} \sim k_B T$ is shown in Figure 5.7. There is a *continuous phase transition* at $K_c = 1$. The point $(K_c = 1, h = 0)$ is called the critical point. Continuous phase transitions are also called *second order phase transition* because the second derivative of the free energy (i.e first derivative of the magnetization jumps). It is interesting to study the behavior of the magnetization close to the critical point. for K close

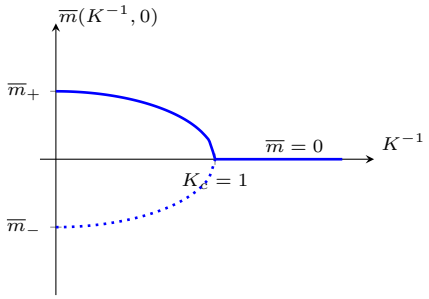
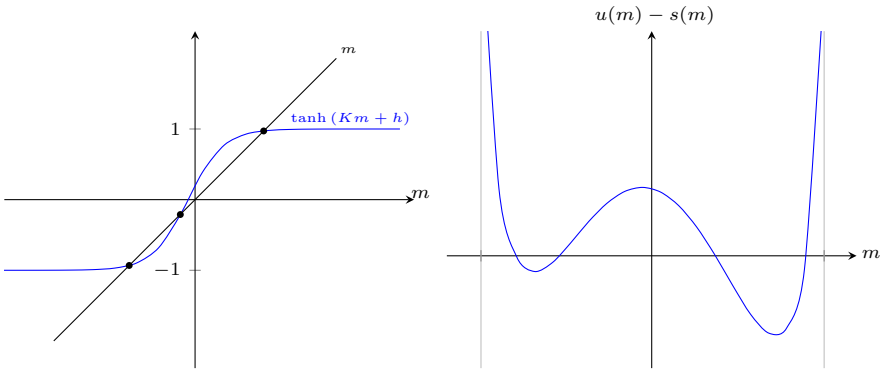


Figure 5.7 Phase transition as a function of $K^{-1} \sim k_B T$ in Curie-Weiss model



(a) Fixed points (b) Free energy

Figure 5.8 Curie-Weiss fixed points, $h > 0, K > 1$

to $K_c = 1$ we have $\overline{m}$ small, so we can expand the Curie-Weiss equation

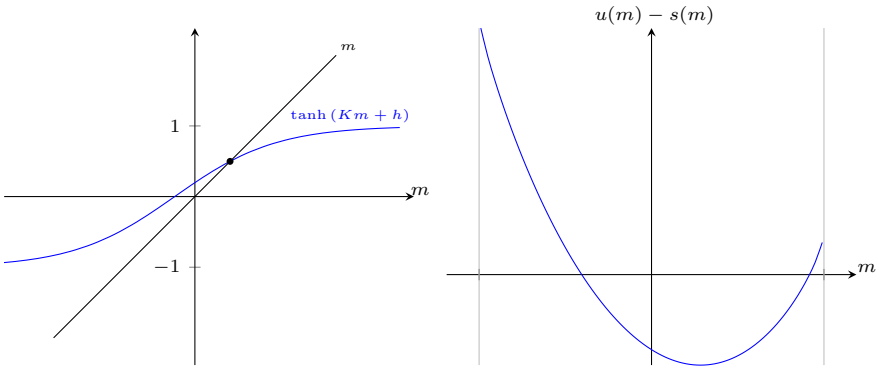
$$m = \tanh Km \approx Km - \frac{K^3}{3} m^3$$

Besides the trivial solution $\overline{m} = 0$ this leads to

$$m \sim \pm 3(K - K_c)^{\frac{1}{2}}$$

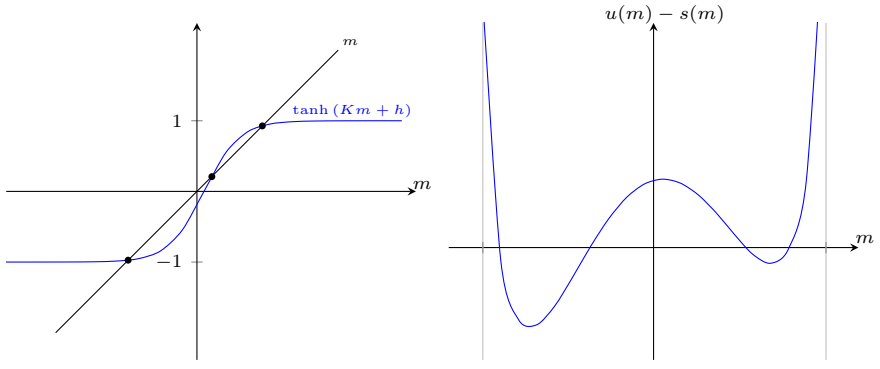
The exponent $\frac{1}{2}$ is called *critical exponent*. Remarkably it often does not depend on the detailed form of the Hamiltonian but only on such things as the dimensionality of the system (here $d = +\infty$), and the underlying symmetries of the Hamiltonian (here the Hamiltonian is invariant under $s_i \to -s_i$ for $h = 0$).

Case $h > 0$ Fixed points and free energy function $f(m)$ are shown in Figure 5.8 where $h > 0$ (h not too large) and $K > 1$. Fixed points and free energy are shown in Figure 5.9 where $h > 0$ (h large) and $K < 1$. Note that the



(a) Fixed points (b) Free energy

Figure 5.9 Curie-Weiss fixed points, $h > 0, K < 1$



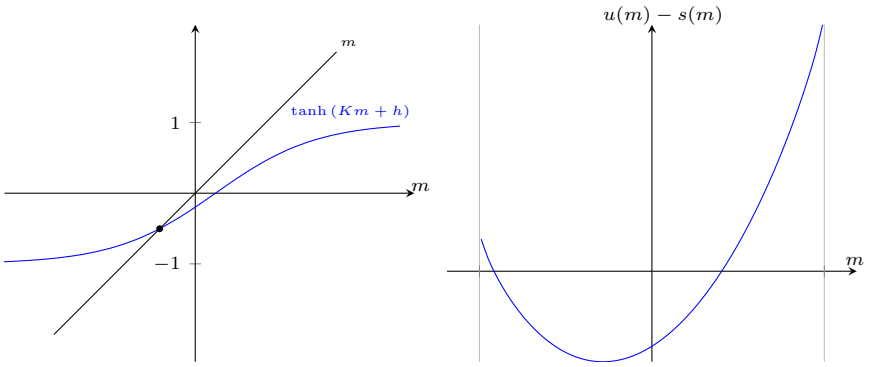
(a) Fixed points (b) Free energy

Figure 5.10 Curie-Weiss fixed points, $h < 0, K > 1$

global minimizer $\overline{m} > 0$.

Case $h < 0$ Fixed points and free energy are shown in Figure 5.10 where $h < 0$ (h not too large) and $K > 1$. Fixed points and free energy function $f(m)$ are shown in Figure 5.11 where $h < 0$ (h large) and $K < 1$. Note that the global minimizer $\overline{m} < 0$.

Phase Transition as a function of h: Summarizing, we see that for $K > 1$, $\overline{m}(K, h)$ 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 (5.12). For $K < 1$, $\overline{m}(K, h)$ is continuous and there is *no* phase transition. At the critical



(a) Fixed points (b) Free energy

Figure 5.11 Curie-Weiss fixed points, $h < 0, K < 1$

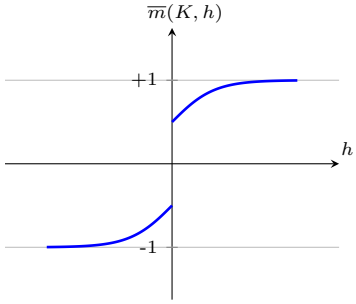


Figure 5.12 Phase transition in Curie-Weiss model when $K > 1$ as a function of h. The phase transition is at $h = 0$

point $(K_c = 1, h = 0)$ the jump disappears and

$$m(K_c = 1, h) \sim |h|^{\frac{1}{3}}, \qquad h \to 0 \tag{5.25}$$

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

5.6 Brief review of the Ising model on $\mathbb{Z}^d$

We briefly review the Ising model in finite dimensions. We point out that this model is still the subject of deep mathematical investigations. This section can be skipped in a first reading.

Existence of the thermodynamic limit

We are interested in analyzing the system in the large size limit for a sequence of graphs. This means that we have to specify a sequence of graphs: physically this can be thought as specifying the "shape" of the sample. The large size limit might depend on the shape. We do not wish to enter into a detailed discussion of this topic here. The simplest case corresponds to taking an Ising model on a cubic grid of equal sides and take free boundary conditions.

THEOREM 5.6.1 *For a sequence of cubic grids $V \subset \mathbb{Z}^d$ with equal side lengths the thermodynamic limit of the free energy exists.*

$$-\lim_{|V|\to+\infty} \frac{1}{|V|} \ln Z_V = \beta f(K,h) \tag{5.26}$$

Moreover it is continuous and concave for all K and h.

Ehrenfest classification of phase transition

Phase transitions can be classified according to derivatives of $f(K,h)$ (this is just one possible classification called the Ehrenfest classfication. It is the most usual one, but there exist other more "modern" ones).

- **First order:** The first derivative of $f(K,h)$ is discontinuous. Recall

$$m(K,h) \equiv \lim_{|V|\to+\infty} \frac{1}{|V|} \sum_{i\in V} \langle s_i \rangle_V = -\frac{\partial}{\partial h} f(K,h)$$

  so the total magnetization per spin, is discontinuous. Figure 5.4a shows a phase transition of first order.
- **Second order:** The second derivative of $f(K,h)$ is discontinuous. Then $m(K,h)$ is continuous but its first derivative is discontinuous. This typically happens when the temperature is varied. Figure 5.4b shows a phase transition of second order.

Evidently one can define higher order transitions within this classification scheme. Even infinite order phase transitions exist where the free energy is infinitely differentiable but not analytic. This hardly manifests itself on the function, but affects correlation functions. More modern classification schemes depend on the symmetry changes that occur at a transition.

Dimensionality dependence

- $d = 1$: No phase transitions (except for interactions with very large range) (Ising 1920).
- $d \geq 2$: First and second order phase transition are present. Qualitatively these are much like those of Curie-Weiss model (Proofs of existence of transition by Peierls 1935, Griffith, Dobrushin 1965-70). Note however that the critical

exponents of second order phase transitions are not the same than in Curie-Weiss.

- $d \to +\infty$: same solution than on the complete graph.

Critical behavior

- $d \geq 4$: exponents of second order transition $\frac{1}{2}$ and $\frac{1}{3}$, same as the ones found for the Curie-Weiss model. Remarkably they do not depend on the microscopic structure of $\mathcal{H}(\underline{s}.)$
- $d = 2, 3$ other critical exponents for the second order transition. For example, for $d = 2$, $m \sim |K - K_C|^{\frac{1}{8}}$. This results from Onsager's famous exact solution of the two dimensional model (1944). For $d = 3$, computing those is the subject of the *renormalization group* which was developed in the 70's (Wilson, Fisher, Kadanoff. Nobel prize to K. Wilson). Their exact values are unknown however and one has to use $d = 4-\epsilon$ expansions and numerical calculations.

Spontaneous magnetization on $\mathbb{Z}^d$, $d \geq 2$.

If one sets $h = 0$ from the outset, one has $\langle s_0 \rangle_V = 0$ (free boundary conditions) and this is also true for the limit $|V| \to +\infty$. However for $K > 1$ the limits $h \to 0$ and $|V| \to +\infty$ do not commute. One defines the spontaneous magnetization as

$$\begin{aligned} \overline{m}_{\pm} &= \lim_{h \to 0_{\pm}} \lim_{|V| \to +\infty} \frac{1}{|V|} \sum_{i \in V} \langle s_0 \rangle \\ &= \lim_{h \to 0_{\pm}} \lim_{|V| \to +\infty} \langle s_0 \rangle_V \end{aligned}$$

On the coexistence line (see phase diagram) the two limits are different. This means that for $K > 1$ an infinitesimally positive magnetic field tilts typical spin configurations to mostly $+1$'s and an infinitesimally negative magnetic field tilts typical spin configurations to mostly -1's. In nature a magnet (say) picks up one of the two limits because of infinitesimal magnetic fields that are always present. This phenomenon is called *spontaneous symmetry breaking*.

Infinite volume Gibbs measures

One can study higher marginals/higher moments of the Gibbs measure in the infinite size limit. For example

$$\lim_{|V| \to +\infty} \langle s_i \rangle_V = \langle s_i \rangle, \qquad \lim_{|V| \to +\infty} \langle s_i s_j \rangle_V = \langle s_i s_j \rangle, \qquad \text{ect...}$$

The set of all limiting marginals defines the infinite volume Gibbs measure. When various phases coexist (on coexistence lines of the phase diagram) the limits of these marginals are different for $h \to 0_{\pm}$. Other formulations of the

same phenomenon use boundary conditions, and then the thermodynamic limits $|V| \to +\infty$ depend on boundary conditions.

From the set of limiting marginals one can reconstruct the "infinite volume Gibbs measure". Hence on the coexistence line the limiting Gibbs measure is also non-unique. Caracterizing the set of infinite volume Gibbs measures is a non-trivial problem. This is a convex set. Extremal points are called *extremal measures* or *pure states*. These describe pure thermodynamic phases (pure water/pure vapour for example). Convex combination of extremal measures describe the coexistence of pure thermodynamic phases (the coexistence of water and vapour for example).

- In $d = 1$ only one infinite volume Gibbs measure for any finite temperature: no phase transitions. The convex set is a point.
- In $d = 2$ only two extremal measures for $K > K_c$ and $h = 0$ (proved in the 80's). The convex set is a segment. For other points of the phase diagram there is only one measure (convex set is a point). At the critical point $K = K_c$ and $h = 0$ the problem is different: seen from large scales the typical configurations look fractal and self-similar. This is the subject of conformal invariance developed by physicist in the 70's-80's. Some of the predictions of conformal invariance have been recently proved by mathematicians (Fields medals in 2006 to Werner, Okhounov and 2010 to Smirnov).
- In $d = 3$ on the coexistence line it is known that there exist more than two extremal states. The convex set is richer than in two dimensions. In particular there are extremal Gibbs measures that describe states with interfaces. Interfaces are stable in three dimensions (and not in $d = 2$).

Problems

5.1 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

$$\beta \mathcal{H}_n = -K \sum_{(i,j)\in E_n} s_i s_j - h \sum_{i \in V_n} s_i \tag{5.27}$$

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

$$m(K,h) = \lim_{n\to+\infty} < s_o >_n = \frac{\sum_{\{s_k, k\in V_n\}} s_o \, e^{-\beta \mathcal{H}_n}}{Z_n} \tag{5.28}$$

The formula $\tanh^{-1} y = \frac{1}{2} \ln \frac{1+y}{1-y}$ might be useful.

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

$$< s_o >_n = \frac{\sum_{\{s_k, k\in V_{n-1}\}} s_o \, e^{-\beta \mathcal{H}'_{n-1}}}{Z'_{n-1}} \tag{5.29}$$

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

$$\beta \mathcal{H}'_n = -K \sum_{(i,j)\in E_{n-1}} s_i s_j - h \sum_{i\in V_{n-1}} s_i - (q-1) \tanh^{-1}(\tanh K \tanh h) \sum_{i \in \text{level } n-1} s_i \tag{5.30}$$

Iterate this calculation and deduce

$$< s_o >_n = \tanh(h + q \tanh^{-1}(\tanh K \tanh u_n)) \tag{5.31}$$

where

$$u_{k+1} = h + (q-1) \tanh^{-1}(\tanh K \tanh u_k), \qquad u_1 = h \tag{5.32}$$

Check that for $q = 2$ you get back the recursion of homework 2.

5.3 *Analysis of the recursion.* We want to analyze the fixed point equation for $q \geq 3$,

$$u = h + (q-1) \tanh^{-1}(\tanh K \tanh u) \tag{5.33}$$

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

- for $K \leq K_c \equiv \frac{1}{2} \ln \frac{q}{q-2} = \tanh^{-1}(q-1)^{-1}$, (5.33) has a unique solution, and that the iterations (5.32) converge to this unique solution.
- for $K > K_c$:
  - for $|h| \geq h_s$, (5.33) has a unique solution (you do not needw3 to compute h_s explicitly although it is possible to find its analytical expression) and that the iterations (5.32) converge to this unique solution.
  - for $|h| < h_s$, (5.33) has three solutions $u_-(h) < u_0(h) < u_+(h)$. Check graphically that for $h > 0$ the iterations (5.32) 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.

5.4 *Phase transitions.* Now we want to discuss the consequences of the results in problem 2 for the phase diagram. On a tree the magnetization is defined as the average spin of the root

$$m(K,h) = \lim_{n\to+\infty} \langle s_o \rangle_n, \tag{5.34}$$

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

- Deduce from the analysis in problem 2 that for $K \le K_c$, $m_+(K) = m_-(K) = 0$.
- Deduce that for $K > K_c$, $m_+(K) \neq m_-(K)$ (jump discontinuity or first order phase transition) and that for $K \to +\infty$ $m_\pm \to \pm 1$.
- Show that for $K \to K_c$ from above, $m_\pm(K) \sim (K-K_c)^{1/2}$. So on the line $h=0$, as a function of K, the spontaneous magnetization is continuous but not differentiable at K_c (second order phase transition).
- Now fix $K = K_c$ and show that $m(K_c,h) \sim |h|^{1/3}$. As a function of h the spontaneous magnetization is continuous but not differentiable at K_c (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 $d \ge 4$ 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.

6 *Summary of Part I*

Explain that all mdoels coding, comprssed sensing and random k sat are of the general form

0.25cm model whose distribution can be factorized as

$$\mu(\underline{x}) = \frac{1}{Z}\prod_a f_a(x_{\partial a}), \qquad Z = \sum_{\underline{x}\in\mathcal{X}^n}\prod_{a=1}^m f_a(x_{\partial a}) \tag{6.1}$$

where $\partial a = \{i | i \in a\}$. For continuous alphabets all sums are replaced by integrals, but otherwise the formalism is the same. These is the structure of fundamental Gibbs distributiosn that describe physical systems. hence we expect convergence of concepts and methods.

Foremost phenomenon is phase transitions. Phase digram will teach us what we can hope to do and not to do whith local algorithmic dynamics ...

We solved CW: important paradigm, easy to solve, and will be useful to know when we discuss compressive sensing. In homework discussed Ising on Tree: easy to solve, important pardigm and will be useful for coding on sparse graphs.

Part II

Analysis of Message Passing

7 Marginalization, Factor Graphs, and Belief Propagation

This chapter is largely about the following question: how can we efficiently compute marginals of multivariate functions. We will see that this question has a natural answer in form of a message-passing algorithm. Perhaps not too surprising, the message-passing paradigm is the basis for the low-complexity algorithms which we will apply to our three running examples.

Much more surprising is the fact that message-passing is also the key for the analysis of the so-called *static threshold* of the three examples (e.g., the MAP threshold for coding or the SAT-UNSAT threshold for K-SAT). A priori there is absolutely no connection between these thresholds and low-complexity algorithms. As we will see, the element which connects these two worlds is the Bethe free energy and the so-called Maxwell construction. We will discuss this connection towards the end of our lectures. It has a fascinating history and a beautiful graphical interpretation.

7.1 Distributive Law

Let $\mathbb{F}$ be a field (think of $\mathbb{F} = \mathbb{R}$) and let $a, b, c \in \mathbb{F}$. By the *distributive law*

$$ab + ac = a(b + c). \tag{7.1}$$

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.

EXAMPLE 6 (Simple Example) Let's start with an example. Consider a function f with factorization

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5). \tag{7.2}$$

We are interested in computing the *marginal* of f with respect to x_1. With some abuse of notation, we denote this marginal by $f(x_1)$:

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

In the previous line we introduced the notation $\sum_{\sim \dots}$ to denote a summation over

all variables contained in the expression *except* the ones listed. This convention will save us from a flood of notation. Assume that all variables take values in a finite alphabet, call it $\mathcal{X}$. Determining $f(x_1)$ 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 $f(x_1)$ as

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

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. ◊

7.2 Graphical Representation of Factorizations

Consider a function and its factorization. Associate with this factorization a *factor graph* as follows. For each variable draw a *variable node* (circle) and for each factor 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. The resulting graph for the function of Example 6 is shown on the left of Figure 7.1. 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 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.

EXAMPLE 7 (Special Case: Tanner Graph) Consider the binary linear code

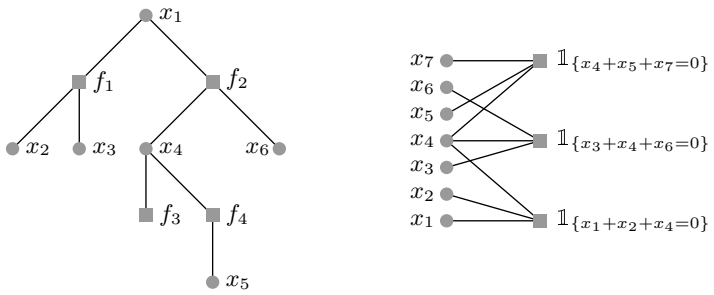


Figure 7.1 Left: Factor graph of f given in Example 6. Right: Factor graph for the code membership function defined in Example 7.

$C(H)$ whose parity-check matrix is

$$H = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \end{array}.$$

Let $\mathbb{F}_2$ denote the binary field with elements $\{0, 1\}$ and let $\underline{x} = (x_1, \ldots, x_7)^T$. Consider the function $f(x_1, \ldots, x_7)$ from $\mathbb{F}_2^7$ to $\{0, 1\} \subset \mathbb{R}$ that is defined by

$$f(x_1, \ldots, x_7) \triangleq \mathbb{1}_{\{\underline{x} \in C(H)\}} \triangleq \begin{cases} 1, & \text{if } H\underline{x} = 0^T, \\ 0, & \text{otherwise.} \end{cases}$$

We can factor f as

$$f(x_1, \ldots, x_7) = \mathbb{1}_{\{x_1+x_2+x_4=0\}} \mathbb{1}_{\{x_3+x_4+x_6=0\}} \mathbb{1}_{\{x_4+x_5+x_7=0\}}.$$

Each term $\mathbb{1}_{\{.\}}$ is an *indicator function*: it is 1 if the condition inside the braces is fulfilled and 0 otherwise. The function f is sometimes also called the *code membership* function since it tests whether a particular word is a member of the code or not. The factor graph of f is shown on the right in Figure 7.1. It is called the *Tanner* graph of H. ◊

It is hopefully clear at this point that *any* (binary) linear block code has a Tanner graph representation.

7.3 Recursive Determination of Marginals

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

g has a generic factorization of the form

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

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

For the function f of Example 6 this factorization is

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

so that $K = 2$. The generic factorization and the particular instance for our running example f are shown in Figure 7.2. Taking into account that the individual

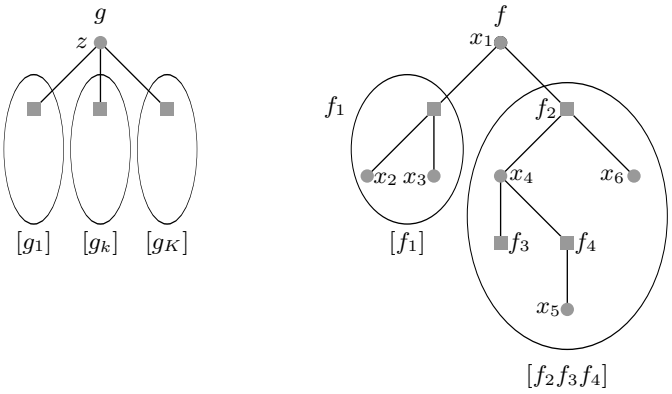


Figure 7.2 Generic factorization and the particular instance.

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

$$\sum_{\sim z} g(z,\dots) = \underbrace{\sum_{\sim z}\prod_{k=1}^{K}[g_k(z,\dots)]}_{\text{marginal of product}} = \underbrace{\prod_{k=1}^{K}\Big[\sum_{\sim z} g_k(z,\dots)\Big]}_{\text{product of marginals}}. \tag{7.3}$$

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

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

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

In general, each g_k is itself a product of factors. In Figure 7.2 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 7.3. Another application of the distributive law gives

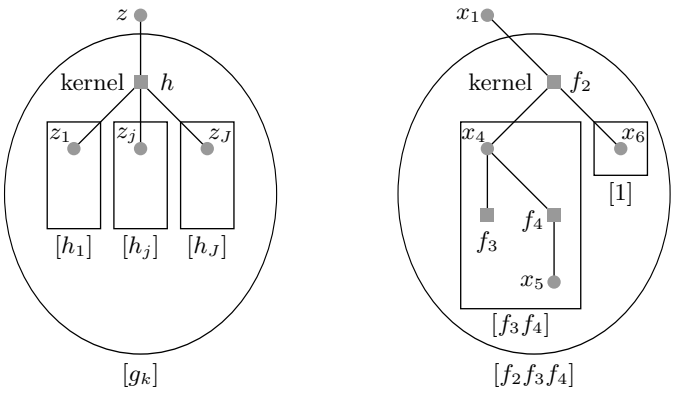


Figure 7.3 Generic factorization of g_k and the particular instance.

$$\begin{aligned}\sum_{\sim z} g_k(z,\dots) &= \sum_{\sim z} h(z,z_1,\dots,z_J) \prod_{j=1}^{J} [h_j(z_j,\dots)] \\ &= \sum_{\sim z} h(z,z_1,\dots,z_J) \underbrace{\prod_{j=1}^{J} \Big[\sum_{\sim z_j} h_j(z_j,\dots)\Big]}_{\text{product of marginals}}. \end{aligned} \tag{7.4}$$

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 (7.4). 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 $f(x_1) = \sum_{\sim x_1} f(x_1, x_2)$. Here, x_2 is the variable leaf node. By the message-passing rule (7.4) the marginal $f(x_1)$ 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.

7.4 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 8 (Message-Passing Algorithm for f of Example 6) Consider this procedure in detail for the case of our running example as shown in Figure 7.4. 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 (7.4), 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 (7.3), which says that the marginal of a product is the product of the marginals. This message, which is a function of x_4, is $f_3(x_4) \sum_{\sim x_4} f(x_4, x_5) = \sum_{\sim x_4} f_3(x_4) f_4(x_4, x_5)$. Next, function node f_2 can forward its message, and, finally, the marginalization is achieved by multiplying all incoming messages at the root node x_1. ◊

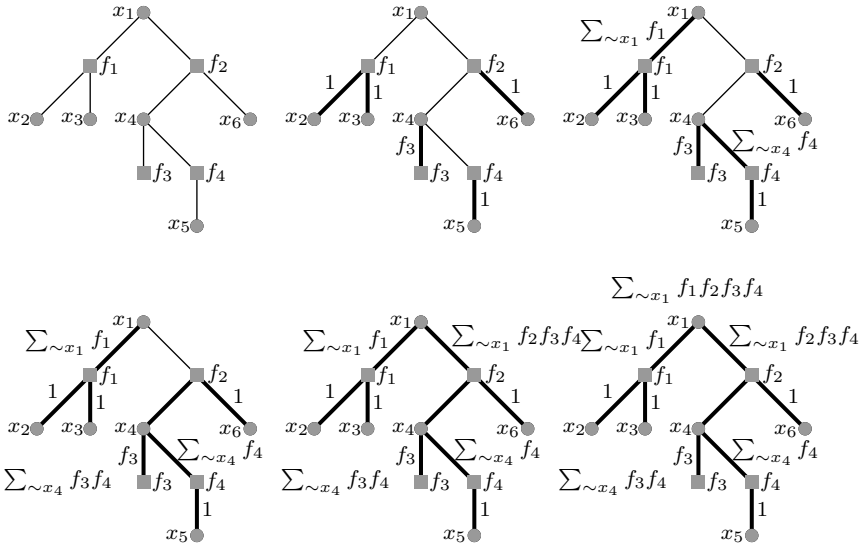


Figure 7.4 Marginalization of function f from Example 6 via message passing. Message passing starts at the leaf nodes. A node that has received messages from all its children processes the messages and forwards the result to its parent node. Bold edges indicate edges along which messages have already been sent.

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.

Messages, which we denote by μ and $\hat{\mu}$, are functions on $\mathcal{X}$. 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 will be denoted by ν. 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

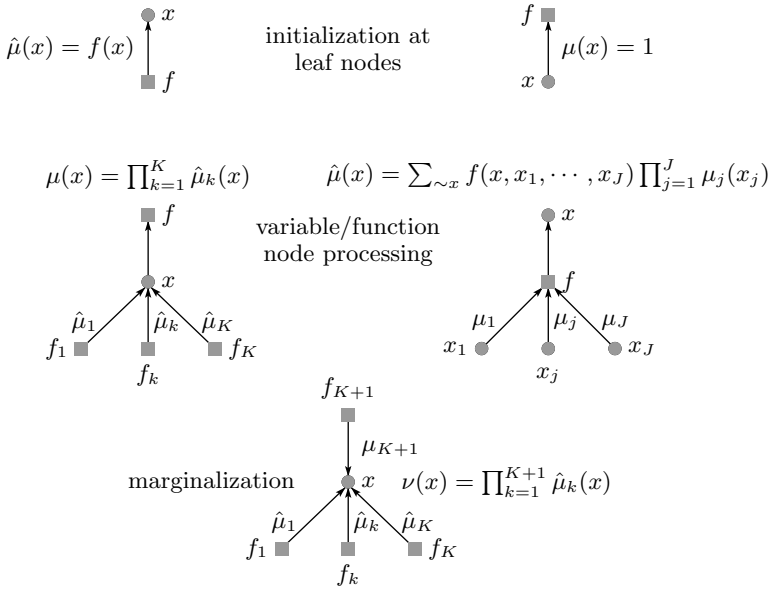


Figure 7.5 Message-passing rules. The top row shows the initialization of the messages at the leaf nodes. The middle row corresponds to the processing rules at the variable and function nodes, respectively. The bottom row explains the final marginalization step.

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 7.5. Since the messages represent 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.

7.5 Coding: Decoding via Message Passing

Assume we transmit over a binary-input ($s_i \in \{\pm 1\}$) memoryless ($p(\underline{y} \mid \underline{s}) = \prod_{i=1}^{n} p(y_i \mid x_i)$) channel using a linear code defined by its parity-check matrix H and assume that codewords are chosen uniformly at random. Recall that the rule (4.8) for the *bit-wise* maximum a posteriori (MAP) decoder reads:

$$\hat{s}_i(y) = \mathrm{argmax}_{s_i \in \{\pm 1\}} p(s_i \mid y) = \mathrm{sign}\langle s_i \rangle. \tag{7.5}$$

Here $p(s_i \mid \underline{y})$ is the marginal of the posterior (4.6),

$$p(\underline{s} \mid \underline{y}) = \frac{1}{Z(\underline{h})} \prod_{c \in \mathcal{C}} \frac{1}{2}(1 + \prod_{i \in c} s_i) \prod_{i=1}^{n} e^{h_i s_i}. \tag{7.6}$$

The partition function $Z(\underline{y})$ is a "constant" with respect to the sums over $\sim s_i$ involved in the marginalization. Thus, to obtain $p(s_i \mid \underline{y})$, it is sufficient to marginalize the numerator in (7.6) and eventually normalize the resulting function of s_i. This numerator has a factorized form with two types of "kernel" functions,

$$f_i(s_i) = e^{h_i s_i}, \quad \text{and} \quad f_c(\{s_i, i \in c\}) = \frac{1}{2}(1 + \prod_{i \in c} s_i). \tag{7.7}$$

The first kernel function 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 9 (Bit-wise MAP Decoding) Consider the parity-check matrix given in Example 7. The corresponding factor graph is shown in Figure 7.6. This graph

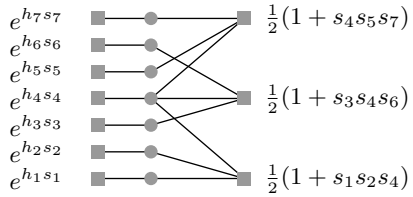


Figure 7.6 Factor graph for the MAP decoding of our running example.

includes the Tanner graph of H but additionally contains the 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 (taking into account the effects of fading, synchronization, mapping of bits to elements of a constellation, or quantization) 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 10 (Message passing algorithm for decoding) We give the first three steps of belief propagation for the tree in Figure 7.6. 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 five 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

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

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

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

As you can see, the sums involved in the messages each involving 2 binary variables, and so each has 4 terms. They are easy to compute. For example the first one is equal to

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

In a real setting, the variables h_i are known numbers and the sums are of course computed numerically and not symbolically.

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 $\nu(s_4)$ by multiplying these incoming messages. Explicitely,

$$\begin{aligned}\nu(s_4) = & e^{h_4 s_4}\big\{(1 + s_4)\cosh(h_1 + h_2) + (1 - s_4)\cosh(h_1 - h_2)\big\} \\ & \times \big\{(1 + s_4)\cosh(h_3 + h_6) + (1 - s_4)\cosh(h_3 - h_6)\big\} \\ & \times \big\{(1 + s_4)\cosh(h_5 + h_7) + (1 - s_4)\cosh(h_5 - h_7)\big\}.\end{aligned}$$

Recall that we did not normalize our messages. Therefore, to get the true marginal $p(s_4 \mid \underline{y})$ one has to normalize $\mu(s_4)$,

$$p(s_4|\underline{y}) = \nu(s_4)/(\nu(+1) + \nu(-1)).$$

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

7.6 Compressive Sensing: Finding a Sparse Vector via Message Passing

Recall the setting in Section 4.4. We want to marginalize the posterior distribution (4.35)

$$p(\underline{x} \mid \underline{y}) = \frac{1}{Z(\underline{y})} \prod_{a=1}^{r} e^{-\frac{1}{2\sigma^2}(y_a - A_a^T \underline{x})^2} \prod_{i=1}^{n} p(x_i), \tag{7.8}$$

in order to get the MMSE estimate

$$\hat{x}_{i,\sigma}(\underline{y}) = \langle x_i \rangle = \int d\underline{x}\, x_i\, p(\underline{x} \mid \underline{y}) = \int dx_i x_i\, p(x_i \mid \underline{y}). \tag{7.9}$$

For compressive sensing the signal $\underline{x} \in \mathbb{R}^n$ is continuous, thus marginalization involves integrals instead of discrete sums. Formally, the distributive law (7.1) is replaced by

$$\int_{\mathbb{R}} dx\, a(x)b(x) + \int_{\mathbb{R}} dx\, a(x)c(x) = \int_{\mathbb{R}} dx\, a(x)(b(x) + c(x)). \tag{7.10}$$

Note that for reasonable priors that decay sufficiently fast as $|x| \to +\infty$ all integrals remain finite. The point is that with (7.10) 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.

As in coding, the partition function $Z(\underline{y})$ is a "constant" with respect to the integrals over $\underline{x} \setminus x_i$ involved in the marginalization. Thus, to obtain $p(x_i \mid \underline{y})$, it is sufficient to marginalize the numerator in (7.8) and eventually normalize the resulting function of x_i. As in the coding case, this numerator has a factorized form with two types of "kernel" functions,

$$f_i(x_i) = p(x_i), \quad \text{and} \quad f_a(\{x_i, i : A_{ai} \neq 0\}) = e^{-\frac{1}{2\sigma^2}(y_a - A_a^T \underline{x})^2}. \tag{7.11}$$

The first factor encodes the prior whereas the second factor encodes the relationships induced by the matrix multiplication.

EXAMPLE 11 (Factor graph for compressive sensing) One can associate a "Tanner" graph to the measurement matrix A. 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. There are also additional factor nodes which represent the prior for the signal. There is one important difference of the compressive sensing model and the coding model. In

coding our analysis will rely heavily on the fact that the graph is sparse, i.e., that the number of edges is linear in the number of variables. As we will see, if we look at very large instances of such graphs, the Tanner graph will not be a tree but it will "locally" be a tree. This will be the key to our analysis. For compressive sensing on the other hand we will assume that the entries of the measurement matrix are iid Gaussian and so the matrix is dense. Indeed the resulting graph is a complete bipartite graph. At first glance it therefore appears that message-passing techiques which explicitly rely on the Tanner graph being a tree are of no use in this context. But 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 fact that the sums here are over the reals, whereas for sums over the binary field this would not be true irrespective how large the sum is. ◊

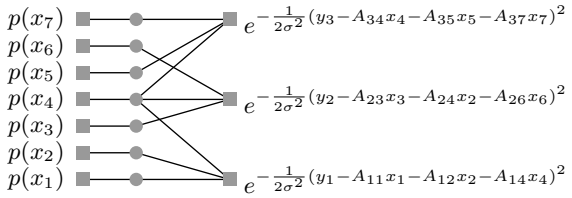


Figure 7.7 Factor graph for compressive sensing. The edges represent the non-zero elements of the measurement matrix. The signal has seven components and there are three measurements.

Let us discuss belief propagation for this example.

EXAMPLE 12 (Message passing algorithm for compressive sensing) We give the first three steps of belief propagation for the tree in Figure 7.7. 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: $\hat{\mu}_{k\to k}(x_k) = p(x_k)$ for $k = 1, \ldots, 7$. At the second step five variables (namely the ones that participate in only one measurement) send messages to factor nodes: $\mu_{1\to 1}(x_1) = p(x_1)$, $\mu_{2\to 1}(x_2) = p(x_2)$, $\mu_{3\to 2}(x_3) = p(x_3)$, $\mu_{5\to 1}(x_5) = p(x_5)$, $\mu_{7\to 1}(x_7) = p(x_7 =$. At the third step the three factor nodes send messages to variable node 4. These are

$$\hat{\mu}_{1\to 4}(x_4) = \int_{\mathbb{R}^2} dx_1 dx_2\, p(x_1)p(x_2)e^{-\frac{1}{2\sigma^2}(y_1-A_{11}x_1-A_{12}x_2-A_{14}x_4)^2},$$

$$\hat{\mu}_{2\to 4}(x_4) = \int_{\mathbb{R}^2} dx_3 dx_6\, p(x_3)p(x_6)e^{-\frac{1}{2\sigma^2}(y_2-A_{22}x_2-A_{23}x_3-A_{26}x_6)^2},$$

$$\hat{\mu}_{3\to 4}(x_4) = \int_{\mathbb{R}^2} dx_5 dx_7\, p(x_5)p(x_7)e^{-\frac{1}{2\sigma^2}(y_3-A_{34}x_4-A_{35}x_5-A_{37}x_7)^2}.$$

We see from this formulation that all integrals are convergent: indeed the expo-

nential is smaller than one and the priors $p(\cdot)$ are of course 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 formulas for messages involving mixtures of Gaussians. 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 analytically tractable integrals because of the absolute value.

At this point we can compute the marginal $\nu(s_4)$. Indeed all messages incoming into variable node 4 are known, and we have

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

To get the true marginal $p(x_4 \mid \underline{y})$ one has to normalize $\mu(s_4)$,

$$p(x_4|\underline{y}) = \frac{\nu(x_4)}{\int_{\mathbb{R}} dx_4\, \nu(x_4)}.$$

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

We remarked in 4.4 that the Lasso estimate can be obtained by taking the prior $p(x_i) = ce^{-\frac{\lambda}{\sigma^2}|x_i|}$, and letting $\sigma \to 0$

$$\lim_{\sigma\to 0} \hat{\underline{x}}_\sigma(\underline{y}) = \text{argmin}_{\underline{x}} \left\{ \frac{1}{2}\|\underline{y} - A\underline{x}\|_2^2 - \lambda\|x\|_1 \right\}. \tag{7.12}$$

Taking the $\sigma \to 0$ limit of the message passing rules developed in this chapter leads to the so-called *min-sum* algorithm. It is instructive to work this out in detail for the current example. But there is also an alternative route how to derive at this result. The belief-propagation (or sometimes also called sum-product algorithm) was derived from the distributed law once we applied 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

$$\min(a+b, a+c) = a + \min(b, c). \tag{7.13}$$

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 consider in detail the min-sum message passing rules and apply it to the Lasso in the homeworks.

7.7 K-SAT: Counting SAT Solutions via Message Passing

In the problems of Chapter 4 you derived the partition function of the K-SAT problem (in the spin notation)

$$Z = \sum_{s_1,\ldots,s_n \in \{-1,+1\}^n} \prod_{a=1}^{M} \Big(1 - \prod_{i \in a} (\frac{1 + s_i J_{ia}}{2})\Big). \tag{7.14}$$

Recall that Z counts the number of solutions of an instance defined by the matrix J_{ia}, $i = 1, \ldots, N$, $a = 1, \ldots, M$. Instead of directly considering this quantity one may first compute

$$\sum_{\sim s_i} \prod_{a=1}^{M} \Big(1 - \prod_{i \in c_a} (\frac{1 + s_i J_{ia}}{2})\Big), \tag{7.15}$$

which is the number of solutions given $s_i = \pm 1$. Of course if one has a method to compute this quantity, it is immediate to deduce Z. In chapter 12 we will see that this marginalization problem allows to, not only count solutions, but also develop an algorithm for finding them[1].

In the present case the marginalization problem involves only one type of kernel function, namely

$$f_a(\{s_i, i : J_{ia} \neq 0\}) = 1 - \prod_{i \in a} (\frac{1 + s_i J_{ia}}{2}). \tag{7.16}$$

There is no prior or bias over the spins like in coding and compressive sensing. In the message passing formalism this corresponds to $f_i(x_i) = 1$. We leave it to the reader to draw her favorite example of a factor graph indicating the factor nodes.

For the K-SAT problem at finite temperature the Gibbs measure

$$\frac{1}{Z} \sum_{\underline{s} \in \{-1,+1\}^n} \prod_{a=1}^{M} \exp\{-\beta \prod_{i \in c_a} (\frac{1 + s_i J_{ia}}{2})\} \tag{7.17}$$

again has a factorized form. It is easy to see that for a tree factor graph the marginalization can be performed with the same message passing rules.

7.8 Summary of message passing equations for general models

In this chapter we learned how to compute the marginals in terms of exact message passing equations on the tree. In the sequel we will use the same set of equations for general loopy graphs. For general graphs, the message passing

[1] This is the belief propagation guided decimation algorithm which finds solutions for $\alpha < 3.86$ in the random 3-SAT problem. This value should be compared to the SAT-UNSAT threshold $\alpha_s(3) \approx 4.26$.

equations are the same but the initial conditions and the schedule differ. Here we summarize the set of equations that we will very often use.

Consider a general model whose "Gibbs" distribution can be factorized as

$$\mu(\underline{x}) = \frac{1}{Z}\prod_a f_a(x_{\partial a}), \qquad Z = \sum_{\underline{x}\in\mathcal{X}^n}\prod_{a=1}^m f_a(x_{\partial a}) \tag{7.18}$$

where $\partial a = \{i | i \in a\}$. For continuous alphabets all sums are replaced by integrals, but otherwise the formalism is the same.

The BP equations are a set of relations linking two types of messages: those flowing from variable to check nodes $\mu_{i\to a}(x_i)$ and those flowing from check to variables node $\hat{\mu}_{a\to i}(x_i)$. These relations read

$$\mu_{i\to a}(x_i) = \prod_{b\in\partial i\setminus a}\hat{\mu}_{b\to i}(x_i) \tag{7.19}$$

$$\hat{\mu}_{a\to i}(x_i) = \sum_{\sim x_i} f_a(x_{\partial a})\prod_{j\in\partial a\setminus i}\mu_{j\to a}(x_j) \tag{7.20}$$

On a tree the messages are uniquely defined by their "initial" values at the leaf nodes. Recall, when the leaf node is a check the outgoing message equals $f_a(x_{\partial a})$ when the leaf node is a check, and equals 1 when the leaf node is a variable. Thanks to messages one can compute the marginals

$$\nu_i(x_i) = \frac{\prod_{a\in\partial i}\hat{\mu}_{a\to i}(x_i)}{\sum_{x_i}\prod_{a\in\partial i}\hat{\mu}_{a\to i}(x_i)} \tag{7.21}$$

$$\nu_a(x_{\partial a}) = \frac{f_a(x_{\partial a})\prod_{i\in\partial a}\mu_{i\to a}(x_i)}{\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu_{i\to a}(x_i)}. \tag{7.22}$$

These are exact on a tree.

We said above will use the same set of equations even for non-tree graphs. There are two points of views which are both useful. The first one is an "algorithmic point of view" that we use throughout Chapters 8-15. One is to fix a natural initial condition depending on the problem at hand, fix a schedule, and compute iterations of the messages, $\mu^{(t)}_{i\to a}(x_i)$, $\hat{\mu}^{(t)}_{a\to i}(x_i)$. At any time t the BP marginals $\nu^{(t)}_i(x_i)$, $\nu^{(t)}_a(x_i)$ are by definition given by the relations (7.19)-(7.20). The "statistical mechanics point of view" which is used throughout Chapters 19-**??** considers (7.19)-(7.20) as fixed point equations for a set of unkowns attached to edges of the graph. Given a fixed point solution one can compute a set of "marginals" from (7.21)-(7.22).

Let us finally stress one important feature of the BP equations that has already been encountered throughout this Chapter. When we compute the marginals it is not important how the messages are normalized. Indeed in (7.21)-(7.22) the normalizations cancel out. We will often exploit this fact and write (7.19)-(7.20) as proportionality relations. This makes many calculations technically easier.

Problems

7.1 *Min-Sum Message Passing rules* In class we discussed how to compute the

marginal of a multivariate function $f(x_1, \ldots, x_n)$ efficiently, assuming that the function can be factorized into factors involving only few variables and that the corresponding factor graph is a tree. We accomplished this by formulating a message-passing algorithm. The messages are functions over the underlying alphabet. Functions are passed on edges. The algorithm starts at the leaf nodes and we discussed how messages are computed at variable and at function nodes.

Recall from the derivation that the main property we used was the *distributive law*. Consider now the following generalization. Consider the so-called *commutative semiring* of extended real numbers (including ∞) with the two operations min and + (instead of the usual operations + and $*$).

(i) Show that both operations are commutative.

(ii) Show that the identity element under min is ∞ and that the identity element under + is 0.

(iii) Show that the distributive law holds.

(iv) If we formally exchange in our original marginalization + with min and $*$ with +, what corresponds to the marginalization of a function?

(v) What are the message passing rules and what is the initialization?

7.2 *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

$$\underline{\hat{x}}^{\text{lasso}}(\underline{y}) = \text{argmin}_{\underline{x}} \left\{ \frac{1}{2} \|\underline{y} - A\underline{x}\|_2^2 - \lambda \|x\|_1 \right\}. \tag{7.23}$$

Consider first the minimum cost given that x_i is fixed.

$$E_i(x_i) = \min_{\sim x_i} \left\{ \frac{1}{2} \|\underline{y} - A\underline{x}\|_2^2 - \lambda \|x\|_1 \right\}. \tag{7.24}$$

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 $E_4(x_4)$.

8 Coding: Belief Propagation

In the last lecture we learned how to marginalize a multivariate function 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. Further, on a tree this messsage-passing algorithm is equivalent to MAP decoding. 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 incoming messages 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 and 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 quickly point out 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 computations considerably more messy.

8.1 Simplification of Message-Passing Rules for Bit-wise MAP Decoding

In the binary case a message $\mu(x)$ can be thought of as a real-valued vector of length two, $(\mu(1), \mu(-1))$ (here we think of the bit values as $\{\pm 1\}$). The initial such message sent from the factor leaf node representing the i-th channel realization to the variable node i is $(p_{Y_i|X_i}(y_i \mid 1), p_{Y_i|X_i}(y_i \mid -1))$ (see Figure 7.6). Recall that at a variable node of degree $K+1$ the message-passing rule calls for a pointwise multiplication:

$$\mu(1) = \prod_{k=1}^{K} \mu_k(1), \qquad \mu(-1) = \prod_{k=1}^{K} \mu_k(-1).$$

Introduce the *ratio* $r_k \triangleq \mu_k(1)/\mu_k(-1)$. The initial such ratios are the likelihood ratios associated with the channel observations. We have

$$r = \frac{\mu(1)}{\mu(-1)} = \frac{\prod_{k=1}^K \mu_k(1)}{\prod_{k=1}^K \mu_k(-1)} = \prod_{k=1}^K r_k;$$

i.e., the ratio of the outgoing message at a variable node is the product of the incoming ratios. If we define the log-likelihood ratios $l_k = \ln(r_k)$, then the processing rule reads $l = \sum_{k=1}^K l_k$.

Consider now the ratio of an outgoing message at a check node which has degree $J+1$. For a check node the associated "kernel" is

$$f(x, x_1, \ldots, x_J) = \mathbb{1}_{\{\prod_{j=1}^J x_j = x\}}.$$

Since in the current context we assume that the x_i take values in $\{\pm 1\}$ (and not $\mathbb{F}_2$) we write $\prod_{j=1}^J x_j = x$ (instead of $\sum_{j=1}^J x_j = x$). We therefore have

$$\begin{aligned} r = \frac{\mu(1)}{\mu(-1)} &= \frac{\sum_{\sim x} f(1, x_1, \ldots, x_J) \prod_{j=1}^J \mu_j(x_j)}{\sum_{\sim x} f(-1, x_1, \ldots, x_J) \prod_{j=1}^J \mu_j(x_j)} \\ &= \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^J x_j = 1} \prod_{j=1}^J \mu_j(x_j)}{\sum_{x_1,\ldots x_J : \prod_{j=1}^J x_j = -1} \prod_{j=1}^J \mu_j(x_j)} = \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^J x_j = 1} \prod_{j=1}^J \frac{\mu_j(x_j)}{\mu_j(-1)}}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^J x_j = -1} \prod_{j=1}^J \frac{\mu_j(x_j)}{\mu_j(-1)}} \\ &= \frac{\sum_{x_1,\ldots,x_J : \prod_{j=1}^J x_j = 1} \prod_{j=1}^J r_j^{(1+x_j)/2}}{\sum_{x_1,\ldots,x_J : \prod_{j=1}^J x_j = -1} \prod_{j=1}^J r_j^{(1+x_j)/2}} = \frac{\prod_{j=1}^J (r_j+1) + \prod_{j=1}^J (r_j - 1)}{\prod_{j=1}^J (r_j+1) - \prod_{j=1}^J (r_j - 1)}. \end{aligned} \tag{8.1}$$

The last step warrants some remarks. If we expand out $\prod_{j=1}^J (r_j + 1)$, then we get the sum of all products of the individual terms r_j, $j = 1, \ldots, J$ (e.g., $\prod_{j=1}^3 (r_j+1) = 1+r_1+r_2+r_3+r_1r_2+r_1r_3+r_2r_3+r_1r_2r_3$). Similarly, $\prod_{j=1}^J (r_j - 1)$ is the sum of all products of the individual terms r_j, where all products consisting of d terms such that $J-d$ is odd have a negative sign (e.g., we have $\prod_{j=1}^3 (r_j - 1) = -1 + r_1 + r_2 + r_3 - r_1r_2 - r_1r_3 - r_2r_3 + r_1r_2r_3$). From this it follows that

$$\prod_{j=1}^J (r_j+1) + \prod_{j=1}^J (r_j - 1) = 2 \sum_{x_1,\ldots,x_J : \prod_{j=1}^J x_j = 1} \prod_{j=1}^J r_j^{(1+x_j)/2}.$$

Applying the analogous reasoning to the denominator, the equality follows. If we divide both numerator and denominator by $\prod_{j=1}^J (r_j + 1)$, we see that (8.1) is equivalent to the statement

$$r = \frac{1 + \prod_j \frac{r_j - 1}{r_j + 1}}{1 - \prod_j \frac{r_j - 1}{r_j + 1}},$$

which in turn implies $\frac{r-1}{r+1} = \prod_j \frac{r_j - 1}{r_j + 1}$. From $r = e^l$ we see that $\frac{r-1}{r+1} = \tanh(l/2)$.

Combining theses two statements we have

$$\tanh(l/2) = \frac{r-1}{r+1} = \prod_{j=1}^{J} \frac{r_j - 1}{r_j + 1} = \prod_{j=1}^{J} \tanh(l_j/2), \quad \text{so that}$$

$$l = 2\tanh^{-1}\big(\prod_{j=1}^{J} \tanh(l_j/2)\big). \tag{8.2}$$

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 log-likelihood ratio (log of the ratio of the two likelihoods) then the processing rules take on a particularly simple form: at variables nodes messages add, and at check nodes the processing rule is stated in (8.2).

8.2 Regular LDPC ensemble on BEC

The BEC is a very special BMSC. As depicted in Fig. 8.1, 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 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 $\{1, ?\}$.

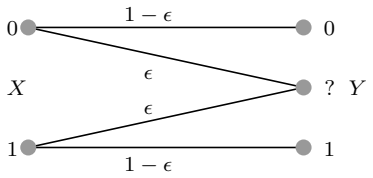


Figure 8.1 Binary erasure channel with parameter ϵ.

The LLRs corresponding to the various channel observations are

$$l = \log\Big(\frac{\mathbb{P}_{Y|X}(y \mid x=1)}{\mathbb{P}_{Y|X}(y \mid x=-1)}\Big) = \begin{cases} \log(\frac{1-\epsilon}{0}) = \infty & y = 1, \\ \log(\frac{\epsilon}{\epsilon}) = 0, & y = ?, \\ \log(\frac{0}{1-\epsilon}) = -\infty, & y = -1. \end{cases}$$

Therefore, the possible values for the LLR are $\{\pm\infty, 0\}$. According to the variable-node rule, the outgoing message from a variable node 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 $\tanh(l/2) \in \{\pm 1, 0\}$, the updating rule of check nodes simplifies (use $\tan(l/2) = \text{sign}(l)$) to the following equation,

$$\text{sign}(l_i) = \prod_{j \in \mathcal{N}(c) \setminus i} \text{sign}(l_j). \tag{8.3}$$

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, the outgoing message is 0.

8.3 Scheduling

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, as we mentioned before, 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 define 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. 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.

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 messaages 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.

8.4 (l, r) Regular LDPC Ensemble

To analyze the performance of the (l, r)-regular LDPC ensemble over the BEC(ϵ), we pick a code, $\mathcal{C}$, uniformly from the ensemble of graphs and run the message passing algorithm. For a given code $\mathcal{C}$ and channel parameter ϵ, let $\mathbb{P}_{\mathrm{BP,b}}(\mathcal{C}, x, \epsilon, \ell)$ denote the average bit error probability of message passing decoder for codeword x at iteration ℓ. We will study the behavior of $\mathbb{P}_{\mathrm{BP,b}}(\mathcal{C}, x, \epsilon, \ell)$ in terms of ϵ and ℓ 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 probabilties.

8.5 Basic Simplifications

Restriction To The All-One Codeword

The first important simplification arises by realizing that we can analyze the error probality of the BP decoder assuming that the all-one codeword (i.e., the codeword, all of its components are 1, where we use the spin language where the components are from the set $\{\pm 1\}$) was transmitted. In formulae, we claim that

$$\mathbb{P}_{\mathrm{BP,b}}(\mathcal{C}, x, \epsilon, \ell) = \frac{1}{|\mathcal{C}|} \sum_{x' \in \mathcal{C}} \mathbb{P}_{\mathrm{BP,b}}(\mathcal{C}, x', \epsilon, \ell) = \mathbb{P}_{\mathrm{BP,b}}(\mathcal{C}, \epsilon, \ell). \tag{8.4}$$

This is true in a general setting. For the statement to hold we need two kinds of symmetry to hold.

- *Channel Symmetry:* First, we need the channel to be symmetric. If we assume that the input is from $\{\pm 1\}$ and that the output is from $\mathbb{R}$, then we need that $p(y \mid x = 1) = p(-y \mid x = -1)$. It is easy to check that all our standard channels, such as the BEC, BSC, or the BAWGNC, are symmetric. Channel symmetry implies exactly what the name suggests, namely that channel "looks" the same from the perspective of a $+1$ input as from the perspective of a -1 input.
- *Decoder Symmetry:* Second, we need that the message-passing decoder preserves this symmetry. For our purposes we are mostly interested in the BP decoder, but in practice one often implements simplified versions. In a nutshell, we require that the decoder does not introduce any bias. More

formally, we require 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 messaages. 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.

For the BEC and BP decoding it is particularly easy to see why (8.4) 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 knwo 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. This is true, since, as [**?**, Thm. 3.30] asserts, the individual behavior of elements of an ensemble is with high probability close to the ensemble average. More precisely,

THEOREM 8.1 (Concentration around Ensemble Average) *Let C, chosen uniformly at random from $\mathcal{C}(n)$, be used for transmission over a BMS channel. Assume that the decoder performs ℓ rounds of message-passing decoding and let $[\mathbb{P}_{BP,b}(\mathcal{C}, \epsilon, \ell)$ denote the resulting bit error probability. Then, for any given $\delta > 0$, there exists an $\alpha > 0$, $\alpha = \alpha(l, r, \delta)$, such that*

$$\mathbb{P}\{|\mathbb{P}_{BP,b}(\mathcal{C}, \epsilon, \ell) - \mathbb{E}_{\mathcal{C}(n)}\left[[\mathbb{P}_{BP,b}(\mathcal{C}, \epsilon, \ell)\right] | > \delta\} \leq \epsilon^{-\alpha n}.$$

In words, the theorem asserts that 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. The proof of the theorem is based on the so-called Hoeffding-Azuma inequality and can be found in [**?**].

8.6 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

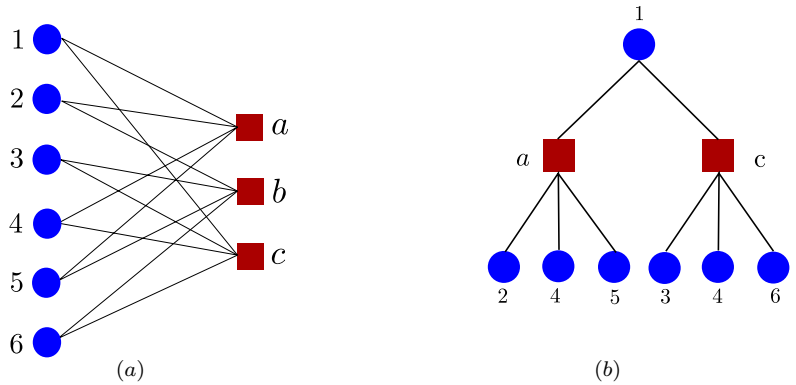


Figure 8.2 (a) The Tanner graph of a $(2,4)$-regular LDPC code with 6 variable nodes; (b) The corresponding computation graph of node 1 for the first iteration.

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 ℓ iterations, the belief of a variable node depends on its initial belief and the beliefs of all the nodes placed within (graph) distance 2ℓ or less. The graph consisting of these nodes is called the computation graph of that variable node of height ℓ. For example, the factor graph of a $(2,4,6)$-regular LDPC code is shown in Fig. 8.2(a) and the computation graph of node 1 with height 1 is also depicted in Fig. 8.2(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 in at most $c\log(n)$ steps, where c is some suitable constant, some node will 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. The following theorem makes this precise.

In particular, it states that computation graphs in sufficiently large LDPC codes are tree-like with high probability.

THEOREM 8.6.1 (Convergence to Locally Tree-Like Graph) *Let T denote the computation graph of a variable node chosen uniformly at random from the set of variable nodes of height ℓ in the (l,r,n)-regular LDPC ensemble. If ℓ is kept fixed then*

$$\lim_{n\to\infty} \mathbb{P}\left(T \textit{ is a tree }\right) = 1. \tag{8.5}$$

Proof We only give a sketch of the proof. We are given the randomly chosen variable node and we construct its computation graph of height ℓ 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 $1 - c/n$, which proves the claim. □

Hence, the above result implies that for the fixed ℓ and as n grows large, the error probability is equal to that observed on a tree.

8.7 Density Evolution

Consider hence the BEC(ϵ). As we just discussed, a random computation graph of a fixed height is tree-like with high probability for large block-lengths. Therefore, the incoming messages to each node of this compuation graph are independent. This simplifies the analysis considerably.

Consider a computation graph T with height ℓ. We divide this computation graph to $\ell + 1$ levels, from 0 to ℓ. 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. 8.3).

Every variable node at the i-th level is the root of a computation tree with height i. However, its root has degree $l - 1$. Let $\{?, +1, -1\}$ denote the outgoing message emitted by variable nodes in the i-th level. It is equal to either `?` (erasure message) with probability x_i or a known value (± 1) with probability $1 - x_i$.

At level $i + 1$, each variable node is connected to $l - 1$ check nodes and each check node is connected to $r-1$ variable nodes of i-th level. The outgoing message of each check node is erasure message, if at least one of its incoming messages is `?`. Since x_i are independent, then the probability that a check node at level $i+1$ sends erasure message is equal to $1 - (1 - x_i)^{r-1}$. The outgoing message from a variable node of $i + 1$-th level, i.e. x_{i+1}, is erasure message if its initial message is erasure message and all of its children (check nodes) at level $i + 1$ also send

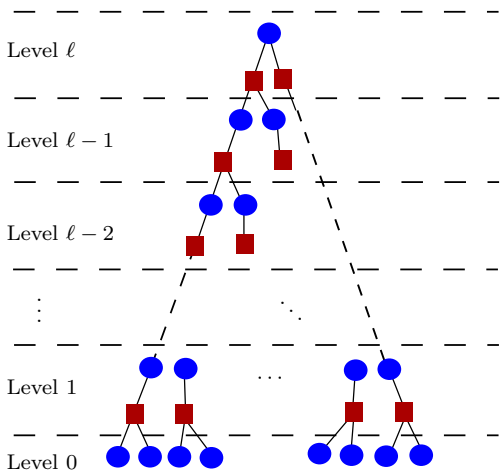


Figure 8.3 A computation graph of $(2,3)$-regular LDPC code with height ℓ. The graph is split to $\ell+1$ levels.

erasure messages. Hence,

$$x_{i+1} = \epsilon(1-(1-x_i)^{r-1})^{l-1} = f(\epsilon, x_i). \tag{8.6}$$

By definition, the outgoing message at level 0 is $x_0 = \epsilon$. Therefore, the erasure probability of the root of T which is connected to l check nodes of level ℓ is

$$\lim_{n\to\infty} \mathbb{P}_{\mathrm{BP,b}}(l,r,n,\epsilon,\ell) = \epsilon(1-(1-x_{\ell-1})^{r-1})^{l} = F(\epsilon, x_{\ell-1}). \tag{8.7}$$

Problems

8.1 *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)?

8.2 *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.

9 Coding: Density Evolution

In the preceeding chapter we have derived the DE equations for a regular ensemble and transmission over the BEC. The task for this chapter is on the one hand to analyse what these equations tell us and on the other hand to explain how to extend this analysis to general BMS channels.

9.1 Density Evolution for the BEC

Recall that we consider a BEC(ϵ) and the (l, r)-regular ensemble. Let $P_{\mathrm{BP,\ b}}(l, r, n, \epsilon, \ell)$ denote the bit error probability of BP decoding for an ensemble of size n, using ℓ iterations (a computation tree of depth ℓ). We know from the last lecture that the limit of this quantity as n goes to infinity is given by $F(\epsilon, x_{\ell-1})$, where

$$F(\epsilon, x) = \epsilon(1 - (1 - x)^{r-1})^l.$$

The quantity x corresponds to the probability of erasure at each iteration, and we can assume $x_0 = 1$. It evolves after each iterations according to the recurrence $x_i = f(\epsilon, x_{i-1})$, where

$$f(\epsilon, x) = \epsilon(1 - (1 - x)^{r-1})^l.$$

We analyze the sequence $\{x_i\}$ and ask whether it converges to 0 or not. In case it does, the decoding is successful, otherwise it is not. Note that convergence depends on ϵ, l, and r.

Remark 9.1 The function $f(\epsilon, x)$ is increasing in ϵ and x for $x, \epsilon \in [0, 1]$.

LEMMA 9.1 (Monotonicity) *Let $2 \leq l \leq r$ and $0 \leq \epsilon \leq 1$. Let $x_0 = 1$ and $x_i = f(\epsilon, x_{i-1})$, $i \geq 1$. Then*

- *The sequence $\{x_i\}$ is decreasing in i.*
- *If $\epsilon \leq \epsilon'$ then $x_i(\epsilon) \leq x_i(\epsilon')$.*

Proof Let us first show thaat the sequence $\{x_i\}$ 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 $x_0 \geq x_1$. Therefore, for $i \geq 2$, we assume $x_{i-1} \leq x_{i-2}$ as the induction hypothesis. Since $f(\epsilon, \cdot)$ is increasing, we obtain $f(\epsilon, x_{i-1}) \leq f(\epsilon, x_{i-2})$. The left hand side is equal to x_i, and the right hand side to x_{i-1}, and we deduce that $x_i \leq x_{i-1}$.

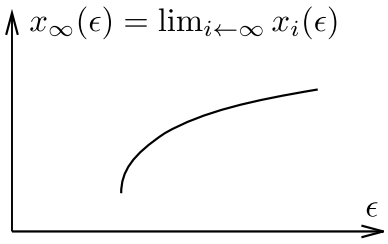


Figure 9.1 Monotonicity of x_∞ as a function of ϵ. For this example, x_∞ jumps at the critical point. There are also examples where x_∞ changes continuously at the critical point. As we have discussed in previous chapters such systems are phase transitions of first and second order, respectively.

To prove the second claim, we use induction once more. Assume that $\epsilon \leq \epsilon'$. Then $x_1(\epsilon) = \epsilon \leq \epsilon' = x_1(\epsilon)$. The general statement is deduced as follows:

$$x_i(\epsilon) = f(\epsilon, x_{i-1}(\epsilon)) \overset{(a)}{\leq} f(\epsilon', x_{i-1}(\epsilon)) \overset{(b)}{\leq} f(\epsilon', x_{i-1}(\epsilon')) = x_i(\epsilon'),$$

where inequality (a) follows from the fact that f is increasing in ϵ, and inequality (b) follows from it being increasing in x, together with the induction hypothesis. □

From the first part of the previous lemma, it follows that $x_i(\epsilon)$ converges in $[0, 1]$. From the second part, it follows that if $x_i(\epsilon) \to 0$ for some ϵ, then $x_i(\epsilon') \to 0$ for all $\epsilon' < \epsilon$. We denote by $x_\infty(\epsilon)$ the limit $\lim_{i\to\infty} x_i(\epsilon)$. Then x_∞ is increasing in ϵ as shown in Figure 9.1. Hence we can define the quantity $\epsilon^{BP} = \sup\{\epsilon : x_\infty(\epsilon) = 0\}$; this is called *the BP threshold*.

There is a graphical way to characterize this threshold. Note that x_∞ is a fixed point of $f(\epsilon, \cdot)$, i.e. $f(\epsilon, x_\infty) = x_\infty$. Thus, if $f(\epsilon, x) - x < 0$ for all $x \in [0, \epsilon]$, then $x_\infty = 0$. For the converese, 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 $(0, 1]$ as $f(\epsilon, x)$ has the form $\epsilon g(\epsilon, x)$, where g is upper bounded by 1. Therefore, if $f(\epsilon, x) - x < 0$ for all $x \in [0, 1]$, then $x_\infty = 0$. For the converese, as soon as there is a fixed point $f(\epsilon, x) = x$ in the interval $(0, 1]$, we have that $x_\infty > 0$. This condition is graphically depicted in Figure 9.2.

EXAMPLE 13 For the $(3, 6)$-regular ensemble, we get $\epsilon^{\text{BP}} \sim 0.4294$. Note that the rate of this ensemble is $R = 1 - \frac{l}{r} = \frac{1}{2}$. Therefore, the fraction 0.4294 has to be compared to the erasure probability that an optimum code could tolerate, which is $\epsilon^{\text{Shannon}} = 1 - R = \frac{1}{2}$. We conclude that already this very simple code,

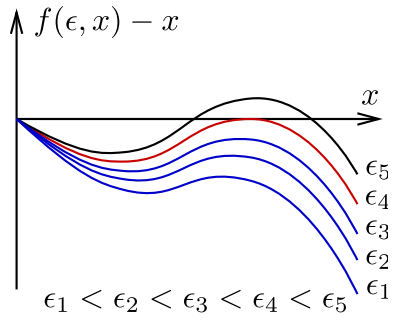


Figure 9.2 The threshold ϵ^{BP} is the largest channel parameter so that $f(\epsilon, x) - x < 0$ for the whole range $x \in [0, 1]$.

together with this very simple decoding procedure can decode up to a good fraction of Shannon capacity.

The previous example immediately suggests several questions? First, how well can we do with such a scheme. E.g., for rate one-half codes, is 0.4294 the highest erasure fraction we can recover from? Second, why does the $(3, 6)$-regular ensemble not achieve capacity? Is this due to a weakness of the code or is the decoding procedure just too simple minded?

For the first question it turns out that in fact we can achieve capacity by slightly tweaking the scheme. There are several ways of doing this. The key step is to look a somewhat more sophisticated ensembles, i.e., ensembles which have some additional structure.

In particular, Luby, Mitzenmacher, Shokrollahi, Spielman and Stemann [2] showed how to construct *irregular* LDPC ensembles which achieve capacity arbitrarily closely under BP. This is very pleasing, since iterative schemes are inherently low complexity.

Let us quickly expand on the main idea without going into details. So far we consider ensembles where every variable node had degree l and every check node had degree r and we called such an ensemble an (l, r)-regular ensemble. From this point of view it is natural to introduce as extension ensembles where we allow nodes of varying degrees. More precisely, define Λ_i as the fraction of variable nodes of degree i in the ensemble; in particular, we have that $\Lambda_i \geq 0$ and $\sum_i \Lambda_i = 1$. Likewise we define R_i as the fraction of check nodes of degree i. The rate of such an ensemble is then quickly determined to be $R = 1 - \frac{\sum_i i\Lambda_i}{\sum_i iR_i}$. The DE equations can be written down in the same manner as we have done this for the regular case and the resulting function $f(\epsilon, x)$, which encodes DE has the same properties as for the regular case. The question is then if we can choose these fractions Λ_i and R_i such that we can approach capacity arbitrarily closely.

This question was answered in the affirmative in [2]. But there is a small price we have to pay. As a function of the gap to capacity, the *average* degree of the nodes has to grow in a logarithmic fashion. This means that also the decoding complexity grows as we approach capacity.

We will come back to the second question later on. In fact, we will see that it connects very nicely to the material in this chapter, despite the fact that we currently look at the performance of a suboptimal algorithm. Just to give a quick preview. Define ϵ^{MAP} as the threshold of the MAP (i.e., the optimal) decoder. By definition, this threshold can only be larger. But how large is it? We will be able to give an analytic answer to this question. E.g., for the $(3,6)$-regular ensemble it will turn out that $\epsilon^{\mathrm{MAP}} \sim 0.4884$. This is considerably larger than $\epsilon^{\mathrm{BP}} \sim 0.4294$ but still falls slightly short of $\epsilon^{\mathrm{Shannon}} = 1 - R = \frac{1}{2}$. We conclude that for the present example both the code as well as the decoder are to blame for the suboptimal performance.

9.2 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 [2]. 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 **??**.

9.3 Density Evolution for General BMS Channels

So far we only considered the very special case of transmission over the BEC. Luckily it turns out that exactly the same type of analysis works for general BMS channels. Again we will be able to derive a DE recursion and this recursion will determine the BP threshold of the scheme. The main difference lies in technicalities. The DE equation for the BEC, encoded by $f(\epsilon, x)$ is a function of two real parameters. Further, this function is monotone in both paramters over the range of interest. This made the analysis simple. For the general case, the DE equation is encoded by a *functional*, i.e., a function of two probability distributions. The first one encodes the channel, in the same way as ϵ encodes the BEC. The second parameter encodes the *state* of the system after a certain number of iterations, and it is quantity equivalent to x.

Rather than explaining all this for the general case let us go through the case of transmission over the BAWGNC. The general case follows along the same steps. The only difference is that in the general case the chennel might not have a description in terms of a density of the log-likelihood ratios but might contain some point masses. For a reader familiar with probability theory this extra complication should not cause any problems. An in case you are not so firm on probability, this is probably not the time and place to start discussing measure theory.

We consider a BAWGNC channel, where the output Y is given in terms of the input X by $Y = X + Z$, where $Z \sim \mathcal{N}(0, \sigma^2)$ is the noise (independent of the input).

If we fix the input to $X = 1$, then $Y \sim \mathcal{N}(1, \sigma^2)$. In this case, the log-likelihood is

$$l = \log \frac{p(y|x=1)}{p(y|x=-1)} = \log \frac{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-1)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y+1)^2}{2\sigma^2}}} = \frac{1}{2\sigma^2}[(y+1)^2 - (y-1)^2] = \frac{2y}{\sigma^2}.$$

Note that in this case, up to rescaling, the log-likelihood is essentially equal to the channel output. Therefore, if we consider the log-likelihood ratio at the output of the channel, conditioned that $X = 1$, as a random variable, its distribution is a Gaussian. Indeed, let L denote this random variable. Then it is distributed like $\mathcal{N}\left(\frac{2}{\sigma^2}, \sigma^2 \left(\frac{2}{\sigma^2}\right)^2\right) \sim \mathcal{N}\left(\frac{2}{\sigma^2}, \frac{4}{\sigma^2}\right)$. Let $a(l)$ denote this density, see Figure 9.3.

Recall that, due to the all-one codeword assumption, positive values of L

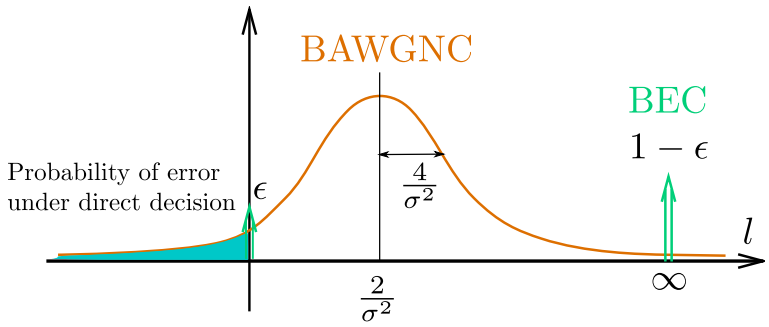


Figure 9.3 The L-density corresponding to the BAWGNC.

indicate that we make a correct decision and negative values that we make an error. Therefore the gray area in Figure 9.3 represents the error probability in case we would make a decision based only on the value received by the channel. More precisely, the bit error probability of the channel is

$$P_b = \int_{-\infty}^{0} a(l)dl = Q\left(\frac{\mu_L}{\sigma_L}\right) = Q\left(\frac{1}{\sigma}\right) \sim e^{-\frac{1}{2\sigma^2}}.$$

Note that P_b tends to 0 as σ approaches 0.

We now look at the message passing rules in the generalized setting.

- At variable nodes, we assume the incoming messages are $L_1, \ldots, L_{l-1}$, with (i.i.d.) distributions $b_i(y)$. The outgoing message is $L = \sum_i^{l-1} L_i$, with L having the distribution $a_{i+1}(y)$. 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.,

$$a_{i+1} = \otimes_{i=1}^{l-1} b_i. \tag{9.1}$$

- At check nodes, we assume the incoming messages are $L_1, \ldots, L_{l-1}$, with (i.i.d.) distributions $a_i(y)$. The outgoing message is $L = 2\tanh^{-1}\left(\prod_{i=1}^{r-1} \tanh\left(\frac{L_i}{2}\right)\right)$, with L having the distribution $b_{i+1}(y)$. Let us write in this case the density of the outgoing random variable in the form

$$b_{i+1} = \oplus_{i=1}^{r-1} a_i. \tag{9.2}$$

  Note that we have written it in a form similar to the convolution which we encountered at variable nodes. Indeed, it turns out that if we are willing to bring all the random variables into a different domain, then again we can write the outgoing random variable as the sume of the incoming random variables and so in this domain again the density of the outgoing

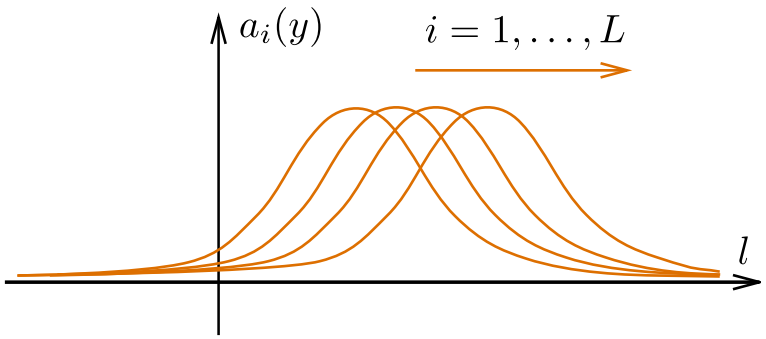


Figure 9.4 The sequence of densities $\{a_i\}$.

random variable is the convolution of the densities of the incoming random variables. This explains the notation. We will not pursue this further here. For our purpose it sufficies to know that there are computationally efficient ways of computing the outgoing density from the incoming densities.

Assume hence that either via a numerical implementaton or by using a population dynamics approach we have succeeded in computing the sequence of densities $\{a_i\}$. A typical such sequence is depicted in Figure 9.4.

As one can see from this figure, as the number of iterations increases, the densities move further and further "to the right." This implies for example that the associated sequence of error probabilities is decreasing, since these error probabilities are given by

$$\lim_{n\to\infty} P_b^{BP}(l, r, \sigma, n, L) = \int_{-\infty}^{0} a_L(y)dy.$$

We would like to proceed in exactly the same manner as we have done this for the BEC. In particular we would like to know if there exists a threshold, i.e., a particular value of the channel parameter so that for elements of the channel family which are "better" than this threshold the sequence of error probabilities converges to zero whereas for elements of this family which are "worse" the error probabilities converge to a non-zero value.

For the BEC the crucial ingredient which facilitated the proof was that there was a simple "order" of all BEC channels – channels with fewer erasures are better than channels with more erasures. We will now see how to introduce a similar notion for the general case.

9.4 Channel Degradation

In order to imitate the proofs that we have obtained for BEC, we need to define an order on the distributions. For two distributions $a(y)$ and $b(y)$, the intuition behind $a(y) \prec b(y)$ is that $a(y)$ is "better" than $b(y)$, and the sequence $a_i(y)$ decreases, as is the case for the BEC.

To define the order relation, we associate to each probability distribution $a(y)$ a binary symmetric channel, with $p(y|x = 1) = a(y)$, and (due to symmetry) $p(y|x = -1) = a(-y)$. A simple calculation shows that if we compute the log-likelihood distribution of this new channel $p(y|x)$ then it is exactly $a(y)$, and thus $a(y)$ representes in fact a channel. We can think of this representation as the canonical representation of a given channel, i.e., two "channels" which might look very different but lead to the same representation, are for all practical purposes the "same."

Let us now define what notion of ordering we are using. Two BMS channels $p(y|x)$ and $q(z|x)$, are said to be ordered by degradation, written as $p(y|x) \prec q(z|x)$, if there exists a memoryless symmetric channel $r(z|y)$, s.t. $q(z|x)$ is the composition of $p(y|x)$ and $r(z|y)$, i.e.,

$$q(z|x) = \sum_y r(z|y)p(y|x). \tag{9.3}$$

As mnemonic. Think of the BEC, then the directon of the sign $\prec$ is the same as if we were thinking of the erasure probabilities of the two corresponding BECs.

CLAIM 1 Given two channels AWGN(σ) and AWGN(σ'), we have that $a_i(y;\sigma) \prec a_i(y,\sigma')$ if and only if $\sigma < \sigma'$.

In other words, the family of AWGNCs is ordered by physical degradation. This is easy to see. In order to convert one AWGNC with noise variance σ^2 into another AWGNC with noise variances $(\sigma')^2$, where $\sigma^2 < (\sigma')^2$, just add some extra AWGN to the first channel, independent of the previous noise and with variance $(\sigma')^2 - \sigma^2$.

The fact that we are dealing with a channel which is ordered by physical degradation gives us now the same properties which we had when we could claim that $f(\epsilon, x)$ is monotone in the first component. In particular, this property ensure that a threshold in fact exists. We also need monotonicity in the second component however. So let us discuss this part as well.

DEFINITION 9.2 We say that $a(y) \prec b(y)$ if the corresponding channels are ordered by physical degragdation.

CLAIM 2 The $a_i(y)$ are monotonically decreasing under degradation, i.e. $a_1(y) \succ a_2(y) \succ \ldots \succ a_i(y) \succ \ldots$. This is similar in the case of BEC, where we had $x_1 > x_2 > \ldots > x_i > \ldots$.

Why is this property true? Consider the computation tree. The point is that

for each of the two types of operations which are involved in density evolution the property of degradedness stays preserved. More precisely, if you for example consider the operaton at a variable node, where the DE operation is the one of a convolution of the densities then if $a \prec b$ then $\otimes a \prec \otimes b$. The same statement is true at check nodes, i.e., $\oplus a \prec \oplus b$. More generally, if you consider any tree and the DE process for this tree, i.e., you look at the density at the output of the root node as a function of the densities at the various variables, this operation preseves degradation, i.e., whenever you replace any of the densities at the variables with a degraded/upgraded density then the output at the root node will be degraded/upgraded with respect to the previous output.

These two properties allow to imitate the proof done over the BEC. They imply that $a_i(y)$ converge weakly towards $a_\infty(y)$ and that the bit error probability behaves as indicated on page 2.

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.

Problems

9.1 *Density Evolution via Population Dynamics* In class we have seen the density evolution (DE) for transmission over the BEC. This was relatively easy since in this case the "densities" are in fact numbers (erasure probabilities). For general channels, DE is more involved since it really involves the evolution of densities. These are the densities of messages which you would see at the various iterations if you implemented the BP message-passing decoder on an infinite ensemble for a fixed number of iterations.

An quick and dirty way of implementing DE for general channels is by means of a population dynamics approach. Here is how this works. Assume that transmission takes place over a given BMS channel and that we are using the (l, r)-regular Gallager ensemble. Pick a population size N. The larger N the more accurate will be your result but the slower it will be.

(i) Pick an *initial* population, call it $\mathcal{V}_0$. This set consists of N iid log-likelihoods associated to the given BMS channel, assuming that the transmitted bit is 1 (we are using spin notation here). More precisely, each sample is created

in the following way. Sample Y according to $p(y \mid x = 1)$. Compute the corresponding log-likelihood value, call it L.

(ii) Starting with $\ell = 1$, where ℓ denotes the iteration number, compute now the densities corresponding to the ℓ-th iteration in the following way.

(iii) To compute $\mathcal{C}_\ell$ proceed as follows. Create N samples iid in the following way. For each sample, call it Y, pick $r-1$ samples from $\mathcal{C}_{\ell-1}$ with repetitions. Let these samples be named $X_1, \ldots, X_{r-1}$. Compute $Y = 2\tanh^{-1}(\prod_{i=1}^{r-1} \tanh(X_i/2))$. Note, these are exactly the message-passing rules at a check node.

(iv) To compute $\mathcal{V}_\ell$ proceed as follows. Create N samples iid in the following way. For each sample, call it Y, pick $l-1$ samples from $\mathcal{C}_\ell$ with repetitions. Let these samples be named $X_1, \ldots, X_{l-1}$. Further, pick a sample from $\mathcal{V}_0$, call it C. Compute $Y = C + \sum_{i=1}^{l-1} X_i$. Note, these are exactly the message-passing rules at a variable node.

We think now of each set $\mathcal{V}_\ell$ and $\mathcal{C}_\ell$ as a sample of the corresponding distribution. E.g., in order to construct this distribution approximately we might use a histogram applied to the set. Recall, that we assume here the all-zero codeword assumption. Hence, in order to see whether this experiments corresponds to a successful decoding, we need to check whether in $\mathcal{V}_\ell$ all samples have positive sign and magnitude which converges (in ℓ) to infinity.

Implement the population dynamics approach for transmission over the BAWGNC(σ) channel using the $(3, 6)$-regular Gallager ensemble. Estimate the threshold using this method. Plot the threshold on the same plot as the simulation results which you performed for your last homework. Hopefully this vertical line, indicating the threshold, is somewhere around where the error probability curves show a sharp drop-off.

10 Interlude: BP to TAP for Sherrington-Kirkpatrick Spin Glass Model

The next two lectures are dedicated to analyzing the performance of compressive sensing under message-passing. The basic outline of the anlysis is very much the same as for coding. But of course, each problem has a few wrinkles on its own.

Recall the basic outline for coding. The decoding problem is an inference problem. The code constraints were encoded in the Tanner graph and the effect of the channel imposed a prior on the bits. This defined the graphical model. We then ran BP on this model to perform the marginalization, since marginals are exactly what the decoder needs to make a decision. If the graph is a tree, this gives us optimal performance. For "real" applications the graph is not a tree but large graphs are "locally tree-like." This allowed us to write down the DE equations. By studying the behavior of the DE equations as a function of the iteration number we were able to define thresholds.

Now compare this to compressive sensing. Again, the decoding problem is an inference problem if we put an appropriate prior on the set of possible "source sequences." Therefore, the next natural step is to run BP on this model. Here we encounter the first difference. Whereas for coding the resulting graphical model was "locall tree-like" this is not at all the case for compressive sensing. Indeed the main part of the graphical model corresponding to the measurement matrix is a complete bipartite (weighted) graph. This is as far as one can get from trees as possible. One might think that this is the end of the story and that BP simply will not work very well on such a model. But in fact BP works quite well. This is true since although we have many loops, every single edge only has a small influence on the outgoing message since there are n such incoming edges and the output is just a weighted and normalized sum of the inputs.

So as we will see not only does BP work very well, but the denseness of the graph leads to significant simplifications for the analysis. In a nutshell, because the outgoing message depends on so many incoming messages, and those messages are to a large degree independent, the outgoing message can be well approximated by a Gaussian and so all we have to determine is the mean and the variance. Several other important simplifications will follow from this picture. This is somewhat reminiscent of how we could simplify the message-passing rules for the binary case by looking at ratios, except that now we are dealing with an approximation which becomes exact as the graph tends to infinity rather than a simplification which is exact per se.

The computations which are necessary to make this simplifications are more complicated though. Therefore, rather than starting right away with compressive sensing, let us look back at a simpler model, namely the Sherrington-Kirkpatrick spin glass model. We will first show how to do the computations in this case. Once the principle is absorbed, the rest involves similar but somewhat more complicated computations. Although we will write down these computations in detail we might not present them in class line by line. This is best left for a rainy Sunday afternoon when you are bored.

10.1 General Spin Systems with Pairwise Interactions

Recall the general setup of Chapter 5. Consider a graph G on n vertices with vertex set V and edge set E. Denote variables by i, $1 \leq i \leq n$, and edges by (i,j). Let the associated Hamiltonian be

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

where J_{ij} are the so-called *coupling* constants associated to each edge $(i,j) \in E$, and the h_i is a site dependent *external magnetic field*. Associated to this Hamiltonian we have our usual Gibbs distribution

$$\mu(\underline{s}) = \frac{e^{-\beta\mathcal{H}_n(\underline{s})}}{Z_n} = \frac{1}{Z_n} \prod_{(i,j)\in E} e^{\beta J_{ij} s_i s_j} \prod_{i\in V} e^{\beta h_i s_i}, \tag{10.1}$$

with $Z_n = \sum_{\underline{s}} e^{-\beta\mathcal{H}_n(\underline{s})}$.

To get the *Curie-Weiss* model of Chapter 5, take G to be the complete graph with J_{ij} normalized and uniform according to $J_{ij} = J/n$, where $J > 0$ is a constant. It turns out that many results are universal and do not depend on the precise distribution of the coupling constants. In the simplest version of the CW model one has a constant external magnetic field $h_i = h$.

To get the *Sherrington-Kirkpatrick* model, choose G also to be the complete graph and $J_{ij} = \tilde{J}_{ij}/\sqrt{n}$, and where the $\tilde{J}_{ij}$ are chosen iid with distribution $\mathcal{N}(0,1)$. Another popular version of the model takes $\tilde{J}_{ij} = \pm 1$ iid Bernoulli(1/2). For the simplest version of the SK model one takes $h_i = h$ constant.

Finally, to get the standard *Ising* model take $G = \mathbb{Z}^d \cap B$, where d is the dimension, and B is a box of some finite side-length. Here the edges $(i,j) \in E$ of the graph consist of all nearest neighbord pairs, $|i-j| = 1$. Further, pick $J_{ij} = J > 0$ for $(i,j) \in E$.

Note that in each of these three problems the normalizations of the constants are different and are chosen in such a way that the free energy has a non-trivial thermodynamic limit.

10.2 BP Equations for General Spin Systems

Let us now write the BP equations for these models. In the following it will be convenient to represent the model in a slightly different way. For every edge $(i,j) \in E$, place a "factor" node on this edge which represents the interaction constant. In this way we get a bipartite graph where every factor node has degree two. Let us denote variables (the vertices of the original graph) by indices like i or j and factor nodes by symbols like a or b.

Let us apply the formalism of 7. Clearly, the Gibbs distribution (10.1) has a factorized form with two types of kernel functions handy

$$f_i(s_i) = e^{\beta h_i s_i}, \qquad \text{and} \qquad f_a(s_i, s_j) = e^{\beta J_{ij} s_i s_j},$$

where $a \equiv (i,j)$. Further, we let $\hat{\mu}_{a\to i}(s_i)$ denote the message which flows from the factor node a to the variable i. It is a function of the spin s_i at position i. In a similar manner, $\mu_{i\to a}(s_i)$ is the message flowing from variable i to factor node a. These messages satisfy the usual BP equations of Chapter 7. Since the messages depend on binary variables $s_i = \pm 1$ we can use the same type of parametrization used for coding in Chapter 8. Let

$$\hat{h}_{a\to i} = \frac{1}{2\beta} \ln\left\{ \frac{\hat{\mu}_{a\to i}(+1)}{\hat{\mu}_{a\to i}(-1)} \right\}, \tag{10.2}$$

$$h_{i\to a} = \frac{1}{2\beta} \ln\left\{ \frac{\mu_{i\to a}(+1)}{\mu_{i\to a}(-1)} \right\}. \tag{10.3}$$

Up to the factor 2β these are the usual log-likelihood variables associated to the messages.[1]

Let us apply our standard message-passing rules to this case. We get

$$h_{j\to a} = h_j + \sum_{b\in\partial j\setminus a} \hat{h}_{b\to j},$$

$$\hat{h}_{b\to j} = \frac{1}{\beta} \operatorname{atanh}\{\tanh(\beta J_{ij}) \tanh(\beta h_{i\to b})\}.$$

These equations are very similar to the ones we discussed in the context of coding theory. The difference to coding is that β is not always equal to 1 and also that there is an extra term $\tanh(\beta J_{ij})$. Note though that this term tends to 1 if J_{ij} tends to infinity. In this limit the constraints become degree two parity check constraints.

The BP-marginal, call it $\hat{\nu}_i^{\mathrm{BP}}(s_i)$, at vertex i is determined from its log-

[1] It is not difficult to see that (10.2) are equivalent to

$$\hat{\mu}_{a\to i}(s_i) = \frac{e^{\beta \hat{h}_{a\to i} s_i}}{2\cosh(\beta \hat{h}_{a\to i})}, \quad \mu_{i\to a}(s_i) = \frac{e^{\beta h_{i\to a} s_i}}{2\cosh(\beta h_{i\to a})}.$$

where the messages have been normalized. These formulas allow to interpret the log-likelihoods as effective magnetic fields.

likelihood variable

$$\eta_i = h_i + \sum_{a \in \partial i} \hat{h}_{a \to i}. \tag{10.4}$$

Explicitly, the normalized marginal is

$$\hat{\nu}_i^{\mathrm{BP}}(s_i) = \frac{e^{\beta \eta_i s_i}}{2\cosh(\beta \eta_i)}.$$

The BP estimate for the magnetization, i.e. the average corresponding to the BP-marginal, is hence equal to

$$m_i^{\mathrm{BP}} = \sum_{s_i \in \{\pm 1\}} s_i \nu_i^{\mathrm{BP}}(s_i) = \tanh(\beta \eta_i).$$

We will call m_i^{BP} the BP-magnetization to distinguish it from the equilibrium (true) magnetization $m_i = \langle s_i \rangle$.

At this point it is useful to give a physical interpretation of this formula. A single spin s in the presence of a magnetic field h has a Hamiltonian $-hs$ and thus a magnetization $\tanh(\beta h)$ (if you have never checked this do it immediately !). Therefore one interprets η_i as an effective magnetic field felt by spin s_i. This is often called the "local field". The local field is the total sum of the external field h_i and "cavity fields" $\hat{h}_{a \to i}$. The later are called cavity fields because their sum represents the field in a cavity left out by the removal of vertex i from the graph.

10.3 BP Algorithm

Since we will apply the BP algorithm to graphs which are not trees (in fact in the SK model the graph is complete) it is important that we specify the schedule. We will opt for a *flooding schedule*. We initialize the iterations with

$$\begin{aligned} h_{j \to a}^0 &= h_j, \\ \hat{h}_{a \to j}^0 &= \operatorname{atanh}\big\{\tanh(\beta J_{ij}) \tanh(\beta h_{i \to a}^0)\big\}, \end{aligned}$$

for all $j \in V, a \in C$. Then at each iteration perform the following operations,

$$\begin{aligned} h_{j \to a}^t &= h_j + \sum_{b \in \partial j \setminus a} \hat{h}_{b \to j}^{t-1}, \\ \hat{h}_{b \to j}^t &= \frac{1}{\beta} \operatorname{atanh}\big\{\tanh(\beta J_{i,j}) \tanh(\beta h_{i \to b}^t)\big\}. \end{aligned}$$

At step t the current BP estimate of the magnetization is

$$m_i^t = \tanh\big\{\beta(h_i + \sum_{a \in \partial i} \hat{h}_{a \to i}^t)\big\}.$$

Since every check has degree exactly two, it is more convenient to write the

whole process in terms of a single step rather than breaking it up into two. Let therefore $\hat{h}_{i\to j} = \hat{h}_{a\to j}$ when $a \equiv (i,j)$. We then get

$$\hat{h}^0_{i\to j} = \frac{1}{\beta}\operatorname{atanh}\big\{(\tanh(\beta J_{ij})\tanh(\beta h_i)\big\}, \tag{10.5}$$

$$\hat{h}^t_{i\to j} = \frac{1}{\beta}\operatorname{atanh}\big\{(\tanh(\beta J_{ij})\tanh(\beta(h_i + \sum_{k\in\partial i\setminus j}\hat{h}^{t-1}_{k\to i}))\big\}. \tag{10.6}$$

As before,

$$m^t_i = \tanh(\beta\eta_i) = \tanh\big\{\beta(h_i + \sum_{j\in\partial i}\hat{h}^t_{j\to i})\big\}. \tag{10.7}$$

Note that in general we have $\Theta(n^2)$ messages we need to update in each iteration. So even a single iteration has quadratic complexity. But for the CW and the SK model one can further simplify the message-passing equations and bring the complexity down to $\Theta(n)$.

10.4 From the BP Algorithm to the CW and the TAP Equations

Both in the CW as well as in the SK model the coupling constants are weak. Indeed, recall that in the CW model we have $J_{ij} = J/n$ and that in the SK model we have $J_{ij} = \tilde{J}_{ij}/\sqrt{n}$. So let us assume in general that the coupling constants J_{ij} are small when $n \to +\infty$, and perform an expansion of the message passing equations. We start with the general case, but at the end we will specialize to the CW and the SK model. In the case of the CW model the simplified message-passing equations are the usual CW equations. More interestingly, for the SK model the simplified message-passing equations are what is called the Thouless-Anderson-Palmer (TAP) equations. As we will see these message passing equations have a complexity of $\Theta(n)$ at each iteration. Thus they provide a linear complexity algorithm to compute the BP-magnetization m^{BP}_i.

Consider the BP iteration (10.5) at step t. Using (10.4) we can rewrite it as

$$\hat{h}^t_{i\to j} = \frac{1}{\beta}\operatorname{atanh}\Big\{\tanh(\beta J_{ij})\tanh(\beta\eta^{t-1}_i - \beta\hat{h}^{t-1}_{j\to i})\Big\}.$$

Now, since J_{ij} is small[2] we linearize both $\tanh(\beta J_{ij}) \sim \beta J_{ij}$ and the inverse hyperbolic tangent. This yields

$$\hat{h}^t_{i\to j} = J_{ij}\tanh(\beta\eta^{t-1}_i - \beta\hat{h}^{t-1}_{j\to i}) + O(\beta^2 J^3). \tag{10.8}$$

Here we abuse notation by writing $O(\beta^2 J^3)$ instead of $O(\beta^2 J^3_{ij})$; this is justified because all the coupling constants are of the same order of magnitude. Equation (10.8) shows that each cavity field is $O(J)$. On the other hand η^{t-1}_i is the sum of

[2] Of order $1/n$ or $1/\sqrt{n}$.

h_i and $n-1$ such cavity fields. Therefore $\hat{h}^{t-1}_{j\to i}$ is much smaller than η^{t-1}_i, so we further expand the hyperbolic tangent in (10.8) to first order in the cavity field,

$$\hat{h}^t_{i\to j} = J_{ij}\tanh(\beta\eta^{t-1}_i) - \beta J_{ij}h^{t-1}_{j\to i}(1-\tanh^2(\beta\eta^{t-1}_i)) + O(\beta^2 J^3).$$

Recalling the expression (10.7) of the BP-magnetization, we can rewrite this formulas as

$$\hat{h}^t_{i\to j} = J_{ij}m^{t-1}_i - \beta J_{ij}\hat{h}^{t-1}_{j\to i}\big(1-(m^{t-1}_i)^2\big) + O(\beta^2 J^3) \tag{10.9}$$

Now we seek an expression for $\hat{h}^{t-1}_{j\to i}$ on the right hand side of this equation, in terms of the BP-magnetization. We note that if we interchange the roles of i and j (note that $J_{ij}=J_{ji}$) and use $\hat{h}^{t-1}_{j\to i}=O(J)$, we get

$$\hat{h}^t_{j\to i} = J_{ij}m^{t-1}_j + O(\beta J^2). \tag{10.10}$$

Replacing (10.10) in (10.9) we obtain

$$\hat{h}^t_{i\to j} = J_{ij}m^{t-1}_i - \beta J^2_{ij}m^{t-1}_j\big(1-(m^{t-1}_i)^2\big) + O(\beta^2 J^3). \tag{10.11}$$

Finally, by replacing this expression in the formula (10.7) for m^t_j we arrive at

$$m^t_j = \tanh\left\{\beta\left(h_j + \sum_{i\in\partial j} J_{ij}m^{t-1}_i - \beta m^{t-1}_j \sum_{i\in\partial j} J^2_{ij}\big(1-(m^{t-1}_i)^2\big)\right)\right\} + O(\beta^3 J^3). \tag{10.12}$$

We have arrived at an approximation of the original BP iterations. The big advantage is that with (10.12) the complexity of each step is $\Theta(n)$, instead of $\Theta(n^2)$ for the BP steps. This comes at a price however. The error terms $O(\beta^3J^3)$ will accumulate as one iterates, and it is not obvious that they can be neglected. Some thought shows that after t iterations the accumulated error for the BP-magnetization is $tO(\beta^3J^3)$. For the CW and SK models $O(\beta^3J^3)$ is $O(n^{-3})$ and $O(n^{-3/2})$, so the error term can be neglected in the regime $n \gg t$. Note that for the standard Ising model on the square grid neglecting this term is not justified. Indeed even if β^3J^3 can be considered small, say at high temperatures, $t\beta^3J^3$ will get large for $t \approx (\beta J)^{-1}$.

CW model

We assume that the error term in (10.12) can be neglected (i.e. we look at the regime $n \gg t$) and discuss the order of magnitude of the terms contributing to the argument of the hyperbolic tangent. For the CW model $J_{ij} = J/n$ and $h_j = h$, so all vertices are equivalent. It is therefore reasonable to seek homogeneous solutions of (10.12) i.e., $m^t_j = m^t$. We observe

$$\sum_{i\in\partial j} J_{ij}m^{t-1}_i = \frac{J}{n}(n-1)m^{t-1} = Jm^{t-1} + O(\frac{1}{n})$$

and

$$\sum_{i\in\partial j} J_{ij}^2(1-(m_i^{t-1})^2) = \frac{J^2}{n^2}(n-1)(1-(m^{t-1})^2) = O(\frac{1}{n}).$$

Thus, in thermodynamic limit $n \to +\infty$ and for fixed t, (10.12) becomes

$$m^t = \tanh\big\{\beta(h + Jm^{t-1})\big\}. \tag{10.13}$$

This is a simple but remarkable result. For the CW model the BP algorithm reduces to (10.13) which is the iterative form of the CW equation (5.24) derived in Chapter 5. This is perhaps a surprizing result. Indeed, the CW equation (5.24) is an equation for the equilibrium magnetization $\frac{1}{n}\sum_{i=1}^n \langle s_i\rangle$, and does not a priori have an "algorithmic meaning". In addition the computations of Chapter 5 are not of algorithmic nature.

Let us summarize what we have learned. We can calculate equilibrium quantities such as the magnetization (and the free energy) from the BP algorithm. Conversely we can guess an iterative algorithm by solving for the equilibrium quantities. This is our first encounter with the "BP-MAP connection" we have alluded to previously in this course. It is a remarkable fact that this connection is valid for a host of more complicated models among which, the SK model, our coding, compressive sensing and random satisfiability models are the main paradigms. In all these cases both the analysis of message passing algorithms and the direct computation of equilibrium quantities are more difficult.[3] We will have to develop various powerful tools and discuss new concepts in the third part of the course to fully uncover the connection.

SK model and TAP equation

Here again the starting point is (10.12) with the error term neglected. We will argue that for the SK model *all terms in the argument of the hyperbolic tangent must be retained*. Recall that $J_{ij} = \tilde{J}_{ij}/\sqrt{n}$ with $\tilde{J}_{ij}$ i.i.d Gaussian of zero mean and unit variance or $\tilde{J}_{ij} = \pm 1$ iid Bernoulli(1/2). Moreover one usually takes $h_i = h$.

Of course m_j^{t-1} depends on the realization of the coupling constants so that $\tilde{J}_{ij}$ and m_j^{t-1} are not independent. In a first stage however we will pretend that they are independent and see how far this leads us. It turns out that although this assumption is far from true, *some of the conclusions* are valid. A rigorous discussion would consist in a course in itself. In Section 10.5 we come back to a few subtle but important issues.

Assuming independence of $\tilde{J}_{ij}$ and m_j^{t-1}

$$\sum_{i\in\partial j} J_{ij} m_j^{t-1} = \frac{1}{\sqrt{n}} \sum_{i\in\partial j} \tilde{J}_{ij} m_j^{t-1} \tag{10.14}$$

[3] The meaning of "more difficult" has to be tuned according to the problem at hand. Random satisfiability and SK being the most difficult representatives which lead to further surprizes.

behaves as a Gaussian variable with zero mean and variance

$$q^{t-1} \equiv \frac{1}{n}\sum_{i=1}^{n}(m_i^{t-1})^2 \tag{10.15}$$

of order one. The quantity q^{t-1} is called the Edwards-Anderson parameter[4]. One expects that the sum in (10.15) concentrates on its mean.

Now consider the term

$$\sum_{i\in\partial j} J_{ij}^2\big(1-(m_i^{t-1})^2\big) \approx \frac{1}{n}\sum_{i\neq j}^{n} \tilde{J}_{ij}^2\big(1-(m_i^{t-1})^2\big) \tag{10.16}$$

Here also, we naively expect that this term concentrates on its mean, and that this mean is of order one. When $\tilde{J}_{ij} = \pm 1$ are Bernoulli(1/2) (10.16) reduces to

$$\sum_{i\in\partial j} J_{ij}^2\big(1-(m_i^{t-1})^2\big) \approx 1 - \frac{1}{n}\sum_{i=1,\neq j}^{n}(m_i^{t-1})^2 \approx 1 - q^{t-1} \tag{10.17}$$

The Edwards-Anderson parameter appears once more.

These naive arguments strongly suggest that both terms (10.14) and (10.16) should be considered of the same order of magnitude and retained. So, apart from neglecting the term $O(\beta^3 J^3)$ equation (10.12) cannot be simplified further.

As pointed out above, this discussion is much too naive. First, it is *never true* that (10.14) behaves as a Gaussian. Second, the Edwards-Anderson parameter (10.15) concentrates on its mean *only in a limited portion of the* (h,β) *plane.* We call this region of the parameter plane the "high temperature phase". This portion corresponds to "high temperatures" and is depicted on figure **??**. It is separated from a low temperature region by a known phase transition line commonly called the Almeida-Thouless line. Third, in the high temperature phase *the whole term* in the argument of the hyperbolic tangent in (10.12) behaves as a Gaussian with variance (10.15). This last fact is remarkable and we come back to it in section 10.5. It is not true in the low temperature phase. In particular, in this phase the Edwards-Anderson parameter does not concentrate.

What is the conclusion of this convoluted discussion? It is that one certainly has to retain the whole argument of the hyperbolic tangent in (10.12). The message passing algorithm now involves $\Theta(n)$ iterations at each step instead of $\Theta(n^2)$. These iterations are

$$m_j^t = \tanh\left\{\beta\left(h_j + \frac{1}{\sqrt{n}}\sum_{i\in\partial j} J_{ij}m_i^{t-1} - \frac{\beta}{n}m_j^{t-1}\sum_{i\in\partial j}\tilde{J}_{ij}^2(1-(m_i^{t-1})^2)\right)\right\} \tag{10.18}$$

[4] Note that here we really have the BP estimate of the Edwards-Anderson parameter. The original Edwards-Anderson parameter is defined through the equilibrium magnetization $m_i = \langle s_i\rangle$.

A popular version of these equations valid for Bernoulli $\tilde{J}_{ij}$ is

$$m_j^t = \tanh\left\{\beta\left(h_j + \frac{1}{\sqrt{n}}\sum_{i\neq j} J_{ij} m_i^{t-1} - \beta m_j^{t-1}(1-q^{t-1})\right)\right\}$$

$$q^{t-1} = \frac{1}{n}\sum_{i=1}^{n}(m_i^{t-1})^2$$

Equation (10.12) is an iterative form of the so-called TAP equations. The original TAP equations concern the equilibrium magnetization and have the same form with $m_i = \langle s_i \rangle$ in place of m_j^t. They are similar to the CW equation except for the extra term

$$-\frac{\beta}{n} m_j^{t-1} \sum_{i\in\partial j} \tilde{J}_{ij}^2 (1-(m_i^{t-1})^2), \qquad \text{or} \qquad -\beta m_j^{(t-1}(1-q^{t-1}).$$

called the *Onsager reaction term*. The usual statistical mechanics derivations the TAP equations and of the Onsager term are not rigorous, but proceed by various methods such as heuristic mean field arguments or high temperature expansions. We will not explicitly show these derivations here.[5]

We contemplate here a second instance of the "BP-MAP connection". Remarkably, both the simplified BP algorithm and the statistical mechanics derivation lead to the same fixed point equation. So message passing allows to guess the statistical mechanical solution of the model, and the statistical mechanical solution allows to guess a low complexity algorithmic scheme. But the situation is more complicated and more interesting than for the Curie-Weiss model. Indeed, firstly it is not clear that the arguments used in the discussion of terms (10.14) and (10.16) are valid. Secondly the mean field calculations are heuristic and the high temperature expansions are not rigorous. There is a good reason for this state of affairs. The replica and cavity methods[6] of statistical mechanics both predict that the conclusions - namely the TAP equations and their consequences - are valid only in the high temperature phase.

10.5 Density evolution for TAP equations

The goal of density evolution is to write down an iterative equation that tracks the evolution of the probability density of the "state" of the system. We review basic results for the SK model that are valid in the high temperature phase where the TAP equations themselves are valid. A rigorous justification is beyond the scope of this chapter. But in the homeworks we propose a numerical justification.

We take the Bernoulli model for which the discussion is slightly simpler. The

[5] Although the mean field derivation can be done on the "back of an envelope".

[6] The replica method is a strange and powerful algebraic method invented by G. Parisi. Its predictions agree with the those of the cavity method which has probabilistic flavor and has been made rigorous for the SK model in the last decade.

results are independent of the precise distribution of $\tilde{J}_{ij}$ for a wide class of distibutions. Recall expression (10.7) for the BP-magnetization,

$$m_i^t = \tanh\{\beta(h + \sum_{j\neq i} \hat{h}^t_{j\to i})\}$$

The TAP approximation consists in replacing the exact cavity field $\hat{h}^t_{i\to j}$ by (see equ. (10.11))

$$\hat{h}^t_{i\to j} \approx \frac{1}{\sqrt{n}}\left\{ \tilde{J}_{ij} m_i^{t-1} - \frac{\beta}{\sqrt{n}} m_j^{t-1}(1-(m_i^{t-1})^2)\right\}$$

The main assumption of density evolution here is that *these cavity fields are sufficiently weakly correlated* so that the sum

$$\sum_{j\neq i} \hat{h}^t_{i\to j} \tag{10.19}$$

is a Gaussian r.v with zero mean and variance

$$\mathbb{E}\left[\left(\tilde{J}_{ij} m_i^{t-1} - \frac{\beta}{\sqrt{n}}\hat{m}_j^{t-1}(1-(m_i^{t-1})^2)\right)^2\right] \approx \mathbb{E}\left[(m_i^{t-1})^2\right] + O(n^{-1/2})$$

The assumption of weak correlation of the cavity fields is non-trivial, and amounts to say that the Onsager reaction term corrects for the *non-Gaussian* nature of the pure Curie-Weiss contribution

$$\frac{1}{\sqrt{n}}\sum_{j\neq i} \tilde{J}_{ij} m_i^{t-1}.$$

When the Onsager reaction term is included the local fiel becomes Gaussian.[7] It is the goal of the homework to check this assumption numerically. Let us discuss one heuristic argument to gain some further intuition. Consider the SK model on a random regular graph of vertex degree d. This is a sparse graph so it is quite natural to consider the BP algorithm in exactly the same way as we did in chapter 8. For a fixed number of iterations t and n large enough the neiborhood of a vertex is a tree with probability $1 - O(d^l/n)$, so that the messages $\hat{h}^t_{i\to j}$ are independent. Now consider the limit $d \to +\infty$. In this limit the meaningful scaling is $J_{ij} = \tilde{J}_{ij}/\sqrt{d}$. Of course it is not necessarily legitimate to interchange the limits $d \to +\infty$ and $n \to +\infty$ but, assuming this is possible then the sum (10.19) behaves as a Gaussian.

Let us now set

$$m^t = \mathbb{E}[(m_i^t)^2], \qquad q^t = \mathbb{E}[(m_i^t)^2]$$

[7] Rigorous proof of this statement appears in recent works of E. Bolthausen (2009) and S. Chaterjee (2010).

Averaging the TAP equation (10.18) we get

$$m^t = \int_{-\infty}^{+\infty} dz \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} \tanh\{\beta(h + z\sqrt{q^{t-1}})\} \tag{10.20}$$

Squaring and then averaging the TAP equation (10.18) we get

$$q^t = \int_{-\infty}^{+\infty} dz \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} \tanh^2\{\beta(h + z\sqrt{q^{t-1}})\}. \tag{10.21}$$

These density evolution equations allow to compute the average magnetization and Edwards-Anderson parameter.

The statistical mechanics solution of the SK model (i.e. the calculation of the free energy, magnetization, etc) proceeds by the replica method (a purely algebraic method) or by the cavity method (which has probabilistic flavor). Quite remarkably there is an exactly known high-temperature region depicted on figure **??** where they both predict that the average magnetization $\mathbb{E}[\langle s_i \rangle]$ and Edwards-Anderson parameter $\mathbb{E}[\langle s_i \rangle^2]$ satisfy the fixed-point form of the density evolution equations (10.20), (10.21). In the low temperature region the theory is much more subtle: let us just mention here that the Edwards-Anderson parameter does not concentrate on its mean but has a non-trivial distribution.

10.6 Notes

In 1936 Onsager was concerned with the dielectric properties of molecular liquids where the so-called "Onsager reaction terms" are important and correct the earlier 1912 theory of Debye. The term "cavity field" was also coined by him. Bethe had similiar insights for magnetism. In 1977 Thouless, Anderson and Palmer (TAP) where the first to point out the importance of the Onsager term in random spin systems. The TAP paper includes a non-algorithmic derivation of the Onsager term through a diagrammatic expansion in the high temperature regime. The SK model has played a very important role in the development of methods and concepts of spin glass theory. These were developed through the 70's and 80's by many physicists and it remained an open mathematical problem for more than 25 years to prove their validity. This was accomplished a decade ago in break through works of Guerra and Talagrand.

Problems

10.1 Distribution of cavity fields in the TAP theory. The goal of this exercise is to numerically justify some of the heuristic arguments of this chapter. When we discuss state evolution for compressive sensing we will encounter similar arguments and hopefully these will seem familiar. Consider the SK model with i.i.d Bernoulli(1/2) coupling constants $\tilde{J}_{ij} = \pm 1$ or $\tilde{J}_{ij}$ Gaussian with zero mean and

unit variance. The TAP approximation to the BP equations reads

$$m_j^t = \tanh\{\beta(h + \sum_{i \neq j} \hat{h}_{i\to j}^t)\}$$

where the update of the cavity fields is

$$\hat{h}_{i\to j}^t = \frac{1}{\sqrt{n}} \tilde{J}_{ij} m_i^{t-1} - \frac{\beta}{n} m_j^{t-1}(1 - (m_i^{t-1})^2)$$

and the initialization $\hat{h}_{i\to j}^{(0)} = 0$.

Take a number $N = 50$ of realizations (coupling constants) of the system of size $n = 500$ or 1000 and an iteration number say $t = 10$. Try values of $(h, T = \beta^{-1})$ in the high temperature regime. The following should be suitable $(h = 0.5, T = 1.2)$ and $(h = 1, T = 0.8)$.

(i) Plot the histogram of the total cavity field

$$\hat{h}_{\text{cav}}^t = \sum_{i \neq j} \hat{h}_{i\to j}^t.$$

This field is equal to a "Curie-Weiss" field to which the "Onsager reaction term" is subtracted. Plot the histogram of the total Curie-Weiss contribution

$$h_{\text{CW}}^t = \sum_{i \neq j} \frac{1}{\sqrt{n}} \tilde{J}_{ij} m_i^{t-1}.$$

(ii) Check that the Edwards-Anderson parameter

$$q^t = \frac{1}{n} \sum_{i=1}^{n} (m_i^t)^2.$$

is concentrated on its empirical mean over the N realizations.

(iii) Compare both histograms with the Gaussian distribution of zero mean and variance equal to the Edwards-Anderson parameter. You should observe that the histogram of the cavity field agrees with this Gaussian.

11 The Conditionning Technique

Bolthauzen's conditionning technique is a method to analyze TAP recursion. Gives analog of DE for TAP equations.

Is the basic tool used to derive state evolution from AMP.

11.1 A toy problem and a basic lemma

11.2 First iteration in TAP

11.3 Main theorem and proof ideas

12 Compressive Sensing: Approximate Message Passing

Let us now look at compressive sensing. Recall from Chapter 4 that a meaningful estimator for the compressive sensing problem is the Lasso estimator, given by

$$\hat{\underline{x}}(\underline{y}, \lambda) = \text{argmin}_{\underline{x}} \Big\{ \frac{1}{2} \|\underline{y} - A\underline{x}\|_2^2 + \lambda \|\underline{x}\|_1 \Big\}. \tag{12.1}$$

We derived this estimator by asking for the estimator which minimizes the mean-squared error in the case where the prior on the components of the signal have a Laplacian distribution (in a small noise limit). But there are also several other "derivations" which end up with this formulation.

We now take a slightly different point of view. We start with the assumption that we want to implement the Lasso estimator. Our previous derivations, showing that the Lasso estimator is optimal under some conditions, serves as motivation for this point of view. But the real "justification" for using this estimator will only be given in hindsight. We will see that this estimator works well in a fairly general setting. Indeed, together with the right structure for the measuring matrix we can in some cases even get optimal performance in terms of its asymptotic (in the size) behavior if we look at the required number of measurements compared to the sparsity of the signal.

It is a long road until we can derive at this conclusion. So for now we will not worry about this. We simply want to implement the Lasso estimator in an efficient manner. The basic idea is straightforward. We first set up a factor graph corresponding to (12.1). Given the factor graph we can mechanically write down the message-passing rules following the general framework about factor graphs set out in Chapter 7, no thinking required. Since the Lasso estimator asks for the best global constellation $\underline{x}$ rather than the best component x_i for each position, our starting point is the min-sum algorithm. This is to some degree a matter of convenience and alternative derivations of the AMP algorithm exist which start with the BP algorithm. Quite surprisingly (the graph is dense and not at all sparse) this works!

In principle this only takes a few lines and we could stop at this point. But there are a few issues. First, there is the issue of complexity. We will see that for the straightforward message-passing algorithm the number of messages which need to be sent in each iteration is quadratic in the graph size. This is true since the graph is dense. The second problem is that the messages are functions and not numbers as was the case for coding. This increases the complexity even

further. So for the rest of the chapter we will see how we can approximate the original message-passing algorithm to (i) first simplify the messages to numbers, and (ii) bring down the number of messages which need to be exchanged in each iteration to a linear number. These calculations are in principle straightforward but they are long. We will see that in order to achieve the second point we can proceed in a fashion very similar to what we did for the SK model where we ended up with the so-called Onsager reaction term. The final algorithm we derive is called AMP, where AMP stand for *approximate message-passing*.

If the simplifications of the message-passing algorithm only had a practical motivation, one could ignore it for the purpose of these lecture. For small examples we could just implement the min-sum algorithm itself and the rest might just be considered engineering. But there is a second, perhaps even more important reason for doing these simplifications. As we will see in the next chapter, for the AMP algorithm we in fact can write down the analysis. This would be out of the question for the original mn-sum algorithm. Finally, even though the AMP algorithm is an approximation, it works very. So we will have derived a relatively simple algorithm which works well and which can be analyzed. All this is well worth the effort!

So without further ado, let us get started.

12.1 Lasso Estimator

From the point of view of statistical physics (12.1) is equivalent to minimizing the Hamiltonian (or cost function)

$$\mathcal{H}(\underline{x}|\underline{y}, A) = \frac{1}{2}\|\underline{y} - A\underline{x}\|_2^2 + \lambda\|\underline{x}\|_1.$$

We explained in Chapter 4 that this cost function can be interpreted as a spin-glass Hamiltonian.

Recall that the matrix A and the observation $\underline{y}$ are random, but once we have a realization they are considered *fixed*. In statistical physics jargon a random variable which is fixed and which we do not average over is called a *quenched* (or frozen) random variable. The degrees of freedom reside in the components x_i. These components are called "continuous spins" since $x_i \in \mathbb{R}$ rather than the usual $s_i \in \{\pm 1\}$;

Recall that the underlying factor graph is the complete bipartite graph with m factor nodes and n variable nodes. It is therefore hopefully clear that this model is, at least superficially, similar to the SK model. Therefore, it should not come as a surprise that the methodology which we follow for the analysis is also similar.

Note that in the formulation above we are looking for the most likely constellation $\underline{x}$. As we pointed out already above, this means that according to the factor graph framework we use the min-sum algorithm (which minimizes the

whole constellation instead of each position). We have seen in Chapter 7, that equivalently we could use the sum-product algorithm applied to the Gibbs distribution $\exp(-\beta\mathcal{H}(\underline{x}|\underline{y}, A))$, and then let β tend to plus infinity. In this "zero temperature" regime, the Gibbs measure is dominated by the minimum energy configurations. But we opt to stick with the min-sum algorithm.

Running min-sum on a complete bi-partite graph with a bi-partition of size n and m respectively, requires $\Theta(mn)$ operations at each iteration, i.e., it is quadratic in the graph size and not linear. For large instances this complexity is prohibitive. We will now show that we can get away with linear complexity. To be sure, the algorithm which we now develop is no longer exact, but it is a good approximation. Further, recall that we are not operating on a tree and so even a full fledged BP is not necessarily optimal. There is therefore no reason to insist on an exact implementation of the BP algorithm.

How can we derive such an approximation? The idea is write down the min-sum equations and then to exploit the fact that $A_{ai} \sim \mathcal{N}(0, \frac{1}{m})$, so that each entry is $O(1/\sqrt{m})$. This leads to significant simplifications. Note that these simplifications will appear even more clearly with the Bernoulli(1/2) ensemble $A_{ai} \in \{+\frac{1}{\sqrt{m}}, -\frac{1}{\sqrt{m}}\}$.

This situation is analogous to that of the SK model. We have seen in the previous chapter that for the SK model we can go from the BP equations to the TAP equations by exploiting the fact that the interaction coefficients are small, explicitly by exploiting that $J_{ij} \sim \mathcal{N}(0, \frac{1}{n})$ or $J_{ij} \sim \text{Ber}(1/2)$ in $\{+\frac{1}{\sqrt{n}}, -\frac{1}{\sqrt{n}}\}$. However, the calculations for the present case are more complicated and some insight can be gained by first looking at a toy problem. This is the subject of the next section.

12.2 Lasso for the Scalar Case

Let $y = x + z$, where $z \sim \mathcal{N}(0, \sigma^2)$. We asssume that the scalar x is "sparse" in the sense that there is a mass of weight $1 - \epsilon$ at $x = 0$ and a mass of weight ϵ distributed for $x \neq 0$. We take the Lasso estimator

$$\hat{x}(y, \lambda) = \text{argmin}_x \Big\{ \frac{1}{2}(y - x)^2 + \lambda|x| \Big\}.$$

This corresponds to the Hamiltonian

$$\mathcal{H}(x|y) = \frac{1}{2}(y - x)^2 + \lambda|x|.$$

Let us check where this Hamiltonian takes on its minimum. For $x > 0$ we have $\mathcal{H}'(x) = -(y - x) + \lambda$. Setting this derivative to 0 we get the solution $\hat{x} = y - \lambda$, which is valid if $y > \lambda$. On the other hand for $x < 0$ we have $\mathcal{H}'(x) = -(y - x) - \lambda$. Setting this derivative to 0 we get the condition $\hat{x} = y + \lambda$, which is valid if $y < -\lambda$. For the remaining case $-\lambda < y < \lambda$ one checks that

$\frac{1}{2}y^2 \le \frac{1}{2}(y-x)^2 + \lambda|x|$ which means that $\hat{x} = 0$. Let us summarize. We get the estimator

$$\hat{x}(y,\lambda) = \begin{cases} y-\lambda, & \text{if } y > \lambda, \\ 0, & \text{if } -\lambda < y < \lambda, \\ y+\lambda, & \text{if } y < -\lambda. \end{cases}$$

This is called the "soft thresholding estimator." Let us express it in terms of the "soft thresholding function" $\eta(y;\lambda)$, where the graph corresponding to $\eta(y;\lambda)$ is shown in Figure 12.1.

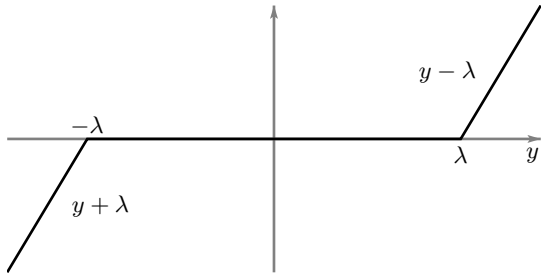


Figure 12.1 Graph of the soft-threshold function $\eta(y;\lambda)$.

In the above estimator we need to choose the threshold λ (specifically if the distribution of x is not known). How shall we choose this value? One possible criterion is to solve the following minimax problem: "Choose the best λ for the worst prior $p_0(x)$." More formally, define

$$\min_{\lambda} \max_{p_0(x)\in\mathcal{F}_\epsilon} \mathbb{E}[|\hat{x}(y,\lambda) - x|^2].$$

Writing it explicitly we get

$$\min_{\lambda} \max_{p_0(x)\in\mathcal{F}_\epsilon} \int dxdy\; p_0(x) \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y-x)^2} (\eta(y,\lambda) - x)^2. \tag{12.2}$$

Here $p_0(\cdot) \in \mathcal{F}_\epsilon$, where $\mathcal{F}_\epsilon$ is the set of distributions of the form $(1-\epsilon)\delta(x) + \phi_0(x)$, where $\phi_0(x)$ is non-negative continuous and has total mass ϵ.

It is natural to set $\lambda = \alpha\sigma$ and to determine α instead of λ (mathematically this is of course equivalent, but the interpretation is that it is natural to choose the threshold on the scale of the noise). The minimax problem (12.2) can be solved exactly. The discussion of its solution is best left to the next chapter.

12.3 Min-Sum Equations

Let us now get back to our main problem. Recall that we want to minimize

$$\mathcal{H}(\underline{x}|\underline{y}, A) = \sum_{a=1}^{m} \frac{1}{2}(y_a - (A\underline{x})_a)^2 + \lambda \sum_{i=1}^{n} |x_i|.$$

We set up a complete bipartite graph with variable nodes i and two types of check nodes corresponding to the factors

$$\frac{1}{2}(y_a - (A\underline{x})_a)^2, \qquad \text{and} \qquad \lambda|x_i|.$$

There are two type of messages flowing from check to variable nodes and from variable to check nodes, call them $\hat{E}_{a\to i}(x_i)$ and $E_{i\to a}(x_i)$. By a straightforward application of the min-sum message passing rules we get the following equations:

$$\begin{cases} E^{t+1}_{i\to a}(x_i) = \lambda|x_i| + \sum_{b\in\partial i\setminus a} \hat{E}^t_{b\to i}(x_i), \\ \hat{E}^{t+1}_{a\to i}(x_i) = \min_{\underline{x}\setminus x_i}\left\{\frac{1}{2}(y_a - (A\underline{x})_a)^2 + \sum_{j\in\partial a\setminus i} E^{t+1}_{j\to a}(x_j)\right\}. \end{cases} \tag{12.3}$$

In addition we have the initialization

$$\begin{cases} E^0_{i\to a}(x_i) = \lambda|x_i|, \\ \hat{E}^0_{a\to i}(x_i) = \min_{\underline{x}\setminus x_i}\{\frac{1}{2}(y_a - (A\underline{x})_a)^2 + \sum_{j\in\partial a\setminus i}\lambda|x_j|\}. \end{cases}$$

The estimate at time t, call it $\hat{x}^t_i(\lambda)$, is computed from

$$\hat{x}^t_i(\lambda) = \operatorname{argmin}_{x_i} E^t_i(x_i),$$

where

$$E^t_i(x_i) = \lambda|x_i| + \sum_{b\in\partial i} \hat{E}^t_{b\to i}(x_i).$$

Recall that in chapter 7 we discussed the BP equations for compressive sensing. As explained there, the min-sum equations (12.3) can be obtained by taking the $\beta\to+\infty$ limit of BP equations. Alternatively one can derive them by a direct application of the distributive law to the min and + operations (see problems in chapter 7.

12.4 Quadratic Approximation

In coding with binary inputs we saw that we could parameterize messages by numbers (the log-likelihood values). In the present case this is a-priori not the case. However, we will now introduce an approximation which admits such a convenient reparameterization. The approximation is called the "quadratic approximation" and it is not yet the AMP algorithm.

The following is a fairly long calculation and somewhat mechanical and technical. In a first reading we recommend that you just look at formulas (12.4) and (12.6) that define the parametrization, and then skip forward directly to the summary of the result in Section 12.4.

Parametrization of messages by real numbers

The crucial observation is that

$$(A\underline{x})_a = \sum_{j=1}^{n} A_{aj} x_j,$$

so that in the message passing expression (12.3) for $\hat{E}^{t-1}_{a\to i}(x_i)$ the x_i dependence enters as $A_{ai}x_i$ and it enters only in the first term. Now $A_{ai}x_i \sim \frac{1}{\sqrt{m}}$. This means this term is small as m tends to infinity. We can therefore consider the Taylor expansion of $\hat{E}^{t+1}_{a\to i}(x_i)$ and only keep the low-order powers of $A_{ai}x_i$.

$$\hat{E}^{t+1}_{a\to i}(x_i) = \hat{E}^{t+1}_{a\to i}(0) - \alpha^{t+1}_{a\to i}(A_{ai}x_i) + \frac{1}{2}\beta^{t+1}_{a\to i}(A_{ai}x_i)^2 + O((A_{ai}x_i)^3), \quad (12.4)$$

where the messages $\alpha^{t+1}_{a\to i}$ and $\beta^{t+1}_{a\to i}$ are real numbers that we will determine later. Equation (12.4) constitutes the parametrization for $\hat{E}^{t+1}_{a\to i}(x_i)$. Replacing this quadratic approximation in the message passing equation (12.3) for $E^{t+1}_{i\to a}(x_i)$ we get

$$\begin{aligned} E^{t+1}_{i\to a}(x_i) &\approx E^{t+1}_{i\to a}(0) + \lambda|x_i| - x_i \sum_{b\in\partial i\setminus a} A_{bi}\alpha^t_{b\to i} + \frac{x_i^2}{2}\sum_{b\in\partial i\setminus a} A^2_{bi}\beta^t_{b\to i} \\ &= E^{t+1}_{i\to a}(0) - \frac{\lambda(a^t_1)^2}{2a^t_2} + \frac{\lambda}{a^t_2}\left\{a^t_2|x_i| + \frac{1}{2}(x_i - a^t_1)^2\right\} \end{aligned} \quad (12.5)$$

where

$$a^t_1 = \frac{\sum_{b\in\partial i\setminus a} A_{bi}\alpha^t_{b\to i}}{\sum_{b\imath\partial i\setminus a} A^2_{bi}\beta^t_{b\to i}}, \qquad a^t_2 = \frac{\lambda}{\sum_{b\in\partial i\setminus a} A^2_{bi}\beta^t_{b\to i}}.$$

Expression (12.5) has been obtained by completing the square. When the right hand side of (12.5) is expanded around its minimum one finds (up to an irrelevant constant)

$$E^{t+1}_{i\to a}(x_i) = \text{Const} + \frac{1}{2\gamma^{t+1}_{i\to a}}(x_i - x^{t+1}_{i\to a})^2 + O((x_i - x_{i\to a})^3) \quad (12.6)$$

where

$$x^{t+1}_{i\to a} = \eta(a^t_1; a^t_2), \qquad \gamma^{t+1}_{i\to a} = \frac{a^t_2}{\lambda}\eta'(a^t_1; a^t_2) \quad (12.7)$$

Equation (12.6) constitutes the parametrization for $E^{t+1}_{i\to a}(x_i)$. In these formulas $\eta(y;\lambda)$ is the same soft thresholding function that was used in the scalar case. The expansion would be exact and the cubic remainder absent for $\lambda = 0$ in which case $\eta(y;0) = y$. For $\lambda \neq 0$ the absolute value is not differentiable at the origin so the derivation involves a few technical subtleties that are worth discussing.[1] Why can one hope that it is a good approximation to expand $E^{t+1}_{i\to a}(x_i)$ near its minimum? One way to understand this is to recall the connection between min-sum and BP.

[1] The rigorous derivation uses a regularization procedure which amounts to work with the BP finite temperature equations, and then take the limit $\beta \to +\infty$.

For $\beta \to +\infty$ the BP messages are proportional to $e^{-\beta E^{t+1}_{i\to a}(x_i)}$, a weight that is dominated by x_i close to the minimum of the exponent. Once this is accepted, it remains to find this minimum and write down the Taylor expansion around it. From the scalar Lasso problem we learn that the minimum of (12.5) over x_i is attained at $x^t_{i\to a} = \eta(a^t_1; a^t_2)$. The expansion is best performed by first assuming that $x^t_{i\to a} > 0$, i.e. $x^t_{i\to a} = \eta(a^t_1; a^t_2) = a^t_1 - a^t_2$. In this case we can set $|x_i| = x_i$ and the first derivative of (12.5) is $\frac{\lambda}{a^t_2}(a^t_2 + (x_i - a^t_1))$ which vanishes at $x^t_{i\to a}$. The second derivative is equal to $\lambda/a^t_2 = \lambda/(a^t_2 \eta'(a^t_1; a^t_2)) = 1/\gamma^t_{i\to a}$. Therefore (12.6) holds when $x^t_{i\to a} > 0$. The reader can work out the case $x^t_{i\to a} < 0$ is a similar way. Finally we consider the singular case $x^t_{i\to a} = 0$, i.e. $\eta(a^t_1; a^t_2) = \eta'(a^t_1; a^t_2) = 0$. At the origin the first derivative of $|x_i|$ has a jump, and the second derivative is formally infinite. Therefore we have to take $\gamma^t_{i\to a} = 0$ which is consistent with $\gamma^t_{i\to a} = \frac{a^t_2}{\lambda}\eta'(a^t_1; a^t_2)$.

The final step is to determine $\alpha^t_{b\to i}$ and $\beta^t_{b\to i}$. For this we replace (12.6) in the second min-sum equation (12.3). Then we compare with the expansion (12.4). After some long but exact algebraic calculations this yields

$$\alpha^t_{a\to i} = \frac{y_a - \sum_{j\in\partial a\setminus i} A_{aj} x^t_{j\to a}}{1 + \sum_{j\in\partial a\setminus i} A^2_{aj}\gamma^t_{j\to a}}, \qquad \beta^t_{a\to i} = \frac{1}{1 + \sum_{j\in\partial a\setminus i} A^2_{aj}\gamma^t_{j\to a}}. \tag{12.8}$$

Let us summarize these calculations. The quadratic approximation assumes that the expansions (12.4) and (12.6) are good approximations and neglect all terms of cubic or higher order. The min-sum equations (12.3) then reduce to message passing equations for real valued messages (12.7), (12.8).

Summary of MinSum equations after the quadratic approximation

Let us now summarize the message-passing rules.

- Variable-to-check messages:

$$\begin{cases} x^{t+1}_{i\to a} = \eta(a^t_1; a^t_2), \\ \gamma^{t+1}_{i\to a} = \frac{a^t_2}{\lambda}\eta'(a^t_1; a^t_2) \end{cases}, \tag{12.9}$$

  where $\eta'(y;\lambda) = \frac{\partial}{\partial y}\eta(y;\lambda)$ and where

$$a^t_1 = \frac{\sum_{b\in\partial i\setminus a} A_{bi}\alpha^t_{b\to i}}{\sum_{b\in\partial i\setminus a} A^2_{bi}\beta^t_{b\to i}}, \qquad a^t_2 = \frac{\lambda}{\sum_{b\in\partial i\setminus a} A^2_{bi}\beta^t_{b\to i}}.$$

- Check-to-variable messages:

$$\begin{cases} \alpha^t_{a\to i} = \frac{y_a - \sum_{j\in\partial a\setminus i} A_{aj} x^t_{j\to a}}{1+\sum_{j\in\partial a\setminus i} A^2_{aj}\gamma^t_{j\to a}}, \\ \beta^t_{a\to i} = \frac{1}{1+\sum_{j\in\partial a\setminus i} A^2_{aj}\gamma^t_{j\to a}}, \end{cases} \tag{12.10}$$

Note that we still have $\Theta(nm)$ equations, i.e., the complexity is still quadratic. But we still gained – we are dealing now with numbers instead of functions $E_{i\to a}(x)$ and $\hat{E}_{a\to i}(x)$.

12.5 Derivation of the AMP Algorithm

Simplications of (12.9) and (12.10)

First, let us simplify further the message passing equations which we have just summarized. Our simplification rests on the assumption that the term in the denominator of (12.10)

$$1 + \sum_{j \in \partial a \setminus i} A_{aj}^2 \gamma_{j \to a}^t$$

can be treated as independent of a and i. Why might this be true? Note that $A_{aj}^2 \sim \frac{1}{m}$ and that we sum over $\Theta(m) = \Theta(n)$ terms. We therefore expect that this sum concentrates on its mean. In the sequel we set

$$1 + \sum_{j \in \partial a \setminus i} A_{aj}^2 \gamma_{j \to a}^t = \frac{\theta_t}{\lambda}$$

and we treat θ_t as independent of a and i. The determination of θ_t is discussed later on.

We also set

$$r_{a \to i}^t = y_a - \sum_{j \in \partial a \setminus i} A_{aj} x_{j \to a}^t, \tag{12.11}$$

so that (12.10) become

$$\alpha_{a \to i}^t = \frac{\lambda}{\theta_t} r_{a \to i}^t, \qquad \beta_{a \to i}^t = \frac{\lambda}{\theta_t}.$$

Let us now look at a_1^t and a_2^t. From $\beta_{b \to i}^t = \frac{\lambda}{\theta_t}$ we deduce that the denominator of a_1^t and a_2^t is equal to

$$\frac{\lambda}{\theta_t} \sum_{b \in \partial i \setminus a} A_{bi}^2$$

Furthermore we note that $\sum_{b \in \partial i \setminus a} A_{bi}^2 \approx 1$ since the A_{bi} are iid $\sim \mathcal{N}(0, \frac{1}{m})$. For the Bernoulli model this sum is exactly equal to $(m-1)/m$ which tends to 1 in the large system size limit. With these remarks we obtain

$$a_1^t = \sum_{b \in \partial i \setminus a} A_{bi} r_{b \to i}, \qquad a_2^t = \theta_t.$$

Replacing in the first message passing equation (12.9) one finds

$$x_{i \to a}^{t+1} = \eta\Big(\sum_{b \in \partial i \setminus a} A_{bi} r_{b \to i}^t ; \theta_t \Big). \tag{12.12}$$

Let us summarize now the current form of the message-passing rules (12.11) and (12.12). We have

$$\begin{cases} x_{i \to a}^{t+1} = \eta\big(\sum_{b \in \partial i \setminus a} A_{bi} r_{b \to i}^t ; \theta_t\big), \\ r_{a \to i}^t = y_a - \sum_{j \in \partial a \setminus i} A_{aj} x_{j \to a}^t. \end{cases}$$

We have simplified (12.9), (12.10) but still have $\Theta(nm)$ updates at each iteration.

At this point the reader should not be surprized that within the quadratic approximation $E_i^t(x_i)$ can be parametrized as follows:

$$E_i^t(x_i) = \frac{1}{2\gamma_i^t}(x_i - \hat{x}_i^t)^2 + O((x_i - \hat{x}_i^t)^3),$$

where

$$\hat{x}_i^t = \eta(\tilde{a}_1^t; \tilde{a}_2^t),$$

and

$$\tilde{a}_1^t = \frac{\sum_{b\in\partial i} A_{bi}\alpha_{b\to i}^t}{\sum_{b\in\partial i} A_{bi}^2\beta_{b\to i}^2}, \qquad \tilde{a}_2^t = \frac{\lambda}{\sum_{b\in\partial i} A_{bi}^2\beta_{b\to i}^t}.$$

This leads to the (Lasso) estimate at time t of the form

$$\hat{x}_i^t = \eta(\sum_{b\in\partial i} A_{bi} r_{b\to i}^t; \theta_t). \tag{12.13}$$

Notice that in (12.13) all messages $r_{b\to i}^t$ entering nodes i are involved, wheras in (12.12) the message $r_{a\to i}^t$ is ommitted.

Finals steps

We are now ready to proceed to the final steps leading to the AMP algorithm. From (12.12) we have

$$\begin{aligned}
x_{i\to a}^{t+1} &= \eta(\sum_{b\in\partial i\setminus a} A_{bi} r_{b\to i}^t; \theta_t) \\
&= \eta(\sum_{b\in\partial i} A_{bi} r_{b\to i}^t - A_{ai} r_{a\to i}^t; \theta_t) \\
&\approx \eta(\sum_{b\in\partial i} A_{bi} r_{b\to i}^t; \theta_t) - A_{ai} r_{a\to i}^t \eta'(\sum_{b\in\partial i} A_{bi} r_{b\to i}^t; \theta_t) \\
&= \hat{x}_i^t - A_{ai} r_{a\to i}^t |\hat{x}_i^t|_0,
\end{aligned}$$

where

$$|\hat{x}_i^t|_0 = \begin{cases} 1, & \text{if } \hat{x}_i^t \neq 0, \\ 0, & \text{if } \hat{x}_i^t = 0. \end{cases}$$

The third approximate equality above is obtained by a taylor expansion to first order in $A_{ai} r_{a\to i}^t \sim 1/\sqrt{m}$. If you go back to chapter **??** you will see that a similar step was performed. This step is crucial and will lead to the "Onsager reaction term". The last equality follows from (12.13) and by remarking that $\eta' = 1$ (resp.

$\eta' = 0$) whenever $\eta \neq 0$ (resp. $\eta = 0$). Replacing this final expression in (12.11)

$$\begin{aligned} r^t_{a\to i} &= y_a - \sum_{j\in\partial a\setminus i} A_{aj} x^t_{j\to a} \\ &= y_a - \sum_{j\in\partial a\setminus i} A_{aj}\hat{x}^{t-1}_j + \sum_{j\in\partial a\setminus i} A^2_{aj} r^{t-1}_{a\to j}|\hat{x}^{t-1}_j|_0 \\ &= (y_a - \sum_{j\in\partial a} A_{aj}\hat{x}^{t-1}_j) + A_{ai}\hat{x}^{t-1}_i + \sum_{j\in\partial a\setminus i} A^2_{aj} r^{t-1}_{a\to j}|\hat{x}^{t-1}_j|_0 \\ &= (y_a - \sum_{j\in\partial a} A_{aj}\hat{x}^{t-1}_j) + \sum_{j\in\partial a} A^2_{aj} r^{t-1}_{a\to j}|\hat{x}^{t-1}_j|_0 + A_{ai}\hat{x}^{t-1}_i - A^2_{ai} r^{t-1}_{a\to i}|\hat{x}^{t-1}_i|_0. \end{aligned}$$

We see that $r^t_{a\to i}$ consists of a main term which is of order one and which is independent of i and the last two terms which do depend on i but which are of order $1/\sqrt{m}$ or $1/m$. So let us write

$$r^t_{a\to i} = r^t_a + \delta r^t_{a\to i}.$$

Up to leading order this yields for the main term

$$r^t_a \approx y_a - \sum_{j\in\partial a} A_{aj}\hat{x}^{t-1}_j + r^{t-1}_a \sum_{j\in\partial a} A^2_{aj}|\hat{x}^{t-1}_j|_0.$$

and for the next order term

$$\delta r^t_{a\to i} \approx A_{ai}\hat{x}^{t-1}_i$$

Using again $A^2_{ai} \sim \frac{1}{m}$ (note again for the Bernoulli model this is exact) the last two equations are summarized as

$$\begin{cases} r^t_a = y_a - \sum_{j\in\partial a} A_{aj}\hat{x}^{t-1}_j + r^{t-1}_a \frac{\|\hat{x}^{t-1}_j\|_0}{m}, \\ \delta r^t_{a\to i} = A_{ai}\hat{x}^{t-1}_i. \end{cases} \tag{12.14}$$

Replacing $r^t_{b\to i} = r^t_b + \delta r^t_{b\to i} = r^t_b + A_{bi}\hat{x}^{t-1}_i$ in the Lasso estimate (12.13) at time t we find

$$\begin{aligned} \hat{x}^t_i &= \eta(\sum_{b\in\partial i} A_{bi} r^t_a + \sum_{b\in\partial i} A^2_{bi}\hat{x}^{t-1}_i; \theta_t) \\ &= \eta(\sum_{b\in\partial i} A_{bi} r^t_a + \hat{x}^{t-1}_i; \theta_t) \end{aligned} \tag{12.15}$$

We can now summarize the final AMP iterative equations (12.15) and (12.14)

$$\begin{cases} \hat{x}^t_i = \eta(\hat{x}^{t-1}_i + \sum_{b\in\partial i} A_{bi} r^t_b; \theta_t), \\ r^t_a = y_a - \sum_{j\in\partial a} A_{aj}\hat{x}^{t-1}_j + r^{t-1}_a \frac{\|\hat{x}^t\|_0}{m}. \end{cases} \tag{12.16}$$

Choice of Threshold θ_t

In the derivations of the previous paragraph we did not precisely specify the threshold θ_t. In fact it is possible to do so by exploiting the equation for $\gamma^t_{i\to a}$

in (12.9). This is the subject of a problem in the homeworks. One finds that the treshold adjusts itself according to the iterations

$$\theta_{t+1} = \lambda + \theta_t \frac{||x^t||_0}{m}. \tag{12.17}$$

However λ still has to tuned suitably.

In this paragraph we discuss a good and *simpler* choice for θ_t that avoids altogether this extra iterative equation. It turns out that the AMP algorithm with the threshold adjustment (12.17) does not offer any significant benefit with respect to the version with the simpler choice. It is not easy to fully justify these points as one first needs the state evolution formalism to ultimately assess the performance of AMP (and its variants).

Let us explain the simpler choice for θ_t. In the scalar case we saw that it is natural to choose the threshold on the scale of the noise, i.e., to set $\lambda = \alpha\sigma$ and then to determine α by solving a minimax problem. In that case, the σ was the standard variation of $y - x$. In the present case it is natural to take θ_t on the scale of $\sqrt{\mathrm{Var}(\sum_{b\in\partial i} A_{bi} r_b^t)}$ which is the term added to the estimate x_i^{t-1} in the first AMP equation. Am rough estimate of this variance is

$$\begin{aligned}\mathrm{Var}(\sum_{b\in\partial i} A_{bi} r_b^t) &= \mathbb{E}\Big(A_{bi}A_{ci} r_b^t r_c^t\Big) \\ &\approx \frac{1}{m}\sum_a (r_a^t)^2 = \frac{1}{m}\|\underline{r}^t\|_2^2.\end{aligned}$$

Therefore we take

$$\theta_t = \alpha \frac{\|r^t\|}{\sqrt{m}}. \tag{12.18}$$

Finally, the parameter α is determined by the minimax problem

$$\inf_\alpha \sup_{p_0(\cdot)\in\mathcal{F}_\epsilon} \mathbb{E}_{\underline{x},\underline{y}}[\|\underline{\hat{x}}(\alpha) - \underline{x}\|_2^2].$$

We will describe the solution of this problem once we have derived the state evolution equations corresponding to the AMP algorithm.

Discussion

We see that the AMP algorithm is almost the same as iterative soft thresholding (IST):

$$\begin{cases} \hat{x}_i^t &= \eta(\hat{x}_i^t + (A^T\underline{r}^t)_i; \theta_t), \\ \underline{r}^t &= \underline{y} - A\underline{\hat{x}}, \end{cases}$$

except for an extra term $\underline{r}^{t-1}\frac{\|\hat{x}^{t-1}\|_0}{m}$. This term, and the way we obtained it, is analogous to the Onsager reaction term in the SK model. This term is crucial. Indeed it is this term that is responsible for the improved performance of AMP

with respect to IST. This performance can be assessed by state evolution which correctly tracks the behavior of the algorithm only if the Onsager term is present. In a nutshell we will see that - analogously to the SK model - $\sum_{j\in\partial a} A_{aj} x_j^{t-1} + r_a^{t-1} \frac{\|\hat{x}^t\|_0}{m}$ has a Gaussian distribution in the large system size limit. This is not true when the Onsager reaction term is ommited.

Although this is not shown in these notes, one can derive the IST algorithm by usual naive mean-field theory arguments and the Onsager reaction term by a TAP-like argument.

Problems

12.1 *A generalization of IST and its connection to Lasso.* The Iterative Soft Thresholding algorithm has the form

$$x_i^{t+1} = \eta(x_i^t + (A^T \underline{r}^t)_i; \lambda)$$
$$\underline{r}^t = \underline{y} - A\underline{x}^t$$

starting from the initial condition $x_i^0 = 0$. Consider the following generalization. Let θ_t and b_t be two sequences of scalars (called respectively "thresholds" and "reaction terms") that converge to fixed numbers θ and b. Construct the sequence of estimates according to the iterations

$$x_i^{t+1} = \eta(x_i^t + (A^T \underline{r}^t)_i; \theta_t)$$
$$\underline{r}^t = \underline{y} - A\underline{x}^t + b_t r^{t-1}$$

The goal of the exercise is to prove that: if x^*, r^* is a fixed point of these iterations, then x^* is a stationary point of the Lasso cost function $L(\underline{x}|\underline{y}, A) = \frac{1}{2}||\underline{y} - A\underline{x}||_2^2 + \lambda||x||_1$ for

$$\lambda = \theta(1 - b)$$

Note that this theorem does not say how to specify suitable sequences b_t and θ_t. The point of AMP is that it specifies unambiguously that one should take $b_t = ||x||_0/m$ (for θ_t there is more flexibility). We will see in the next chapter that with this choice *state evolution correctly tracks the average behavior of the iterative algorithm*, which allows to assess its performance.

The proof proceeds in two steps.

(i) Show that the stationarity condition for the Lasso cost function is

$$A^T(\underline{y} - A\underline{x}^*) = \lambda \underline{v}, \qquad v_i = \text{sign}(x_i^*)$$

where $v_i = \text{sign}(x_i^*)$ for $x_i^* \neq 0$ and $v_i \in [-1, +1]$ for $x_i^* = 0$.

(ii) Show that the fixed point equations corresponding to the iterations above are

$$x_i^* + \theta v_i = x_i^* + (A^T \underline{r}^*)_i$$
$$(1 - b)\underline{r}^* = \underline{y} - A\underline{x}^*$$

Remark that these two equations implies the stationary condition in item (i).

12.2 *AMP with automatic adjustment of threshold* In class, starting from the min-sum equations, we derived an AMP algorithm of the form

$$\hat{x}_i^t = \eta(\hat{x}_i^{t-1} + (A^T \underline{r})_i; \theta_t)$$
$$\underline{r}^t = \underline{y} - A\underline{\hat{x}}^t + \frac{||\underline{\hat{x}}^t||_0}{m} r^{t-1}$$

We argued that a reasonable choice for $\theta_t = \alpha||r^t||_2/\sqrt{m}$. There are however other choices that yield good performance. In particular, one of them follows directly from the min-sum equations. The resulting algorithm is slightly more complex and it turns out there is no benefit in performance.

Deduce from the message passing equations obtained after the quadratic approximation, that one can adjust the threshold according to the iterations

$$\theta_{t+1} = \lambda + \theta_t \frac{||\underline{\hat{x}}^t||_0}{m}$$

Use the same assumption done in class, namely that $1 + \sum_{j \in \partial a \setminus i} A_{aj}^2 \gamma_{j \to a}^t$ is independent of a and i.

13 Compressive Sensing: State Evolution

In the context of coding we were able to assess the performance of the BP algorithm thanks to DE. Recall that in the large-size limit the state of the algorithm is given in terms of a distribution (density). DE then allows us to track this state as a function of the iteration.

It is possible to develop a similar formalism for the AMP algorithm. In the context of compressive sensing, this formalism is called *state evolution* (SE). As we will see, one can derive recursive equations for the MSE whose average behavior is tracked by SE.

An important application of SE is a principled way to compute an optimal threshold parameter λ. We will aslo discuss a related application which consists of determining a "phase transition" line in the "phase diagram" of compressive sensing.

All these derivations have been the subject of extensive numerical as well as analytical work in the last 15 years. They are fairly complicated and here we will limit ourselves to a general description of the main results. Some of these will be supported by only intuitive arguments, some we will do explicitly and rigorously.

13.1 The role of the Onsager term in the TAP and the AMP equations

We begin with a few general analogies between the TAP equations and the AMP equations. Recall the TAP equations (10.18). As explained in Chapter 10, the total cavity field, namely

$$h_{j,\text{cav}} = \frac{1}{\sqrt{n}} \sum_{i=1;i\neq j}^{n} \tilde{J}_{ij} m_i^{(t-1)} - \beta m_j^{(t-1)}(1-q^{(t-1)}), \tag{13.1}$$

becomes Gaussian, more precisely $\mathcal{N}(0, q^{(t-1)})$, as $n \to \infty$. Recall that this would not be the case when the Onsager term $-\beta m_j^{(t-1)}(1-q^{(t-1)})$ is omitted. So it is exactly this term which removes the correlations in the first sum, so that for the remaining sum a "central limit" theorem applies. You checked this numerically in one of the homeworks.

The situation is perfectly analogous for the AMP equations. Recall the AMP

equations (12.16)

$$\begin{cases} \hat{x}_i^{(t+1)} = \eta(\hat{x}_i^{(t)} + \frac{1}{\sqrt{m}} \sum_{b=1}^{m} \tilde{A}_{bi} r_b^t; \theta^{(t)}), \\ r_a^{(t)} = y_a - \frac{1}{\sqrt{m}} \sum_{j=1}^{n} \tilde{A}_{aj} \hat{x}_j^{(t-1)} + r_a^{t-1} \frac{\|\hat{x}^{(t)}\|_0}{m}, \end{cases}$$

where[1] $\tilde{A}_{aj} \sim \mathcal{N}(0,1)$ and $\theta^{(t)} = \alpha \frac{\|r^{(t)}\|_2}{\sqrt{m}}$. One can check numerically (see homework) that the unthresholded estimate

$$\hat{x}_i^{(t)} + \frac{1}{\sqrt{m}} \sum_{b=1}^{m} \tilde{A}_{bi} r_b^{(t)},$$

has a Gaussian distribution, and that this is not true if the Onsager term $r_a^{(t-1)} \frac{\|\hat{x}^t\|_0}{m}$ is omitted. Again, this term in effect cancels all the correlations present among the terms of the sums in the AMP equations.

13.2 Heuristic Derivation of State Evolution

The rigorous derivation of state evolution is based on a technique introduced by E. Bolthausen for the TAP equations. Roughly speaking, this technique allows us to show the following. The behavior under the TAP equations is the same as if we removed the Onsager term but in turn replaced the frozen values $\tilde{J}_{ij}$ by new independent realizations $\tilde{J}_{ij}^{(t)}$ at each time step t. Of course, the latter system is much easier to analyse since for this system the cavity field (13.1) is replaced by

$$\frac{1}{\sqrt{n}} \sum_{i=1; i \neq j}^{n} \tilde{J}_{i,j}^{(t)} m_i^{(t-1)},$$

and we can apply to this sum the central limit theorem. Indeed, the $m_i^{(t-1)}$ are independent of the $\tilde{J}_{ij}^{(t)}$, so that the sum has distribution $\mathcal{N}(0, q^{(t-1)})$.

For the AMP equations we apply the same principle. We remove the Onsager term and at the same time we replace the frozen variables $\tilde{A}_{bi}$ by new and independent realizations $\tilde{A}_{bi}^{(t)}$ chosen from $\mathcal{N}(0,1)$ at each time step. This means that we replace the AMP equations by the equations

$$\begin{cases} \hat{x}_i^{(t+1)} = \eta\left(\hat{x}_i^{(t-1)} + \frac{1}{\sqrt{m}} \sum_{b=1}^{m} \tilde{A}_{bi}^{(t)} r_b^{(t)}; \theta^{(t)}\right), \\ r_a^{(t)} = y_a - \frac{1}{\sqrt{m}} \sum_{j=1}^{n} \tilde{A}_{aj}^{(t)} \hat{x}_j^{(t-1)}. \end{cases}$$

It is convenient for the subsequent discussion to merge the two equations in a single one. Therefore we write

$$\hat{x}_i^{(t+1)} = \eta\left(\hat{x}_i^{(t)} + \frac{1}{\sqrt{m}} \sum_{b=1}^{m} \tilde{A}_{bi}^{(t)} y_b - \frac{1}{m} \sum_{j=1}^{n} (\tilde{A}^{(t)\top} \tilde{A}^{(t)})_{ij} \hat{x}_j^{(t-1)}; \theta_t\right).$$

[1] Here we set $\tilde{A}_{aj} = \frac{1}{\sqrt{m}} A_{aj}$.

In order to be consistent we should also replace $\underline{y} = A\underline{x}_0 + \underline{z}$, where $\underline{x}_0$ is the measured signal by $\underline{y} = \tilde{A}^{(t)}\underline{x}_0 + \underline{z}$. This leaves us with

$$\hat{x}_i^{(t+1)} = \eta\Big(x_{0i} + \frac{1}{\sqrt{m}}\sum_{b=1}^{m}\tilde{A}_{bi}^{(t)}z_b + \sum_{j=1}^{n}(\delta_{ij} - \frac{1}{m}(\tilde{A}^{(t)\top}\tilde{A}^{(t)})_{ij})(\hat{x}_j^{(t-1)} - x_{0j}); \theta^{(t)}\Big). \tag{13.2}$$

One can easily check that the threshold $\theta^{(t)}$ (12.18) becomes

$$\theta^{(t)} = \frac{\alpha}{\sqrt{m}}\|\frac{1}{\sqrt{m}}\tilde{A}^{(t)}(\underline{x}_0 - \underline{\hat{x}}^{(t)}) + \underline{z}\|_2. \tag{13.3}$$

Let us now discuss the behavior of each sum in (13.2), in the limit $m \to \infty$. Clearly, given $\underline{z}$, from the central limit theorem

$$\frac{1}{\sqrt{m}}\sum_{b=1}^{m}\tilde{A}_{bi}^{(t)}z_b \tag{13.4}$$

tends to a Gaussian with zero mean and variance $\frac{1}{m}\sum_{b=1}^{m}z_b^2 \to \sigma^2$. Next, again by the central limit theorem, one shows that the matrix entries $(\delta_{ij} - \frac{1}{m}(\tilde{A}^{(t)\top}\tilde{A}^{(t)})_{ij})$ tend to a zero mean Gaussian with variance $1/m$. By looking at the covariance of these entries we see that they are independent to first order. Thus the term

$$\sum_{j=1}^{n}(\delta_{ij} - \frac{1}{m}(\tilde{A}^{(t)\top}\tilde{A}^{(t)})_{ij})(\hat{x}_j^{(t)} - x_{0j}) \tag{13.5}$$

is also zero mean Gaussian and has variance

$$\frac{1}{m}\sum_{j=1}^{n}(\hat{x}_j^{(t)} - x_{0j})^2 = \frac{1}{\delta}\frac{1}{n}\|\underline{\hat{x}}^{(t)} - \underline{x}_0\|_2^2,$$

where $\delta = \frac{m}{n}$ is the undersampling rate. At this point we define the normalized AMP estimate of the MSE at time t,

$$\tau^{(t)} = \lim_{n\to+\infty}\frac{1}{n}\|\underline{\hat{x}}^{(t)} - \underline{x}_0\|_2^2 \tag{13.6}$$

In thermodynamic limit the variance of (13.5) is equal to $(\tau^{(t)})^2/\delta$. Finally, one can look at the covariance of the two approximate Gaussian variables in (13.4) and (13.5) and show that they are approximately independent.

Let us summarize. We have obtained that in the thermodynamic limit (13.4) is $\mathcal{N}(0, \sigma^2)$, that (13.5) is $\mathcal{N}(0, \frac{1}{\delta}(\tau^{(t)})^2)$, and that they are independent. Thus their sum is $\mathcal{N}(0, \sigma^2 + \frac{1}{\delta}(\tau^{(t)})^2)$. Thus in the thermodynamic limit the first argument of the thresholding function in (13.2) tends to the random variable

$$x_0 + (\sigma^2 + \frac{1}{\delta}(\tau^{(t)})^2)^{1/2}z \tag{13.7}$$

where $z \sim \mathcal{N}(0,1)$ and $x_0 \sim p_0(x)$.

Let us now discuss the fate of $\theta^{(t)}$ in (13.3). Expanding the norm we have

$$\begin{aligned}(\theta^{(t)})^2 &= \frac{\alpha^2}{m}\sum_{b=1}^{m}\left(z_b + \frac{1}{\sqrt{m}}\sum_{i=1}^{n} A_{bi}^{(t)}(x_{0i} - \hat{x}_i^{(t)})\right)^2 \\ &= \frac{\alpha^2}{m}\sum_{b=1}^{m} z_b^2 + 2\frac{\alpha^2}{m\sqrt{m}}\sum_{b=1}^{m}\sum_{i=1}^{n} z_b \tilde{A}_{bi}^{(t)}(x_{0i} - \hat{x}_i^{(t)}) \\ &+ \frac{\alpha^2}{m^2}\sum_{b=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{A}_{bi}^{(t)}\tilde{A}_{bj}^{(t)}(x_{0i} - \hat{x}_i^{(t)})(x_{0j} - \hat{x}_j^{(t)})\end{aligned}$$

The first term tends to $\alpha^2\sigma^2$. The second term can be shown to tend to zero and the third term tends to $\frac{\alpha^2}{\delta}(\tau^{(t)})^2$. Thus in the thermodynamic limit we obtain

$$\theta^{(t)} = \alpha\sqrt{\sigma^2 + \frac{1}{\delta}(\tau^{(t)})^2}. \tag{13.8}$$

From the limits (13.7) and (13.8) of the two arguments of η in (13.2) we conclude that each component $x_i^{(t)}$ tends to the random variable

$$\hat{x}^{(t)} = \eta\left(x_0 + (\sigma^2 + \frac{1}{\delta}(\tau^{(t)})^2)^{1/2} z; \theta^{(t)}\right), \tag{13.9}$$

In this equation $z \sim \mathcal{N}(0,1)$, $x_0 \sim p_0(x)$, $\tau^{(t)})$ is the normalized MSE (13.6), and $\theta^{(t)}$ is given by (13.8). The normalized MSE can be replaced by $|\hat{x}^{(t)} - x_0|^2$. Thus this is a closed equation for a random variable $x^{(t)}$ which plays the role of a state.

An observable of prime importance that one can compute thanks to this formalism is the normalized MSE. From (13.9) we deduce that it satisfies the recursion

$$(\tau^{(t+1)})^2 = \mathbb{E}\left[\left(\eta(x_0 + (\sigma^2 + \frac{1}{\delta}(\tau^{(t)})^2)^{1/2} z; \theta^{(t)}) - x_0\right)^2\right]. \tag{13.10}$$

This equation tracks the MSE as a function of time, and is called the SE equation.[2]

It is sometimes more convenient to work with the following equivalent equation. Set

$$(\tilde{\tau}^{(t)})^2 = \sigma^2 + \frac{1}{\delta}(\tau^{(t)})^2.$$

Then

$$(\tilde{\tau}^{(t+1)})^2 = \sigma^2 + \frac{1}{\delta}\mathbb{E}\left[\left(\eta(x_0 + \tilde{\tau}^{(t)} z; \alpha\tilde{\tau}^{(t)}) - x_0\right)^2\right].$$

It is not difficult to analyse the corresponding fixed point equation. Indeed the right hand side is an increasing and concave function of $\tilde{\tau}$ for all reasonable distributions $p_0(x)$. Moreover, for $\tilde{\tau} = 0$ the right hand side equals σ^2, so is

[2] This is a slight abuse of language; the evolution of the state is given by (13.9).

strictly positive. As a consequence, you can see graphically that there exists a unique solution $\tilde{\tau}^*(\delta, \rho, \alpha, p_0, \sigma)$ in the extended positive real line $[0, +\infty]$.

13.3 Performance of the AMP

Recall some notation. The undersampling ratio is $\delta = \frac{m}{n}$, $\rho = \frac{k}{m}$ is the number of non-zero components per measurement. We call $\mathcal{F}_\epsilon$ the class of distributions with mass $1-\epsilon$ at 0. Note that $\epsilon = \rho\delta$ is the fraction of non-zero components of the signal.

Minimax Criterion

To analyse the performance of the AMP algorithm we have to decide on a criterion of how to choose the threshold α. We already alluded to the choice of the minimax criterion in Chapter 12. The idea is to tune α to the best value when $p_0(x) \in \mathcal{F}_\epsilon$ is the worst distribution. More formally, one solves the following minimax problem,

$$\inf_{\alpha \geq 0} \sup_{p_0 \in \mathcal{F}_\epsilon} \tau^{*2}(\delta, \rho, \alpha, p_0, \sigma), \tag{13.11}$$

where τ^* is the solution of the SE fixed point equation (13.10). As shown latter on

$$\tau^{*2}(\delta, \rho, \alpha, p_0, \sigma) = \sigma^2 \tau^{*2}(\delta, \rho, \alpha, \tilde{p}_0, 1), \tag{13.12}$$

where $\tilde{p}_0(x) = \sigma p_0(\sigma x)$. Then, notice that the class of distributions $\mathcal{F}_\epsilon$ is scale invariant. Indeed

$$\begin{aligned} \tilde{p}_0(x) &= (1-\epsilon)\sigma\delta(\sigma x) + \sigma\Phi_0(\sigma x) \\ &= (1-\epsilon)\delta(x) + \sigma\Phi_0(\sigma x) \\ &= (1-\epsilon)\delta(x) + \tilde{\Phi}_0(x), \end{aligned}$$

thus if $p_0 \in \mathcal{F}_\epsilon$ then $p_0 \in \mathcal{F}_\epsilon$ and vice-versa. Consequently (13.11) is equal to

$$\sigma^2 \inf_{\alpha \geq 0} \sup_{p_0 \in \mathcal{F}_\epsilon} \tau^{*2}(\delta, \rho, \alpha, p_0, 1) = \sigma^2 M(\rho, \delta).$$

The quantity $M(\rho, \delta)$ is sometimes called the *noise sensitivity*. It is the rate of change of the minimax-MSE under changes of the external noise.

It is worth showing (13.12). For this, write explicitly the SE fixed point equation (13.10)

$$\tau^2 = \int dx p_0(x) \int dz \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} \Big(\eta\big(x + (\sigma^2 + \frac{1}{\delta}\tau^2)^{1/2} z; \alpha(\sigma^2 + \frac{\tau^2}{\delta})^{1/2}\big) - x \Big)^2. \tag{13.13}$$

Set $\tau = \sigma\tau_1$. We have to show that τ_1 satisfies the same fixed point equation

with σ and p_0 replaced by 1 and $\tilde{p}_0$ respectively. This is easily seen by making the change of variables $x \to \sigma x$ and using the property $\eta(\sigma y; \sigma\lambda) = \sigma\eta(y; \lambda)$.

Our next goal is to compute the noise senitivity $M(\rho, \delta)$. This is not a trivial task since one has to first compute the minimax of $\tau^*(\delta, \rho, \alpha, p_0, \sigma = 1)$, which itself satisfies a non-trivial fixed point equation, and then we have to optimize over α and $p_0(x)$. Remarkably, there is a closed form expression that can be expressed in terms of the analogous quantity for the scalar case. We thus revisit the scalar case first.

Minimax of the scalar case

If you have a look at the equation (12.2) in Chapter 12 and set $y = x_0 + \sigma z$ (with $z \sim \mathcal{N}(0, 1)$) then you easily see that for the scalar case the minimax is equal to

$$\begin{aligned}\sigma^2 \inf_{\alpha \geq 0} \sup_{p_0 \in \mathcal{F}_\epsilon} \int dx p_0(x) \int dz \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} (\eta(x+z;\alpha) - x)^2 \\ = \sigma^2 \text{min}_\alpha M_{\text{scalar}}(\epsilon, \alpha) \\ = \sigma^2 M_{\text{scalar}}(\epsilon).\end{aligned}$$

The solution of this problem is already non-trivial in itself (see Donoho 1994). For fixed α the worst case distribution turns out to be

$$p_{0,\text{worst}}(x_0) = (1-\epsilon)\delta(x_0) + \frac{\epsilon}{2}\delta_{+\infty}(x_0) + \frac{\epsilon}{2}\delta_{-\infty}(x_0).$$

If we plug this distribution into the minimax one finds after a few calculations that it is equal to $\inf_{\alpha \geq 0} M_{\text{scalar}}(\epsilon, \alpha)$, where

$$M_{\text{scalar}}(\epsilon, \alpha) = \epsilon(1+\alpha^2) + (1-\epsilon)\big(2(1+\alpha^2)\Phi(-\alpha) - 2\alpha\frac{e^{-\frac{\alpha^2}{2}}}{\sqrt{2\pi}}\big), \tag{13.14}$$

where $\Phi(-\alpha) = \int_{-\infty}^{-\alpha} \frac{e^{-\frac{u^2}{2}}}{\sqrt{2\pi}} du$. Setting the derivative of $M_{\text{scalar}}(\epsilon, \alpha)$ with respect to α to zero we obtain

$$\epsilon = \frac{2\big(\frac{e^{-\frac{\alpha^2}{2}}}{\sqrt{2\pi}} - \alpha\Phi(-\alpha)\big)}{\alpha + 2\big(\frac{e^{-\frac{\alpha^2}{2}}}{\sqrt{2\pi}} - \alpha\Phi(-\alpha)\big)} \tag{13.15}$$

One can check that the right hand side is a monotone decreasing function of α. Thus, given ϵ there exist a unique optimal $\alpha_{\text{best}}(\epsilon)$. One can then find the minimax-MSE for the scalar problem as

$$M_{\text{scalar}}(\epsilon) = M_{\text{scalar}}(\epsilon, \alpha_{\text{best}}(\epsilon)) \tag{13.16}$$

Minimax for the vector case and the notion of noise sensitivity phase transition

As said before, it is possible to compute the minimax for the vector case. Before indicating how this can be done, we discuss the result which is is remarkably

simple,

$$M(\delta,\rho)=\begin{cases}\frac{M_{\text{scalar}}(\rho\delta)}{1-\frac{1}{\delta}M_{\text{scalar}}(\rho\delta)} & \rho<\rho_c(\delta)\\ +\infty & \rho>\rho_c(\delta),\end{cases} \tag{13.17}$$

where $\rho_c(\delta)$ is the solution of the equation

$$\delta = M_{\text{scalar}}(\rho\delta). \tag{13.18}$$

Figure 13.1 shows a plot of the curve $\rho_c(\delta)$ in the (δ,ρ)-plane. This curve sepa-

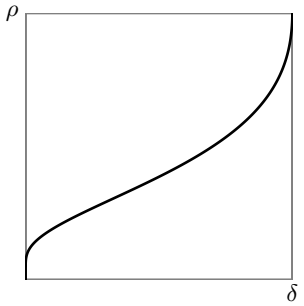


Figure 13.1 The function $\rho_c(\delta)$ in the (δ,ρ)-plane.

rates the (δ,ρ) plane in two regions where $M(\delta,\rho)=+\infty$ and where $M(\delta,\rho)$ is finite. In other words on can recover the sparse signal with finite error only for $\rho<\rho_c(\delta)$. From the point of view of statistical physics, Figure 13.1 is a phase diagram and the separating curve a phase transition curve.

It is easy to write this curve in parametrized form. Indeed with (13.15) and (13.16) we see that (13.18) is equivalent to

$$\begin{cases}\delta = M_{\text{scalar}}(\rho\delta,\alpha)\\ \delta\rho = \dfrac{2\left(\frac{e^{-\frac{\alpha^2}{2}}}{\sqrt{2\pi}}-\alpha\Phi(-\alpha)\right)}{\alpha+2\left(\frac{e^{-\frac{\alpha^2}{2}}}{\sqrt{2\pi}}-\alpha\Phi(-\alpha)\right)}.\end{cases}$$

Using (13.14), a bit of algebra leads to the more pleasant form

$$\begin{cases}\delta = 2\dfrac{e^{-\frac{\alpha^2}{2}}}{\sqrt{2\pi}}\dfrac{1}{\alpha+2\left(\frac{e^{-\frac{\alpha^2}{2}}}{\sqrt{2\pi}}-\alpha\Phi(-\alpha)\right)}\\ \rho = 1-\sqrt{2\pi}\alpha e^{\frac{\alpha^2}{2}}\Phi(-\alpha).\end{cases}$$

We conclude this paragraph with a calculation justifying formula (13.17). The starting point is again the fixed point equation (13.13) and a scaling argument. The change of variables $x\to(\sigma^2+\frac{1}{\delta}\tau^2)^{1/2}x$ leads to

$$\tau^2=(\sigma^2+\frac{1}{\delta}\tau^2)\int dx p_0^{(\tau)}(x)\int dz\frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}}(\eta(x+z,\alpha)-x)^2 \tag{13.19}$$

where

$$p_0^{(\tau)}(x) = (\sigma^2 + \frac{1}{\delta}\tau^2)^{1/2} p_0((\sigma^2 + \frac{1}{\delta}\tau^2)^{1/2} x)$$

The integral in (13.19) is the scalar MSE for a scaled signal distribution and $\sigma = 1$. Let us denote it by $M_{\text{scalar}}(\epsilon, \alpha, p_0^\tau)$. Remark that scale invariance of the set $\mathcal{F}_\epsilon$ implies

$$\begin{aligned}\sup_{p_0 \in \mathcal{F}_\epsilon} M_{\text{scalar}}(\epsilon, \alpha, p_0^\tau) &= \sup_{p_0 \in \mathcal{F}_\epsilon} M_{\text{scalar}}(\epsilon, \alpha, p_0) \\ &= M_{\text{scalar}}(\epsilon, \alpha)\end{aligned}$$

where the last equality is attained for $p_{0\text{worst}}$. Suppose first that $M_{\text{scalar}}(\epsilon, \alpha) > \delta$. Then $\tau|_{p_{0\text{worst}}} = +\infty$ (because (13.19) cannot have a finite solution) so that $\sup_{p_0 \in \mathcal{F}_\epsilon} \tau = +\infty$. Consider now the case $M_{\text{scalar}}(\epsilon, \alpha) < \delta$. In particular this means that $M_{\text{scalar}}(\rho\delta) < \delta$ or $\rho < \rho_c(\delta)$. For such α's the solution τ of (13.19) is finite for all $p_0 \in \mathcal{F}_\epsilon$, and satisfies,

$$\tau^2 = \sigma^2 \frac{M_{\text{scalar}}(\epsilon, \alpha, p_0^\tau)}{1 - \frac{1}{\delta} M_{\text{scalar}}(\epsilon, \alpha, p_0^\tau)}$$

Now, we maximize both sides of this equation over $p_0 \in \mathcal{F}_\epsilon$. Since the set $\mathcal{F}_\epsilon$ is scale invariant we can replace p_0^τ by p_0 in the maximization. Thus far we have obtained

$$\sup_{p_0 \in \mathcal{F}_\epsilon} \tau^2 = \begin{cases} \sigma^2 \sup_{p_0 \in \mathcal{F}_\epsilon} \left\{ \frac{M_{\text{scalar}}(\epsilon, \alpha, p_0)}{1 - \frac{1}{\delta} M_{\text{scalar}}(\epsilon, \alpha, p_0)} \right\}, & M_{\text{scalar}}(\epsilon, \alpha) < \delta \\ +\infty, & M_{\text{scalar}}(\epsilon, \alpha) > \delta \end{cases}$$

Now we wish to further minimize this expression over $\alpha \geq 0$. Formally, under a variation of the parameters $\Delta\alpha$ and Δp_0 the variation of the ratio in the first equation is equal to

$$\frac{\Delta M_{\text{scalar}}}{(1 - \frac{1}{\delta} M_{\text{scalar}})^2},$$

so the stationnary point satisfies $\Delta M_{\text{scalar}} = 0$, just like for the pure scalar problem, whose solution solution α_{best} and $p_{0\text{worst}}$ was discussed in Section 13.3. Using this stationnary point we find

$$\inf_{\alpha > 0} \sup_{p_0 \in \mathcal{F}_\epsilon} \tau^2 = \begin{cases} \sigma^2 \frac{M_{\text{scalar}}(\rho\delta)}{1 - \frac{1}{\delta} M_{\text{scalar}}(\rho\delta)}, & M_{\text{scalar}}(\rho\delta) < \delta \\ +\infty, & M_{\text{scalar}}(\rho\delta) > \delta. \end{cases}$$

This is formula (13.17).

13.4 Discussion

It remains to discuss a point, namely the relationship between the true Lasso estimator (i.e., obtained by performing an exact minimization of the Lasso Hamiltonian) and the AMP estimator?

This question is analogous to the situation in coding theory where we want to compre the BP threshold to the MAP threshold. For coding we will look at this question in later chapters, but for compressive sensing the answer is remarkably simple. In a previous homework, you proved a simple but important theorem. This theorem states that a fixed point of the AMP equations $(\underline{\hat{x}}^*, \underline{r}^*, \theta^*)$ is a stationary point of the Lasso cost function for

$$\lambda = \theta^*(1 - \frac{\|\underline{\hat{x}}^*\|_0}{m})$$

In other words, running the AMP algorithm yields the current minimum of Lasso for

$$\lambda(\alpha) = \alpha\frac{\|\underline{r}^*\|_2}{\sqrt{m}}(1 - \frac{\|\underline{\hat{x}}^*\|_0}{m}).$$

Therefore we can conclude that the "true Lasso" estimation $\underline{\hat{x}}(\lambda)$ has an MSE of

$$\lim_{n\to\infty} \frac{1}{n}\|\underline{\hat{x}}(\lambda) - \underline{x}_0\|_2^2 = \tau^2,$$

which satisfies the state evoluiton fixed point equation for

$$\tilde{\tau}^2_{\text{Lasso}} = \sigma^2 + \frac{1}{\delta}\tau^2_{\text{Lasso}}.$$

$$(\tilde{\tau}^*)^2 = \sigma^2 + \frac{1}{\delta}\mathbb{E}\left[\left(\eta(x_0 + (\tilde{\tau}^*)^2 z; \alpha\tilde{\tau}) - x_0\right)^2\right]$$

provided that we take

$$\lambda(\alpha) = \alpha\tilde{\tau}^*\left(1 - \frac{1}{\delta}\mathbb{E}\left[\eta'(x_0 + (\tilde{\tau}^*)^2 z; \alpha\tilde{\tau})\right]\right).$$

This relationship bewen λ and α has been called "calibration map" in the literature.

At this point we again see that there is a close connection between message passing solutions and exact solutions.We explicitly saw this for the CW model and we discussed this for the SK model. We will come back to such a connection in the case of coding and the K-SAT problem in the third part of this course.

Last but not least there is one more remarkable feature of the AMP algorithm. The phase transition curve $\rho_c(\delta)$ is exactly the same as the one derived by Donoho and Tanner by solving exactly the l_1-minimization problem

$$\underline{\hat{x}}^{(l_1)} = \text{argmin}_{A\underline{x}=\underline{y}}\|\underline{x}\|_1.$$

From the perpective of message passing techniques that we have developped so far this is not completely surprizing. Indeed one can reformulate this minimization problem as the study of a "Gibbs" measure

$$\frac{1}{Z}\exp\left\{-\frac{\beta_2}{2}\|\underline{y} - A\underline{x}\|_2^2 + \beta_1\|\underline{x}\|_1\right\} \tag{13.20}$$

with two inverse "temperatures" and study this problem by going through a

BP and AMP formalism similar to what we have presented in this chapter. The connection with l_1 minimization boils down to note that

$$\underline{\hat{x}}^{(l_1)} = \lim_{\beta_1 \to +\infty} \lim_{\beta_2 \to +\infty} \langle \underline{x} \rangle$$

for *finite* n. The coincidence of the Donoho-Tanner curve and the AMP phase transition curves means that one can exchange the thermodynamic and zero temperature limits, a fact that is often non-trivial to prove in the context statistical mechanics.

Problems

13.1 *Statistics of AMP and IST un-threshoded estimates.* Consider a sparse signal $\underline{x}_0$ with n iid components distributed as $(1-\epsilon)\delta(x_0)+\frac{\epsilon}{2}\delta(x-1)+\frac{\epsilon}{2}\delta(x+1)$. Generate m noisy measurements $\underline{y} = \frac{1}{\sqrt{m}}\tilde{A}\underline{x} + \underline{z}$ where $\tilde{A}_{ai}$ are iid uniform in $\{+1, -1\}$ and z_a are iid Gaussian zero mean and variance σ^2.

Consider the AMP iterations

$$\begin{cases} \hat{x}_i^{(t+1)} = \eta(\hat{x}_i^{(t)} + \frac{1}{\sqrt{m}} \sum_{b=1}^{m} \tilde{A}_{bi} r_b^t; \theta^{(t)}), \\ r_a^{(t)} = y_a - \frac{1}{\sqrt{m}} \sum_{j=1}^{n} \tilde{A}_{aj} \hat{x}_j^{(t-1)} + r_a^{t-1} \frac{\|\hat{x}^{(t)}\|_0}{m}, \end{cases}$$

with the choice $\theta^{(t)} = \alpha \|\underline{r}^{(t)}\|_2/\sqrt{m}$. In class we derived through heuristic means that the i-th component, given $\underline{x}_0$, of the un-thresholded estimate

$$\hat{x}_i^{(t)} + \frac{1}{\sqrt{m}} \sum_{b=1}^{m} \tilde{A}_{bi} r_b^{(t)},$$

has Gaussian statistics. The mean is x_{0i} and the variance $\sigma^2 + (\tilde{\tau})^{(2)}$ where $(\tilde{\tau})^{(2)} = \|\underline{x}^{(t)} - \underline{x}_0\|_2^2/n$.

Perform an experiment to check this numerically. Compute also the statistics of the un-thresholded estimate for the IST iterations, i.e. when the Onsager term $r_a^{t-1} \frac{\|\hat{x}^{(t)}\|_0}{m}$ is removed. Compare the two histograms.

Indications: Fix a signal realization $\underline{x}_0$. Try $n = 4000$, $m = 2000$, $\epsilon = 0.125$ and 40 instances for A and $\underline{z}$. Try various values for σ and α. Look at the i-th components of the un-thresholded estimate for components such that say $x_{0i} = +1$ (or -1, or 0).

14 K-SAT: Unit Clause Propagation and the Wormald Method

The satisfiability problem is considerably more difficult to analyze than either coding or compressive sensing. One reason for this difficulty is that random K-SAT is not an inference problem. Indeed, in the regime where a random formula is SAT with high probability (i.e., in the regime where the number of clauses per Boolean function is sufficiently small) there are exponetially many solutions contrary to coding or compressive sensing where we typically only have one valid solution. At first we might guess that this makes the problem easy: We are not asking for a *particular* solution – *any* solution will do! But in fact it is exactly this lack of uniqueness which makes the problem hard.

Why does this non-uniqueness cause trouble? Pick a specific Boolean variable. From the perspective of this variable this means that there are typically solutions for which this variable takes on the value 0 but also solutions for which it takes on the value 1. In fact, of the exponentially many solutions there are typically roughly equally many of either type. So even if the message-passing algorithm succeeded in computing the marginals of all bits correctly (here we assume that we put a uniform measure on all solutions and compute the marginal wrt this measure) all these marginals would we uniform and we cannot extract from them a globally valid solution. Therefore a straighforward application of a message-passing algorithm does not work. A new ingredient is needed.

One approach is quite natural given the above description. Assume for a second that message-passing is capable of accurately computing marginals. Then we can proceed as follows. Compute the marginal for one variable. As long as this marginal does not put all mass on either 0 or 1 it means that there are solutions which take on the value 0 as well as solutions which take on the value 1 for this variable. So in this case choose any value for this variable and reduce the formula, by eliminating this variable and all clauses which are now satisfied. This reduction is called the *decimation* step. If the marginal puts all its mass on 0, then pick the value 0, and if it puts all its mass on 1 then choose 1. Again, decimate. It is clear that this procedure will succeed in finding a satisfiable formula if one exists.

The above description assumed that message-passing is capable of exactly computing the marginals. Since this might not be the case we proceed slightly differently. Compute the marginals of all variables. Then pick a variable with maximal bias and decimate according to this bias. The hope is that by picking

variable with maximal bias we minimize the chance of making a mistake. This will be true as long as the message-passing algorithm predicts the marginals with reasonable accuracy. The above idea is what is used in *BP-guided* decimation. We will talk in more detail about this algorithm in the next chapter. Unfortunately, currently there does not exist a rigorous analysis for this algorithm. We consider a simpler algorithm in this chapter and show how to analyse it rigorously.

As we mentiond before, the K-SAT problem is the most difficult of our three running examples. Even very basic questions, like the *existence* of a SAT/UNSAT threshold, are currently not settled rigorously. We therefore will not be able to give a complete "solution" to this problem. The literature on this problem splits into two categories. On the one hand there are rigorous results typically concerning lower and upper bounds on the threshold, thresholds for some simple algorithms, as well as some basic structural properties of the problem. On the other hand, there are statistical physics calculations which make much more precise predictions and suggests sophisticated algorithms but which are not rigorous.

The aim of the current chapter is to introduce and rigorously analyse a very simple algorithm, called the *unit-clause propagation* algorithm. This algorithm has a somewhat mediocre performance, i.e., the threshold up to which it works is much below the actual SAT/UNSAT threshold as predicted by statistical physics. But it is relatively easy to analyze and it will give us the excuse of introducing a very powerful general machinery of analyzing such types of processes, called the *Wormald* method. In the next chapter we will then introduce a much more powerful message-passing algorithm based on belief-propagation and decimation. This algorithm has significantly better performance but currently no rigorous analysis exists.

Before we start with our analysis we give a quick tour of what is known about the problem. Readers, who are mostly interested in techniques, and not so much in the problem itself, can skip the next section.

14.1 A Brief Overview

As we mentioned earlier, satisfiability was the first problem proved to be NP-complete. Practically speaking, this means that there is no known algorithm which can efficiently decide for all SAT formulas if a satisfiable assignment exists or not and it is doubtful that such an algorithm can be found. Here, by efficient algorithm we mean an algorithm whose running time is polynomial in the number of Boolean variables.

The preceding paragraph concerns the *worst case*, i.e., algorithms that must succeed *always*. An alternative approach is to look at suitably defined *random* instances and to ask that the algorithm succeeds with high probability. For instance, suppose we construct a K-SAT formula by choosing each of the clauses uniformly at random from the set of all possible K-clauses. Hence, rather than considering deviously designed opponents (formulas), we are given an *ensemble*

of formulas, i.e., a set of formulas endowed with a probabilistic structure. We can now ask how hard it is to decide for a *typical* formula. In the following, we introduce the most famous of such probabilistic ensembles, namely the K-SAT ensemble.

Consider N Boolean variables and $M = \lfloor \alpha N \rfloor$ clauses of length K. The number α is positive and real and is called the *clause density*. To choose an instance from the K-SAT ensemble, we proceed as follows. Each of the M clauses picks uniformly at random a subset of length K of the variables and flips a fair coin to decide whether or not to negate each variable. Each of the above steps are taken independently of each other. The above procedure puts a uniform distribution on the set of all K-SAT formulas. In the following, we use $\mathrm{SAT}(N, K, \alpha)$ to denote the ensemble of random K-SAT formulas with size N and density α.

Due to its simple probabilistic structure and the importance of the satisfiability problem, the K-SAT ensemble has become a central topic of collaborations between computer scientists, mathematicians and statistical physicists. As we will see later, random K-SAT formulas enjoy a number of intriguing properties, some of which have been proven rigorously, but many of which are still awaiting a mathematical proof.

Most of the ideas and intuitions about this ensemble have been extended to other constraint satisfaction problems such as graph coloring (COL). One can argue whether or not random ensembles are good models for the highly structured SAT formulas which one finds in engineering and in the "real world." However, it is worth mentioning that random K-SAT instances are computationally hard for a certain range of densities, and this makes them a popular benchmark for testing and tuning SAT algorithms. In fact, some of the better practical ideas in use today come from insights gained by studying the performance of algorithms on random K-SAT instances [**?**].

We proceed by a brief detour of the current state of the art for the K-SAT problem.

The Threshold Conjecture

Pick a random formula from the K-SAT ensemble. What is the probability that such a formula is satisfiable? A moment of thought shows that this probability is a non-increasing function of α. Also, for small α we expect that most of the formulas are satisfiable whereas for α tending to infinity we expect most of the formulas to be un-satisfiable. What more can we say? In particular, what happens when the size of these formulas grows unbounded, i.e., when $N \to \infty$? Numerical experiments, physical arguments (as we will see later) as well as the experience from simpler constraint satisfaction problems sugggest that when the density crosses a critical threshold, these formulas undergo a *phase transition*. More precisely, as we increase α from zero to infinity the probability transitions from being almost certainly satisfiable to almost certain unsatisfiable and it does so in a jump at one *critical* value of α. Despite all evidence and effort, the conjecture

in this strong form is yet unproved for $K \geq 3$, and hence has remained as a conjecture known as the *satisfiability conjecture.*

Conjecture 14.1.1 (The Satisfiability Conjecture) For $K \geq 2$, there exists a constant $\alpha_s(K)$ such that the following holds

$$\lim_{N\to\infty} \Pr\{\text{SAT}(N, K, \alpha) \text{ is satisfiable}\} = \begin{cases} 1 & \text{if } \alpha < \alpha_s(K), \\ 0 & \text{if } \alpha > \alpha_s(K). \end{cases} \tag{14.1}$$

For $K = 2$, the satisfiability conjecture is known to be true and we have $\alpha_s(2) = 1$ [**?**]. The following theorem is the closest we know regarding the existence of such a threshold.

THEOREM 14.1 (Friedgut [**?**]) *For $K \geq 2$ there exists a sequence of numbers $\alpha_s(K, N)$ such that for all $\epsilon > 0$*

$$\lim_{N\to\infty} \mathbb{P}\{F(N, N(\alpha_s(N, K) - \epsilon)) \textit{ is SAT}\} = 1,$$
$$\lim_{N\to\infty} \mathbb{P}\{F(N, N(\alpha_s(N, K) + \epsilon)) \textit{ is SAT}\} = 0.$$

Theorem 14.1 comes very close to proving the satisfiability conjecture except that the sequence $\alpha_s(K, N)$ is not known to converge to a well-defined limit. In particular, there remains the possibility that such a sequence oscillates in a small window and hence may not converge. From now on, we let $\alpha_s(K)$ denote both the satisfiability threshold from Conjecture 14.1.1 and also the threshold sequence of Theorem 14.1, and leave the corresponding interpretation to the interest of the reader.

The consequences of Theorem 14.1 are not confined merely to the satisfiability conjecture. Another main application of this theorem is in providing bounds on $\alpha_s(K)$ in the following way. Suppose there exists a method that proves for some density $\alpha_{\text{method}}(K)$,

$$\lim_{N\to\infty} \Pr\{\text{SAT}(N, K, \alpha_{\text{method}}(K)) \text{ is satisfiable}\} \geq C, \tag{14.2}$$

where C is a positive constant. Then, from Theorem 14.1 we conclude that for any $\alpha \leq \alpha_{\text{method}}(K)$ we have

$$\lim_{N\to\infty} \Pr\{\text{SAT}(N, K, \alpha) \text{ is satisfiable}\} = 1.$$

In particular, this would show that $\alpha_s(K) \geq \alpha_{\text{method}}(K)$. Similarly, if $\alpha_{\text{method}}(K)$ is such that the inequality (14.2) holds in the opposite direction, then the probability that a random formula is satisfiable at densities above $\alpha_{\text{method}}(K)$ tends to 0 and we obtain that $\alpha_s(K) \leq \alpha_{\text{method}}(K)$.

This consequence of Theorem 14.1 has been the main venue for providing lower bounds on $\alpha_s(K)$. We now proceed by reviewing various methods and bounds on the threshold.

Various Bounds and the Asymptotic Behavior of the Threshold

Let us begin by a simple, but important, upper bound. For a random K-SAT formula F we denote by $X(F)$ its number of satisfying assignments (if $X(F)$ is zero then the formula is un-satisfiable). It is an easy exercise to show that

$$\mathbb{E}[X] = 2^N(1 - \frac{1}{2^K})^M.$$

As a result, by noticing $M = N\alpha$, if we choose

$$\alpha > \frac{-\ln 2}{\ln(1 - \frac{1}{2^K})},$$

then the value of $\mathbb{E}[X]$ is exponentially small in N and hence by an application of the Markov inequality we deduce that the probability of satisfiability is exponentially small. We thus have

$$\alpha_s(K) \leq \frac{-\ln 2}{\ln(1 - \frac{1}{2^K})} \leq 2^K \ln 2 - \frac{\ln 2}{2} - O(2^{-K}). \tag{14.3}$$

The above method, which is based on the first moment of X is called the *first moment* method. In fact, this simple upper bound can be made slightly sharper [**?**, **?**]

$$\alpha_s(K) \leq 2^K \ln 2 - \frac{1 + \ln 2}{2} - o(1), \tag{14.4}$$

where the $o(1)$ term is asymptotic in K. To obtain a lower bound, the *second moment* can be used [**?**, **?**]. The idea is that by an application of the Cauchy-Schwarz inequality we can show that

$$\Pr(X > 0) \geq \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}. \tag{14.5}$$

Now, if we find densities α for which the value $\frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}$ is bounded by a constant, it is immediate that such a value of α is a lower bound for $\alpha_s(K)$. However, on the negative side, for the choice of $X = X(F)$ to be the number of solutions, it can be shown that for any value of α, the quantity $\frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}$ decays to 0 by N. In other words, the number of solutions does not concentrate around its average. On the positive side, one can choose other candidates for X, rather than the number of solutions, to plug into (14.5). For instance, instead of giving an equal weight to all solutions of a formula F (as done in counting the number of solution), one can assign different weights to different solutions. This is called the *weighted second order method*. Using this method, it can be shown [**?**] that

$$\alpha_s(K) \geq 2^K \ln 2 - (K + 1)\frac{\ln 2}{2} - 1 - o(1). \tag{14.6}$$

Very recently, by a new version of the weighted second order method, a new lower bound has been obtained in [**?**]

$$\alpha_s(K) \geq 2^K \ln 2 - \frac{3 \ln 2}{2} - o(1). \tag{14.7}$$

K	3	4	5	7	10
Upper bound from (14.3)	5.19	10.74	21.83	88.37	709.44
Best upper bound [**?**]	4.51	10.23	21.33	87.88	708.94
Lower bound from [**?**]	2.68	7.91	18.79	84.82	704.94
Best algorithmic bound	3.52	5.54	9.63	33.23	172.65

Table 14.1 Best known rigorous bounds for the location of the satisfiability threshold $\alpha_s(K)$ for some small values of K. The last row gives the largest density for which a polynomial-time algorithm has been proven to find satisfying assignments.

To summarize: for large K we have

$$2^K \ln 2 - \frac{3 \ln 2}{2} - o(1) \leq \alpha_s(K) \leq 2^K \ln 2 - \frac{1 + \ln 2}{2} - o(1), \tag{14.8}$$

where the $o(1)$ term is asymptotic in K. These bounds indicate that for large values of K, the value of $\alpha_s(K)$ is just a small constant away from $2^K \ln 2$. For smaller values of K, the bounds derived from these methods are given in Table 14.1.

A different venue to find lower bounds is to provide algorithms capable of solving a random formula with a positive probability. We will have more to say about these algorithms and the methods used to analyze them later. In a nutshell, most of these algorithms act in the following way. Given a random formula, they set the variables one at a time using heuristics that use very little, and completely local, information about the variable-clause interactions. Of course, such a confinement is also what enables their analysis. Table 14.1 contains the best such algorithmic lower bounds from [**?**] and [**?**].

Outline

We will see later on that the "real" 3-SAT threshold is around $\alpha = 4.26$. This threshold is currently not provable but only "computable" by statistical physics calculations. If we use BP-guided decimation, we will find an algorithmic threshold of $\alpha_{\mathrm{BP}} = 3.86$. Even this threshold can currently only be asserted by large-scale simulations or by statistical physics calculations.

The aim of this lecture is to derive a lower bound which can be asserted rigorously. We will do so by analyzing a very simple algorithm, called unit-clause propagation (UCP). As we will see, it has a threshold of $\alpha = \frac{8}{3}$. This is not the best known lower bound. More sophisticated algorithms have been analyzed and yield a threshold of $\alpha = 3.52$. But these algorithms are considerably harder to analyze.

14.2 The Unit-Clause Propagation Algorithm

Let us now come back to the main object of study for the current chapter. We will introduce and anlyse a simple algorithm to solve K-SAT formulas. The algorithm does not have record-shattering performance. But it is natural can can be analysed by a standard and important method, called the *Wormald* method.

The Unit Clause propagation algorithm, or UC for short, is a (randomized) algorithm which sets one variable at a time. Compared to the DPLL algorithm, the UC algorithm never backtracks. Once a variable is fixed, the value stays fixed and is never changed. In brief, the algorithm works as follows: Represent a K-SAT formula in the usual way by a bipartite graph G consisting of N literals, or variable nodes, and $M = N\alpha$ clauses, or check nodes. The algorithm starts with G and in each step removes some nodes from the graph. In more detail, the UC algorithm consists of two main steps:

- *Free step:* If there does not exist a check node (clause) in the graph of degree one we perform a *free* step: Choose a variable uniformly at random and set its value uniformly at random to either 0 or 1. Remove the chosen variable node as well as any check node corresponding to a clause which is now fulfilled through the choice of the value, as well as all edges emanating from any of the removed nodes.
- *Forced Step:* If there exists a check node (clause) of degree one we perform a *forced* step: Choose a check node of degree one uniformly at random from all such check nodes. Set the value of the adjacent variable node to the unique value which fulfills the clause (hence the name "forced"). Remove from the graph the check node, the variable, all further check nodes which in addition might now be fulfilled, as well as all edges emanating from any of the removed nodes.

It is easy to see that the UC algorithm fails in finding a solution if only it generates a clause of degree 0 at some point in the course of the algorithm. In fact, once a 0-clause appears, the algorithm can halt and return a message "unable to find a solution."

The progress of UC algorithm can be predicted in terms of the solution of a set of differential equations. This method, called the Wormald method, is broadly applicable. Therefore, in the next section we describe this technique in general, before coming back to the analysis of the UC algorithm in the subsequent section.

14.3 The Wormald Method

A Simple Example

Let us start with a very simple example to illustrate the idea. Consider N particles in a box of volume V. Assume that time is discrete and takes integer values.

Assume that at each time instant and for each pair of particles (i, j) present, the probability that these two particles annihilate each other is equal to

$$\frac{1}{V^2} = \frac{N^2}{V^2}\frac{1}{N^2} = \frac{\rho^2}{N^2},$$

where ρ is the initial density of particles. Let $N(t)$ denote the number of particles which are left at time t, with $N(0) = N$. How will the number of particles evolve? We have the relationship

$$N(t+1) = N(t) - 2\sum_{(i,j)} \mathbb{1}_{\{(i,j) \text{ is annihilated between } t \text{ and } t+1\}}.$$

The evolution of this process is of course stochastic, but it is easy to write down the expected progress in one time step given the current state. We have

$$\begin{aligned}\mathbb{E}[N(t+1) \mid N(t)] &= N(t) - 2\frac{N(t)(N(t)-1))}{2}\frac{\rho^2}{N^2} \\ &= N(t) - \rho^2\frac{N(t)(N(t)-1))}{N^2}.\end{aligned}$$

This means that

$$\mathbb{E}[N(t+1) - N(t) \mid N(t)] = -\rho^2\frac{N(t)(N(t)-1))}{N^2}.$$

Assume that the process evolves exactly according to its expected progress. This means that we drop the expectations and the conditioning. This gives us

$$N(t+1) - N(t) = -\rho^2\frac{N(t)(N(t)-1))}{N^2} \approx -\rho^2\frac{N(t)^2}{N^2}.$$

Now set $t = \tau N$, where $\tau \in \mathbb{R}^+$ so that $N(t) = N(\tau N)$. Further, scale $N(t)$ by the initial number of particles, i.e., write $N(N\tau) = Nn(\tau)$. We can then write

$$Nn(\tau + 1/N) - Nn(\tau) \approx -\rho^2\frac{n(\tau)^2}{n(0)^2}.$$

With $N = \frac{1}{d\tau}$ this leads us the consider the differential equation

$$\frac{dn(\tau)}{d\tau} = -\rho^2 n(\tau)^2,\ n(0) = 1.$$

This differential equation has the solution

$$n(\tau) = \frac{1}{\rho^2(\tau + \frac{1}{\rho^2})},$$

which is best seen by direct verification. If we go back to $N(t)$ then we see that according to this model we have

$$N(t) = \frac{1}{\frac{t}{V^2} + \frac{1}{N}}.$$

In the above derivation we have waved our hands like a drunken sailor. In particular, we have replace what by its very nature was a stochstic process by a

deterministic description. Clearly, this cannot be strictly correct. But one might hope that the behavior of specific instances of $N(t)$ are "close" to this deterministic solution. Indeed, this is correct, as we will see in the next section.

The Wormald Theorem

There are myriads of versions of increasing sophistication. We will be content with stating and applying one particular incarnation. In the computer science literature the basic approach is typically referred to as the *Wormald* method. In the economics literature it is sometimes called the *Kurtz* method. Although perhaps phrased less formally, the physics community has applied this techniques for an even longer time.

THEOREM 14.2 (Wormald) *Let $Y_i^{(n)}(t)$ be a sequence (indexed by n) of real valued random processes, $1 \le i \le k$, where k is fixed, so that for all $1 \le i \le k$, all $0 \le tm(n)$, and all $n \in \mathbb{N}$*

$$|Y_i^{(n)}(t)| \le Bn, \text{ for some constant } B.$$

- *Let $H(t)$ denote the history up to time t, i.e., $H(t) = \{\underline{Y}^{(n)}(0), \ldots, \underline{Y}^{(n)}(t)\}$.*
- *Let $I = \{(y_1, \ldots, y_k) : \mathbb{P}\{\underline{Y}^{(n)}(0) = (y_1 n, \ldots y_k n)\} > 0, \text{for some } n\}$.*
- *Let D be some open connected bounded set containing the closure of $\{(0, y_1, \cdots, y_k) : (y_1, \cdots, y_k) \in I\}$.*
- *Let $f_i : \mathbb{R}^{k+1} \to \mathbb{R}$, $1 \le i \le k$:*
  1. *(Trend) For all i and uniformly for all $t < m$*

$$\mathbb{E}[Y_i(t+1) - Y_i(t) \mid H(t)] = f_i(\frac{t}{n}, \frac{Y_1^{(n)}(t)}{n}, \cdots, \frac{Y_k^{(n)}(t)}{n}) + o(1).$$

  2. *(Tail) For all i and uniformly for all $t < m$*

$$\Pr(|Y_i^{(n)}(t+1) - Y_i^{(n)}(t)| > n^{\frac{1}{5}} \mid H(t)) = o(n^{-3}).$$

  3. *(Lipschitz) For each i, the function f_i is a Lipschitz continuous on D. That is, there exists a constant L such that for any pair $x, y \in D$,*

$$|f_i(x) - f_i(y)| \le L||x - y||_1 = L \sum_{i=1}^{k} |x_i - y_i|.$$

Then we have:

(a)(Differential equation)For $(0, \hat{z}_0, \cdots, \hat{z}_k) \in D$ the system of differential equations

$$\frac{dz_i}{d\tau} = f_i(\tau, z_1, \cdots, z_k), \;\; 1 \le i \le k,$$

has a unique solution in D for $z_i : \mathbb{R} \to \mathbb{R}$ passing through $z_i(0) = \hat{z}_i$, $1 \le i \le k$, and which extends to points arbitrarily close to the boundary of D.

(b)(Concentration) Almost surely

$$Y_i^{(n)}(t) = z_i(\frac{t}{n})n + o(n),$$

uniformly for $0 \leq t \leq \min\{m(n), n\tau_{max}\}$ *and for each* i*, where* $z_i(\tau)$ *is the solution in (a) with* $\hat{z}_i(0) = \frac{Y_i^{(n)}(0)}{n}$ *and where* τ_{max} *is the maximum time until the solution can be extended before reaching in* L_1*-distance* ϵ*-close to the boundary of* Dm *where* ϵ *is arbitrary but strictly positive.*

14.4 Analysis of the UC Algorithm

Let us begin by introducing the necessary notation for the analysis:

- We let t denote current "time" of the algorithm. The term "time" means the total number of variables fixed so far.
- We let $C_i(t)$, $i \in \{1, \cdots, K\}$, denote the the number of clauses of degree i that the remaining formula at time t contains.

One important fact for the analysis of such algorithms is the so called *uniform randomness property*. In brief, this property means that at any time t, each clause of length i in the remaining formula is uniformly distributed among all the possible clauses of length i. In other words, conditioned on the number of variables and clauses of different length, the formula is uniformly random. An intuitive justification for the randomness property in our case stems from the fact that at any step (free or forced) in the UC algorithm, no information, what-so-ever, can be deduced about the structure of the remaining formula. The exact proof of the uniform randomness property in our case can be easily deduced from [**?**, Lemma 3].

We are now ready to write the set of differential equations for C_i's. Let us for simplicity assume $K = 3$ and bear in mind that for general K the analysis follows along the same path. Recall that we start with N Boolean variables. In the process we consider, at each step in the process we remove exactly one variable node. Let time t be discrete and increasing, starting at $t = 0$. Let $N(t)$ be the number of variables which are left at time t. Then we have $N(t) = N - t$.

We start with $C_3(t)$. At any time t, a variable is chosen among the $N - t$ remaining ones and is given a permanent value. This variable can either be chosen due to a forced step or due to a free step. Note that in both cases the degree distribution of the chosen variable is essentially the same. In more detail. At time t there are $N - t$ variables left and $C_1(t) + 2C_2(t) + 3C_3(t)$ edges left. Further, each edge is connected uniformly at random to each variable node. So the distribution of the number of edges for a randomly chosen variable node is equal to $C_1(t) + 2C_2(t) + 3C_3(t)$ independent Bernoulli trials with success probability $1/(N - t)$. In particular, in expectation, a randomly chosen variable node has $\frac{C_1(t)+2C_2(t)+3C_3(t)}{N-t}$ edges connected to it. Even more, in a forced step,

when we consider a random variable which is connected to a clause of degree 1, the expected number of *additional edges* is $\frac{C_1(t)+2C_2(t)+3C_3(t)-1}{N-t}$, which for large N is essentially the same number. For this reason we can treat both cases, namely free and forced step in the same way.

Now consider what happens when we fix the value of the variable. This variable is connecected in expectation to

$$\frac{C_1(t)+2C_2(t)+3C_3(t)-1}{N-t}\frac{3C_3(t)}{C_1(t)+2C_2(t)+3C_3(t)-1}=\frac{3C_3(t)}{N-t}.$$

clauses of degree 3. Therefore

$$\mathbb{E}[C_3(t+1)-C_3(t)|C_3(t)]=-\frac{3C_3(t)}{N-t}. \tag{14.9}$$

Among the 3-clauses that contain the chosen variable, half of them (in expectation) are satisfied and hence removed from the formula and the other half are shortened to two 2-clauses. We claim that

$$\mathbb{E}[C_2(t+1)-C_2(t)|C_3(t),C_2(t)]=\frac{3C_3(t)}{2(N-t)}-\frac{2C_2(t)}{(N-t)}. \tag{14.10}$$

We have already seen where the first term on the right comes from. The second term has a similar interpretation. Each variable node is connected in expectation to

$$\frac{C_1(t)+2C_2(t)+3C_3(t)-1}{N-t}\frac{2C_2(t)}{C_1(t)+2C_2(t)+3C_3(t)-1}=\frac{2C_2(t)}{N-t}$$

clauses of degree 2. In expectation, half of them will be fulfilled through the choice of the value of the variable node, and the other half will become 1-clauses.

Finally, look at the evolution of degree-1 clauses. We claim that we have

$$\mathbb{E}[C_1(t+1)-C_1(t)|C_3(t),C_2(t),C_1(t)]=\frac{C_2(t)}{N-t}-\mathbb{1}_{\{C_1(t)>1\}}. \tag{14.11}$$

This equation is somewhat more subtle. If at time t there are degree-1 clauses then we will eliminate one for sure. In this case we will also add in expectation $\frac{C_2(t)}{N-t}$ new ones. If on the other hand we do not have a degree-1 node then we only add in expectation $\frac{C_2(t)}{N-t}$ such clauses.

Note that in order to predict the evolution of $C_3(t)$ and $C_2(t)$ we only need to know $(C_3(t),C_2(t))$ but not $C_1(t)$. Therefore, let us just solve the differential equation for these two higher degrees.

At this step we need to check that all the conditions of the Wormald theorem are fulfilled. We leave this task to the reader. Most conditions are trivially fulfilled. E.g., the process starts in a bounded state and all quantities decrease and stay non-negative. Hence the process is trivially bounded. Also the initial condition is determinstic. Further, steps are small with high probability, so the tail condition is also easy to check. Further, the trend condition is also trivially fulfilled. The only condition which needs checking is that the function which gives the trend is Lipschitz. A quick check shows that this is true until almost

towards the end of the algorithm. At the very end, the denominator $1-\tau$ tends to zero, which causes problems. So according to the Wormald theorem, actual instances will behave close to the prediction given by the solution of the differential equation up to any fixed time strictly bounded away from $\tau = 1$.

As a next step let us write down the differential equations corresponding to this evolution. We get, using $\tau \equiv \frac{t}{N}, c_2(\tau) \equiv \frac{C_2(t)}{N}, c_3(\tau) \equiv \frac{C_3(t)}{N}$,

$$\frac{\mathrm{d}c_3(\tau)}{\mathrm{d}\tau} = -3\frac{c_3(\tau)}{1-\tau}, \tag{14.12}$$

$$\frac{\mathrm{d}c_2(\tau)}{\mathrm{d}\tau} = \frac{3}{2}\frac{c_3(\tau)}{1-\tau} - 2\frac{c_2(\tau)}{1-\tau}, \tag{14.13}$$

with initial conditions

$$c_3(0) = \alpha, \tag{14.14}$$

$$c_2(0) = 0. \tag{14.15}$$

The solution to the above set of equations can easily be found to be

$$c_3(\tau) = \alpha(1-\tau)^3,$$
$$c_2(\tau) = \frac{3}{2}\alpha\tau(1-\tau)^2.$$

Figure **??** compares the solutions of the differential equations with their counterpart in performing the UC algorithm over an actual random K-SAT formula.

Now let us see what this differential equation tell us about the threshold of this algorithm. We claim that the threshold is $\alpha^* = \frac{8}{3}$. Let us first show that it is at most $\frac{8}{3}$. Assume that we are operating with a higher value of α. Note that

$$\frac{C_2(t)}{N-t} = \frac{3}{2}\alpha\frac{t}{N}(1-\frac{t}{N}) + o(1), \tag{14.16}$$

Note that at time $t = \frac{1}{2}$ we have according to this prediction $\frac{C_2(t)}{N-t}\Big|_{t=\frac{N}{2}} = \frac{3}{8}\alpha + o(1)$. But note that $\frac{C_2(t)}{N-t}$ is the density of 2-clauses at time t. In other words, if we choose α greater than $\frac{8}{3}$ then at this point in time the density of 2-clauses is above 1. Using the uniform randomness property we see that what we would have at this point is a random 2-SAT formula with density larger than 1 with some additional 3-clauses. But such a formula is unsatisfiable with high probability. So in particular the UCP cannot possible succeed.

Now let us prove that if we pick α strictly smaller than $\frac{8}{3}$ then with high probability the algorithm succeeds. Recall that the algorithm succeeds if and only if no degree-0 clause is prodcued at any point in time. Consider the equation for the evolution of the degree-1 clauses. Note that $\frac{2C_2(t)}{N-t}$ is not only the 2-clause density, but it is also the expected number of new degree-1 clauses which are generated at time t. If this number is strictly less than 1 over the whole time interval then with high probability $C_1(t)$ is at any point in time at most a constant but never becomes linear in N. This means that the chance that when

we set a variable that this variable is connected to two such degree-1 clauses of opposite sign vanishes. Hence, we never create a degree-0 clause.

Some care has to be taken to make this argument completely rigorous. In particular, as we discussed we can only guarantee the accuracy of the prediction up to a time very close to $\tau = 1$. So we need in addition an argument which guarantees that the remaining formula is satisfiable with high probability. This is somewhat analogous to decoding, where we sometimes need an argument which guarantees that we can decode all bits assuming that we have decoded most of them. The argument for the present case goes as follows. If we look at the solution of the differential equation, we see that if we run the algorithm long enough for $\alpha < \frac{8}{3}$ then there is a time strictly before $\tau = 1$ where the sum of the 2-density plus the 3-density is strictly less than 1. We can now argue as follows. Drop a random variable from each 3-clause. Then the resulting formula is satisfiable with high probability.

Problems

14.1 (Peferential Attachment). The purpose of this homework is to use the Wormald method to study a model for "preferential attachment." Consider n nodes. Initially all nodes have degree 0. Assume that we allow a maximum degree of $d_{\max}$. We proceed as follows. At every step pick two nodes from the set of all nodes which have degree at most $d_{\max} - 1$. Rather than picking them with uniform probability pick them proportional to their current degree. More precisely, assume that at time t you have $D_i(t)$ nodes of degree i. Then pick a node of degree i with probability

$$\begin{cases} \frac{D_i(t)}{\sum_{j=0}^{d_{\max}-1} d_j(t)}, & 0 \le i < d_{\max}, \\ 0, & i = d_{\max}. \end{cases}$$

Initially, we have $D_0(t = 0) = n$ and $D_i(t = 0) = 0$ for $i = 1, \cdots, d_{\max}$. Note that at time $t = nd_{\max}/2$ all nodes will have maximum degree. Pick $d_{\max} = 4$.

(i). Write down the set of differential equations for this problem. Are the conditions fulfilled?

(ii). Plot the evolution of the degree distribution as a function of the normalized time for $\tau = t/n \in [0, d_{\max}/2]$

HINT: In general one cannot expect to solve the system of differential equations analytically. But it is typically easy to solve them numerically. Here is how you do it in Mathematica. The following lines set up the differential equation we discussed in class and plots the solution.

```
(* initial conditions      *)
cnds = {n[0] == 1};
(* set of diff equations   *)
eqns = {n'[u] == - rho n[u]^2};
(* put the two together    *)
eqnspluscnds = Flatten[Join[eqns, cnds]];
```

```
(* solve up to this point  *)
umax=10;
(* solve the diff equation *)
sol=Flatten[NDSolve[eqnspluscnds, {n}, {u, 0, umax} ]]
(* plot the solution       *)
Plot[Evaluate[{n[u]} /. sol], {u, 0.0, umax}]
```

If you have more than one variable then it is convenient to call them

```
d[0][u], d[1][u], d[2][u], ...
```

In this case you might have something like

```
cnds = {d[0][0] == ..., d[1][0]==..., ...};
eqns = {d[0]'[u] == ..., d[1]'[u]==..., ...};
eqnspluscnds = Flatten[Join[eqns, cnds]];
umax=...;
sol =
Flatten[NDSolve[eqnspluscnds, {d[0], d[1], ...}, {u, 0, umax} ]]
Plot[Evaluate[{d[0][u], d[1][u], ...} /. sol], {u, 0.0, umax}]
```

15 K-SAT: BP-Guided Decimation

In the preceding chapter we have introduced and analysed a very simple algorithm, called unit clause propagation. This analysis established a non-trivial lower bound for the SAT/UNSAT threshold and this threshold is in particular algorithmic, i.e., we have an efficient algorithm which works up to this threshold. On the downside, the UCP algorithm is not very powerful and so the threshold is quite low.

The aim of the current chapter is to introduce and to discuss a more powerful algorithm, called BP-guided decimation. The basic idea is similar to what that of the UCP algorithm. At each step we pick a variable node (or several of them) and fix its value, i.e., we decimate the formula. The difference lies in how we choose the variable we decimate. In the UCP algorithm, the choice was either forced upon us or we chose randomly. In the BP-guided decimation algorithm we use the BP algorithm to guide the selection.

We first introduce a version of the algorithm which is guaranteed to succeed if the formula is satisfiable and if the factor graph corresponding to the formula is a tree. As we did for coding, we then introduce a more convenient parametrization of the messages. Finally, we apply the algorithm to formulas in the ensemble, even though of course in this case the factor graphs are far from trees. As we discussed previously, the K-SAT problem is considerably harder than either coding or compressive sensing. Many of the basic questions are still open from a mathematical point of view. E.g., currently there exists no rigorous analysis of BP-guided decimation. We will therefore have to be content with a somewhat more heuristic approach.

15.1 Simple Example

Let us start with a very simple example. Suppose we are given the formula

$$F = x_1 \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3). \tag{15.1}$$

The corresponding factor graph is shown in Figure 15.1. Dashed lines means the variable appears negated in the corresponding clause.

F is a Boolean function. However, we can slightly modify F and model it as a binary function that can take either of 0 or 1 values. In this case, we can write F

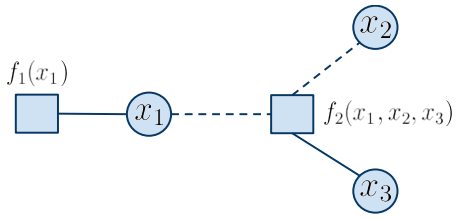


Figure 15.1 Factor graph of the equation $F = x_1 \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$

x_1	x_2	x_3	$f_1(x_1)$	$f_1(x_1,x_2,x_3)$	F
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	1	1	1

Table 15.1 Satisfiability of F, given by equation (15.1), for all possible combination of x_1, x_2 and x_3.

as the product of two other binary functions: $f_1 = x_1$ and $f_2 = \text{sign}(\overline{x_1}+\overline{x_2}+x_3)$, where sign is the normal sign function with $\text{sign}(0) = 0$.

Note that in order to see if F is satisfiable, we can compute the "partition function"

$$\sum_{x_1,x_2,x_3} f_1(x_1)f_2(x_1,x_2,x_3).$$

This is true since the partition function counts the number of satisfying configurations. Hence, if the partition function is non-zero then the formula is satisfiable. For the current case there are 3 SAT solutions. Table 15.1 illustrate the satisfiability of F for all possible combination of x_1, x_2 and x_3.

We can also look at marginals with respect to different variables, for instance

$$\sum_{\sim x_1} f_1(x_1)f_2(x_1,x_2,x_3).$$

This marginal counts the number of satisfying clauses given that x_1 has a particular fixed value. From Table 15.2 we see that $\mu(x_1 = 0) = 0$ and $\mu(x_1 = 1) = 3$; $\mu(x_2 = 0) = 2$ and $\mu(x_2 = 1) = 1$; $\mu(x_3 = 0) = 1$ and $\mu(x_3 = 1) = 2$. Note that the factor graph is a tree. Therefore BP can compute the partition function as well as the marginals *exaclty*. Table 15.2 summarizes the messages exchanged in

Iteration number	Messages
1	$\mu_{f_1 \to x_1} = f_1,\ \mu_{x_2 \to f_2} = 1,\ \mu_{x_3 \to f_2} = 1$
2	$\mu_{f_2 \to x_1} = \sum_{x_2, x_3} f_2(x_1, x_2, x_3)$

Table 15.2 Messages exchanged in each iteration of the belief propagation performed over the factor graph given in Fig. 15.1.

each iteration of the belief propagation in order to compute the marginal with respect to x_1, denoted by $\mu(x_1)$, for the factor graph given in Fig. 15.1. Let us illustrate the use of message passing rules for the derivation of $\mu(x_1)$. The first line of the table gives the initial messages at the leaf nodes. First we compute

$$\mu_{2\to 1} = \sum_{\sim x_1} f_2(x_1, x_2, x_3).\underbrace{\mu_{2\to 2}(x_2)}_{1}\underbrace{\mu_{3\to 2}(x_3)}_{1} = 4 \text{ if } x_1 = 0 \text{ and } 3 \text{ if } x_1 = 1 \tag{15.2}$$

and

$$\mu_{1\to 1} = f(x_1) = 0 \text{ if } x_1 = 0 \text{ and } 1 \text{ if } x_1 = 1 \tag{15.3}$$

Finally,

$$\mu(x_1) = \mu_{2\to 1}(x_1)\mu_{1\to 1}(x_1) = 0 \text{ if } x_1 = 0 \text{ and } 3 \text{ if } x_1 = 1 \tag{15.4}$$

which agrees with Table 15.2.

Let us summarize. The marginals count the number of satisfying solutions with a particular assignment for the given Boolean variable. If we are interested in the *fraction* of satisfying solutions with a particular assignment for the given Boolean variable we can just normalize the messages. Also, we will see shortly, that if we can accurately compute marginals, we can also find SAT assignments.

Notation: We denote clauses by $a, b, c, \ldots$ and variables by $i, j, k, \ldots$. Furthermore, we denote the neighborhood of a node x by ∂x. The same neighborhood excluding a particular node y is indicated by $\partial x \setminus y$.

Having these notations in mind, we start by modifying the message-passing rules. In the original message passing scheme, the message from variable i to clause a is given by equation (15.5).

$$\mu_{i\to a}(x_i) = \prod_{b \in \partial i \setminus a} \mu_{b\to i}(x_i) \tag{15.5}$$

However, since we are interested in the *fraction* of the solutions with $x_i = 0$ and $x_i = 1$, we require the new messages $\widetilde{\mu}_{i\to a}(x_i)$ to satisfy the following equation.

$$\widetilde{\mu}_{i\to a}(x_i = 0) + \widetilde{\mu}_{i\to a}(x_i = 1) = 1$$

Therefore, it is sufficient to set $\widetilde{\mu}_{i\to a}(x_i)$ according to equation (15.6).

$$\widetilde{\mu}_{i\to a}(x_i) = \frac{\mu_{i\to a}(x_i)}{\mu_{i\to a}(x_i = 0) + \mu_{i\to a}(x_i = 1)} \tag{15.6}$$

Figure 15.2 BP Guided Decimation over Trees

1. Run belief propagation on F and compute the all marginals $\mu(x_i)$ for all of the variables.
2. Pick a variable i. If $\mu(x_i = 0) > 0$ (there exists an assignement with $x_i = 0$), then:
   1Set $x_i = 0$ in all clauses.
   2Eliminate all those clauses that x_i appears negated in them.
   3Remove x_i from the other clause.
   If on the other hand $\mu(x_i = 0) = 0$ (there doesnt exist an assignmenet with $x_i = 0$), then:
   1Set $x_i = 1$ in all clauses.
   2Eliminate all those clauses that x_i appears unnegated in them.
   3Remove x_i from the other clause.
3. Repeat the process until no variables are left.

At this point, it seems as if we have to once calculate $\mu_{i\to a}(x_i)$ for $x_i = 0, 1$ and then normalize the messages. However, it is easy to show that we can directly calculate $\widetilde{\mu}_{i\to a}(x_i)$. To simplify the notations, we omit the normalization factor and write the messages as

$$\widetilde{\mu}_{i\to a}(x_i) \propto \prod_{b\in\partial i\setminus a} \widetilde{\mu}_{b\to i}(x_i). \tag{15.7}$$

15.2 From Counting the Number of Solutions to Finding a Solution

Given a SAT problem F, assume that the factor graph of F is a tree and F has a satisfying solution. Then algorithm 1 will find a solution that satisfies F.

Note that in each step of the above algorithm we must run BP. So in total we might need to run BP n times.

Terminology: Since we use belief propagation and eliminate a variable in each iteration, the algorithm is called **BP-guided decimation**.

Algorithm 1 is only guaranteed to give accurate marginals if we have a tree. But what about the more general cases? We will introduce a modified version of the above algorithm in the next section to deal with general factor graphs.

Applying BP Guided Decimation to General Factor Graphs

In this section, we apply a modified version of the BP guided decimation algorithm to general factor graphs. However, note that the graph in this section should be sparse as before.

Over a tree, the previous algorithm yields exact marginals and we can pick anyone of them in each iteration. However, in general graphs it is not the case any more. As a result and in order to deal with the inherent uncertainty in marginals, in each iteration we pick a node i such that the difference $|\mu(x_i = 0) - \mu(x_i = 1)|$ is **maximized**. This way, we hope that this node has such a clear bias that its marginals are are quite exact despite the graph not being a tree.

Figure 15.3 BP Guided Decimation over General Graphs
1. Run BP and calculate all marginals.
2. Pick a node i such that $|\mu(x_i = 0) - \mu(x_i = 1)|$ is maximized.
3. Set x_i to the most likely value, i.e. $x_i = 0$ if $\mu(x_i = 0) > \mu(x_i = 1)$ and to 1 otherwise.
4. Eliminate all clauses that the particular choice of x_i make them satisfied. Remove x_i from the other clause.
5. Recurse until all variables are eliminated.

The rest of the algorithm is the same, summarized below:
Some remarks about running BP on general graphs are in order:

- *Initialization* The typical way of initializing messages is to set all of them equal to 1/2.
- *Scheduling* In contrast to BP guided decimation over a tree, the choice of node i affect the solution and the whole algorithm. Therefore, scheduling matters. We usually use flooding as a means of scheduling. In other words, in each iteration every node sends its messages over its outgoing links.

Figure 15.4 illustrates two kinds of probabilities as a function of α (ratio of nb of clauses to variables). One can run pure BP over many instances and compute the empirical probability that it converges. This yields the upper curves in figure 15.4. For $K = 3$ we get a convergence threshold $\alpha_{\text{BP}\approx 3.86}$ and for $K = 4$ we get $\alpha_{\text{BP}} \approx 10.3$. Now, one can run BP guided decimation (algorithm 2) over many instances and derive the empirical probability of success. The corresponding threshold must in general be lower than α_{BP} since BP must at least converge after each decimation step. This empirivcal probability is given by the lower curve in figure 15.4. For $K = 3$ the threshold is approximately identical to α_{BP} but for $K = 4$ it is smaller and approximately equal to 9.3.

The actual SAT-UNSAT threshold is for $K = 3$, $\alpha_{\text{sat}-\text{unsat}} \approx 4.26$ and for $K = 4$, $\alpha_{\text{sat}-\text{unsat}} \approx 9.93$. We will see in future lectures how to obtain these thresholds by *survey propagation* algorithms.

15.3 Convenient Re-parametrization

To write down the BP equations in simple form it is convenient to use the reformulation in terms of spin variables exposed in Chapter 4. Recall that a weight $J_{ia} = +1$ (resp. -1) is associated to full (resp. dashed) edges for which x_i appears un-negated (negated) in clause a. Recall also that $s_i = (-1)^{x_i}$. With these definitions $s_i = J_{ia}$ means that the assignment s_i does not satisfy a, and $s_i = -J_{ia}$ means that it satisfies a.

We parametrize the messages as follows

$$\mu_{i\to a}(s_i = \pm J_{ia}) = \frac{1 \mp \tanh h_{i\to a}}{2}, \quad \hat{\mu}_{a\to i}(s_i = \pm J_{ia}) = \frac{1 \mp \tanh \hat{h}_{i\to a}}{2}. \quad (15.8)$$

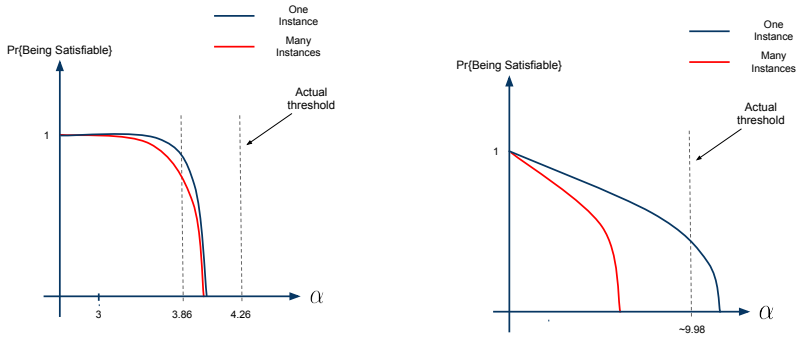


Figure 15.4 Probability of $3-SAT$ and $4-SAT$ being satisfied by BP guided decimation.

The interpretation of this notation is that $(1-\tanh h_{i\to a})/2$ is the probability that x_i/s_i has a value which *does not satisfy* the clause corresponding to node a. Similarly, $(1-\tanh \hat{h}_{i\to a})/2$ represents the probability that x_i/s_i is *not free* to be chosen arbitrarily since the clause a is not satisfied yet.

We need one more bit of notation. Consider a fixed edge ia with some edge weight J_{ia}. Let S_{ia} be the subset of variable nodes in ∂a that have the same edge type (weight) J_{ia}. Likewise, let U_{ia} be the subset of variable nodes in ∂a with a different edge type i.e., $-J_{ia}$.

The original message passing equations for messages from variable to check nodes is given by:

$$\begin{aligned}\mu_{i\to a}(s_i=\pm J_{ia}) &\propto \prod_{b\in\partial i\setminus a}\hat{\mu}_{b\to i}(s_i=\pm J_{ia})\\ &\propto \prod_{b\in S_{ia}}\hat{\mu}_{b\to i}(s_i=\pm J_{ib})\prod_{b\in U_{ia}}\hat{\mu}_{b\to i}(s_i=\mp J_{ib})\end{aligned}\tag{15.9}$$

Hence,

$$\frac{1\pm\tanh h_{i\to a}}{2}\propto\left(\prod_{b\in S_{ia}}\frac{1\pm\tanh\hat{h}_{b\to i}}{2}\right)\left(\prod_{b\in U_{ia}}\frac{1\mp\tanh\hat{h}_{b\to i}}{2}\right)\tag{15.10}$$

Taking the ratio of these two equations we find

$$h_{i\to a}=\sum_{b\in S_{ia}}\hat{h}_{b\to i}-\sum_{b\in U_{ia}}\hat{h}_{b\to i}\tag{15.11}$$

The original message passing rules for messages from constraint to variable nodes yield

$$\hat{\mu}_{a\to i}(s_i=\pm J_{ia})\propto\sum_{\sim s_i=\pm J_{ia}}f_a(s_{\partial a})\prod_{j\in\partial a\setminus i}\mu_{j\to i}(s_j)\tag{15.12}$$

As noted at the beginning of this section for $s_i = -J_{ia}$ the clause a is satisfied irrespective of other variables, i.e. $\psi(s_{\partial a}) = 1$. As a result, the sum of some products in (15.12) factorizes into

$$\hat{\mu}_{a\to i}(s_i = -J_{ia}) \propto \prod_{j\in\partial a\setminus i} \sum_{s_j} \mu_{j\to a}(s_j) \propto 1. \tag{15.13}$$

In other words $(1 + \tanh \hat{h}_{a\to i})/2 \propto 1$. Now we calculate (15.12) for $s_i = J_{ia}$. For this assignment the variable s_i can be eliminated from the kernel function f_a since this variable does not satisfy a. In (15.12) we have to sum over all assignments of remaining variables $j \in \{\partial a \setminus i\}$ such that at least one of them has value $s_j = -J_{ja}$. It is easy to see that this yields

$$\hat{\mu}_{a\to i}(s_i = J_{ia}) \propto 1 - \prod_{j\in\partial a\setminus i} \mu(s_j = J_{ja}). \tag{15.14}$$

Dividing out relations (15.12) and (**??**) allows to eliminate the normalization factors and one finds

$$\hat{h}_{a\to i} = -\frac{1}{2} \ln\left\{1 - \prod_{j\in\partial a\setminus i} \frac{1 - \tanh_{j\to a}}{2}\right\} \tag{15.15}$$

Equations (15.11)-(15.15) are the BP equations for K-SAT. The reader will appreciate the similarity with coding.

Problems

15.1 You will implement Belief Propagation (BP) for K-SAT (say $K = 3$ and $K = 4$) The first one is to find a convenient parametrization of the BP messages. This was done in class. The second is to investigate numerically the convergence of BP as a function of α (the clause density). The third is to implement a decimation algorithm that finds satisfying assignments for α not too large.

15.Belief Propagation Equations for K-SAT Go through the derivation, especially if this was not done in detail during class.

15.2 Implementation of BP] You will implement BP according to the flooding (or parallel) schedule. initialize the messages uniformly randomly in $[0, 1]$. One iteration means that you send messages from nodes to clauses and back from clauses to variables. Define the following "convergence criterion": declare that the messages have "converged" if there is an iteration number (time) $t_{\rm conv}(\delta)$ such that no messages changes by more than δ at $t_{\rm conv}(\delta)$ (take the smallest such time).

Perform the following experiment. Take 100 K-SAT instances of length say $N = 5000$ and 10000 variables and for each instance implement BP as explained above with $\delta = 10^{-2}$. If the iterations do not converge stop them at a large time say $t_{\max} \approx 1000$. When they converge, they should do so in a shorter time $t_{\rm conv}(\delta) < t_{\max}$ that does not change much with N.

Plot as a function of α the empirical probability that the iterations converge.

You should see that this probability is large for $\alpha < \alpha_{\rm BP}$ and drops abruptly around some threshold $\alpha_{\rm BP}$. For $K = 3$, $\alpha_{\rm BP} \approx 3.85$ and $K = 4$, $\alpha_{\rm BP} \approx 10.3$.

15.3 BP guided decimation] Now you will implement the following algorithm for finding SAT assignments. It uses the above BP procedure as a guide to take decisions on how to fix values for the variables. Once a variable has been fixed the K-SAT formula is suitably reduced - this step is called "decimation" - and BP is run again.

- Initialize with a K-SAT formula $\mathcal{F}$ of length N.
- For n= 1, ..., N do:
  - Run BP on an instance, as in the previous exercise (with the same convergence criterion).
  - If BP does not converge, return "assignment not found" and exit.
  - If BP converges, for each variable j compute its bias (express it in terms of $\hat{z}eta$ variables!)

$$\pi_j = \mu_j(1) - \mu_j(0) = \frac{\prod_{a\in\partial j}\mu_{a\to j}(1) - \prod_{a\in\partial j}\mu_{a\to j}(0)}{\prod_{a\in\partial j}\mu_{a\to j}(1) + \prod_{a\in\partial j}\mu_{a\to j}(0)}$$

  - Pick a variable $j(n)$ that has the largest absolute bias $|\pi_{j(n)}|$.
  - If $\pi_{j(n)} \geq 0$ fix $x_{j(n)} = 1$. Otherwise fix $x_{j(n)} = -1$.
  - Replace $\mathcal{F}$ by the K-SAT formula obtained by decimating variable $j(n)$.
- End-For
- Return all fixed variables.

Give for several values of α, the empirical success probability of this algorithm when tested over 100 instances. Compare this empirical success probability with the empirical convergence probability of the previous exercise. You should observe that $K = 3$ and $K = 4$ do not behave on the same way. Try to find an approximate threshold α_t beyond which the algorithm does not find SAT assignments.

16 Maxwell Construction

The Maxwell construction is a paradigm to guess the "true" (optimal/physical) behavior of a system from a simple model. For us the "simple model" is the description in terms of message-passing quantities and this setting is well-suited for this construction. Once the Maxwell construction has given us a guess, this guess can then often be converted into a rigorous statement. The important point here is that typically the proof uses the guess as an essential input. I.e., the Maxwell construction is typically a crucial first step in the proof.

We will discuss several instances of this paradigm in this chapter. Note that whenever this program works, then this means that the message-passing algorithm is not just a convenient low-complexity algorithm but plays a fundamental role in characterizing the problem.

16.1 The Original Maxwell Construction

The original Maxwell construction goes back to the 19th century struggle of trying to understand the liquid-vapor phase transition for simple substances (say H_2O). Quite surprisingly, even though this problem seems to have little to do with our three examples, there is a very straightforward analogy between the Maxwell construction for this problem and the Maxwell construction in our case. It is therefore worth to quickly review the problem.

Assume that we have a gas consisting of N molecules in a volume of V cubic meters under a pressure of p pascals and a temperature of T Kelvins. How are these quantities related? The *ideal* gas law states that

$$pV = NkT, \tag{16.1}$$

where k is the Boltzmann constant. The left picture in Figure 16.1 shows this relationship at different temperatures T. As one can see from this picture, as we decrease the volume, the pressure increases. The derivation of this ideal gas law is based on several simplifying assumptions. In reality the molecules[1] interact via

[1] The reader should not underestimate that the atomic and molecular constitution of matter acquired the status of scientific truth, as opposed to philosophical assumption, only in the 19th century thanks to the work of numerous chemists.

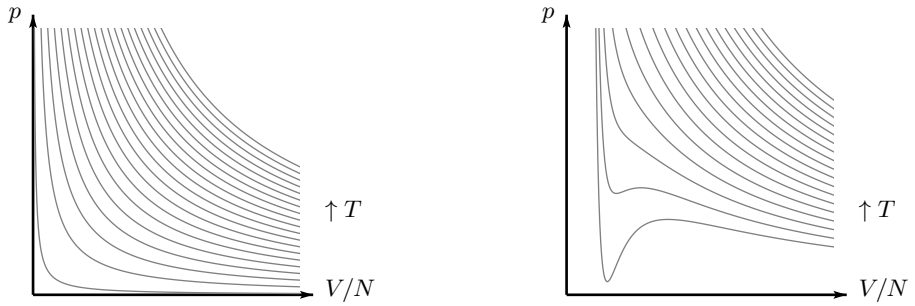


Figure 16.1 Left: Isotherms of the ideal gas equation of state. Right: Isotherms of the van der Waals equation of state. Note that below a critical temperature, the isotherms are no longer monotone.

forces of quantum mechanical origin.[2] These forces have a very short range and strong repulsive part and a weak long range attractive part. Because of the short range strong repulsion it is good model to assume that the molecules have an "effective volume". The ideal gas law simply *neglects* this effective volume as well as the attractive part of the force (so it neglects all forces hence the name ideal). The relation expressed in (16.1) is an *equation of state*, since it relates quantities that define the thermodynamic "state" of the system (namely, (p, V, T, N)).

In 1873, Johannes Diderik van der Waals derived a more accurate equation of state taking into account the non-zero effective size of the molecules as well as the weak long range attracting forces. His derivation resulted in the equation

$$(p + a\frac{N^2}{V^2})(V - bN) = NkT.$$

This equation is very similar in structure to the ideal gas law, but both the volume as well as the pressure terms are modified. The constant b takes into account the effective finite size of each molecule. Due to this finite size the *effective volume of the box* which is available to the N molecules shrinks from V to $V - bN$. The constant a takes into account attractive forces between molecules. It is assumed that these attractive forces act only between molecule of the gas but not between the wall and gas molecules. Therefore, close to a boundary, a molecule has more neighbors away from the boundary then towards the boundary and this creates an effective force "inwards," reducing the pressure of the gas. Note that the van der Waals equation is equivalent to $p = NkT/(V - bN) - a\frac{N^2}{V^2}$ so that the pressure is reduced by $a\frac{N^2}{V^2}$. The reduction is proportional to N^2 because each molecule close to the wall feels the effect of approximately N other molecules and there are of the order of N molecules close to the wall. To obtain an intensive quantity (pressure is intensive, i.e. independent of system size) we have to divide by V^2 which is the only other extensive quantity besides N. Another way to understand the form of this term is to assume that that the reduction in

[2] So it is only much later, in 1920-1930, that the true origin and proper way to model these forces was understood!

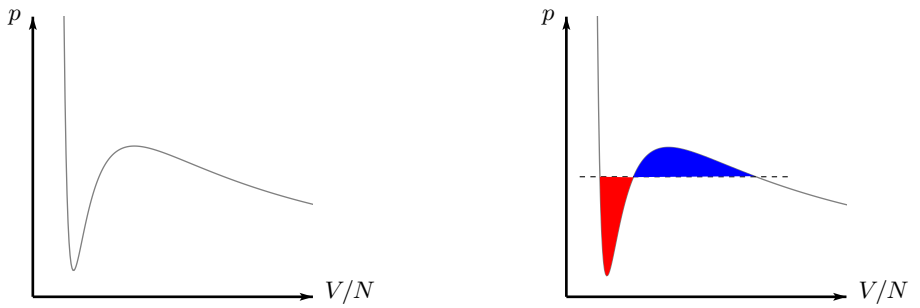


Figure 16.2 The original Maxwell construction. Left: One isotherm of the van der Waals equation of state. Right: The same isotherm, where a part of the curve is replaced by a horizontal line which is placed so that the two enclosed areas are in balance.

pressure is only a function of the density N/V close to the wall. For somewhat low densities (at least in the gas phase) one can expand this function in powers of N/V. The first order term must vanish because the attracting forces involve pairs of particles, leaving us with the second order term. Higher order terms are then neglected in the van der Waals theory.[3]

Let us write the above equation as $(p + a\frac{N^2}{V^2})(V/N - b) = kT$. Note that now all involved quantities, namely p, V/N, as well as T are *intensive* quantities, i.e., they are independent of the system size.

The right-hand side picture in Figure 16.1 shows the van der Waals isotherms for some choice of constants a and b and for various choices of T. Comparisons with measurements show that the predictions of the van der Waals equation are for the most part more accurate compared to the predictions of the ideal gas equation. But a closer look at Figure 16.1 shows a somewhat curious and non-physical behavior. Below a "critical" temperature, the isotherms are no longer relating the pressure p and the density V/N in a monotone fashion, i.e., below this critical temperate, there is a section where a decrease in density leads to a *decrease* in pressure. Clearly, the physical process is not described accurately in this range.

It was Maxwell who in 1875 suggested a modification of the van der Waals isotherms to account for this unphysical behavior. Consider Figure 16.2. The picture on the left shows one isotherm which shows a non-physical oscillating behavior. The idea of Maxwell was to modify this curve by replacing part of the curve by a horizontal line. This line is placed in such a way that the two areas (painted in red and blue in the picture) are in balance. Note that these two areas represent work since the pressure is measured in Newtons per square meters and the volume in meters cubed. So the product is Newton times meter, the units

[3] Note that such "virial expansions" in powers of density are computed in the framework of statistical mechanics once a precise model for the repulsive and attractive forces is fixed. These expansions relate coefficients like a and b to the expressions of the forces; and by experimentally measuring the equation state one extracts information about the forces.

of work. Roughly speaking, the basic thermodynamic argument to support the equality of the two areas is that the work done by compressing the gas (starting at large volumes) along the curved path and the work gained by relaxing the volume along the straight line back to its original value should be equal because the system has returned to its initial state, and no net work should have been gained or done (otherwise we would have a perpetuum mobile). The horizontal line segment corresponds to a phase in the system where the gas co-exists in two phases, namely as liquid as well as vapor. Along the line the percentage of each component changes from all vapor to all liquid. Note that as soon as all the gas is in liquid form, any further decrease in volume leads to a very large increase in pressure.

It is important to realize that for this physical system neither the ideal gas equation, nor the van der Waals equation, and not even the modified van der Waals equation with the Maxwell construction describe the system *exactly*. They are all increasingly accurate descriptions, taking into account more and more physical effects, and they agree reasonably well with experimental measurements.

For our applications we are in a somewhat easier situation. Our aim is not to find a correct theoretical description for a real physical system. Rather, we *start* with a model and this model is by definition *exact*. Therefore, in such a situation we can hope that also the Maxwell construction gives us an exact result.

16.2 Curie-Weiss Model

For the Curie-Weiss model we have in fact already "seen" the Maxwell construction, we just never mentioned it.

In Chapter 5 we computed the exact relationship between the magnetization m and the external magnetic field h for a particular interaction strength K. We saw in (5.15) that for a fixed h and K, m takes on a value which minimizes (the free energy function)

$$-(\frac{K}{2}m^2 + hm) - h_2(\frac{1+m}{2}). \tag{16.2}$$

If we take the derivative of the above expression, we see that m is a solution of the fixed-point equation

$$m = \tanh\big\{h + Km\big\}. \tag{16.3}$$

For $K < 1$, this fixed-point equation has only a single solution for each h, but for $K > 1$ it has up to three, depending on h. Note that even though there might be many solutions of m for each h, there is always exactly one solution of h for each m. The left picture in Figure 16.3 shows this relationship (which is a smooth curve) between m and h for $K = 2$. The dashed part of the curve are points (h, m) which are solutions to the fixed-point equation but where m is not the minimizer of (16.2).

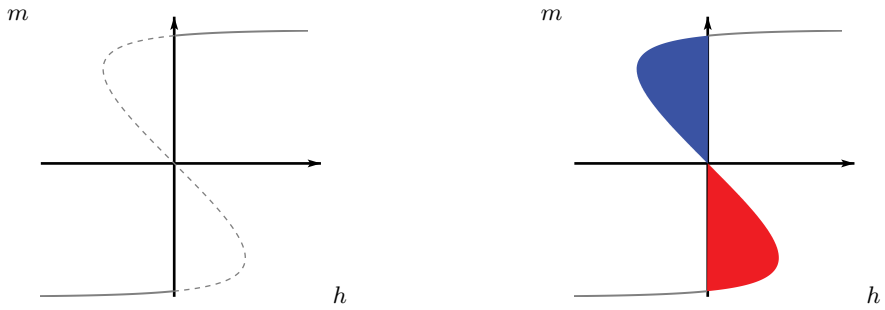


Figure 16.3 Phase transition in Curie-Weiss model when $K > 1$ as a function of h. The phase transition is at $h = 0$.

In Chapter 10 we attacked the CW (and SK) model via a message-passing approach. We first wrote down the message-passing equations. We then simplified the message-passing equations and derived the TAP equations. Note that the simplification itself was expected to be "loss-less" since it was based on the realization that only the leading terms in the message-passing equations contribute in the thermodynamic limit, the remaining terms tend to 0 with increasing system size.

But the graph corresponding to the CW model is not a tree. In fact it is as far away from a tree as one can get since it is a complete graph. It is therefore far from clear how well a message-passing analysis can capture the behavior. We saw, to our surprise, that the resulting message-passing equation, written as a fixed point equation is in fact equal to (16.3). But in the message-passing world we do not know that we "should" minimize (16.2). From the message passing perspective we start with a particular value of m and then we iterate.

Note that if we consider h as a function of m we again have in some range an unphysical behavior, namely in the branch where h decreases but m increases. It is therefore very natural to "correct" this unphysical part by a Maxwell construction, where we replace this unphysical part with a straight line which cuts the BP curve. Note that by symmetry we again have a balance of the two areas and that this Maxwell construction results in the correct phase diagram.

Let us see where we are. We have seen the Maxwell construction now for two examples, but so far it is perhaps not very convincing. For the gas model the Maxwell construction might appear like a kludge – a rough fix for an obvious problem. For the CW model, on the other hand, it might appear like a very lucky coincidence, but it did not tell us anything new.

It would be much more compelling if we could start with the BP equations and then from these equations could prove that the actual equation of state and phase transition threshold have to be of the form predicted by the Maxwell construction. In particular, this will be compelling if the actual equation of state and phase transition threshold is difficult to compute directly.

In the next section we discuss exactly such a case – namely the case of coding. Here the Maxwell construction does indeed give the correct prediction for the

MAP threshold and it is the starting point for a rigorous derivation of this quantity. More importantly, this is currently the only way of computing and proving the MAP threshold.

16.3 Coding: The Maxwell Construction for the BEC

Let us now consider coding, using elements of the (l, r)-regular LDPC ensemble, transmission over the BEC, and BP decoding. For this case we will see how we can determine the MAP threshold exactly. The Maxwell construction plays a crucial role in this determination.

As we saw in Chapter 8, the threshold for this case is determined by means of the fixed points (FP) of the equation

$$x = \epsilon f(\epsilon, x),$$

where $f(\epsilon, x) = \epsilon(1-(1-x)^{r-1})^{l-1}$. This leads us to consider the curve $(\epsilon(x), x)$ for $0 \leq x \leq 1$. Recall how from this curve we can determine the threshold – the threshold is the smallest value of ϵ which we see along this curve,

$$\epsilon^{\mathrm{BP}} = \min_{0 \leq x \leq 1} \epsilon(x) = \min_{0 \leq x \leq 1} \frac{x}{(1-(1-x)^{r-1})^{l-1}}.$$

Instead of plotting the curve $(\epsilon(x), x)$ let us plot the curve $(\epsilon(x), (1-(1-x)^{r-1})^{l})$. Note that $(1-(1-x)^{r-1})^{l})$ is the erasure probability of the best estimate of a randomly chosen variable nodes we can make if we only use the "internal" messages but ignore the directly received observation of this bit (since we ignore the direct observation the factor ϵ is missing; on the other hand we have a power of l in the expression and not just $(l-1)$ as for the density evolution equations since we take all internal inputs into account). This is the "correct" curve to which to apply the Maxwell construction as we will see now. This curve is known as the *EXIT* curve in the literature.

LEMMA 16.1 (Graphical Characterization of Thresholds) *The left-hand side of Figure 16.4 shows the so-called BP EXIT curve associated to the* $(3, 6)$*-regular ensemble. This is the curve given by* $\{\epsilon(x), (1-(1-x)^{r-1})^{l}\}$, $0 \leq x \leq 1$. *For all regular ensembles with* $l \geq 3$ *this curve has a characteristic "C" shape. It starts at the point* $(1, 1)$ *for* $x = 1$ *and then moves downwards until it "leaves" the unit box at the point* $(1, x_u(1))$ *and extends to infinity.*

The right-hand side of Figure 16.4 shows the Maxwell construction for this case. The MAP threshold is constructed from the curve by inserting a vertical line. The line is inserted at that unique spot so that area of the BP EXIT curve to the left of the vertical line is equal to the area of this curve to the right.

The Maxwell conjecture only gives us a guess of the MAP threshold. To prove this conjecture needs considerably more work. We will first show that the conjectured threshold is always an upper bound on the MAP threshold. To prove

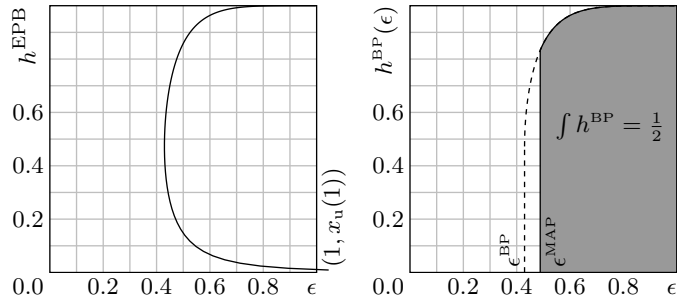


Figure 16.4 Left: The BP EXIT curve h^{BP} of the $(l = 3, r = 6)$-regular ensemble. The curve goes "outside the box" at the point $(1, x_{\mathrm{u}}(1))$ and tends to infinity. Right: The BP EXIT function $h^{\mathrm{BP}}(\epsilon)$. Both the BP as well as the MAP threshold are determined by $h^{\mathrm{BP}}(\epsilon)$.

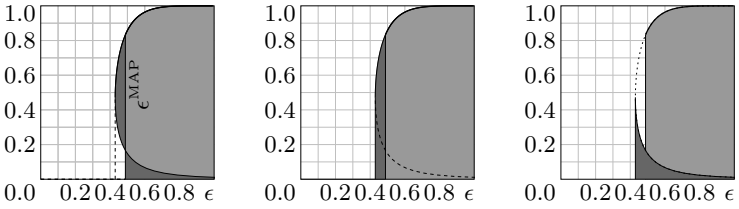


Figure 16.5 Maxwell construction.

that it is also a lower bound, and hence exact, needs different techniques, and we will discuss this later on.

Let C be a fixed code from the (l, r)-regular LDPC ensemble of length n. Let X denote the codeword, chosen uniformly at random from the set of all codewords and let Y be the received word, i.e., Y is the result of transmitting X over a BEC with parameter ϵ. We claim that

$$\frac{\mathrm{d}H(X|Y(\epsilon))}{n\mathrm{d}\epsilon} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{P}\{\hat{x}_i^{\mathrm{MAP}}(y_{\sim i}) =?\} \tag{16.4}$$

To see this, assume that each bit i is transmitted over a BEC with parameter ϵ_i.

So we have

$$\begin{aligned}\frac{1}{n}\frac{\mathrm{d}H(X|Y(\epsilon_1,\cdots,\epsilon_n))}{\mathrm{d}\epsilon} &= \sum_{i=1}^{n}\left.\frac{\partial H(X|Y(\epsilon_1,\cdots,\epsilon_n))}{\partial\epsilon_i}\right|_{\epsilon_i=\epsilon}\\ &\overset{(a)}{=} \frac{1}{n}\sum_{i=1}^{n}\left.\frac{\partial H(X_i|Y(\epsilon_1,\cdots,\epsilon_n))}{\partial\epsilon_i}\right|_{\epsilon_i=\epsilon}\\ &\overset{(b)}{=} \frac{1}{n}\sum_{i=1}^{n}\mathbb{P}\{\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i})=?\}\\ &\overset{(c)}{\leq} \frac{1}{n}\sum_{i=1}^{n}\mathbb{P}\{\hat{x}_i^{\mathrm{BP}}(y_{\sim i})=?\}.\end{aligned}$$

To see (a) note that

$$H(X\mid Y)=H(X_i\mid Y)+H(X_{\sim i}|X_i,Y)=H(X_i\mid Y)+H(X_{\sim i}|X_i,Y_{\sim i}),$$

where in the last step we can drop the Y_i in $H(X_{\sim i}|X_i,Y)$ since the channel is memoryless. Now note $H(X_{\sim i}|X_i,Y_{\sim i})$ does not depend on ϵ_i so that this term drops when we take the derivative. For step (b),

$$H(X_i\mid Y)=\mathbb{P}\{Y_i=?\}\underbrace{\mathbb{P}\{\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i})=?\}}_{\text{not a function of }\epsilon_i}=\epsilon_i\mathbb{P}\{\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i})=?\}. \tag{16.5}$$

Finally, step (c) follows since the MAP decoder is optimal and hence has the lowest error probability of all decoders.

Let us now look closer at the last expression. Define

$$h^{\mathrm{BP}}(\epsilon)=\lim_{\ell\to\infty}\lim_{n\to\infty}\mathbb{E}_{\mathrm{LDPC}}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbb{P}\{\hat{x}^{\mathrm{BP},\ell}(y_{\sim i})=?\}\right]. \tag{16.6}$$

This limit exists and is given by density evolution. In fact, $h^{\mathrm{BP}}(\epsilon)$ is essentially the EXIT function which we just discussed above. This derivation makes it clearer why the EXIT function is the "right" quantity on which to apply the Maxwell construction.

Let us discuss this all in some more detail. As we discussed above, define

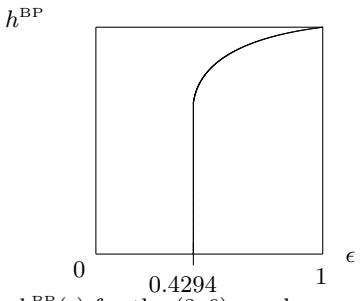


Figure 16.6 The function $h^{\mathrm{BP}}(\epsilon)$ for the $(3,6)$-regular ensemble.

$$\epsilon(x)=\frac{x}{(1-(1-x)^{r-1})^{l-1}},\quad h^{\mathrm{BP}}(x)=(1-(1-x)^{r-1})^{l},$$

and let us plot $(\epsilon(x), h^{\mathrm{BP}}(x))_{x=0}^{1}$, see Figure 16.6: Then the "envelope" of this

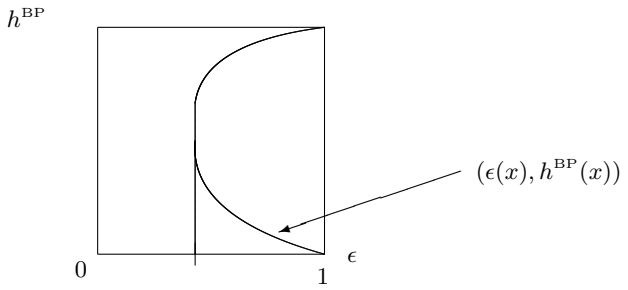


Figure 16.7 The curve $(\epsilon(x), h^{\mathrm{BP}}(x))$ and its "envelope."

curve is equal to $h^{\mathrm{BP}}(\epsilon)$ as a function of ϵ. It will be convenient to have a notation for the integral under this curve. To this end define the so called *trial entropy*:

$$P(x) = \int_0^x (1-(1-x)^{r-1})^l \epsilon'(x) dx. \tag{16.7}$$

$$= x + \frac{1}{r}(1-x)^{r-1}(l + l(r-1)x - rx) - \frac{l}{r}. \tag{16.8}$$

Note that $P(x)$ is the areas under the EXIT curve from the point $x = 0$ (this corresponds to a point at $+\infty$) until the point which is parameterized by x as indicated in Figure 16.8. Note that $P(0) = 0$. The function $P(x)$ is decreasing

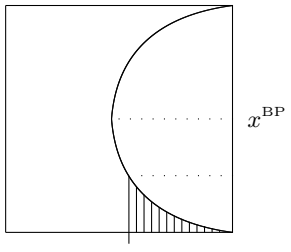


Figure 16.8 The trial entropy $P(x)$.

until $x = x^{\mathrm{BP}}$, where x^{BP} is that unique parameter so that $\epsilon^{\mathrm{BP}} = \epsilon(x^{\mathrm{BP}})$. For $x^{\mathrm{BP}} \le x \le 1$, $P(x)$ is increasing and $P(1) = 1 - \frac{l}{r}$, as a direct check shows.

It follows that there is a unique value of x in the region $[x^{\mathrm{BP}}, 1]$, call it x^A, so that $P(x^A) = 0$. We call $\epsilon(x^A)$ the *area threshold*, and write $\epsilon^A = \epsilon(x^A)$.

We now have the following sequence of inequalities:

$$
\begin{aligned}
&1-\frac{l}{r}-\liminf_{n\to\infty}\ \mathbb{E}_{\mathrm{LDPC}}[\frac{1}{n}H(x\mid y(\epsilon=\tilde{\epsilon}))]\\
&\overset{(a)}{=}\lim_{n\to\infty}\mathbb{E}_{\mathrm{LDPC}}[\frac{1}{n}H(x\mid y(\epsilon=1))]-\liminf_{n\to\infty}\ \mathbb{E}_{\mathrm{LDPC}}[\frac{1}{n}H(x\mid y(\epsilon=\tilde{\epsilon}))]\\
&\overset{(a)}{=}\limsup_{n\to\infty}\ \mathbb{E}_{\mathrm{LDPC}}[\frac{1}{n}\{H(x\mid y(\epsilon=1))-H(x\mid y(\epsilon=\tilde{\epsilon}))\}]\\
&\overset{(b)}{=}\limsup_{n\to\infty}\ \mathbb{E}\left[\int_{\tilde{\epsilon}}^{1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{P}\{\hat{x}_i^{\mathrm{MAP}}(y'_{\sim i})=?\}\right]d\epsilon\\
&\overset{(c)}{=}\limsup_{n\to\infty}\ \int_{\tilde{\epsilon}}^{1}\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbb{P}\{\hat{x}_i^{\mathrm{MAP}}(y'_{\sim i})=?\}\right]d\epsilon\\
&\leq\int_{\tilde{\epsilon}}^{1}\limsup_{n\to\infty}\ \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbb{P}\{\hat{x}_i^{\mathrm{MAP}}(y'_{\sim i})=?\}\right]d\epsilon\\
&\overset{(d)}{\leq}\int_{\tilde{\epsilon}}^{1}\lim_{\ell\to\infty}\lim_{n\to\infty}\ \mathbb{E}\left[\frac{1}{n}\mathbb{P}\{\hat{x}_i^{\mathrm{BP},\ell}(y'_{\sim i})=?\}\right]d\epsilon\\
&\overset{(e)}{=}P(1)-P(\tilde{\epsilon})\\
&=1-\frac{l}{r}-P(\tilde{\epsilon}).
\end{aligned}
$$

In step (a) note that $\frac{1}{n}H(x\mid y(\epsilon=1))$ is equal to the logarithm of the size of the code normalized by the length. It is intuitive that the limit of this quantity when $n\to\infty$, and averaged over the ensemble, is equal to the "design rate" of the code which is $1-\frac{l}{r}$. Even though this is intuitive, this needs some proof. Since the proof is purely combinatorial we skip the steps. But this transition is valid for all (l,r)-regular ensembles with $2\leq l\leq r$.

In step (b) we write the conditional entropy as an integral of its derivative and replace the derivative with the sum as we previously discussed. Since the integral is non-negative, we can exchange the order of the two integrals by Tonelli. This is step (c). In step (d) we apply the Fatou-Lebesgue theorem by observing that the integrand is bounded. Step (d) follows by the optimality of the MAP decoder, and in the final two steps we have used the definition of the trial entropy.

Equivalently,

$$
\liminf_{n\to\infty}\ \mathbb{E}_{\mathrm{LDPC}}\left[\frac{1}{n}H(x\mid y(\epsilon(x))\right]\geq P(x). \tag{16.9}
$$

DEFINITION 16.2 (MAP Threshold) [4] The *MAP threshold* of the (l,r)-regular ensemble for the BEC is denoted by $\epsilon^{\mathrm{MAP}}(l,r)$ and is defined by

$$
\inf\{\epsilon\in[0,1]:\liminf_{n\to\infty}\mathbb{E}[H(X_1^n\mid Y_1^n(\epsilon))/n]>0\}.
$$

□

[4] Define $\mathrm{P}_{e,i}=\Pr\{X_i\neq\hat{X}_i(Y_1^n)\}$, where $\hat{X}_i(Y_1^n)$ is the MAP estimate of bit i based on the observation Y_1^n. Note that by the Fano inequality we have $H(X_i\mid Y_1^n)\leq h_2(P_{e,i})$. Assume

We conclude that $\epsilon^{\mathrm{MAP}} \geq \epsilon^A = \epsilon(x^A)$, the area threshold.

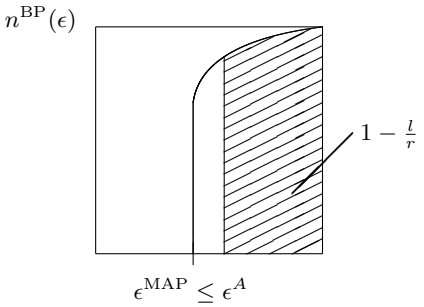


Figure 16.9 aa

So far we have seen that the threshold given by the Maxwell construction is an upper bound on the MAP threshold. There are several ways of proving the reverse inequality. For the specific case at hand, namely transmission over the BEC, one can give a purely combinatorial proof. The idea is to prove that with high probability the matrix which we get if we start with the parity-check matrix and remove all columns which correspond to non-erased bits has rank equal to the number of erased bits. This shows that with high probability the codeword can be reconstructed by solving the corresponding linear system of equations, i.e., with high probability the MAP decoder succeeds. Since this proof is very specific to the erasure channel we skip it. There is a second more conceptual

that we are transmitting above $\epsilon^{\mathrm{MAP}}(l, r)$ so that $\mathbb{E}[H(X_1^n \mid Y_1^n)/n] \geq \delta > 0$.[5] Then

$$h_2(\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n} \mathrm{P}_{e,i}]) \geq \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n} h_2(\mathrm{P}_{e,i})] \geq \mathbb{E}[\sum_{i=1}^{n} H(X_i \mid Y_1^n)/n]$$
$$\geq \mathbb{E}[H(X_1^n \mid Y_1^n)/n] \geq \delta > 0.$$

In words, if we are transmitting *above* the MAP threshold, then the ensemble average bit-error probability is lower bounded by $h_2^{-1}(\delta)$, a strictly positive constant. This ensemble is therefore not suitable for reliable transmission above this threshold.
In general we cannot conclude from $\mathbb{E}[H(X_1^n \mid Y_1^n)/n] \leq \delta$ that the average error probability is small. This is possible if we have the slightly stronger condition $\mathbb{E}[\sum_{i=1}^n H(X_i \mid Y_1^n)/n] \leq \delta$. In this case $\delta \geq \frac{1}{n}\mathbb{E}[\sum_{i=1}^n H(X_i \mid Y_1^n)] = \frac{1}{n}\mathbb{E}[\sum_{i=1}^n \mathbb{E}_{Y_1^n}[h_2(\min_x p(x \mid Y_1^n))]] \geq \frac{1}{n}\mathbb{E}[\sum_{i=1}^n \mathbb{E}_{Y_1^n}[2\min_x p(x \mid Y_1^n)]] = \frac{1}{n}\mathbb{E}[\sum_{i=1}^n 2\mathrm{P}_{e,i}]$, so that $\frac{1}{n}\mathbb{E}[\sum_{i=1}^n \mathrm{P}_{e,i}] \leq \frac{1}{2}\delta$. The last step in the previous chain of inequalities follows since under MAP decoding the error probability conditioned that we observed y_1^n is equal to $\min_x p(x \mid y_1^n)$. An alternative way to prove this is to realize that $H(X_i \mid Y_1^n)$ represents a BMS channel with a particular entropy and to use extremes of information combining to find the worst error probability such a channel can have. The extremal channel in this case is the BEC. But for the codes we consider we will see that below ϵ^{MAP} we can indeed decode correctly with high probability, which justifies the choice of our definition.
The reader might wonder why we did not start with an operational interpretation of the MAP threshold as the channel parameter below which a MAP decoder can decode with high probability. As pointed out above, for the codes we consider the given definition is in fact equivalent to the operational one. But in addition it has the advantage that the conditional entropy connects directly to the quantities which appear in our analysis, in particular to the generalized EXIT curve.

approach using spatial coupling and the interpolation technique which applies to all such problems. We will get back to this point in the next chapter.

16.4 Compressive Sensing

Also for compressive sensing there is a Maxwell construction. As a starting point however one has to consider the compressive sensing problem for a fixed and known source distribution, rather than looking for a universal algorithm.

16.5 Random K-SAT

As always, for K-SAT the situation is the most complicated. Again it is possible to write down a Maxwell construction. However, the starting point is not the BP-guided decimation algorithm but a more sophisticated algorithm, called *survey propagation*.

16.6 Discussion

Besides the original example, we have given two explicit examples of the Maxwell construction. For the CW model, the Maxwell construction appears somewhat like a coincidence. We first computed the exact relationship between average magnetization and the external field and then we computed the same relationship from a message-passing perspective. Comparing the two expressions we see that they are related by a Maxwell construction, just like in the original construction for an ideal gas.

Even more interesting is the situation if we cannot in fact compute the exact free energy expression but, starting with the message-passing formulation, can construct it using a Maxwell construction. This was the case for our second example, namely coding. There is currently no classical way of computing the MAP threshold. We have seen that the Maxwell construction gives us a guess of where this phase transition appears and we have also seen how we can prove that this guess is an *upper bound* on the MAP threshold. In the third part of these notes we will see how we can further show that this guess is also a *lower bound* on the MAP threshold using the concepts of spatial coupling and the so-called interpolation method. So in this case, the Maxwell construction, together with further techniques, allows us to solve, what from a classical perspective seems to be a hard problem.

This is a general theme. But, there is no trivial recipe for how to apply the Maxwell construction and how to prove that it is indeed correct. Each case requires some slightly different tricks and techniques. In fact, it is easy to construct examples (like K-SAT with BP guided decimation) where the predictions given

by the Maxwell construction are not even correct. But with a little bit of experience the Maxwell construction is a powerful paradigm.

Problems

16.1 *Magnetization of the Ising model on a d-regular graph with large girth.* In this problem we consider the ferromagnetic Ising model on a d-regular graph with large girth. Using the probabilistic method Erdős and Sachs proved that there exist a graphs $G_{n,d}$ on n vertices, with all vertex degrees equal to d and with a girth $g_{n,d} \geq (1 - o(1)) \log_{d-1} n$ (here $o(1)$ stands for a function that goes to zero as $n \to +\infty$). We recall that the girth is the length of the shortest loop in the graph.

Consider the Gibbs distribution of the Ising model on $G_{n,d}$

$$\mu_{n,d}(\underline{s}) = \frac{1}{Z_{n,d}} \exp\left(\frac{\beta J}{d} \sum_{\{i,j\} \in \text{edges}} s_i s_j + \beta h \sum_{i=1}^{n} s_i \right)$$

The Hamiltonian is given by the contribution of all ferromagnetic interactions associated to edges $\{i, j\}$, and a contribution from a constant magnetic field. The strength of the interaction is scaled by d for later convenience. Note that $J > 0$ but h can take both signs.

Recall that the magnetization at a vertex o is defined as $\langle s_o \rangle_{n,d}$ where $\langle - \rangle_{n,d}$ is the usual Gibbs average. This quantity is non trivial to compute. On the other hand we can run BP and compute the BP estimates of the magnetization.

(i) The second Griffith-Kelly-Sherman correlation inequality states that for Ising models with all interaction coefficients and all magnetic fields positive the magnetization can only decrease when one coefficient decreases. In the present case this inequality implies that the magnetization decreases when an edge is removed from $G_{n,d}$. Now consider the neighborhood of a vertex o, namely $N = \{i \in G_{n,d} | \text{dist}(o, i) \leq g_{n,d} - 1\}$. Define $\langle - \rangle_N$ the Gibbs average for the Ising model restricted to N. Show that for $h \geq 0$

$$\langle s_o \rangle_{n,d} \geq \langle s_o \rangle_N$$

and that for $h \leq 0$

$$\langle s_o \rangle_{n,d} \leq \langle s_o \rangle_N$$

Hint: for the second inequality use symmetry properties under the operation $h \to -h$.

(ii) The average $\langle s_o \rangle_N$ can be computed exactly from the BP recursion. Why? Show that this recursion is:

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

$$u^{(t)} = \beta h + (d-1) \tanh^{-1}(\tanh \frac{\beta J}{d} \tanh u^{(t-1)}), \qquad u^{(0)} = h$$

and that $\langle s_o \rangle_N = m^{(g_{n,d}-1)}$.

Remark: go back to homework 4 and observe this is the same recursion that you had derived by "other means".

(iii) Take now a fixed sequence of graphs $G_{n,d}$ with respect to n. Observe from above that for $h > 0$ and all t,

$$\liminf_{n\to+\infty} \langle s_o \rangle_{n,d} \geq m^{(t)},$$

and for $h \geq 0$

$$\limsup_{n\to+\infty} \langle s_o \rangle_{n,d} \leq m^{(t)}.$$

We want to look at the limit $d \to +\infty$. Show that

$$\lim_{d\to+\infty} \liminf_{n\to+\infty} \langle s_o \rangle_{n,d} \geq \lim_{t\to+\infty} m_{\mathrm{CW}}^{(t)},$$

and for $h \leq 0$ and all t

$$\lim_{d\to+\infty} \limsup_{n\to+\infty} \langle s_o \rangle_{n,d} \leq \lim_{t\to+\infty} m_{\mathrm{CW}}^{(t)},$$

where $m_{\mathrm{CW}}^{(t)}$ is the BP-magnetization of the CW model and satisfies the recursion

$$m_{\mathrm{CW}}^{(t)} = \tanh(\beta(h + J m_{\mathrm{CW}}^{(t-1)}))$$

with the initial condition $m_{\mathrm{CW}}^{(o)} = \tanh \beta h$.

Remark: These inequalities suggest the conjecture

$$\lim_{d\to+\infty} \liminf_{n\to+\infty} \langle s_o \rangle_{n,d} = \lim_{d\to+\infty} \limsup_{n\to+\infty} \langle s_o \rangle_{n,d} = \langle s_o \rangle_{\mathrm{CW}}$$

where $\langle s_o \rangle_{\mathrm{CW}}$ is the true CW magnetization.

17 *Summary of Part II*

Let us summarize what we have seen in this part.

We have considered three problems, coding, compressive sensing, and constraint satisfaction. In all three problems we started with a basic message-passing algorithm (BP) which is guaranteed to work if the factor graph is a tree and then we applied it to non-tree like factor graphs.

Coding and compressive sensing are inference problems. A signal is selected, the signal is sent over a channel, and we get to see noisy observations. From these noisy observations we want to infer the signal either exactly or give an estimate of the signal which is "close" to the sent one.

Even though both problems are inference problems, there are nevertheless substantial differences.

In the coding problem we started with a criterion (maximum a posteriori criterion) which is optimal in the sense that it minimizes the error probability. We then formulated a message-passing algorithm which implements the criterion in case the factor graph is tree-like. Finally, we applied it to non tree-like factor graphs and analyzed its behavior. As it turns out, even though the initial criterion was optimal, the actual performance we get is not optimal. More precisely, the threshold we determined is lower than the threshold which we could achieve if we used a decoder which would implement the MAP criterion exactly. In fact, in Chapter 16 we have seen how we can determine the MAP threshold and so we can determine now for each ensemble how much we loose by using a sub-optimal algorithm. In addition to this algorithmic loss, there is also the loss which we incur due to the fact that we use a sparse graph code. Due to this sparseness, the even the MAP threshold of a given ensemble is not equal to the Shannon threshold.

Compare this now to what happened in compressive sensing. There we started with the LASSO criterion. This in fact is *not* the optimal criterion since the regularization term involves the L_1 norm and not the L_0 norm. The reason for starting with LASSO is that is a natural starting point for message-passing algorithms. We then formulated a message-passing algorithm which was inspired by the LASSO criterion and which would again be exact if the factor graph was a tree. The second important difference to the coding problem is that for compressive sensing the factor graph is a complete graph, i.e., it is as far away from a locally tree-like graph as one could imagine. Nevertheless we applied our

algorithm to this case and it works very well. In fact, once all was said and done, we could conclude that in terms of threshold behavior the algorithm works as well as exactly implementing the LASSO criterion. This is not just surprising given that the graph is not at all tree-like, it is also surprising since we did not in fact analyze the natural message-passing algorithm which corresponds to the LASSO algorithm, but we analyzed an algorithm which was a considerable simplification of the original algorithm. We had two reasons to look for such a simplification. First, this brings down the complexity per iteration from square to linear. Second, only because we brought it down to a much simpler algorithm, where we able to in fact analyze the behavior of the algorithm. The basic approach we used for compressive sensing was very simple. What made the story somewhat long was that we required long calculations to determine how we could simplify the algorithm.

To summarize, whereas in coding we started with an optimal criterion but got suboptimal performance due to using a simple message-passing algorithm, in compressive sensing we started with a suboptimal criterion but then did not loose any further performance by using a message-passing algorithm.

Finally, for the K-SAT problem we are not dealing with an inference problem. The problem is much closer to e.g., lossy source coding where we also have an exponential number of essentially equivalent solutions and any solution will do. As we discussed, it is this non-uniqueness which makes the problem hard to solve. The standard solution approach is to use a type of decimation algorithm where we decide on the value of one variable at a time. What makes the analysis so difficult is that we have to run BP many times on slightly altered versions of the graph. Since each time we are dealing with essentially the same graph, there is substantial correlation in the system. There are currently no known mathematical techniques that can be applied for a rigorous analysis for this case.

Let us now look ahead to see what is to come. In all three cases the application of message-passing algorithms leads to good but suboptimal performance as it turns out. So how good would each of these systems work if in fact we did not have any constraints on complexity and could implement the optimal criteria? This is an important engineering question. If the gap to the optimal performance is very small then it is probably not worth thinking of improved algorithms or spending higher computational resources to solve the problem. But if the gap is substantial, it is an entirely different story. Also, if we are able to answer this question, then perhaps we can take a more active approach. Particularly in coding and compressive sensing, we are not forced to use a particular code or sensing matrix. We often have a choice and can design the system. Therefore, at least in these two problems we can ask what we can do in order to narrow the performance gap. This is sometimes also called as "engineering the phase transition." For this purpose, we will discuss a generic and very useful construction, called spatial coupling, which allows us to construct graphical models which perform particularly well under message-passing algorithms.

Part III

Advanced Topics

18 Spatial Coupling and Nucleation Phenomenon

So far we have seen that a variety of problems can be phrased in a natural way in terms of marginalizing a highly-factorized function. Message-passing algorithms are then the logical choice to accomplish this marginalization and we have seen how such algorithms perform in the thermodynamic limit.

Perhaps more surprisingly, we saw that the same quantities which were important for the analysis of the suboptimal message-passing algorithm reappeared when we looked at the seemingly more fundamental question of determining static thresholds, like the MAP threshold or the SAT/UNSAT threshold. The Maxwell construction is a graphical representation of this phenomenon.

We will now tie these two threads together. We will discuss a generic construction, called spatial coupling, which can be applied to a wide range of graphical models. The idea is to take many copies of a graphical model, to place them next to each other on a line and then to start connecting these models by "exchanging edges" in such a way that the local structure of the graphical model remains unchanged but that globally we create a larger graphical model which forms a one-dimensional chain. If in addition we impose suitable conditions at the boundaries of the model, this larger graphical model behaves very well under message-passing. Roughly speaking, the performance of the large spatially-coupled model under message-passing (in terms of the resulting threshold) is as good as if we had done optimal processing on the original graphical model.

For the most part we will only discuss the phenomenon but we will not give proofs. We will see how this phenomenon has again a nice physical interpretation. In fact – it is what is called the *nucleation* phenomenon in physics. Nucleation explains amongst other things how crystals grow, starting with a *seed* or *nucleus*.

We will discuss two important consequences of the nucleation phenomenon.

First, whenever we are in control of the graphical structure and the size of the graph is not very crucial, it is natural to construct the graph according to the above recipe. This results in graphs which are well suited for message-passing processing and give very good performance. E.g., for the coding problem this construction makes it possible to design codes which, under BP decoding, are not only provably capacity-achieving for a particular channel, but are in fact universally so, i.e., they are capacity-achieving for the whole class of BMS channels. A similar construction is possible for the compressive sensing problem.

There is a second, equally important application of the idea, namely to use

spatial coupling as a proof technique. Consider e.g. the case of the K-SAT problem. Also in this case we can use spatial coupling. This means we can construct spatially-coupled K-SAT formulas, and it is easier to find satisfiable solutions for such formulas than for the uncoupled ones. But what is the use of this? In coding, we were in charge of picking the code, and so we can pick coupled ones. The same thing applies for compressive sensing. We do not have the same degree of freedom for the constraint satisfaction problem where the formula is given to us. The idea is the following. If we are able to analyze the performance of a message-passing algorithm on coupled formulas then we can use the so-called *interpolation* method to show that this algorithmic threshold is also a lower bound on the SAT/UNSAT threshold of the uncoupled ensemble. So in this case we use spatial coupling only as a thought experiment. Indeed, the same method can be used in the context of coding to prove that the MAP threshold of the uncoupled formula is at least as large as the area threshold. Together with the upper bound on the MAP threshold which we derived in Chapter 16 this shows that the MAP threshold of the uncoupled ensemble is equal to the area threshold.

In the remainder of the chapter we go over our three running examples. In each case we describe the construction, the performance of the coupled system, as well as the consequences for our problem at hand.

18.1 Coding

There are many possible ways of constructing coupled graphical models from uncoupled ones. The "saturation phenomenon" is fairly robust with respect to the exact way of how we construct coupled models. So the difference lies mostly in how convenient the construction is either from a practical perspective or for the purpose of proofs. We present below two generic ways to achieve the spatial coupling. We start with the "protograph" construction. It has a very good performance and the additional structure is well suited for implementations. Our second construction is a "random" model. This model is well suited for proofs. Indeed, in the sequel we exclusively use the random model when it comes to showing plots and to formulating theorems.

Protograph Construction

To start, consider a protograph of a standard $(3, 6)$-regular ensemble (see [**?**, **?**] for the definition of protographs). It is shown in Figure 18.1. There are two variable nodes and there is one check node. Let M denote the number of variable nodes at each position. For our example, $M = 100$ means that we have 50 copies of the protograph so that we have 100 variable nodes at each position. For all future discussions we will consider the regime where M tends to infinity.

Next, consider a collection of $(2L+1)$ such protographs as shown in Figure 18.2. These protographs are non-interacting and so each component behaves just like

Figure 18.1 Protograph of a standard $(3, 6)$-regular ensemble.

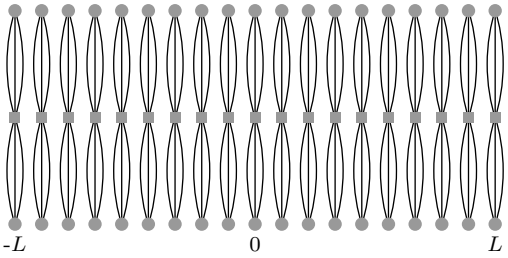


Figure 18.2 A chain of $(2L + 1)$ protographs of the standard $(3, 6)$-regular ensembles for $L = 9$. These protographs do not interact.

a standard $(3, 6)$-regular component. In particular, the belief-propagation (BP) threshold of each protograph is just the standard threshold, call it $\epsilon^{\mathrm{BP}}(l = 3, r = 6)$. Slightly more generally: start with an $(l, r = kl)$-regular ensemble where l is odd so that $\lfloor l/2 \rfloor = (l - 1)/2 \in \mathbb{N}$.

We will now "coupled" these copies. To achieve this coupling, connect each protograph to $\lfloor l/2 \rfloor$ protographs "to the left" and to $\lfloor l/2 \rfloor$ protographs "to the right." This is shown in Figure 18.3 for the two cases $(l = 3, r = 6)$ and $(l = 7, r = 14)$.

Note that $\lfloor l/2 \rfloor$ extra check nodes are added on each side to connect the "overhanging" edges at the boundary. This reduces the rate of this ensemble from $1 - \frac{l}{r} = \frac{k-1}{k}$ to

$$\begin{aligned} R(l, r = kl, L) &= \frac{(2L + 1) - (2(L + \lfloor l/2 \rfloor) + 1)/k}{2L + 1} \\ &= \frac{k - 1}{k} - \frac{2\lfloor l/2 \rfloor}{k(2L + 1)}, \end{aligned}$$

Note that this rate loss decreases with the length of the chain. Therefore, in practice we want to pick the length not too small. Of course, this increases the blocklength and so there is a natural trade-off between the block length and the rateloss due to the boundary.

In the above construction we had to assume that l was odd and also the "width" of the connection was linked directly to the degree l. In this case the construction leads to the very symmetric ensemble. It is not very hard to extend this construction to cases where l is even and so that "width" of the connection

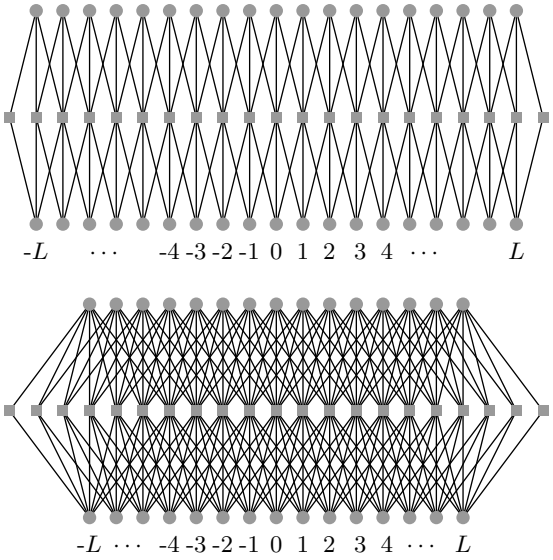


Figure 18.3 Two coupled chains of protographs with $L = 9$ and $(l = 3, r = 6)$ (top) and $L = 7$ and $(l = 7, r = 14)$ (bottom), respectively.

is no longer directly linked to l. But instead of following this path, let us directly go to another extreme and introduce an ensemble which includes much more randomness.

Random Construction

For the purpose of analysis, the following random ensemble is much betters suited. Let us assume that $r \geq l$, so that the ensemble has a non-trivial design rate.

We assume that the variable nodes are at positions $[-L, L]$, $L \in \mathbb{N}$. At each position there are M variable nodes, $M \in \mathbb{N}$. Conceptually we think of the check nodes to be located at all integer positions from $[-\infty, \infty]$. Only some of these positions actually interact with the variable nodes. At each position there are $\frac{l}{r}M$ check nodes. It remains to describe how the connections are chosen.

Rather than assuming that a variable at position i has exactly one connection to a check node at position $[i - \lfloor l/2 \rfloor, \ldots, i + \lfloor l/2 \rfloor]$, we assume that each of the l connections of a variable node at position i is uniformly and independently chosen from the range $[i, \ldots, i + w - 1]$, where w is a "smoothing" parameter. In the same way, we assume that each of the r connections of a check node at position i is independently chosen from the range $[i - w + 1, \ldots, i]$. We no longer require that l is odd.

More precisely, the ensemble is defined as follows. Consider a variable node at position i. The variable node has l outgoing edges. A *type* t is a w-tuple of non-negative integers, $t = (t_0, t_1, \ldots, t_{w-1})$, so that $\sum_{j=0}^{w-1} t_j = l$. The operational

meaning of t is that the variable node has t_j edges which connect to a check node at position $i+j$. There are $\binom{l+w-1}{w-1}$ types. Assume that for each variable we order its edges in an arbitrary but fixed order. A *constellation* c is an l-tuple, $c=(c_1,\ldots,c_l)$ with elements in $[0,w-1]$. Its operational significance is that if a variable node at position i has constellation c then its k-th edge is connected to a check node at position $i+c_k$. Let $\tau(c)$ denote the type of a constellation. Since we want the position of each edge to be chosen independently we impose a uniform distribution on the set of all constellations. This imposes the following distribution on the set of all types. We assign the probability

$$p(t)=\frac{|\{c:\tau(c)=t\}|}{w^l}.$$

Pick M so that $Mp(t)$ is a natural number for all types t. For each position i pick $Mp(t)$ variables which have their edges assigned according to type t. Further, use a random permutation for each variable, uniformly chosen from the set of all permutations on l letters, to map a type to a constellation.

Under this assignment, and ignoring boundary effects, for each check position i, the number of edges that come from variables at position $i-j$, $j\in[0,w-1]$, is $M\frac{l}{w}$. In other words, it is exactly a fraction $\frac{1}{w}$ of the total number Ml of sockets at position i. At the check nodes, distribute these edges according to a permutation chosen uniformly at random from the set of all permutations on Ml letters, to the $M\frac{l}{r}$ check nodes at this position. It is then not very difficult to see that, under this distribution, for each check node each edge is roughly independently chosen to be connected to one of its nearest w "left" neighbors. Here, "roughly independent" means that the corresponding probability deviates at most by a term of order $1/M$ from the desired distribution. As discussed beforehand, we will always consider the limit in which M first tends to infinity and then the number of iterations tends to infinity. Therefore, for any fixed number of rounds of DE the probability model is exactly the independent model described above.

LEMMA 18.1 (Design Rate) *The design rate of the ensemble (l,r,L,w), with $w\le 2L$, is given by*

$$R(l,r,L,w)=(1-\frac{l}{r})-\frac{l}{r}\frac{w+1-2\sum_{i=0}^{w}\left(\frac{i}{w}\right)^r}{2L+1}.$$

Proof Let V be the number of variable nodes and C be the number of check nodes that are connected to at least one of these variable nodes. Recall that we *define* the design rate as $1-C/V$.

There are $V=M(2L+1)$ variables in the graph. The check nodes that have potential connections to variable nodes in the range $[-L,L]$ are indexed from $-L$ to $L+w-1$. Consider the $M\frac{l}{r}$ check nodes at position $-L$. Each of the r edges of each such check node is chosen independently from the range $[-L-w+1,-L]$. The probability that such a check node has at least one connection in the range $[-L,L]$ is equal to $1-\left(\frac{w-1}{w}\right)^r$. Therefore, the expected number of check nodes

at position $-L$ that are connected to the code is equal to $M\frac{l}{r}(1-\left(\frac{w-1}{w}\right)^r)$. In a similar manner, the expected number of check nodes at position $-L+i$, $i=0,\dots,w-1$, that are connected to the code is equal to $M\frac{l}{r}(1-\left(\frac{w-i-1}{w}\right)^r)$. All check nodes at positions $-L+w,\dots,L-1$ are connected. Further, by symmetry, check nodes in the range $L,\dots,L+w-1$ have an identical contribution as check nodes in the range $-L,\dots,-L+w-1$. Summing up all these contributions, we see that the number of check nodes which are connected is equal to

$$C = M\frac{l}{r}[2L - w + 2\sum_{i=0}^{w}(1-\left(\frac{i}{w}\right)^r)].$$

□

Discussion: In the above lemma we have *defined* the design rate as the normalized difference of the number of variable nodes and the number of check nodes that are involved in the ensemble. This leads to a relatively simple expression which is suitable for our purposes. But in this ensemble there is a non-zero probability that there are two or more degree-one check nodes attached to the same variable node. In this case, some of these degree-one check nodes are redundant and do not impose constraints. This effect only happens for variable nodes close to the boundary. Since we consider the case where L tends to infinity, this slight difference between the "design rate" and the "true rate" does not play a role. We therefore opt for this simple definition. The design rate is a lower bound on the true rate.

Density Evolution

The protograph construction has a slightly better performance if we look at codes of finite length and also, due to the extra structure, it might be easier to implement. On the other hand, the random ensemble is easier to deal with when it comes to proofs. Since asymptotically they behave essentially the same, we concentrate in the sequel on the random case.

The (l,r,L,w) ensemble is just an LDPC ensemble with some additional structure. It's asymptotic performance can hence again be assessed via density evolution. Therefore, as a first step let us write down the density evolution equations. The only difference compared to the DE equations of the uncoupled ensemble is that now we have a potentially different erasure probability for *every position*. The state is therefore no longer a scalar quantity but a vector of the length of the chain.

DEFINITION 18.2 (Density Evolution of (l,r,L,w) Ensemble) Let x_i, $i\in\mathbb{Z}$, denote the average erasure probability which is emitted by variable nodes at position i. For $i\notin[-L,L]$ we set $x_i=0$. For $i\in[-L,L]$ the FP condition

implied by DE is

$$x_i = \epsilon\Big(1 - \frac{1}{w}\sum_{j=0}^{w-1}\big(1 - \frac{1}{w}\sum_{k=0}^{w-1} x_{i+j-k}\big)^{r-1}\Big)^{l-1}. \tag{18.1}$$

If we define

$$y_i = \Big(1 - \frac{1}{w}\sum_{k=0}^{w-1} x_{i-k}\Big)^{r-1}, \tag{18.2}$$

then (18.1) can be rewritten as

$$x_i = \epsilon\Big(1 - \frac{1}{w}\sum_{j=0}^{w-1} y_{i+j}\Big)^{l-1}.$$

EXIT Curves

As for uncoupled ensembles we can draw EXIT curves for the coupled case. Recall that in the uncoupled case, the EXIT curve is a plot of the channel parameter ϵ as a function of the EXIT value $(1-(1-x)^{r-1})^l$, see e.g., Figure 16.4. In the uncoupled case we had a simple analytical formula for this curve. For the coupled case, no such formula exists, but one can compute the curves numerically.

Figure 18.4 shows the EXIT curves for the $(l = 3, r = 6, L)$ for $L = 1, 2, 4, 8, 16, 32, 64$, and 128. Note that these EXIT curves show a dramatically different behavior

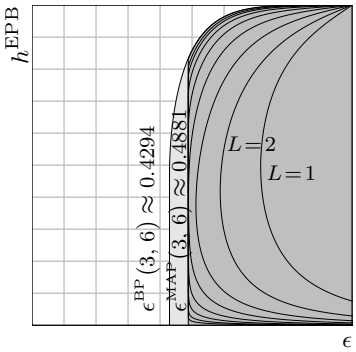


Figure 18.4 EBP EXIT curves of the ensemble $(l = 3, r = 6, L)$ for $L = 1, 2, 4, 8, 16, 32, 64$, and 128. The BP/MAP thresholds are $\epsilon^{\mathrm{BP/MAP}}(3, 6, 1) = 0.714309/0.820987$, $\epsilon^{\mathrm{BP/MAP}}(3, 6, 2) = 0.587842/0.668951$, $\epsilon^{\mathrm{BP/MAP}}(3, 6, 4) = 0.512034/0.574158$, $\epsilon^{\mathrm{BP/MAP}}(3, 6, 8) = 0.488757/0.527014$, $\epsilon^{\mathrm{BP/MAP}}(3, 6, 16) = 0.488151/0.505833$, $\epsilon^{\mathrm{BP/MAP}}(3, 6, 32) = 0.488151/0.496366$, $\epsilon^{\mathrm{BP/MAP}}(3, 6, 64) = 0.488151/0.492001$, $\epsilon^{\mathrm{BP/MAP}}(3, 6, 128) = 0.488151/0.489924$. The light/dark gray areas mark the interior of the BP/MAP EXIT function of the underlying $(3, 6)$-regular ensemble, respectively.

compared to the EBP EXIT curve of the underlying ensemble. These curves appear to be "to the right" of the threshold $\epsilon^{\mathrm{MAP}}(3, 6) \approx 0.48815$. For small values of L one might be led to believe that this is true since the design rate of such

an ensemble is considerably smaller than $1 - l/r$. But even for large values of L, where the rate of the ensemble is close to $1 - l/r$, this dramatic increase in the threshold is still true. Empirically we see that, for L increasing, the EBP EXIT curve approaches the MAP EXIT curve of the underlying $(l = 3, r = 6)$-regular ensemble. In particular, for $\epsilon \approx \epsilon^{\text{MAP}}(l, r)$ the EBP EXIT curve drops essentially vertically until it hits zero.

Decoding Wave

"The" key to understanding why spatially coupled ensembles perform so well is to study their FPs under density evolution. Recall that for uncoupled ensembles the FPs are scalars. For the coupled case the state of the system is no longer a scalar but a vector, where the length of the vector is equal to the length of the chain. Due to this fact, there are some very interesting FPs which appear.

Assume we are operating much above the threshold. Let us assume that we decode until we are stuck and let us plot the final erasure probability at each section along the chain. Then it is reasonably to expect that this erasure probability is equal to the erasure probability which we would observe for an uncoupled ensemble. The only exception are positions very close to the boundary where the behavior is a little bit better due to the extra information we have there. The top picture in Figure 18.6 shows this situation together with the position of the FP on the EXIT curve. Since the FP is symmetric with respect to the middle of the chain, only one half is shown. Imagine that we now slowly lower the erasure probability of the channel. Due to the improved conditions at the boundary, the "effective" erasure probability at the boundary will at some point be below the BP threshold of the uncoupled ensemble and the BP decoder will be able to decode the bits at the boundary. But once these bits are decoded this will lower the "effective" erasure probability for bits a little bit further into the chain. This effect propagates like a wave and the whole chain will get decoded. The middle and the bottom picture in Figure 18.6 show the wave in various stages.

The perhaps the most surprising aspect is that the BP threshold for the coupled chain is exactly the area threshold of the uncoupled one.

¡¡¡¡¡¡¡ .mine Figure 18.6 shows the FP for various parameters of the channel together with the position of the FP on the EXIT curve. Since the FP is symmetric with respect to the middle of the chain, only one half is shown.

Main Statement

THEOREM 18.3 (BP Threshold of the (l, r, L, w) Ensemble) *Consider transmission over the BEC(ϵ) using random elements from the ensemble (l, r, L, w). Let $\epsilon^{BP}(l, r, L, w)$ denote the BP threshold and let $R(l, r, L, w)$ denote the design rate of this ensemble.*

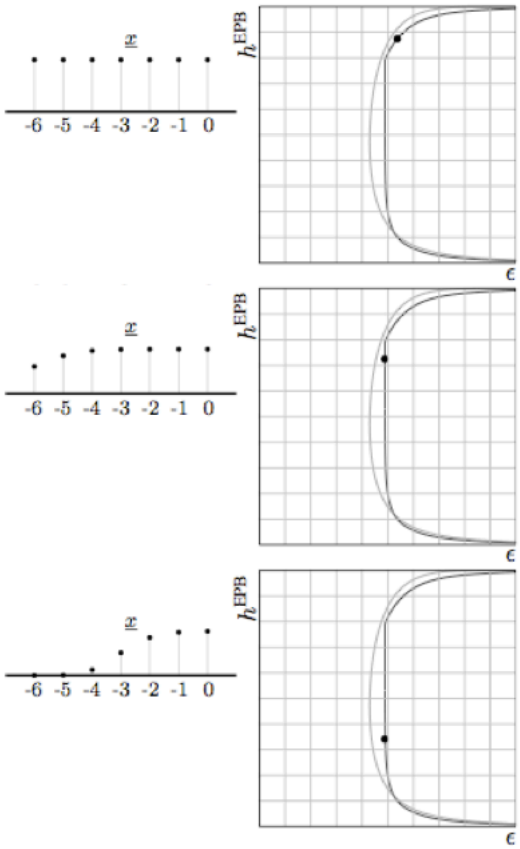


Figure 18.5 FPs for various parameters of the channel together with the position of the FP on the EXIT curve.

Figure 18.6 FPs for various parameters of the channel together with the position of the FP on the EXIT curve.

Then, in the limit as M tends to infinity, and for w sufficiently large

$$\epsilon^{BP}(l,r,L,w) \leq \epsilon^{MAP}(l,r,L,w) \leq \epsilon^{MAP}(l,r) + \frac{w-1}{2L(1-(1-x^{MAP}(l,r))^{r-1})^{l}}, \quad (18.3)$$

$$\epsilon^{BP}(l,r,L,w) \geq \left(\epsilon^{MAP}(l,r) - w^{-\frac{1}{8}} \frac{8lr + \frac{4rl^2}{(1-4w^{-\frac{1}{8}})^r}}{(1-2^{-\frac{1}{r}})^2}\right) \times \left(1-4w^{-1/8}\right)^{rl}. \quad (18.4)$$

In the limit as M, L and w (in that order) tend to infinity,

$$\lim_{w\to\infty}\lim_{L\to\infty} R(l,r,L,w) = 1-\frac{l}{r}, \tag{18.5}$$

$$\begin{aligned}\lim_{w\to\infty}\lim_{L\to\infty} \epsilon^{BP}(l,r,L,w) &= \lim_{w\to\infty}\lim_{L\to\infty} \epsilon^{MAP}(l,r,L,w)\\ &= \epsilon^{MAP}(l,r).\end{aligned} \tag{18.6}$$

Roughly speaking, the above theorems states that the BP threshold of the coupled chain is equal to its MAP threshold and also to the MAP threshold of the uncoupled chain. The statements in the theorem are considerably weaker than what can be observed empirically. In particular, the convergence with respect to the coupling width is conjectured to be exponential in w.

A very similar statement can be shown to hold for transmission over general channels. In particular, one can show that these ensembles are good universally for the whole class of BMS channels.

18.2 Compressive Sensing

The idea of spatial coupling can also be used in compressive sensing to attain optimal performance by message passing. In a nutshell, the idea is to construct appropriate sensing matrices that correspond to a "spatially coupled" factor graph and then to apply an AMP type algorithm. The performance of the algorithm is then analyzed through a state evolution recursion tailored to the spatially coupled graph. This turns out to be a one-dimensional recursion which displays similar phenomena than those described for the BEC.

In Chapter 12 our starting point was the Lasso estimator which is a reasonable starting point to develop a universal algorithm that doe not assume a prior knowledge of the signal distribution in the class $\mathcal{F}_\epsilon$. Recall that the state evolution equation in Chapter 13 has at most one fixed point. Therefore, intuitively, one does not expect that any improvement in performance can be obtained by spatial coupling. This has indeed been corroborated by numerical simulations. We will therefore turn our attention to a setting where the prior distribution of the signal is known.

AMP when the prior is known

We assume that the signal distribution is from the class $\mathcal{F}_\epsilon$ and that it is known. In other words $p_0(x) = (1-\epsilon)\delta_0(x) + \epsilon\phi_0(x)$ for a known $\phi_0(x)$ (for example a Gaussian distribution). As explained in Chapter 4, in this setting the optimal estimator is the MMSE estimator (4.30). In Chapter 7 we went through the belief propagation equations in Example 12. This approach can be systematically developed in order to recursively compute the BP-estimate. Furthermore, following the same route as in Chapter 12, these message-passing equations can be

simplified in order to arrive at an AMP algorithm that is very similar to (12.16). By skimming through the previous chapters one can almost guess the form of the new algorithm.

In (12.16) the update of the AMP-estimate uses the soft thresholding function $\eta(y,\lambda)$ found by solving the scalar Lasso problem. The reader should not be too surprised that now the AMP updates involve a thresholding function given by the MMSE estimator of the scalar case. Consider a scalar measurement $y = x + \nu z$ of "signal" x affected by Gaussian noise with variance ν^2 (so $Z \sim N(0,1)$) the thresholding function is

$$\eta_0(y,\nu) = \mathbb{E}[X|X+\nu Z = y] = \frac{\int dx\, x\, p_0(x) e^{-\frac{(y-x)^2}{2\nu^2}}}{\int dx\, p_0(x) e^{-\frac{(y-x)^2}{2\nu^2}}}.$$

We stress that $\eta_0(y,\nu)$ is not universal and depends on the prior. Here ν plays the role of a threshold level analogous to λ in the Lasso case. It will be adjusted at each AMP iteration. The mean square error for this optimal estimator (of the scalar problem) is the MMSE function[1]

$$\begin{aligned}\mathrm{mmse}(\nu^{-2}) &= \mathbb{E}\big[(X - \mathbb{E}[X|X+\nu Z])^2\big] \\ &= \int dx\, p_0(x) \int dz\, \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} (x - \eta_0(x+\nu z,\nu))^2.\end{aligned}$$

The AMP updates are the same than in Chapter 12 except η is replaced by η_0,

$$\hat{x}_i^{(t+1)} = \eta_0\big(x_i^{(t)} + \sum_{a=1}^{m} A_{ai} r_a^{(t)}, \nu^{(t)}\big), \tag{18.7}$$

$$r_a^{(t)} = y_a - \sum_{j=1}^{n} A_{aj} \hat{x}_j^{(t-1)} + b^{(t)} r_a^{t-1}. \tag{18.8}$$

If you go back to the derivation of the Onsager term in Chapter 12 you will see that it can be traced back to a derivative of the soft thresholding function. You can guess that now

$$b^{(t)} = \frac{1}{\delta n} \sum_{i=1}^{n} \eta_0'\big(x_i^{(t-1)} + \sum_{a=1}^{m} A_{ai} r_a^{(t-1)}, \nu^{(t)}\big). \tag{18.9}$$

Similarly recall that in Chapter 13 we expressed the threshold level $\nu^{(t)}$ thanks to the MSE through (13.8). Here one arrives at the same conclusion, namely

$$(\nu^{(t)})^2 = \sigma^2 + \frac{1}{\delta}(\tau^{(t)})^2, \tag{18.10}$$

where $\tau^{(t)2}$ is the average (normalized) MSE of the AMP algorithm $(\tau^{(t)})^2 = \lim_{n\to+\infty} \frac{1}{n}\mathbb{E}\|\underline{\hat{x}}^{(t)} - \underline{x}_0\|_2$. We can track its evolution thanks to the recursion (same as (13.10) with correct η_0-function)

$$(\tau^{(t+1)})^2 = \mathrm{mmse}((\nu^{(t)})^{-2}). \tag{18.11}$$

[1] By convention the argument of the MMSE function is a signal-to-noise-ratio, here ν^{-2}.

In hindsight one can develop an interpretation for this equation: at time $t+1$ the total quadratic error $(\tau^{(t+1)})^2$ for the AMP estimate is given by the MMSE of a scalar signal with effective noise variance $\sigma^2 + \frac{1}{\delta}(\tau^{(t)})^2$ at time t.

Let us summarize. Equations (18.11) and (18.10) give the evolution of the MSE and the threshold level. These quantities can be precomputed. Equations (18.8) and (18.9) define the AMP algorithm, and allow to compute the estimates for the signal.

Construction of the measurement matrix

Let us first explain the general idea. In the standard case considered so far, the measurement matrices have iid entries $A_{ai} \sim \mathcal{N}(0, \frac{1}{\sqrt{m}})$ so that "their factor graph" is a complete bipartite graph with m checks and n variables. The ratio $\delta = m/n$ is the sampling rate. Inspired by the construction of spatially coupled codes one may try to use matrices associated to a spatial chain of L complete bipartite graphs coupled across a window of size w. This turns out to be a successful idea! The sampling rate is still equal to δ in the bulk of the chain. At the boundary one has to add extra check nodes or equivalently one has to oversample. Indeed, in order to create a seed that gets the nucleation process started one needs a good estimate of the first few components of the signal. The increase in sampling rate is negligible in the thermodynamic limit.

In practice, because the AMP algorithm updates purely local quantities (the BP messages flowing along edges have been eliminated), one can forget about the factor graph and specify directly the sensing matrix. You can convince yourself that the sensing matrix described here has a factor graph that is a chain of coupled complete bipartite graphs. There are many possible constructions and ways to optimize the finite length performance. But these issues will not concern us here, and we discus a similar construction which is similar to the one presented in the coding case.

The signal has n components in total and we make m measurements. The measurement matrix has n columns and m rows. Think of n given and m to be determined later. Partition the columns in L groups[2] $c \in \{1, \dots, L\}$ with N columns each, so $N = n/L$. Consider $L + w - 1$ groups of rows $r \in \{-(w-2), \dots, 0, 1, \dots, L\}$, each with $M = \delta N$ rows. The total number of measurements is $m = (w-1)M + ML = \delta n(1 + (w-1)/L)$. The contribution of the oversampling rate to the total rate $m/n = \delta(1 + (w-1)/L)$ vanishes for large L.

Now consider an $(L + w - 1) \times L$ matrix of variances J_{rc}. A simple choice is

$$J_{r,c} = \begin{cases} \frac{1}{2w-1} & \text{if } \ c \in \{r - w + 1, \dots, r + w - 1\} \\ 0 & \text{otherwise} \end{cases}$$

Here we use a simple square-like and symmetric shape function for $J_{r,c}$. One can generalize this to $J_{r,c} = \rho \mathcal{J}(\rho|r - c|)$ with $\rho = (2w-1)^{-1}$ and a shape function

[2] One can visualize the groups as positions along the chain.

$\mathcal{J}(z)$ that is positive, supported on $[-1,+1]$ and $\int_{-1}^{+1} dz\, \mathcal{J}(z) = 1$. Let us also note that taking larger variances for the seeding part of the matrix may lead to better performance. In the sequel all equations are valid for general choices of $J_{r,c}$.

To specify the matrix elements of A_{ai}, we introduce the notation $R(a)$ and $C(i)$ for the groups (r and c) to which row a and column i belong. A simple choice is to take iid entries

$$A_{ai} \sim (0, \frac{1}{M} J_{R(a),C(i)})$$

We notice that by construction we have the normalization $\sum_i A_{ai}^2 \approx 1$, as in the standard (uncoupled) case. This matrix has a band structure with a band of height and width $wM \times wN$. However the correct regime in which the spatially coupled model is used is $N >> L$ so effectively the matrix is "full".

Spatially coupled AMP

The starting point - the BP equations - are exactly the same except they are applied to a bigger factor graph. The derivation of the coupled AMP algorithm then proceeds in the usual way by retaining only important terms *in the regime* $N \to +\infty$ and L fixed.

It turns out that the resulting equations have a few extra complications. Namely, due to coupling, the sensing matrix elements get "renormalized" and the threshold level as well as the Onsager term get "averaged". The AMP equations now read

$$\hat{x}_i^{(t+1)} = \eta_0\big(x_i^{(t)} + \sum_{a=1}^{m} Q^{(t)}_{R(a),C(i)} A_{ai} r_a^{(t)}, \nu^{(t)}_{C(i)}\big) \tag{18.12}$$

$$r_a^{(t)} = y_a - \sum_{j=1}^{n} A_{aj}\hat{x}_j^{(t-1)} + b^{(t)}_{R(a)} r_a^{t-1} \tag{18.13}$$

where

$$b^{(t)}_{R(a)} = \frac{1}{\delta}\sum_{c=1}^{L} J_{R(a),c} Q^{t-1}_{R(a),c}\left\{\frac{1}{N}\sum_{i\,\mathrm{s.t}\,C(i)=c} \eta_0'\big(x_i^{(t)} + \sum_{b=1}^{m} Q^{(t)}_{R(b),C(i)} A_{bi} r_b^{(t)}, \nu^{(t)}_{C(i)}\big)\right\}$$

The threshold levels $\nu^{(t)}_{C(i)}$ and the weights $Q_{R(a),C(i)}$ depend only on the *local* MSE $(\tau_c^{(t)})^2 = \lim_{N\to+\infty} \frac{1}{N}\sum_{i\,\mathrm{s.t}\,C(i)=c} \mathbb{E}\|\hat{x}_i^{(t)} - x_{0,i}\|_2^2$. These quantities can all be pre-computed from state evolution. The threshold level is given by (a generalization of (18.10))

$$(\nu_c^{(t)})^{-2} = \sum_r J_{r,c}\big(\sigma^2 + \frac{1}{\delta}\sum_c J_{r,c}(\tau_c^{(t)})^2\big)^{-1}, \tag{18.14}$$

This equation says that the threshold for estimates of the signal components in group c is given by an average of the signal to noise ratios for measurements in

the groups $r \in \{c-w+1, \dots, c+w-1\}$, and the later are themselves given by an average of the local MSE in the groups $c \in \{r-w+1, \dots, r+w-1\}$. The sensing matrix gets renormalized by weights

$$Q_{r,c} = \frac{\left(\sigma^2 + \frac{1}{\delta}\sum_c J_{r,c}(\tau_c^{(t)})^2\right)^{-1}}{\sum_r J_{r,c}\left(\sigma^2 + \frac{1}{\delta}\sum_c J_{r,c}(\tau_c^{(t)})^2\right)^{-1}}.$$

Finally, the local MSE evolves as

$$(\tau_c^{(t+1)})^2 = \mathrm{mmse}((\nu_c^{(t)})^{-2}), \qquad c = 1, \dots, L \tag{18.15}$$

Equations (18.14)-(18.15) are the one dimensional state evolution recursion and can be used to derived the performance of AMP on the spatially coupled model. The reader should ponder on this recursion and realize that its structure is perfectly analogous to the DE recursion in coding for the BEC.

Analysis of Performance and Phase Diagram

The discussion in this paragraph is valid for a fairly wide class of functions $\phi_0(x)$, but a good exercise for the reader is to verify the claims for a Gaussian $\phi_0(x)$. This can be done analytically for the uncoupled case and numerically in the coupled case. Notice that in this case $\eta_0(y,s)$ can be explicitly be computed.

Consider the recursion (18.11) and look at the corresponding fixed point equation. Let

$$\tilde{\delta}(p_0) \equiv \sup_{\nu}\{\nu^{-2}\mathrm{mmse}(\nu^{-2})\} > \epsilon$$

Here the equality is definition. The inequality is a fact, which follows by remarking $\lim_{\nu\to 0}\nu^{-2}\mathrm{mmse}(\nu^{-2}) = \epsilon$. For a sampling rate $\delta > \tilde{\delta}(p_0)$ there exists only one fixed point solution $(\tau_{\mathrm{good}})^2 = O(\sigma^2)$. This corresponds to correct reconstruction in the small noise limit $\sigma \to 0$. Now, decrease the sampling rate in the range $\epsilon < \delta < \tilde{\delta}(p_0)$. One finds two or more stable fixed points (as well as unstable ones) for all $\sigma^2 > 0$. Besides the "good" fixed point satisfies $(\tau_{\mathrm{good}})^2 = O(\sigma^2)$ there is a "bad" one, i.e. $(\tau_{\mathrm{bad}})^2 = \Theta(1)$ as $\sigma \to 0$. Under the (natural) initial condition $(\tau^0)^2 = +\infty$ one always tends to $(\tau_{\mathrm{bad}})^2$. This means that the noise sensitivity $\lim_{\sigma\to 0} MSE/\sigma^2$ diverges, and exact reconstruction is not possible even for very small noise. In this context $\tilde{\delta}(p_0)$ is the algorithmic threshold of AMP. The analogous quantity in our coding model is ϵ_{BP} and it the CW model it is the spinodal point.

This threshold is lower than the Lasso (or l_1) threshold derived in Chapter 13. This is not too surprising since the later concerns the worst case distribution for $p_0 \in \mathcal{F}_\epsilon$. It is instructive to compute the phase diagram and plot the optimal, Lasso and AMP phase transition lines in the (ϵ, δ) plane.

Let us now turn our attention to the coupled model. The performance is analyzed through the one dimensional recursion (18.14)-(18.15) which gives the evolution of the MSE profile $\tau_c^{(t)}$, as a function of time t and position along the

chain $c = 1, \dots, L$. For $\delta > \tilde{\delta}(p_0)$ the local MSE tends to $(\tau_{c,\text{good}})^2 = O(\sigma^2)$ uniformly along the chain. The advantage brought by spatial coupling appears for a sampling rate in the range $\epsilon < \delta < \tilde{\delta}(p_0)$. For $L \to +\infty$ and fixed $w \geq 2$ there is a $\tilde{\delta}(p_0, w) < \tilde{\delta}(p_0)$ such that for $\delta > \tilde{\delta}(p_0, w)$ the local MSE per position is bounded by $O(\sigma^2)$, and in particular the noise sensitivity remains finite. Because of the oversampling of the first few signal components, the MSE falls down to a level $O(\sigma^2)$ for these components, and then an estimation wave propagates along the chain. Eventually the local MSE converges to the good fixed point for all positions $\tau_{\text{good},c} = O(\sigma^2)$. Furthermore one observes that $\tilde{\delta}(p_0, w) \to \epsilon$ as $w \to +\infty$. In other words in the regime $N >> L >> w >> 1$ the dynamical AMP threshold saturates towards the optimal phase transition threshold. Figure **??** illustrates the phase diagram and the phase transition lines in teh (ϵ, δ) plane for various values of L and w.

18.3 K-SAT

For the random K-SAT problem we discussed several algorithms. The best one is BP-guided decimation. We described this algorithm and its empirical performance in Chapter 15. If we apply spatial coupling to this algorithm we see no boost in performance. This does not mean that spatial coupling does not help for this problem. It just means that BP-guided decimation is not the right setting for the nucleation phenomenon. The "right" setting is in fact a more sophisticated algorithm called *survey propagation*.

Rather than pursuing this avenue, let us go to a simpler algorithm, namely the UCP algorithm which we discussed in Chapter 14. We will see that spatially coupled formulas have a significantly higher threshold under UCP than uncoupled ones. Combined with the interpolation method this gives good lower bounds on the SAT/UNSAT threshold of uncoupled systems.

Construction

As for the case of coding, there are various ways of constructing coupled K-SAT formulas. E.g., Figure 18.7 shows the equivalent of a protograph ensemble for the case $K = 3$ where each clause at position i has exactly one connection to a variable at position i, $i + 1$, and $i + 2$.

For the purpose of analysis it is again more convenient to consider a random ensemble. As before, let w be a window size. Then, for each clause at position i and for each of its K connections we independently and uniformly pick a variable at a position in the range $[i, i + w - 1]$ and connect it to this variable with a uniformly chosen sign. This is the ensemble which we consider in the sequel.

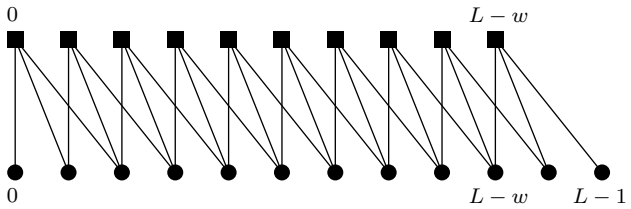


Figure 18.7 A "protograph"-like coupled K-SAT ensembles or $K = 3$.

Performance under the UCP Algorithm

Let us now focus on the UC algorithm for the coupled formulas. As for the un-coupled case, the UC algorithm consists of two main steps: free and forced. The operation of the algorithm at a forced step is clear: remove all the unit-clauses until no further unit-clause exists. However, at a free step, depending on how we might want to use the chain structure of the formula, we can have different *schedules* for choosing a free variable. For a coupled formula, the schedule within which we are choosing a variable in a free step is important

Consider for instance the following naive schedule – at a free step, pick a variable uniformly at random from all the remaining variables and fix it by flipping a coin. Computer experiments indicate that this naive schedule has no threshold gain compared to the un-coupled ensemble. This is not surprising since this schedule does not exploit the spatial (chain) structure of the formula. Hence, in order for the UC algorithm to have a threshold improvement over the coupled ensemble, we need to come up with schedules that exploit the additional spatial structure of the formula. We proceed by illustrating one such successful schedule.

In the very beginning of the algorithm, all the check nodes have degree K and there are no unit clauses. Hence, we are free to fix the variables in the first few steps of the algorithm. Let us fix the variables from the left-most position (i.e., the boundary). If we do this then we are creating in effect a seed at the boundary of the chain. Continuing this action at the free steps, we will eventually create unit clauses and at these forced steps a natural choice is just to clear all the unit clauses. However, when we are confronted with a free step, we will again try to help this seed to grow inside the chain, i.e., we always fix variables from the left-most possible position. Consequently, the schedule that we apply is as follows.

- At a *free step*, pick a variable randomly from the left-most position at which variables exists and fix it permanently by flipping a fair coin.
- At a *forced step*, remove unit clauses as long as they exist.

Computer experiments show that this schedule indeed exhibits a threshold improvement over the un-coupled ensemble. E.g., for the coupled 3-SAT problem, experiments suggest that the threshold of the UC algorithm is around 3.67. This

is a significant improvement compared to the threshold of UC for the un-coupled ensemble which is $\frac{8}{3}$.

To prove that indeed this schedule leads to this threshold we use again the Wormald method. This means, we write down a set of differential equations which describe the expected progress of the algorithm. Not surprisingly, the number of differential equations we need scales linearly in the chain length.

Phases, Types, and Rounds

For the coupled ensemble, the analysis of the evolution of UC is much more involved than the un-coupled ensemble. This is because of the fact that the schedule we have used prefers the left-most variable position in a free step. Hence, the number of variables in different positions will evolve differently. As an example, one can easily see that during the algorithm, the first position that all its variables are set is the left-most position (i.e., position 0). After the evacuation of position 0, position 1 becomes the left-most position of the graph and hence, the second position that becomes empty of variables is position 1. Continuing in this manner, the last position that is evacuated is position $L + w - 2$. With these considerations, we consider $L+w-1$ *phases* for this algorithm (see Figure 18.8). At phase $p \in \{0, 1, \cdots, L + w - 2\}$, all the variables at positions prior to p have been set permanently and as a result, at a free step we will pick a variable from position p.

This statistical asymmetry in the number of variables at each position also affects the the behavior of the number of check nodes in each position. As a result, we consider *types* for the check nodes. For instance, consider a degree two check node. It is easy to see that the probability that this degree two check node is hit (removed or shortened) is greatly dependent on the position of variables that it is connected to. This means that, dependent on the variable positions to which they are connected, we have different types of degree two check nodes. Clearly, the same statement holds for clauses of degree three, four, etc.

Let us now formally define the ingredients needed for the analysis. The notation we use here is slightly hard to swallow immediately. Thus, for the sake of maximum clarity, we try to uncover the details as smoothly as possible. We consider *rounds* for this algorithm. Each round consists of one free step followed by the forced steps that follow it. More precisely, at the beginning of each round we perform a free step and then we clear out all the unit-clauses as long as they exist (forced steps). We let time t be the number of rounds passed so far. This time variable will be called *round time*. The relation between t and the *natural time* (the total number of permanent fixes) is not linear. We also let $L_i(t)$ be the *number of literals* left in variable position $i \in \{0, 1, \cdots, L + w - 2\}$.

We now define the check types. Consider a coupled K-SAT formula to begin with. For such a formula there are L sets of check nodes placed at positions $\{0, 1, \cdots, L\}$. Let us consider a specific position $i \in \{0, 1, \cdots, L\}$ and look at the check nodes at position i. Each of these check nodes can potentially be connected

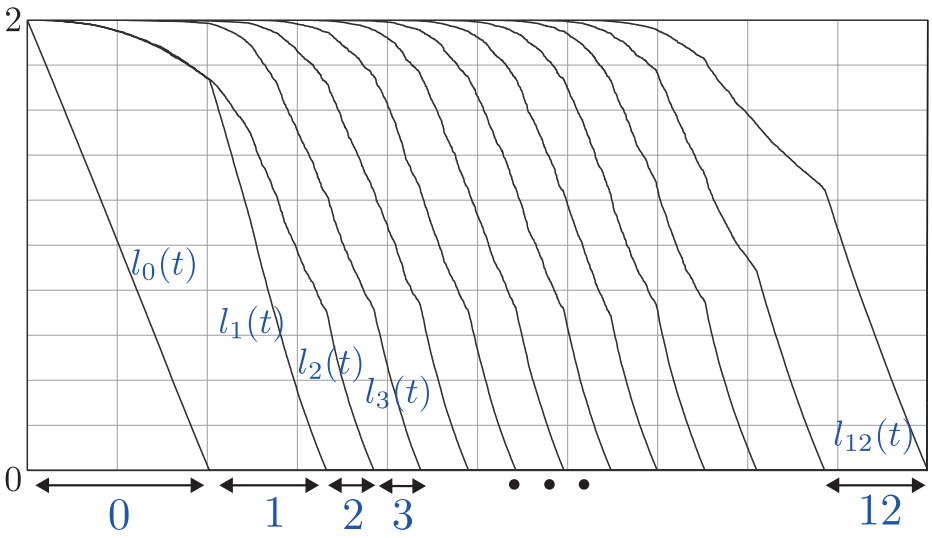


Figure 18.8 A schematic representation of how the literals at each of the positions vary in time. The horizontal axis corresponds to time t which is the number of free steps. Here we have $L = 11$ and $w = 3$. This plot corresponds to an implementation of the UC algorithm on a random coupled instance. The blue numbers below the plot are the phases of the algorithm. In the beginning of the algorithm, we are in phase 0. This phase lasts until all the literals in the first position are peeled off and as a result $\ell_0(t)$ reaches 0. We then go immediately to phase 1 and this phase lasts till $\ell_1(t)$ reaches 0 and so on. We have in total $L + w - 1 = 13$ phases.

to any set of K variables resting in variable positions $\{i, i+1, \cdots, i+w-1\}$. Some thought shows that there are various types of check nodes depending on the variable positions that they are connected to. For example, there is a type of check nodes for which all of the K edges go only into a single variable position $j \in \{i, i+1, \cdots, i+w-1\}$ or there is a type for with some of its edges go to position i and the rest go to position $i+1$ and so on. Also, as we proceed through the UC algorithm, some of these checks are shortened to create new types of checks with degrees less than K. We now explain a natural way to encode these various types.

By $C(t, i, \underline{\tau})$ we mean the number of check nodes at check position $i \in \{0, 1, \cdots, L\}$ that have type $\underline{\tau}$ at round tme t. The type $\underline{\tau} = (\tau_0, \cdots, \tau_{w-1})$ is a w-tuple and indicates that relative to position i, how many edges the check has in (variable) positions $i, i+1, \cdots, i+w-1$. The best way to explain $\underline{\tau}$ is through an example. Let us assume $w = 4$ and consider the set of check nodes at check position 20 that are only connected to variable positions $20, 22, 23$ in the following way. For each of these check nodes there are exactly two edges going to position 20, and 1 edge going to position 22 and 1 edge going to position 23 (thus each of these checks have degree 4). Figure 18.9 illustrates a generic check node of this set.

We denote the number of these checks at time t by $C(t, 20, (1, 0, 2, 1))$. In

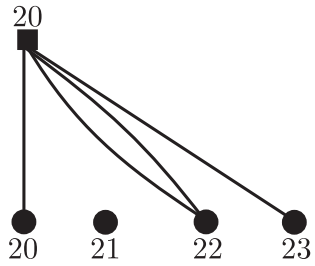


Figure 18.9 A schematic representation of checks which contribute to $C(t, 20, (1, 0, 2, 1))$. All the check nodes that contribute to $C(t, i, \underline{\tau})$, were initially (at time 0) degree K check nodes resting at check position i. However, the algorithm has evolved in a way that these check nodes have been deformed (possibly shortened or remained unchanged) to have a specific type $\underline{\tau}$.

other words, the type is computed as follows: the check position number that the check rests in is 20. This check is connected to a variable at position 20, and 2 variables at position 22, and a variable at position 23. So, relative to the check position 20, we see the edge-tuple $(1, 0, 2, 1)$. Let us now repeat and generalize: By $C(t, i, \underline{\tau})$ we mean the number of check nodes, at time t, which rest in position i, and $\underline{\tau}$ is a w-tuple that indicates relative to variable position i, the number of edges that go to positions $i, i+1, \cdots, i+w-1$, respectively. One can easily see that by summing up elements of the w-tuple $\underline{\tau} = (\tau_0, \cdots, \tau_{w-1})$, we find the degree of the corresponding check type. We denote the degree of a type $\underline{\tau}$ by $\deg(\underline{\tau})$. It is also easy to see that there are $\binom{d+w-2}{d-1}$ different types of degree d for $d \in \{2, 3, \cdots, K\}$. We are now ready to write the differential equations. Our approach is as follows. Assume the phase of the algorithm is p and we are in a round t. At a free step, we fix a variable at position p (free step). This will create a number of forced steps in each of the positions $p, p+1, \cdots, L+w-1$. We first compute the average of these forced fixes in each variable position as a function of the number of degree two check nodes. Using these averages, we then update the average number of check and variable nodes at each position. We proceed by explaining a key property for the analysis.

The Differential Equations

Now, having the vector $\underline{\beta}$ we can find how the number of variables and checks evolve. For all $i \geq 0$,

$$\Delta L_i(t) = L_i(t+1) - L_i(t) = -2\beta_i(t). \tag{18.16}$$

To see how the check types evolve, we note that for a given check type there are two kinds of flows to be considered. A negative flow going out and a positive flow coming in from the checks of higher degrees. In this regard, for a type $\underline{\tau} = (\tau_0, \cdots, \tau_{w-1})$ with $\deg(\underline{\tau}) < K$ let $\partial\underline{\tau}$ be the set of types of degree $\deg(\underline{\tau})+1$

such that by removing one edge from them we reach to the type $\underline{\tau}$. The set $\partial\underline{\tau}$ consists of w types which we denote by $\underline{\tau}^d$, $d \in \{0, 1, \cdots, w-1\}$, such that

$$\underline{\tau}^d = \underline{\tau} + (0, \cdots, \overset{d}{1}, \cdots, 0), \tag{18.17}$$

where $+$ denotes vector addition in the field of reals. Thus, if $\deg(\underline{\tau}) < K$, we obtain

$$\Delta C(t, i, \underline{\tau}) = -2 \sum_{d=0}^{w-1} \beta_{i+d}(t) \frac{\tau_d c(t, i, \underline{\tau})}{L_{i+d}(t)} + \sum_{d=0}^{w-1} (1 + \tau_d) \beta_{i+d}(t) \frac{C(t, i, \underline{\tau}^d)}{L_{i+d}(t)}. \tag{18.18}$$

The right-hand side of (18.18) has two parts. The first part corresponds to the flow that is going out of $C(t, i, \underline{\tau})$ and has negative sign. The right part is the incoming flow from the check nodes of higher degrees. In the case where $\deg(\underline{\tau}) = K$, we only have an outgoing flow since no check node with higher degrees exist. Hence, for the case $\deg(\underline{\tau}) = K$ we can write

$$\Delta C(t, i, \underline{\tau}) = -2 \sum_{d=0}^{w-1} \beta_{i+d}(t) \frac{\tau_d c(t, i, \underline{\tau})}{L_{i+d}(t)}. \tag{18.19}$$

We now write the initial conditions for the variables and check types. Firstly, note that $L_i(0) = 2N$. In the beginning of the algorithm, all checks are of degree K, thus for types $\underline{\tau}$ such that $\deg(\underline{\tau}) < K$, we have $C(0, i, \underline{\tau}) = 0$. For $\deg(\underline{\tau}) = K$ we have

$$C(0, i, \underline{\tau}) = \alpha N \frac{\binom{K}{\tau_0, \tau_1, \cdots, \tau_{w-1}}}{w^K}. \tag{18.20}$$

In order to write the differential equations, we re-scale the (round) time by N, i.e.

$$t \leftarrow \frac{t}{N}, \tag{18.21}$$

and also normalize all our other numbers by N, i.e.,

$$c(t, \cdot, \cdot) = \frac{C(Nt, \cdot, \cdot)}{N} \text{ and } \ell_i(t) = \frac{L_i(Nt)}{N}. \tag{18.22}$$

We then obtain for $i \in \{0, 1, \cdots, L + w - 2\}$,

$$\frac{d\ell_i(t)}{dt} = -2\beta_i(t). \tag{18.23}$$

For $i \in \{0, 1, \cdots, L-1\}$ and $\deg(\underline{\tau}) < K$ we have

$$\frac{dc(t, i, \underline{\tau})}{dt} = -2 \sum_{d=0}^{w-1} \beta_{i+d}(t) \frac{\tau_d c(t, i, \underline{\tau})}{\ell_{i+d}(t)} + \sum_{d=0}^{w-1} (1 + \tau_d) \beta_{i+d}(t) \frac{c(t, i, \underline{\tau}^d)}{\ell_{i+d}(t)}, \tag{18.24}$$

and otherwise if $\deg(\underline{\tau}) = K$ we have

$$\frac{dc(t, i, \underline{\tau})}{dt} = -2 \sum_{d=0}^{w-1} \beta_{i+d}(t) \frac{\tau_d c(t, i, \underline{\tau})}{\ell_{i+d}(t)}. \tag{18.25}$$

K	3	4	5
$\alpha_{\mathrm{UC}}(K)$	2.66	4.50	7.58
$\alpha_{\mathrm{UC},L=50,w=3}(K)$	3.67	7.81	15.76

Table 18.1 *First line*: The thresholds for UCP on the uncoupled ensemble. *Second line*: UCP threshold for a coupled chain with $w = 3$, $L = 50$.

The vector $\bar{\beta}$ is also found as follows. For p being the current phase, we have

$$\underline{\beta}(t) = (\beta_0(t), \cdots, \beta_{L+w-2}(t))^T = (I - A)^{-1} e_p, \tag{18.26}$$

where $A = [A_{i,j}]_{(L+w-1)(L+w-1)}$ has the form

$$A_{i,j} = \frac{1}{\ell_j(t)} \begin{cases} \sum_{k=i-w+1}^{i} 2c(t, k, \pi_{i-k,i-k}) & i = j, \\ \sum_{k=j-w+1}^{i} c(t, k, \pi_{i-k,j-k}) & 0 <\mid i - j \mid < w, \\ 0 & \text{otherwise} \end{cases} \tag{18.27}$$

Finally, the initial conditions are given by:

$$\begin{aligned} &\ell_i(0) = 2, \text{ for } 0 \le i \le L + w - 2 \\ &c(0, i, \underline{\tau}) = \begin{cases} \alpha \frac{\binom{K}{\tau_0, \tau_1, \cdots, \tau_{w-1}}}{w^K} & \text{if } \deg(\underline{\tau}) = K \text{ and } 0 \le i \le L - 1, \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{18.28}$$

Numerical Implementation

We have implemented the above set of differential equations in C. We define the threshold $\alpha_{\mathrm{UC},L,w}(K)$ as the highest density for which the spectral norm (largest eigenvalue) of the matrix A is strictly less than one throughout the whole algorithm. A practical point to notice here is that, for the sake of implementation, we assume a phase p finishes when its corresponding variable $\ell_p(t)$ goes below a (very) small threshold $\epsilon > 0$. In our implementations, we have typically taken $\epsilon = 10^{-5}$. However, it can be made arbitrarily small as long as the computational resources allow.

Table 18.1 shows the value of $\alpha_{\mathrm{UC},L,w}(K)$ with $L = 50$ and $w = 3$ for different choices of K. As we observe from Table 18.1, for the UC algorithm with the specific schedule mentioned above, there is a significant threshold improvement over the un-coupled ensemble.

For $L = 50, w = 3, K = 3$ and several values of α, we have plotted in Figure 18.10 the evolution of largest eigenvalue of A as a function of round time t.

In order to characterize analytically the ultimate threshold for the UC algorithm when L and w grow large, we proceed by further analyzing the set of differential equations.

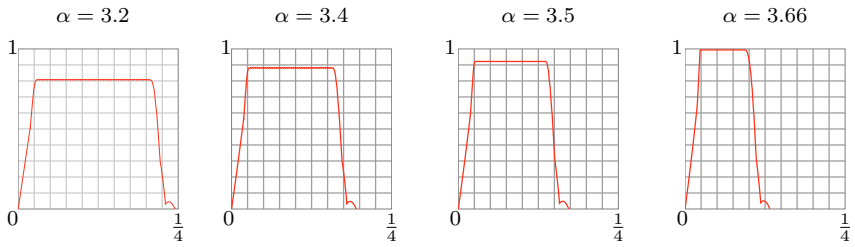


Figure 18.10 The largest eigenvalue of the matrix A, plotted versus the round time t (the number of rounds divided by the total number of variables NL). The plots correspond to an actual implementation of the UC algorithm for the 3-SAT coupled ensemble with $L = 50$ and $w = 3$. As we observe, for $\alpha < 3.67$, there is a gap between the largest eigenvalue of A and the value 1 throughout the UC algorithm. By increasing α this gap shrinks to 0. For $\alpha = 3.66$ (the right-most plot) this gap is around 0.006.

19 Variational Formulation and the Bethe Free Energy

In our previous lectures we have discussed how we can analyze the performance of various low-complexity algorithms, in particular algorithms of message-passing type. We have seen that in the limit of infinite system size, such algorithms have threshols and we were able to characterize these thresholds quantitatively. Such thresholds are often called *dynamical* thresholds since they are associated to the *dynamics* of a process (for us this is the algorithm).

But there is typically also a *static* phase transition. This corresponds to a phase transition which describes a change of the system behavior itself, independent of any algorithmic question. E.g., in coding we can ask how much noise we can add so that with high probability there is a unique codeword which is "compatible" with the received information. In communicatons jargon, this corresponds to the MAP threshold. For compressive sensing we can ask how the number of measurements has to scale with the number of unknowns so that with high probability there is a unique sparse vector which is compatible with the measurements. Finally, in K-SAT we can ask how many constraints we can have per Boolean variable so that with high probability a random formula is satisfiable. This is usally refered to as the SAT-UNSAT threshold.

Why are we interested in these quantities? Some systems are given to us and we cannot change them (e.g., K-SAT). In this case it is important to know how well a computationally unbounded system could do in order to gauge how well our algorithm is performing. But often we are actually in control of the system itself. E.g., think of the coding problem or also compressive sensing. It is typically us who designs the code or the measurement matrix. So in these cases it is important to know that the system itself is designed in such a way that at least in principle (if we had unbounded computational resources at our disposal) it has a good performance (comparable to the optimal one). E.g., in coding we can then compare the MAP threshold to the ultimate limit, namely the Shannon threshold and hopefully these two thresholds are close.

As we will see, there are two basic themes which appear. First, static thresholds are in general much harder to compute than the dynamical ones. This is why we have postponed this discusson towards the end. In a few cases we will be able to derive rigorous quantitative statements. In some other ones, we will have to be content with computations which are believed to yield the correct value but fall short of a mathematical proof. The second, perhaps more surprising theme

is that the analysis of the static threshold can often be done by looking at the behavior of the message-passing algorithm! Why message-passing, a sub-optimal algorithm, should have any bearing on the behavior of the optimal algorithm is at first glance puzzling.

As we will see, the key object which connects these two themese is the so-called Bethe free energy. It is an "approximation" to the true free energy which itself depends on the fixed points of the message-passing algorithm. In some instances the static thresholds predicted by the Bethe free energy can be shown to be indeed correct.

Let us discuss this in more detail. Computing the true free energy for general graphical models (or statistical mechanics models) is an impossible task. An important approximation philosophy is the so-called "mean-field theory." In this theory, when looking at the interactions of a "spin" with the rest of the system, we only take into account very close neighbors exactly, but model influences of the remaining system simply by a "mean field," i.e., a field which models the average influence of this part of the system. For models defined on sparse graphs that are locally tree-like, a very good form of mean filed theory was developed by Bethe and Peierls. This leads to the so-called Bethe free energy approximation. We note that this is already a "sophisticated" version of the most basic mean field theory.

As we will see the Bethe-Peierls theory involves fixed point equations that are the same as those occurring in Belief-Propagation. Their use and to some extent interpretation are however different. Note that there is a clash of initials (BP) that is solely due to an historical accident. We hope that this will not cause major confusions.

In this chapter we treat in detail the case of graphical models with a discrete alphabet $\mathcal{X}$. As a direct application we will look more closely at the cases of coding and K-SAT. For models with a continuous alphabet such as those occuring in the context of compressive sensing the ideas are conceptually the same, but the calculations have to be slightly adapted. We consider a general Gibbs measure of the form

$$\mu\left(\underline{s}\right) = \frac{1}{Z}\prod_{a} f_a\left(x_{\partial a}\right), \tag{19.1}$$

where the variables $x_i \in \mathcal{X}$, $i = 1, \ldots, n$ and f_a, $a = 1, \ldots, m$ are kernel functions associated to check nodes which depend on $x_{\partial a} = \{x_i, i \in \partial a\}$. In Chapter 7 we discussed the sum-product algorithm that computes *BP-marginals* for such measures. Recall that these are the *exact marginals* when the graph is a tree. Similarly we will see that on a tree the free energy

$$f = -\frac{1}{\beta n}\ln Z, \tag{19.2}$$

can be expressed exactly in terms of the marginals of the measure. This is the starting point of the formalism developped in this chapter.

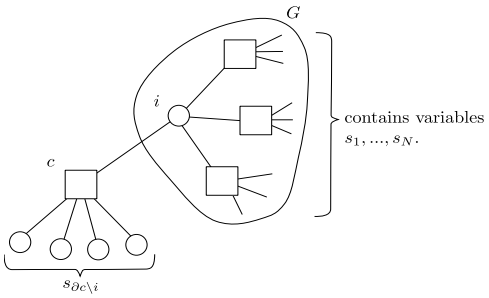


Figure 19.1 Induction procedure: G is the original tree to which we add check c connected to i such that the new graph is a tree

19.1 The Gibbs measure on trees

Consider the (exact) marginals

$$\nu_i(x_i) = \sum_{\sim x_i} \mu(x_1, ..., x_N), \qquad \nu_a(x_{\partial a}) = \sum_{\sim x_{\partial a}} \mu(x_1, ..., x_N).$$

As explained in Chapter 7 on a tree these can be computed exactly by the sum-product algorithm. More is true.

Lemma 19.1.1 The Gibbs measure on a tree can be expressed in terms of its marginals as follows,

$$\mu(\underline{x}) = \prod_a \nu_a(x_{\partial a}) \prod_i (\nu_i(x_i))^{1-d_i} \tag{19.3}$$

where d_i is the degree of node i.

Proof We prove (19.3) by induction over number the number m of check nodes. For $m = 1$ the unique clause is connected to variable nodes with $d_i = 1$. Thus (19.3) is trial in this case. Now, we assume (19.3) is true for a tree graph G with m check nodes and prove that it also holds for the new Gibbs measure

$$\mu_{\text{new}}(x_{\partial c \setminus i}, x_1, ..., x_n) = \frac{1}{Z_{\text{new}}} f_c(x_{\partial c}) \prod_a f_a(x_{\partial a}) \tag{19.4}$$

obtained when one adds one check node c connected to a variable node i in such a way that the new graph[1] is a tree. The original tree G and the new tree are depicted on figure19.1

Consider the conditional probability $\Pr(x_{\partial c \setminus i} \mid x_1, ..., x_n)$ of an assignement $x_{\partial c \setminus i}$ given $x_1, \ldots, x_n$. We observe that

$$\begin{aligned} \Pr(x_{\partial c \setminus i} \mid x_1, ..., x_n) &= \Pr(x_{\partial c \setminus i} \mid x_i) \\ &= \frac{\nu_{\text{new},c}(x_{\partial c})}{\nu_{\text{new},i}(x_i)}. \end{aligned}$$

[1] We do not discuss the somewhat trivial case where the new check is disconnected.

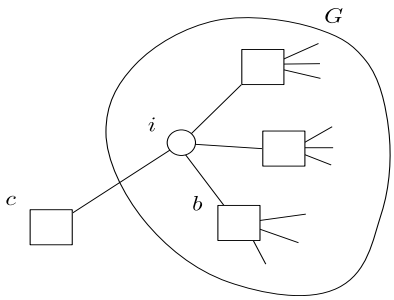


Figure 19.2 Factor graph for the marginal distribution (19.6). We select an arbitrary check $b \in \partial i \setminus c$.

Therefore, denoting by $\nu_{\text{new}}(x_1, ..., x_n)$ the marginalisation of (19.4) over the variables $x_{\partial c \setminus i}$,

$$\begin{aligned}\mu_{\text{new}}\left(x_{\partial c \setminus i}, x_1, ..., x_n\right) &= \Pr\left(x_{\partial c \setminus i} \mid x_1, ..., x_N\right) \nu_{\text{new}}(x_1, ..., x_n) \\ &= \nu_{\text{new},c}(x_{\partial c}) \left(\nu_{\text{new},i}(x_i)\right)^{-1} \nu_{\text{new}}(x_1, ..., x_n). \end{aligned} \tag{19.5}$$

Now, by definition of $\nu(x_1, ..., x_n)$ we have

$$\begin{aligned}\nu_{\text{new}}(x_1, ..., x_N) &= \frac{1}{Z_{\text{new}}} \sum_{x_{\partial c \setminus i}} f_c(x_{\partial c}) \prod_a f_a(x_{\partial a}) \\ &= \frac{1}{Z_{\text{new}}} \widetilde{f}_c(x_i) \prod_a f_a(x_{\partial a}). \end{aligned} \tag{19.6}$$

where we have set $\sum_{x_{\partial c \setminus i}} f_c(x_{\partial c}) = \widetilde{f}_c(x_i)$. This distribution has the factor graph depicted on figure 19.2 This tree still has $m+1$ check nodes. However c can be absorbed in any arbitrarily selected check $b \in \partial i \setminus c$:

$$\begin{aligned}\nu_{\text{new}}(x_1, ..., x_n) &= \frac{1}{Z_{\text{new}}} \widetilde{f}_c(x_i) \prod_a f_a(x_{\partial a}) \\ &= \frac{1}{Z_{\text{new}}} \widetilde{f}_c(x_i) f_b(x_{\partial b}) \prod_{a \neq b} f_a(x_{\partial a}) \\ &= \frac{1}{Z_{\text{new}}} \widetilde{f}_b(x_{\partial b}) \prod_{a \neq b} f_a(x_{\partial a}) \end{aligned}$$

where we have set $\widetilde{f}_c(x_i) f_b(x_{\partial b}) = \widetilde{f}_b(x_{\partial b})$. We recognize this expression as a Gibbs measure defined on a tree with m check nodes, so that we can apply the induction hypothesis

$$\nu_{\text{new}}(x_1, \ldots, x_n) = \prod_a \tilde{\nu}_{\text{new},a}(x_{\partial a}) \prod_i \left(\nu_{\text{new},i}(x_i)\right)^{1-d_i}.$$

Here $\nu_{\text{new},a}$ and $\nu_{\text{new},i}$ are the marginals of ν_{new}. But clearly, they are also the marginals of ν_{new} in (19.4). Combining this last formula with (19.5) yields the desired result. □

19.2 The free energy on trees

We begin with a general and important expression for the free energy which is universally valid, and in particular is not restricted to trees. This formula is best understood when the Gibbs measure (19.1) is expressed in its traditional physics form

$$\mu\left(\underline{x}\right)=\frac{1}{Z}\exp\left(-\beta\mathcal{H}\left(\underline{x}\right)\right). \tag{19.7}$$

The formal relation between the Hamiltonian and the kernel functions is

$$\beta\mathcal{H}\left(\underline{x}\right)=-\sum_{a}\ln f_{a}(x_{\partial a}) \tag{19.8}$$

Replacing (19.7) in the definition of the freee energy (19.2) one easily finds for the un-normalized free energy $F\equiv nf$,

$$F=\left\langle \mathcal{H}\right\rangle -\beta^{-1}S\left[\mu\right] \tag{19.9}$$

where

$$\begin{aligned}\left\langle \mathcal{H}\right\rangle &=\sum_{x_{1},...,x_{N}}\mathcal{H}\left(x_{1},...,x_{N}\right)\mu\left(x_{1},...,x_{N}\right)\\ S\left[\mu\right] &=-\sum_{x_{1},...,x_{N}}\mu\left(x_{1},...,x_{N}\right)\ln\mu\left(x_{1},...,x_{N}\right).\end{aligned}$$

Here $\left\langle \mathcal{H}\right\rangle$ is the average value of the Hamiltonian. Physically this represents the total average internal energy that the system possesses, and is commonly called the internal energy. $S[\mu]$ is called the Gibbs entropy. This is nothing else than a special form of Shannon's entropy written down for the Gibbs measure. In thermodynamics one shows that the free energy is the amount of work that a system can perform. Equ. (19.9) says that this is equal to the the total internal energy minus an unsuable part equal given by the temperature times the entropy.

We now apply formula (19.9) to the Gibbs measure on a tree graph. This leads to

PROPOSITION 19.1 On a tree graphical model the (un-normalized) free energy $F=nf$ can be expressed in terms of its marginals as

$$F=\sum_{a}\sum_{x_{\partial a}}\nu_{a}\left(x_{\partial a}\right)\ln\frac{\nu_{a}\left(x_{\partial a}\right)}{f_{a}\left(x_{\partial a}\right)}+\sum_{i}\left(1-d_{i}\right)\sum_{x_{i}}\nu_{i}\left(s_{i}\right)\ln\nu_{i}\left(x_{i}\right) \tag{19.10}$$

Proof Using (19.8) the internal energy contribution yields

$$\begin{aligned}\left\langle \mathcal{H}\right\rangle _{\mu} &=-\sum_{a}\sum_{x_{1},...,x_{N}}\mu\left(x_{1},...,x_{N}\right)\ln f_{a}(x_{\partial a})\\ &=-\sum_{a}\sum_{x_{\partial a}}\nu\left(x_{\partial a}\right)\ln f_{a}\left(x_{\partial a}\right).\end{aligned}$$

Note that this formula is completely general and does not depend on having a tree graph.

To compute the contribution of the enetropy we use (19.3) in lemma 19.1.1. This gives

$$\begin{aligned} S[\mu] &= -\sum_a \sum_{x_1,\ldots,x_N} \mu(x_1,\ldots,x_N)\left(\ln \nu_a(x_{\partial a})\right) \\ &\quad + \sum_i (1-d_i) \sum_{x_1,\ldots,x_N} \mu(x_1,\ldots,x_N) \ln\left(\nu_i(x_i)\right) \\ &= -\sum_a \sum_{x_{\partial a}} \nu_a(x_{\partial a}) \ln \nu_a(x_{\partial a}) + \sum_i (1-d_i) \sum_{x_i} \nu_i(x_i) \ln \nu_i(x_i) \end{aligned}$$

Combining the the energetic and entropic contributions gives (19.10) □

In chapter 7 we learned how to compute the marginals in terms an exact message passing equations on the tree. Recall that we have two types of messages: those flowing from variable to check nodes $\mu_{i\to a}(x_i)$ and those flowing from check to variables node $\mu_{a\to i}(x_i)$. The exact marginals are given by

$$\begin{aligned} \nu_i(x_i) &= \frac{\prod_{a\in\partial i} \hat{\mu}_{a\to i}(x_i)}{\sum_{x_i} \prod_{a\in\partial i} \hat{\mu}_{a\to i}(x_i)} \\ \nu_a(x_{\partial a}) &= \frac{f_a(x_{\partial a}) \prod_{i\in\partial a} \mu_{i\to a}(x_i)}{\sum_{x_{\partial a}} f_a(x_{\partial a}) \prod_{i\in\partial a} \mu_{i\to a}(x_i)}. \end{aligned}$$

and the messages by the sum-product equations by

$$\begin{aligned} \mu_{i\to a}(x_i) &= \prod_{b\in\partial i\setminus a} \hat{\mu}_{b\to i}(x_i) \\ \hat{\mu}_{a\to i}(x_i) &= \sum_{\sim x_i} f_a(x_{\partial a}) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(x_j) \end{aligned}$$

Moreover the messages are uniquely defined by their "initial" values at the leaf nodes. Recall, when the leaf node is a check the outgoing message equals $f_a(x_{\partial a})$ when the leaf node is a check, and equals 1 when the leaf node is a variable.

Using these expressions in (19.10), a straightforward calculation leads to the alternative expression for the free energy

PROPOSITION 19.2 On a tree graphical model the (un-normalized) free energy $F = nf$ can be expressed in terms of the BP messages as a sum of three contributions associated to variable nodes, check nodes and edges

$$F = \sum_i F_i + \sum_a F_a - \sum_{(i,a)} F_{ia},$$

where the three contributions are

$$F_i = \ln\left\{\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}_{b\to i}\left(x_i\right)\right\}$$

$$F_a = \ln\left\{\sum_{x_{\partial a}} f_a\left(x_{\partial a}\right)\prod_{i\in\partial a}\mu_{i\to a}\left(x_i\right)\right\}$$

$$F_{ia} = \ln\left\{\sum_{x_i}\mu_{i\to a}\left(x_i\right)\hat{\mu}_{a\to i}\left(x_i\right)\right\}$$

We stress that in this formula the messages do not have to be normalized. Indeed they were not normalized in the first place in the sum-product equations. The anxious reader can check that F is invariant under the renormalizations $\hat{\mu}_{a\to i}\to\hat{z}_{a\to i}\hat{\mu}_{a\to i}$ and $\mu_{i\to b}\to\hat{z}_{i\to a}\mu_{i\to a}$ for any arbitrary numbers $\hat{z}_{a\to i}$ and $z_{i\to a}$.

19.3 Bethe free energy for general graphical models

We now turn our attention to general graphical models of the type (19.1) with a factor graph that is not necessarily a tree, and introduce a definition. We assign to each edge two ditributions $\mu_{i\to a}(s_i)$ and $\mu_{a\to i}(s_i)$. The set of all distributions forms two vectors denoted by μ and $\underline{\hat{\mu}}$. The notation is the same than for the BP messages for reasons that will become clear, however the reader should bear in mind that conceptually these are general distributions, not necessarily equal to the BP messages (for one thing the BP equations do not necessarily have a unique solution). The *Bethe free energy* is by definition the functional

$$F_{\text{Bethe}}\left[\underline{\mu},\underline{\hat{\mu}}\right] = \sum_i F_i\left[\{\mu_{i\to b}, b\in\partial i\}\right] + \sum_a F_a\left[\{\mu_{i\to a}, i\in\partial a\}\right] - \sum_{ai} F_{ai}\left[\{\mu_{i\to a},\hat{\mu}_{a\to i}\}\right]. \tag{19.11}$$

with the three contributions associated to variable and check nodes, and edges.

$$F_i = \ln\left\{\sum_{s_j}\prod_{b\in\partial i}\hat{\mu}_{b\to i}\left(s_i\right)\right\} \tag{19.12}$$

$$F_a = \ln\left\{\sum_{s_{\partial a}} f_a\left(s_{\partial a}\right)\prod_{i\in\partial a}\mu_{i\to a}\left(s_i\right)\right\} \tag{19.13}$$

$$F_{ai} = \ln\left\{\sum_{s_i}\mu_{j\to a}\left(s_i\right)\hat{\mu}_{a\to j}\left(s_i\right)\right\}. \tag{19.14}$$

what is the idea behind this definition? The Bethe free energy exactly gives the true free energy for factor graphs that are trees. For a loopy factor graph it may seem a reasonable idea to propose the Bethe free energy as an ansatz (an educated guess) that hopefully approximates the true one. However there

are various problems that immediately arise. The most urgent is: how does one choose the messages? The BP equations do not necessarily have a unique solution for loopy graphs. The rule of thumb is to take the messages that minimize the Bethe functional. Were does this rule of thumb come from? In the standard physics variationnal approaches the true free energy is always lower than the ansatz. Then minimizing the ansatz over a set of open parameters is the best possible choice. This is not true for the Bethe free energy, so the usual rule of thumb has be considered with a grain of salt. We stress that there is no general inequality that states that the true free energy is always smaller than the Bethe functional. In general, quantifying the difference between the true and minimal bethe free energy is a hard problem about which we do not know much.

The discussion above suggests that a first important step is to look at stationnary points of the Bethe functional. One then discovers the follwing important result.

PROPOSITION 19.3 The stationary points of the Bethe free energy satisfy the sum-product message passing equations and conversely the solutions of the sum-product equations are stationary points of the Bethe free energy.

Proof For a finite system with a discrete alphabet the Bethe free energy functional is really a function of many variables, namely $\mu_{i\to a}(x_i)$, $\mu_{a\to i}(x_i)$ for $x_i \in \mathcal{X}$. Thus the stationnarity conditions are simply

$$\frac{\partial F_{\text{Bethe}}}{\partial \mu_{i\to a}(x_i)} = 0, \qquad \frac{\partial F_{\text{Bethe}}}{\partial \widehat{\mu}_{a\to i}(x_i)} = 0$$

For the first derivative there is a contribution from F_a and F_{ia},

$$\frac{\partial F_{\text{Bethe}}}{\partial \mu_{i\to a}(x_i)} = \frac{\hat{\nu}_{a\to i}(x_i)}{\sum_{x_i} \mu_{i\to a}(x_i)\,\hat{\mu}_{a\to i}(x_i)} - \frac{\sum_{\sim x_i} f_a(x_{\partial a}) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(x_j)}{\sum_{x_{\partial a}} f_a(x_{\partial a}) \prod_{j\in\partial a} \mu_{j\to a}(x_j)},$$

and for the second one the contribution comes from F_i and F_{ia},

$$\frac{\partial F_{\text{Bethe}}}{\partial \widehat{\mu}_{a\to i}(x_i)} = \frac{\nu_{i\to a}(x_i)}{\sum_{x_i} \mu_{i\to a}(x_i)\,\hat{\mu}_{a\to i}(x_i)} - \frac{\prod_{b\in\partial i\setminus a} \hat{\mu}_{b\to i}(x_i)}{\sum_{x_i} \prod_{b\in\partial i} \hat{\mu}_{b\to i}(x_i)}.$$

If we set the two derivatives to zero we find

$$\begin{aligned}
\widehat{\mu}_{a\to i}(x_i) &\propto \sum_{\sim x_i} f_a(x_{\partial a}) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(x_j) \\
\mu_{i\to a}(x_i) &\propto \prod_{b\in\partial i\setminus a} \widehat{\mu}_{b\to i}(x_i).
\end{aligned}$$

which are equivalent to the sum-product equations. Conversely it is easy to revert these calculations and show that the sum-product equations imply the stationnarity condition. □

19.4 Application to coding

We explained in Chapter 7 that the posterior measure used for MAP decoding is

$$\frac{1}{Z(\underline{h})}\prod_a \frac{1}{2}(1+\prod_{i\in\partial a} s_i)\prod_{i=1}^n e^{h_i s_i}.$$

where $s_i \in \mathcal{X} = \{-1,+1\}$. There are two types of kernel functions

$$f_i(s_i) = e^{h_i s_i}, \quad \text{and} \quad f_a(\{s_i, i\in\partial a\}) = \frac{1}{2}(1+\prod_{i\in\partial a} s_i), \tag{19.15}$$

associated to leaf checks and usual parity checks. An example with the corresponding factor graph is shown in figure 7.6.

The messages flowing on edges connecting variable nodes and parity checks can be parametrized as

$$\mu_{i\to a}(s_i) \propto e^{h_{i\to a}s_i}, \qquad \widehat{\mu}_{a\to i}(s_i) \propto e^{\widehat{h}_{a\to i}s_i} \propto 1 + s_i \tanh \widehat{h}_{a\to i}.$$

The messages flowing on edges connecting leaf checks and variable nodes are

$$e^{h_i s_i}, \qquad \prod_{a\in\partial i} e^{\widehat{h}_{a\to i}s_i} \propto \prod_{a\in\partial i}(1+s_i\tanh\widehat{h}_{a\to i}).$$

As pointed out above the normalization factors of teh messages cancel out in The Bethe free energy. This is why our parametrization only involves proportionality relations.

Replacing these messages in expressions (19.12)-(19.14) it is possible to perform exactly all sums over the spins, and express the Bethe free energy as function of $(\underline{h},\underline{\hat{h}}) = \{h_{i\to a}, \hat{h}_{a\to i}\}$. We give the main steps of this calculation. From (19.12) the contribution of variable nodes is

$$\begin{aligned} F_i &= \ln\Big\{\sum_{s_i=\pm 1} e^{h_i s_i}\prod_{a\in\partial i}(1+s_i\tanh\hat{h}_{a\to i})e^{h_i s_i}\Big\} \\ &= \ln\Big\{e^{h_i}\prod_{a\in\partial i}(1+\tanh\hat{h}_{a\to i}) + e^{-h_i}\prod_{a\in\partial i}(1-\tanh\hat{h}_{a\to i})\Big\}. \end{aligned} \tag{19.16}$$

From (19.13), for parity checks we have

$$F_a = \ln\Big\{\sum_{s_{\partial a}}\frac{1}{2}(1+\prod_{i\in\partial a} s_i)\prod_{i\in\partial a}(1+s_i\tanh h_{i\to a})\Big\}.$$

Observe that

$$\begin{aligned} \sum_{s_{\partial a}}\prod_{i\in\partial a}(1+s_i\tanh h_{i\to a}) &= \prod_{i\in\partial a}\sum_{s_i=\pm 1}(1+s_i\tanh h_{i\to a}) \\ &= 2^{|\partial a|} \end{aligned}$$

and

$$\sum_{s_{\partial a}} \prod_{i\in\partial a} s_i \prod_{i\in\partial a} (1 + s_i \tanh h_{i\to a}) = \prod_{i\in\partial a} \sum_{s_i=\pm 1} (s_i + \tanh h_{i\to a}) \\ = 2^{|\partial a|} \prod_{i\in\partial a} \tanh h_{i\to a}.$$

Now we compute the contribution of checks. The contribution of parity checks is

$$F_a = \ln\left\{\frac{1}{2}(1 + \prod_{i\in\partial a} \tanh h_{i\to a})\right\} + |\partial a| \ln 2. \tag{19.17}$$

There is also a contribution from leaf check nodes that happens to be given by (19.16), and also happens to cancel with the contribution of edges connecting variable and leaf check nodes. There remains the contribution of edges connecting variable and parity check nodes

$$F_{ai} = \ln\left\{\sum_{s_i=\pm 1} (1 + s_i \tanh h_{i\to a})(1 + s_i \tanh \hat{h}_{a\to i})\right\} \\ = \ln\left\{1 + \tanh h_{i\to a} \tanh \hat{h}_{a\to i}\right\} + \ln 2. \tag{19.18}$$

The Bethe free energy is given by the sum of the three types of contributions (19.16), (19.17) and (19.18)

$$F_{\text{Bethe}}(\underline{h}, \underline{\hat{h}}) = \sum_i \ln\left\{e^{h_i} \prod_{a\in\partial i} (1 + \tanh \hat{h}_{a\to i}) + e^{-h_i} \prod_{a\in\partial i} (1 - \tanh \hat{h}_{a\to i})\right\} \\ + \sum_a \ln\left\{\frac{1}{2}(1 + \prod_{j\in\partial a} \tanh\ h_{j\to a})\right\} \\ + \sum_{ai} \ln\left\{1 + \tanh h_{i\to a} \tanh \hat{h}_{a\to i})\right\} \tag{19.19}$$

As an exercise the reader can check that the stationary points of the Bethe functional satisfy the BP equations, in other words

$$\begin{cases} h_{i\to a} = h_i + \sum_{b\in\partial i\setminus a} \hat{h}_{b\to i} \\ \hat{h}_{a\to i} = \tanh^{-1}\left\{\prod_{j\in\partial a\setminus i} \tanh\ h_{j\to a}\right\} \end{cases}$$

We will see that the average over the channel outputs and the graph ensemble of the Bethe free energy allows to derive the so-called replica-symmetric (RS) formula for the average free energy[2]. It is known that for a large class of LDPC codes and BMS channels the RS free energy is equal to the exact free energy.

[2] The adjective "replica-symmetric" is due to historical reasons. indeed these formulas were first derived thanks to the so-called replica method which we do not cover in this course. The approach of the replica method is algebraic in nature but mathematically more mysterious.

In particular it allows to correctly predict the MAP noise threshold. In the next chapters we will derive the RS formula with the specific application of the BEC in mind, and partly prove that the RS formula is exact.

19.5 Application to compressive sensing

To do.

19.6 Application to K-SAT

Recall from Chapter 4 the partition function of K-SAT (at finite temperature) which counts the number of solutions.

$$Z = \sum_{s_1,\dots,s_n \in \{-1,+1\}^n} \prod_{a=1}^{M} \Big(1 - (1 - e^{-\beta}) \prod_{i \in a} (\frac{1 + s_i J_{ia}}{2}) \Big). \tag{19.20}$$

The Bethe free energy here serves as a first ansatz for $-(\beta n)^{-1} \ln Z$. Recall that for $\beta = +\infty$, Z counts the number of solutions. Thus as long as there exist at least one solution and $\ln Z$ is well defined for $\beta = +\infty$ one can also use the Bethe formula to write down an ansatz for the entropy of the uniform measure over solutions (the Boltzman entropy!).

To compute the Bethe free energy we replace the kernel function

$$f_a(\{x_i, i \in \partial a\}) = 1 - (1 - e^{-\beta}) \prod_{i \in a} (\frac{1 + s_i J_{ia}}{2}).$$

in (19.12)-(19.14) and use the parametrization (15.8) introduced in Chapter 15. Let $\partial_{J_{ia}} i$ the the set of checks connected to i by an edge such that $J_{ia} = -1$ (dashed) or $J_{ia} = 1$ (full). The resulting expressions are easily found to be

$$F_{\text{Bethe}}(\underline{h}, \underline{\hat{h}}) = \sum_i F_i(\{h_{j \to a}, j \in \partial a\}) + \sum_a F_a(\{\hat{h}_{b \to i}, i \in \partial b\}) \tag{19.21}$$

$$- \sum_{ia} F_{ia}(h_{i \to a}, \hat{h}_{a \to i}) \tag{19.22}$$

with

$$F_i = \ln\Bigg\{ \prod_{a\in\partial_- i} (1-\tanh\hat{h}_{a\to i}) \prod_{a\in\partial_+ i} (1+\tanh\hat{h}_{a\to i}) + \prod_{a\in\partial_- i} (1+\tanh\hat{h}_{a\to i}) \prod_{a\in\partial_+ i} (1-\tanh\hat{h}_{a\to i}) \Bigg\} \tag{19.23}$$

$$F_a = \ln\left\{ 1-(1-e^{-\beta}) \prod_{i\in\partial a} \frac{1-\tanh h_{i\to a}}{2} \right\} \tag{19.24}$$

$$F_{ai} = \ln\left\{ 1+\tan h_{i\to a} \tanh\hat{h}_{a\to i} \right\} \tag{19.25}$$

Again, the reader can easily check that the stationnary points of $F_{\rm Bethe}(\underline{h},\underline{\hat{h}})$ satisfy the BP equations presented in Chapter 15 ((15.11)-(15.15) are written down for $\beta=+\infty$).

In the next chapter we discuss an important application of these formulas. When $-\beta F_{\rm Bethe}[\underline{\xi},\underline{\hat{\xi}}]/n$ is averaged over the graph ensemble one get a specific prediction for the entropy of the K-SAT ensemble. This prediction is not consistent with rigorous upper bounds on the SAT-UNSAT threshold. This means that the Bethe formulas and the corresponding BP equations are not good enough to inform us on the SAT-UNSAT transition. But this is not the end of the story. We will see that it is necessary to further develop the approach taken in this chapter and wander into the cavity method.

20 Replica Symmetric Free Energy Functionals

The main idea behind density or state evolution analysis of message passing algorithms is to track their average behaviour. This allows to analyze their performance and derive their algorithmic (or dynamic) phase transition thresholds. But we also saw that one can guess the (static) phase transition threshold through a Maxwell construction. For example for coding, at least for the BEC, we defined an EXIT curve computable from DE, on which a Maxwell construction gives the MAP threshold. However we did not provide any clear general principle for deciding what are the correct variables[1]for which the Maxwell construction works. For the CW model the guess was quite trivial, for the BEC and compressive sensing it was less so. For K-SAT we have to postpone the discussion after the cavity method is introduced.

We will see in this chapter that by carrying the variational approach one step further we will be able to provide some clues for these questions. In particular we will be able to provide certain guiding lines determining the static phase transition threshold, and the variables on which the Maxwell construction works. In fact the variational approach allows to reformulate the Maxwell construction in a less ambiguous and useful way.

We have seen that the sum-product or BP equations are the stationarity conditions for the Bethe free energy. We will see in this chapter that the density and state evolution equations are the stationarity conditions of an averaged form of an averaged form of the Bethe free energy. This averaged form is called the *replica symmetric free energy functional*. The adjective "replica symmetric" mostly comes from historical reasons but, it has a meaning which we will explain once we have gone through the cavity method. We will explain how this functional allows to predict the algorithmic as well as static phase transition thresholds. Until recently this prediction was rigorously proved only in somewhat special cases or was supported by bounds. Recent proof techniques such as the interpolation method and spatial coupling have allowed to provide relatively simple and intuitive proofs in the cases of coding and compressive sensing. Such proof techniques are the subject of chapter 21. For K-SAT we will see that the predictions of the replica symmetric free energy functional are wrong. In-

[1] In physics parlance determining the "correct variables" for the description of a phase transition is part of a more general and deep problem, called the determination of the *order parameter* (see notes).

stead of being a curse this makes the subject even more fascinating. We will see in Chapter 22 that the correct thresholds and Maxwell constructions are given by pushing the notions of Bethe and replica free energy functionals "one level up". That these predictions are correct for K-SAT and other similar constraint satisfaction problems is still an open and alive problem.

We refrain from giving a completely general definition of the replica symmetric free energy functional because this immediately leads to cumbersome notations. Rather we directly treat our three paradigms in the next paragraphs. In fact each one has its own features and going through each of them allows to cover most essential cases.

20.1 Coding

We first discuss the general definition of the replica symmetric free energy functional for the regular Gallager (l, r) ensemble over a BMS channel $p_{Y|X}$, and then specialize to the case of the BEC where the functional simply becomes a function of a real variable. Recall the notation $c(.)$ for the distribution of half-loglikehod ratios $h(y) = \frac{1}{2} \ln p_{Y|X}(y|1)/p_{Y|X}(y|-1)$.

Replica symmetric functionals for BMS channels

The main idea is to pretend that in expression (19.19) the messages $h_{i\to a}$, are iid random variables distributed according to a trial distribution x(.), and that $\hat{h}_{a\to i}$ are dependent random variables defined through the BP equation

$$\hat{h}_{a\to i} = \tanh^{-1}\Big\{ \prod_{j\in\partial a\setminus i} \tanh\ h_{j\to a} \Big\}$$

Then one averages (19.19) which yields a functional of $x(.)$.

Let us give the formal definition. Here x(.) is a fixed trial probability distribution over $\mathbb{R}$. Pick r iid copies of $H \sim$ x(.), and call them H_k, $k = 1, \dots r$. Let

$$\hat{H} = \tanh^{-1}\{ \prod_{k=1}^{r} \tanh\ H_j \} \tag{20.1}$$

Pick l iid copies $\hat{H}_k$, $k = 1, \dots l$. Let

$$\begin{aligned} f(h, \underline{H}, \underline{\hat{H}}) = &\ln\Big\{ e^{h} \prod_{k=1}^{l} (1 + \tanh \hat{H}_k) + e^{-h} \prod_{a=1}^{k} (1 - \tanh \hat{H}_k) \Big\} \\ &+ \frac{l}{r} \ln \frac{1}{2}\Big\{ 1 + \prod_{k=1}^{r} \tanh H_k \Big\} - l \ln\Big\{ 1 + \tanh H \tanh \hat{H} \Big\} \end{aligned}$$

The RS free energy functional is defined as:

$$f_{\rm RS}[\mathrm{x}(.)] = \mathbb{E}[f(h, \underline{H}, \hat{\underline{H}}]$$

where the expectation is with respect to $h \sim c(.)$ and $\underline{H} \sim \mathrm{x}(.)$ (and $\hat{\underline{H}} \sim \hat{\mathrm{x}}(.)$ the induced distribution that depends on $\mathrm{x}(.)$). For an irregular LDPC ensemble (l, r) are random and one has an extra average over their distribution. The RS entropy functional is defined as

$$h_{\rm RS}[\mathrm{x}(.)] = -f_{\rm RS}[\mathrm{x}(.)] + \mathbb{E}[h] \tag{20.2}$$

The motivation for introducing the functional $h_{\rm RS}[\mathrm{x}(.)]$ will become clear in the next paragraph (see equ. (20.4)).

How to determine the MAP threshold

Recall that the (true) average free energy is given by the thermodynamic limit $-\lim_{n\to+\infty} \mathbb{E}[\ln Z]/n$ where Z is the partition function for coding (4.7). The replica symmetric formula states that

$$-\lim_{n\to+\infty} \frac{1}{n}\mathbb{E}[\ln Z] = \inf_{\mathrm{x}\in\mathcal{S}} f_{\rm RS}[\mathrm{x}(.)] \tag{20.3}$$

In this formula $\mathcal{S}$ is the space of (Nishimori) symmetric distributions (see Chapter 4). That the infimum can be restricted to this space of distributions is a special feature coming from channel symmetry. Such formulas relating a free energy to a replica functional have been long standing conjectures since the mid 70's in the field of spin glass models (on sparse and complete graph models) but much progress have been made in the last fifteen years towards their proofs. The present one is a case where we have a partial proof that combines interpolation methods with spatial coupling. This will be sketched in the subsequent chapter. In the next sub-section we take a closer look at (20.3) for the BEC, and show that it is equivalent to the Maxwell construction.

The MAP threshold is defined as the smallest ϵ such that $\liminf_{n\to\infty} \mathbb{E}[H(\underline{X} \mid \underline{Y}(\epsilon))/n] > 0$ (see definition 16.2). Recall also the relationship (4.39)

$$\frac{1}{n}\mathbb{E}[H(\underline{X} \mid \underline{Y}(\epsilon))] = -\frac{1}{n}\mathbb{E}[\ln Z] + \mathbb{E}[h] \tag{20.4}$$

Equation (20.3) has two consequences. One can replace $\liminf$ by $\lim$ in the definition of the MAP threshold, but more importantly,

$$\lim_{n\to+\infty} \frac{1}{n}\mathbb{E}[H(\underline{X} \mid \underline{Y}(\epsilon))] = \sup_{\mathrm{x}\in\mathcal{S}} h_{\rm RS}[\mathrm{x}(.)] \tag{20.5}$$

and

$$\epsilon_{\rm MAP} = \inf\{\epsilon \in [0,1] : \sup_{\mathrm{x}\in\mathcal{S}} h_{\rm RS}[\mathrm{x}(.)] > 0\}$$

In order to concretly calculate the MAP threshold one has to solve the variational problem consisting in minimizing (or maximizing) the replica symmetric

free energy (or entropy). It is easy to write down the stationary point conditions (homework) and one finds the density evolution fixed point equations (see Equ. (10.20)-(10.21))

$$\mathrm{x} = c \otimes \hat{\mathrm{x}}^{\otimes(l-1)}, \qquad \hat{\mathrm{x}} = \mathrm{x}^{\oplus(r-1)} \tag{20.6}$$

Remark that $\hat{\mathrm{x}}(.)$ is the distribution of $\hat{H}$ in Equ. (20.1). This is not surprising: the stationary points of the Bethe free energy are given by the BP equations and the stationary points of the replica functional are given by the density evolution equations. Once stationary points, i.e. fixed points of (20.1) have been found one selects the one that yields the largest $h_{\mathrm{RS}}[\mathrm{x}(.)]$ (or smallest $f_{\mathrm{RS}}[\mathrm{x}(.)]$) and determines ϵ_{MAP}. Since in practice fixed points are found by iterative methods, it is fortunate that we only need to find *stable* fixed points. Indeed the maximum of $h_{\mathrm{RS}}[\mathrm{x}(.)]$ (or minimum of $f_{\mathrm{RS}}[\mathrm{x}(.)]$) is necessarily a stable fixed point.

But that is not all. We already know that allow to determine the BP threshold. The BP threshold is the smallest noise for which a non-trivial fixed point is reached under iterations initialized with $\mathrm{x}(.) = c(.)$. Therefore this information is also contained in the RS functional. The BP threshold is the smallest noise such that the RS functional has a non trivial stationary point.

To summarize, the RS functional contains all the information we want. In particular it allows to deduce the DE equations. To determine the BP threshold it suffices to solve the DE equation. But, to evaluate the MAP threshold we have to solve the DE equations *and* to evaluate corresponding largest RS entropy or smallest RS free energy.

In the next paragraph we specialize this discussion to the case of the BEC. This will also allow us to derive the Maxwell construction in a more principled way.

20.2 Explicit Case of the BEC

A bit transmitted through the BEC is either perfectly transmitted with probability ϵ or earsured with probability $1-\epsilon$. This implies that $c(h) = \epsilon\delta(h) + (1-\epsilon)\delta_\infty(h)$, and that we can restrict the RS functionals to distributions parametrized as

$$\mathrm{x}(H) = x\delta(H) + (1-x)\ \delta_\infty(H)$$

where x is the erasure probabaility emanting from variables. This also implies that $\hat{\mathrm{x}}(\hat{H}) = \hat{x}\delta(\hat{H}) + (1-\hat{x})\delta_\infty(\hat{H})$ with $\hat{x} = 1-(1-x)^{r-1}$ the erasure probability emanating from checks. With this parametrization one can compute each term in the RS expression for the free energy. One easily finds the contributions of "check nodes"

$$\mathbb{E}[\ln \frac{1}{2}(1 + \prod_{k=1}^{r} \tan H_k)] = (1-x)^r \ln 2 - \ln 2$$

and "edges"

$$\mathbb{E}[\ln(1+\tan H\tan\hat{H})] = (1-x)(1-\hat{x})\ln 2$$

For the BEC, one should include the term $\mathbb{E}[h]$ in (20.2) directly in the contribution of "variable nodes" in order to avoid working with infinite quantities. One finds

$$\begin{aligned}
&\mathbb{E}[\ln(\prod_{k=1}^{l}(1+\tanh\hat{H}_k)+e^{-2h}\prod_{k=1}^{l}(1-\tanh\hat{H}_k))] \\
&= (1-\epsilon)\sum_{e=0}^{l}\binom{l}{e}\hat{x}^e(1-\hat{x})^{l-e}\ln 2^{l-e} + \epsilon\sum_{e=0}^{l-1}\binom{l}{e}\hat{x}^e(1-\hat{x})^{l-e}\ln 2^{l-e} \\
&\quad + \epsilon\binom{l}{l}\hat{x}^l(1-\hat{x})^{l-l}\ln 2 \\
&= \sum_{e=0}^{l}\binom{l}{e}\hat{x}^e(1-\hat{x})^{l-e}(l-e)\ln 2 + \epsilon\hat{x}^l\ln 2 \\
&= (1-\hat{x})\sum_{e=0}^{l}\hat{x}^e\frac{d}{dy}y^{l-e}\mid_{y=1-\hat{x}}\ln 2 + \epsilon\hat{x}^l\ln 2 \\
&= (1-\hat{x})\frac{d}{dy}(\hat{x}+y)^l\mid_{y=1-\hat{x}}\ln 2 + \epsilon\hat{x}^l\ln 2 \\
&= l(1-\hat{x})\ln 2 + \epsilon\hat{x}^l\ln 2
\end{aligned}$$

Putting these results together one finds the replica symmetric entropy function for the BEC

$$\frac{h_{\mathrm{RS}}(x;\epsilon)}{\ln 2} = (\frac{l}{r}-l)(1-x)^r + l(1-x)^{r-1} + \epsilon(1-(1-x)^{r-1})^l - \frac{l}{r}$$

According to (20.3) the conditionnal entropy is given by

$$\lim_{n\to+\infty}\frac{1}{n}\mathbb{E}[H(\underline{X}\mid\underline{Y}(\epsilon))] = \max_{0\le x\le 1} h_{\mathrm{RS}}(x;\epsilon) \tag{20.7}$$

and the MAP threshold can be calculated from $\epsilon_{\mathrm{MAP}} = \inf\{\epsilon : \max_{0\le x\le 1} h_{\mathrm{RS}}(x;\epsilon) > 0\}$. It is immediate to check that the stationnary points are given by the usual density evolution fixed point equation $x = \epsilon(1-(1-x)^{r-1})^{l-1}$.

As pointed out before, the function $-h_{\mathrm{RS}}$ contains all the information about the BP and MAP thresholds, so it is very useful to have an idea of the shape of the RS function. Figure **??** shows $-h_{\mathrm{RS}}$ as a function of x, for various values of ϵ.[2] We prefer to plot *minus* the RS entropy function[3] because this quantity is the free energy (up to an irrelevant term) and is better suited to make the physical analogies more transparent. For all ϵ there is a trivial minimum at $x = 0$, which

[2] This plot is generic only for regular ensembles with $l \ge 3$. Irregular ensembles can have a richer behavior and the corresponding discussion is more complicated. The case $l = 2$ is somewhat special because $\epsilon_{\mathrm{BP}} = \epsilon_{\mathrm{MAP}}$.

[3] To avoid any confusion let us stress that there is no reason why $h_{\mathrm{RS}}(x)$ should be non-negative. It is only $\max_{0\le x\le 1} h_{\mathrm{RS}}(x)$ that has to be non-negative.

is also the trivial stable fixed point of DE. For $\epsilon < \epsilon_{\text{BP}}$ this minimum is unique (hence global). At $\epsilon = \epsilon_{\text{BP}}$ the function develops a flat inflexion point and a second (local) minimum as well as a (local) maximum branch of. The local minimum is the stable non-trivial fixed point of density evolution, $x_{\text{st}}(\epsilon)$, and the local maximum is the unstable fixed point $x_{\text{un}}(\epsilon)$. As one increases ϵ further the local minimum at $x_{\text{st}}(\epsilon)$ decreases until it touches the horizontal axis for ϵ_{MAP}. At this threshold value there are two global minima, $h_{\text{RS}}(0;\epsilon_{\text{MAP}}) = h_{\text{RS}}(x_{\text{st}}(\epsilon_{\text{MAP}};\epsilon_{\text{MAP}})$. Finally, $\epsilon > \epsilon_{\text{MAP}}$ it is $x_{\text{st}}(\epsilon)$ that becomes the unique global minimum.

To summarize, one should retain from this discussion that the RS function contains all the information we want. The BP threshold is found by searching values of ϵ where the function develops flat inflexion points, and the MAP threshold is found by looking at values of ϵ where the two minima are at the same height. The reader should go back to the exact solution of the CW model in Chapter 5 and notice the intimate structural analogies with the present situation. The CW free energy is given by a variational problem $\min_{-1\leq m\leq 1} f(m)$ whose solutions determine both the phase transition ("MAP") threshold $h = 0$ and the spinodal ("BP") points $\pm h_{\text{sp}}$.

We conclude this paragraph by casting (20.7) in an equivalent form. For $\epsilon > \epsilon_{\text{MAP}}$ the derivative of the right hand side of $\max_{0\leq x\leq 1} h_{\text{RS}}(x;\epsilon)$ equals

$$\begin{aligned}\frac{d}{d\epsilon}h_{\text{RS}}(x_{\text{st}};\epsilon) &= \frac{\partial}{\partial\epsilon}h_{\text{RS}}(x_{\text{st}};\epsilon) + \frac{\partial}{\partial x}h_{\text{RS}}(x_{\text{st}};\epsilon)\frac{dx_{\text{st}}}{d\epsilon}\\ &= \frac{\partial}{\partial\epsilon}h_{\text{RS}}(x_{\text{st}};\epsilon)\end{aligned}$$

The second equality is valid because x_{st} is a stationnary point of h_{RS} and $\frac{dx_{\text{st}}}{d\epsilon}$ is finite for $\epsilon \in]\epsilon_{\text{MAP}}, 1]$. This last point can be checked rather explicitly for the BEC but for other channels this is much more difficult. We obtain

$$\frac{d}{d\epsilon}\lim_{n\to+\infty}\frac{1}{n}\mathbb{E}[H(\underline{X}\mid\underline{Y}(\epsilon))] = \begin{cases}0, \epsilon < \epsilon_{\text{MAP}}\\ \frac{\partial}{\partial\epsilon}h_{\text{RS}}(x_{\text{st}}(\epsilon);\epsilon) = (1-(1-x_{\text{st}}(\epsilon))^{r-1})^{l}, \epsilon > \epsilon_{\text{MAP}}\end{cases}$$

Note that for $\epsilon > \epsilon_{\text{MAP}}$ the derivative of the conditional entropy coincides with the EXIT curve introduced somewhat arbitrarily in Chapter 16.

20.3 Back to the Maxwell Construction

The Maxwell construction identifies the MAP threshold ϵ_{MAP} with the area threshold ϵ_{A} on the EXIT curve. We are now in a position to show that this identity is equivalent to the equality of the two minima of $-h_{\text{RS}}(x;\epsilon)$ when $\epsilon = \epsilon_{\text{MAP}}$. Apart from the conceptual importance of this result, this shows that for coding a proof of the Maxwell construction boils down to the one of the RS formula.

Consider $\epsilon > \epsilon_{\text{BP}}$. The non-trivial minimum and maximum of $-h_{\text{RS}}(x;\epsilon)$, namely $x_{\text{st}}(\epsilon)$ and $x_{\text{un}}(\epsilon)$, form a curve in the (ϵ, x)-plane. This curve is precisely $(\epsilon(x), x)$ where $\epsilon(x) = x/(1-(1-x)^{r-1})^{l-1}$ (since the stationary points

of $-h_{\rm RS}(x;\epsilon)$ are given by DE). Now consider the path starting from $(\epsilon_{\rm MAP},0)$ to $(+\infty,0)$ on the horizontal axis and then along the curve till $(\epsilon(x),x)$ for some x. Look at the total change in RS entropy along this path. We have

$$\begin{aligned} h_{\rm RS}(x;\epsilon(x)) - h_{\rm RS}(0;\epsilon_{\rm MAP}) &= \int_{\rm path} dh_{\rm RS} = \int_0^x dx \frac{d}{dx} h_{\rm RS}(x;\epsilon(x)) \\ &= \int_0^x dx (\frac{\partial}{\partial x} h_{\rm RS}(x;\epsilon(x)) + \epsilon'(x) \frac{\partial}{\partial \epsilon} h_{\rm RS}(x;\epsilon(x))) \\ &= \int_0^x dx \epsilon'(x) \frac{\partial}{\partial \epsilon} h_{\rm RS}(x;\epsilon(x)) \\ &= \int_0^x dx \epsilon'(x) (1-(1-x)^{r-1})^l \end{aligned}$$

The last integral is recognized as the trial entropy $P(x)$, the area under the EXIT curve $(\epsilon(x),(1-(1-x)^{r-1})^l)$ (see (16.7)).

Let us highlight the main points of this discussion. The natural definition of the EXIT curve in parametric form is,

$$(\epsilon(x), \frac{\partial}{\partial \epsilon} h_{\rm RS}(x;\epsilon(x))).$$

and satisfies

$$h_{\rm RS}(x;\epsilon(x)) - h_{\rm RS}(0;\epsilon_{\rm MAP}) = \int_0^x dx \epsilon'(x) \frac{\partial}{\partial \epsilon} h_{\rm RS}(x;\epsilon(x)).$$

The right hand side is the area under the EXIT curve and the left hand side is the corresponding change in entropy. On one hand the area threshold is by definition $\epsilon_{\rm A} = \epsilon(x_{\rm A})$ such that the area under the EXIT curve vanishes, and on the other hand the MAP threshold is $\epsilon_{\rm MAP} = \epsilon(x_{\rm MAP})$ such that the minima of $-h_{\rm RS}$ are at the same height $h_{\rm RS}(x_{\rm MAP};\epsilon(x_{\rm MAP})) - h_{\rm RS}(0;\epsilon_{\rm MAP}) = 0$. Therefore these two thresholds are identical.

20.4 Compressive Sensing

Write RS free energy (can be derived by integrating out state evolution). Illustrate thresholds it predicts. Discuss that RS is exact. Do it for Lasso or for know prior case ?

20.5 K-SAT

Recall that in Chapter 19 we gave the Bethe expression for the free energy of K-SAT. From this expression one also gets a Bethe formula for the entropy density. There is natural RS functional associated to this formula, which leads to a natural conjecture for the entropy density. We will see that, contrary to

coding and compressive sensing, the conjecture cannot be fully correct.[4] This is one of the main motivations for developping a better theory, namely the cavity method.

The construction of the natural RS functional for K-SAT proceeds like in the coding case: one takes as a starting point the Bethe expression (19.21) and treats the messages $h_{i\to a}$ as independent random variables distributed according to a trial distribution Q(.). The message passing equation (15.15),

$$\hat{h}_{a\to i} = -\frac{1}{2}\ln\Big\{1 - \prod_{j\in\partial a\setminus i}\frac{1-\tanh h_{j\to a}}{2}\Big\} \tag{20.8}$$

induces the distribution $\hat{Q}(.)$. In the coding case we discussd the case of regular Gallager (l, r) ensembles. One difference here is that while the check nodes have degree K, the variable node degrees are (asymtotically) Poisson distributed with average degree αK.

Here is the formal definition of the RS functional for the entropy. Fix a trial distribution $Q(.)$ on $\mathbb{R}$. Pick K iid copies of the random variable $H \sim Q(.)$. Call them $H_1, \ldots, H_K$. Define the random variable

$$\hat{H} = -\frac{1}{2}\ln\Big\{1 - \prod_{k=1}^{K-1}\frac{1-\tanh H_k}{2}\Big\}. \tag{20.9}$$

Pick two Poisson distributed integers p and q with average αK, and pick $p+q$ iid copies of $\hat{H}_k$, $k = 1, \ldots, p+q$. Let

$$\begin{aligned} s(\underline{H}, \underline{\hat{H}}, p, q) = \ln\Big\{ &\prod_{k=1}^{p}(1-\tanh\hat{H}_k)\prod_{k=p+1}^{p+q}(1+\tanh\hat{H}_k) \\ &+ \prod_{k=1}^{p}(1+\tanh\hat{H}_k)\prod_{k=p+1}^{p+q}(1-\tanh\hat{H}_k)\Big\} \\ &+ \ln\Big\{1 - \prod_{k=1}^{K}\frac{1-\tanh H_k}{2}\Big\} \\ &- \ln\Big\{1 + \tanh H \tanh\hat{H}\Big\} \end{aligned}$$

The RS entropy functional is defined as

$$s_{\mathrm{RS}}(Q(.)) = \mathbb{E}[s(\underline{H}, \underline{\hat{H}}, p, q)] \tag{20.10}$$

where the expectation is over all random variables $p, q, \underline{H}, \underline{\hat{H}}$.

The replica symmetric prescription for computing the entropy density is to

[4] While in coding and compressive sensing it is quite hard to prove the RS formulas are exact, in K-SAT it is relatively easier to prove that they cannot be correct or at least fully correct.

take

$$s_{\rm RS}(\alpha) \equiv \sup_{Q(.)} s_{\rm RS}(Q(.))$$

The stationnary points of (20.10) yields an integral equation for $Q(.)$. Similarly to coding, this can be split in two integral equations linking $Q(.)$ and $\hat{Q}(.)$ where $\hat{Q}(.)$ is the distribution of $\hat{H}$. These two equations can equivalently be written as (homework)

$$H \stackrel{d}{=} \sum_{k=1}^{p} \hat{H}_k - \sum_{k=p+1}^{p+q} \hat{H}_k, \quad \hat{H} \stackrel{d}{=} -\frac{1}{2} \ln\Big\{1 - \prod_{k=1}^{K-1} \frac{1-\tanh H_k}{2}\Big\}.$$

where $\stackrel{d}{=}$ means equality in distribution. The second relation is of course the same as (20.9), and you will derive the first one in the homeworks. These equations can be solved numerically (e.g. by the population dynamics method of homework). This allows to find the maximizer of the RS functional and compute $s_{\rm RS}(\alpha)$.[5] Figure **??** shows that $s_{\rm RS}(\alpha)$ for $K = 3$. the function decreases as the clause density increases, and vanishes at $\alpha \approx 4.677$. Thus the present replica symmetric analysis predicts that there exist exponentially many solutions at least until this value of α, and that in particular the SAT-UNSAT threshold should be larger. However it is known that this is wrong. For example in problem **??** we guide you through the proof of $\alpha_{\text{sat-unsat}} \leq 4.666$ for $K = 3$. In fact, as we will see in Chapter 22 the cavity method proposes that the RS formula is exact till a threshold value $\alpha_c < \alpha_{\text{sat-unsat}}$, called the "condensation threshold", and that another one called RSB formula[6] holds in the range $\alpha_c < \alpha < \alpha_{\text{sat-unsat}}$. At the condensation threshold there is a genuine phase transition: $\lim n^{-1}\mathbb{E} \ln Z$ is not analytic, in other words the same (analytic) formula cannot hold both above and below α_c. For $K = 3$ we have $\alpha_c \approx 3.86$ and $\alpha_{\text{sat-unsat}} \approx 4.26$. None of these claims have been proven so far.

20.6 Notes

A few words about the concept of order paramter. Like for many physical concepts there is no rigid definition, and finding the correct order parameter is an art validated by experiment. Depending on the problem at hand this can seem more or less obvious like in fluids (the volume per particle) or in magnetism (the magnetization), but can be much more subtle like in superconductivity (the "wave function" of Cooper pairs). The Higgs field is the order parameter associated to the electroweak phase transition that occurred at an early epoch of the universe. The recently discovered Higgs bosons are elementary excitations of this

[5] Note the global maximum necessarily corresponds to a stable fixed point and therefore iterative methods to solve the density evolution equations can find it. Similarly global minima of the free energy necessarily correspond to stable fixed point of density evolution.

[6] As we will see "B" stands for broken.

field, much like spin flips are elementary excitations associated to magnetization. As we will see K-SAT is one of these problems for which the guess of the order parameter requires a stretch of imagination: probability distributions of random probability distributions.

Problems

20.1 *RS analysis for K-SAT* Derive the density evolution equations for K-SAT. Use population dynamics (as seen in homeworks of Chapter **??**) to compute the RS prediction for $\alpha_{\text{sat-unsat}}$.

20.2 *Upper bounds on the SAT-UNSAT threshold.* Upper bounds for the SAT-UNSAT threshold, we call it α_s, are usually derived by counting arguments. The first exercise develops the simplest such argument. In the second exercise you will study a more subtle counting argument which leads to an important improvement[7]. This method can be further refined and has led to better bounds.

An assignment is a tuple $\underline{x} = (x_1, \ldots, x_n)$ where $x_i = 0, 1$ of n variables. The total number of possible clauses with k variables is equal to $2^k \binom{n}{k}$. A random formula F is constructed by picking, with replacement, uniformly at random, m clauses. Thus there are $(2^k \binom{n}{k})^m$ possible formulas.

We set $m = \alpha n$ and think of n and m as tending to ∞ with α fixed. This is the regime displaying a SAT-UNSAT threshold.

It is useful to keep in mind that $\mathbb{P}[A] = \mathbb{E}[1(A)]$ where $1(A)$ is the indicator function of event A. In what follows probabilities and expectations are with respect to the random formulas F.

20.3 Crude upper bound by counting all satisfying assignments Let $S(F)$ be the set of all assignments satisfying F and let $|S(F)|$ be its cardinality. Since F is a random formula, $|S(F)|$ is an integer valued random variable.

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

b) Fix an assignement $\underline{x}$. Show that $\mathbb{P}[\underline{x} \,\text{satisfies}\, F] = (1 - 2^{-k})^m$. Then deduce that

$$\mathbb{E}[|S(F)|] = 2^n (1 - 2^{-k})^m.$$

c) Deduce the upper bound

$$\alpha_s < \frac{\ln 2}{|\ln(1 - 2^{-k})|}.$$

For $k = 3$ this yields $\alpha_s < 5.191$.

20.4 Bound by counting a restricted set of assignments] We define the set $S_{\text{m}}(F)$ of *maximal* satisfying assignments as follows. An assignment $\underline{x} \in S_{\text{m}}(F)$ iff:

- $\underline{x}$ satisfies F,

[7] by Kirousis, Kranakis, Krizanc and Stamatiou, *Approximating the Unsatisfiability Threshold of Random Formulas*, in Random Struct and Algorithms (1998).

- for all i such that $x_i = 0$ (in $\underline{x}$), the *single flip* $x_i \to 1$ yields an assignment - call it $\underline{x}^i$ - that *violates* F.

a) Show that if F is satisfiable then $S_m(F)$ is not empty. *Hint*: proceed by contradiction.

b) Show as in the first exercise the Markov inequality $\mathbb{P}[F \text{ satisfiable}] \leq \mathbb{E}[|S_m(F)|]$

c) Show that

$$\mathbb{E}[|S_m(F)|] = (1-2^{-k})^m \sum_{\underline{x}} \mathbb{P}[\cap_{i:x_i=0} (\underline{x}^i \text{ violates } F) \mid \underline{x} \text{ satisfies } F].$$

d) Fix $\underline{x}$. The events $E_i \equiv (\underline{x}^i \text{ violates } F)$ are negatively correlated, i.e

$$\mathbb{P}[\cap_{i:x_i=0} E_i \mid \underline{x} \text{ satisfies } F] \leq \prod_{i:x_i=0} \mathbb{P}[E_i \mid \underline{x} \text{ satisfies } F]$$

For the full proof which uses a correlation inequality (of FKG type) we refer to the reference given above. Here is a rough intuition for the inequality. First note that if $x_i = 0$ and $\underline{x}^i$ violates F, there must be some set S_i of clauses (in F) that are satisfied *only* by this variable $x_i = 0$ (this set might contain only one clause). This restricts the possible formulas contributing to the event E_i. Second note that sets S_i, S_j corresponding to different such variables $x_i = 0$, $x_j = 0$ must be *disjoint*. This "repulsion" between the sets S_i and S_j puts even more restrictions on the possible formulas, compared to a hypothetical situation where the events (and thus the sets S_i and S_j) would have been independent.

e) Now show that

$$\mathbb{P}[E_i \mid \underline{x} \text{ satisfies } F] = 1 - \left(1 - \frac{\binom{n-1}{k-1}}{(2^k-1)\binom{n}{k}}\right)^m.$$

Hint: note that in the event E_i there must be at least one clause containing $x_i = 0$ and containing other variables that do not satisfy it.

f) Deduce from the above results that $\lim_{n\to 0} \mathbb{P}[F \text{ satisfiable}] = 0$ as long as α satisfies

$$(1-2^{-k})^\alpha (2 - e^{-\frac{\alpha k}{2^k-1}}) < 1.$$

The improvement compared with the first exercise resides in the factor $e^{-\frac{\alpha k}{2^k-1}}$. A numerical evaluation for $k = 3$ yields the bound $\alpha_s < 4.667$.

21 Interpolation Method

22 Cavity Method: Basic Concepts

Message passing and spatial coupling techniques have been very successful in providing efficient algorithms in the realm of coding and compressive sensing. Furthermore the variational method has allowed us to derive the phase diagram for these models, and the Maxwell construction ties the two approaches together. On the other hand these methods are not as successful for constraint satisfaction problems such as K-SAT. For example, plain BP does not allow to find solutions and had to be supplemented by a decimation process. BP guided decimation finds solutions up to some density, but it is not clear if this limitation corresponds to some sort of fundamental dynamic threshold, similar to the BP threshold say. Also (for the moment) we are not able to find the SAT-UNSAT threshold by a sort of Maxwell construction or spatial coupling technique. At the same time the RS entropy functional does not count correctly the number of solutions.

The success of message passing marginalization is related to the absence of long range correlations between dynamical variables. In constraint satisfaction problems such as K-SAT long range correlations are present and it is not possible to only take into account a tree like neighborhood of a node when its marginal is computed. The boundary conditions at the leaf nodes of the tree like neighborhood somehow matter. Often, in statistical mechanics, when long range correlations are present, the key to the analysis comes from the concept of extremal measure and convex decomposition of the Gibbs measure into extremal measures. While these notions are relatively well understood and mathematically precise for low dimensional deterministic Ising models on regular grids, the mathematical theory in the context of spin glass type models is still very much of an open challenge. As we will see the cavity method boldly pushes the idea of convex decomposition of the Gibbs distribution to its limit in the sense that we will have to deal with a convex superposition with an exponentially large number of extremal measures. Once this is accepted, the theory, although technically challenging, flows. Indeed it turns out this convex superposition defines a new factor graph model which can again be analyzed by the message passing, variational free energy and spatial coupling techniques. That we can again apply these techniques "one level up" is one of the fascinating aspects of the subject.

22.1 Notion of Pure State

The concept of extremal measure or pure state has not been introduced nor used explicitely yet, but this is the time to do so. We start by a very brief discussion in the context of the Ising model because this is the simplest best understood non-trivial paradigmatic situation. We then turn our attention to the CW model, for which this notion is somewhat special due to the absence of geometry, but allows to introduce a very useful heuristic point of view that lends itself to generalizations.

A diggression on the Ising model

The construction of infinite volume Gibbs measures is a non-trivial problem whose mathematical theory is developped mainly for Ising type models on regular grids, say $\mathbb{Z}^d$. Here we summarize very briefly and informally the main picture for the classical two dimensional Ising model with nearest neighbord ferromagnetic interactions. for which the theory is fully controlled, and the interested reader will find pointers to the litterature in the notes. The phase diagram of this model is qualitatively the same as the one of CW. The mathematical theory of the Gibbs states for infinite volume starts with the Gibbs distribution on a finite square grid $\Lambda \subset \mathbb{Z}^d$ with specified boundary conditions. The boundary conditions amount to fix the spin assignments on vertices of $\partial\Lambda$. One computes the infinite volume limit of all marginals, given the boundary conditions, and the set of these marginals defines the infinite volume Gibbs state. For any point of the (T, h) plane the set of all possible infinite volume Gibbs states is convex. Away from the coexistence line this set is trivially a point i.e, the infinite volume limits of the marginals is independent of boundary conditions. On the coexistence line the set of infinite volume limits is non-trivial. It has two extremal measures obtained by the all $+1$ and all -1 boundary conditions, and in particular $\langle s_i\rangle_\pm = \pm m \neq 0$. All other states on the coexistence line are of the form $\langle - \rangle_w = w\langle - \rangle_+ + (1-w)\langle - \rangle_-$. Extremal states have correlations that satisfy the exponential decay property; this holds when the state is unique and for the $+$ and $-$ states. For example, $|\langle s_i s_j\rangle_\pm - \langle s_i\rangle_\pm \langle s_j\rangle_\pm| \leq \text{const}\, e^{-|i-j|/\xi(T)}$ where $\xi(T)$ is a finite correlation length.[1] On the other hand mixed states with $w \neq 0, 1$ have long range order which means $\lim_{|i-j|\to+\infty}(\langle s_i s_j\rangle_w - \langle s_i\rangle_w \langle s_j\rangle_w) \neq 0$. As a good exercise one can check that the clustering property of pure states implies this limit is equal to $4w(1-w)m^2$.

The CW model revisited

On the complete graph there is no boundary so we simply start with the model on a finite graph with a fixed constant magnetic field. We saw in Chapter 5 that

[1] This length diverges when T approaches the critical temperature.

in the (T, h) plane there is a the coexistence line on which the magnetization can take two different values in the sense that $\lim_{h\to 0_\pm} \lim_{n\to+\infty} \langle s_i \rangle = \pm m \neq 1$. The magnetization is uniquely defined away from this line in the sense that it is an analytic function of h and T. It is not difficult to show that this feature is shared by any average $\langle s_{i_1} \dots s_{i_k} \rangle$, for any finite set of spins. In this sense the infinite Gibbs state is unique and "pure" away from the coexistence line, and is not unique on this line. There, one can define two "pure states" $\langle s_{i_1} \dots s_{i_k} \rangle_\pm = \lim_{h\to 0_\pm} \lim_{n\to+\infty} \langle s_{i_1} \dots s_{i_k} \rangle$, and also any convex superposition $\langle - \rangle_w = w\langle - \rangle_+ + (1-w)\langle - \rangle_-$ for $0 < w < 1$. For the CW model the "pure" states satisfy an extreme form of clustering where variables *decouple* in thermodynamic limit. For example for $k = 2$ $\langle s_i s_j \rangle_\pm - \langle s_i \rangle_\pm \langle s_j \rangle_\pm = 0$.[2] Genuine superpositions (mixed states) have correlations that do not vanish in the thermodynamic limit. For example, the decoupling property implies $\langle s_i s_j \rangle_w - \langle s_i \rangle_w \langle s_j \rangle_w = 4w(1-w)m^2$ on the coexistence line for any $i \neq j$. Remark for the Ising model the same relation is obtained for $|i - j| \to +\infty$.

For the CW model there is a one to one correspondence between "pure states" and minima of the free energy function $f(m)$ appearing in the variational expression for $-n^{-1} \ln Z$. This is an extremely simple instance of the landscape picture discussed in the next paragraph.

The landscape picture

For spin glass models the situation is not "as simple". It is not known how to define a mathematically sound notion of extremal state. For models on complete or sparse locally tree like graphs one heuristic and intuitive approach identifies the extremal states with global or quasi-global minima of the TAP or Bethe type free energy functionals[3]. Let $(\mu^{(p)}_{i\to a}, \hat{\mu}^{(p)}_{a\to i}) = (\underline{\mu}^{(p)}, \underline{\hat{\mu}}^{(p)})$ be the corresponding solutions of the sum-product equations where p indexes the minima. From these mesages one can reconstruct marginals $\nu^{(p)}(.)$ which define the "extremal measure". To distinguish this measure from the usual notion of extremal state and to avoid confusions we will call this an *extremal or pure Bethe measure*. One has to think of it as a "proxy" for an ideal notion of pure state. When message passing iterations converge one expects that there are a small number of fixed points with well defined bassins of attraction and the number of pure Bethe states is small. However when these iterations do not converge this may be due to the presence of a very large number of fixed points, and thus to a very large number of minima in the Bethe free energy. In such situations one expects a large number of pure Bethe states. This happens in the TAP approach to the SK model for the region of the phase diagram below the AT line. This

[2] The CW model is a bit special in this respect because the complete graph wipes out any trace of geometry. For finite n and any h one has $\langle s_i s_j \rangle - \langle s_i \rangle \langle s_j \rangle = O(n^{-1})$. Since there is a unit distance between any two variables, one may interpret this as a exponential decay of correlations on a length scale $O(1/\ln n)$.

[3] It is debated wether such an approach is valid for low dimensional spin glasses e.g the Edwards-Anderson model

also happens in K-SAT for clause densities slightly above the ones found by BP guided decimation. The reason for the failure of BP guided decimation is the proliferation of minima in the Bethe free energy. Free energy functions with a proliferation of numerous minima are often called free energy landscapes. Figure **??** serves as a useful mental picture summarizing these ideas.

22.2 The Level-One Model

The convex decomposition ansatz

We formalize the heuristic landscape picture. The cavity method assumes that: (i) The Gibbs distribution is a convex sum of "pure states"; (ii) Pure states are identified with the Bethe measures corresponding to minima of the free energy; (iii) The weights of the convex superposition are determined by the Bethe free energy minima. We write

$$\mu\left(\underline{x}\right)=\sum_{p=1}^{\mathcal{N}}\frac{e^{-xF^{(p)}}}{Z(x)}\mu^{(p)}(\underline{x}),\qquad Z(x)=\sum_{p=1}^{\mathcal{N}}e^{-xF^{(p)}} \tag{22.1}$$

The sum runs over p which indexes the minima $\{\mu^{(p)}_{a\to i},\hat{\mu}^{(p)}_{a\to i}\}=(\underline{\mu}^{(p)},\underline{\hat{\mu}}^{(p)})$ of the Bethe free energy functional. The weights are determined by the free energy of these minima $F^{(p)}=F_{\rm Bethe}\left(\underline{\mu}^{(p)},\underline{\hat{\mu}}^{(p)}\right)$. The "pure" Bethe measures $\mu^{(p)}(\underline{x})$ are defined through the collection of all their marginals, which themselves are determined from $(\underline{\mu}^{(p)},\underline{\hat{\mu}}^{(p)})$. The role of x, called the "Parisi parameter", turns out to be quite subtle.[4] For the moment one can think of it as a multiplicative "renormalization" of the temperature. In a large portion of the phase diagram the naive choice $x=1$ is correct. However we will see that there are regions of the phase diagram where values $0<x<1$ are forced upon us.

Level-one auxiliary model

In order to make technical progress with the convex decomposition ansatz we make one more assumption. One expects that at low temperatures when there are an exponential number of minima, these are exponentially more numerous than maxima and saddle points. Therefore we assume: (iv) the sum over p runs over *all* stationnary points of the Bethe free energy i.e, fixed points solutions of the sum-product equations.

The partition function (22.1) can be thought of as the one of a statistical mechanics system with dynamical variables $(\underline{\mu}^{(p)},\underline{\hat{\mu}}^{(p)})$ and effective Hamiltonian

[4] The notation x is traditional and should not be confused with the one for configurations $\underline{x}$. This parameter was first introduced by parisi in the context of the replica approach. There its role is even more mysterious an appears as an integer that is anlytically continued to values in $]0,1[$.

given by the Bethe free energy. Using assumption (iv) we are led to study the Gibbs probability distribution of an auxiliary model, called the "level-one model"

$$\mu_1(\underline{\mu},\underline{\hat{\mu}}) = \frac{1}{Z_1(x)} e^{-xF_{\text{Bethe}}\,(\underline{\mu},\underline{\hat{\mu}})}\,\mathbb{1}_{\text{sp}}(\underline{\mu},\underline{\hat{\mu}}) \tag{22.2}$$

and

$$Z_1(x) = \sum_{\underline{\mu},\underline{\hat{\mu}}} e^{-xF_{\text{Bethe}}\,(\underline{\mu},\underline{\hat{\mu}})}\,\mathbb{1}_{\text{sp}}(\underline{\mu},\underline{\hat{\mu}}) \tag{22.3}$$

The indicator function $\mathbb{1}_{\text{sp}}(\underline{\mu},\underline{\hat{\mu}})$ selects solutions of the sum product fixed point equations. Recall that in the sum-product equations and the Bethe free energy the normalization of the messages is arbitrary. In order for the sum in (22.3) to be well defined we have to fix a normalization. We will take the most natural one, namely $\sum_{x_i}\mu_{i\to a}(x_i) = \sum_{x_i}\hat{\mu}_{a\to i}(x_i) = 1$. With this normalization the sum product equations used in subsequent calculations read

$$\mu_{i\to a}(x_i) = \frac{\prod_{b\in\partial i\setminus a}\hat{\mu}_{b\to i}(x_i)}{\sum_{x_i}\prod_{b\in\partial i\setminus a}\hat{\mu}_{b\to i}(x_i)} \tag{22.4}$$

$$\hat{\mu}_{a\to i}(x_i) = \frac{\sum_{\sim x_i} f_a(x_{\partial a})\prod_{j\in\partial a\setminus i}\mu_{j\to a}(x_i)}{\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{j\in\partial a\setminus i}\mu_{j\to a}(x_i)} \tag{22.5}$$

Let us immediately give a few definitions that will be useful to us later on. Averages with respect to (22.2) are denoted by the usual bracket notation $\langle - \rangle_1$. The level-one free energy is defined as usual $f_1(x) = -\frac{1}{nx}\ln Z_1(x)$. As in Chapter 3, the free energy allows to compute numerous other quantities by differentiations with respect to the inverse temperature, here with respect to x. The level-one internal energy is $u_1(x) = \langle F_{\text{Bethe}}\rangle_1/n = \frac{\partial}{\partial x} f_1(x)$. The Shannon-Gibbs entropy associated to (22.2) is equal to $\Sigma(x) = x^2\frac{\partial}{\partial x} f_1(x) = u_1(x) - x^{-1}\Sigma(x)$.

Choice of the Parisi parameter

Small paragraph to be written. Explain briefly. Interpret $\Sigma(x)$.

22.3 Message passing, Bethe free energy and complexity one level up

Message passing

We now show how the level-one model is solved in practice. The main idea is to first recognize that the model is defined on a sparse factor graph and apply again the sum-product and Bethe formulas. If $\Gamma = (V, C, E)$ is the original factor graph, then the level-one model has the factor graph $\Gamma_1 = (V_1, C_1, E_1)$ described on Fig. 22.1. We use the shorthand notation $\mathbb{1}_i$ and $\hat{\mathbb{1}}_a$ for the indicator functions

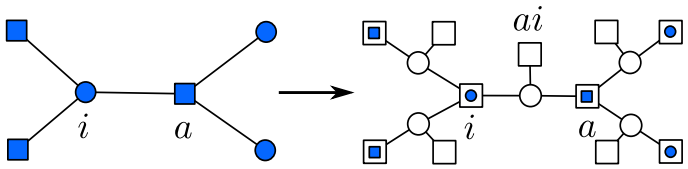


Figure 22.1 On the left, an exemple of an original graph Γ. On the right its corresponding graph Γ_1 for the level-one model.

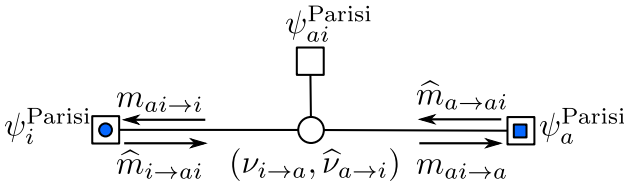


Figure 22.2 Messages are labbeled with m if they outgoing from a Parisi variable node are and with $\widehat{m}$ if they are outgoing from a Parisi function node.

forcing equations (22.4)-(22.5). Thus $\mathbb{1}(\underline{\mu}, \underline{\hat{\mu}}) = \prod_i \mathbb{1}_i \prod_a \hat{\mathbb{1}}_a$. A variable node $i \in V$, becomes a function node $i \in C_1$, with the function

$$\psi_i = e^{-xF_i} \prod_{a \in \partial i} \mathbb{1}_i. \tag{22.6}$$

A function node $a \in C$ remains a function node $a \in C_1$ with factor

$$\psi_a = e^{-xF_a} \prod_{i \in \partial a} \hat{\mathbb{1}}_a. \tag{22.7}$$

An edge $(a, i) \in E$, becomes a variable node $(a, i) \in V_1$. There is also an extra function node attached to each variable node of the new graph, or equivalently attached to each edge of the old graph. The corresponding function is

$$\psi_{ai} = e^{+xF_{ai}}. \tag{22.8}$$

With these definitions (22.2) can be written as

$$\mu_1(\underline{\mu}, \underline{\widehat{\mu}}) = \frac{1}{Z_1(x)} \prod_{i \in V} \psi_i \prod_{a \in C} \psi_a \prod_{ai \in E} \psi_{ai}. \tag{22.9}$$

The sum product equations for (22.9) involve four kind of messages shown on figure 22.2. Messages flowing from a new function node to a new variable node satisfy (the symbol $\simeq$ means equal up to a normalization factor)

$$\begin{aligned} \widehat{m}_{a\to ai} &\simeq \sum_{\sim(\mu_{i\to a}, \widehat{\mu}_{a\to i})} \psi_a \prod_{aj \in \partial a \setminus ai} m_{aj\to a} \\ &= e^{xF_{ai}} \sum_{\sim(\mu_{i\to a}, \widehat{\mu}_{a\to i})} \hat{\mathbb{1}}_a(\widehat{\mu}_{a\to i}) e^{-x(F_a - F_{ai})} \prod_{aj \in \partial a \setminus ai} m_{aj\to a} \end{aligned}$$

and

$$\begin{aligned}\widehat{m}_{i\to ai} &\simeq \sum_{\sim(\mu_{i\to a},\widehat{\mu}_{a\to i})} \psi_i \prod_{bi\in\partial i\setminus ai} \widehat{m}_{bi\to i} \\ &= e^{xF_{ai}} \sum_{\sim(\mu_{i\to a},\widehat{\mu}_{a\to i})} \mathbb{1}_i(\mu_{i\to a}) e^{-x(F_i-F_{ai})} \prod_{bi\in\partial i\setminus ai} \widehat{m}_{bi\to i}\end{aligned}$$

Messages from a new function node to a new variable node satisfy

$$m_{ai\to i} \simeq e^{xF_{ai}}\widehat{m}_{a\to ai}, \qquad m_{ai\to a} \simeq e^{xF_{ai}}\widehat{m}_{i\to ai}.$$

Notice that $m_{ai\to i}$ and $m_{ai\to a}$ are independent of $\widehat{\mu}_{a\to i}$ and $\mu_{i\to a}$ respectively; this allows us to simplify the message passing equations. To achieve the simplification define two distributions

$$Q_{i\to a}(\mu_{i\to a}) = m_{ai\to a}, \qquad \widehat{Q}_{a\to i}(\widehat{\mu}_{a\to i}) = m_{ai\to i}$$

These flow on the edges of the original factor graph $\Gamma = (V, C, E)$ and are called *cavity messages*. It is easy to see that they satisfy

$$\widehat{Q}_{a\to i}(\widehat{\mu}_{a\to i}) \simeq \sum_{\underline{\mu}} \hat{\mathbb{1}}_a(\widehat{\mu}_{a\to i}) e^{-x(F_a-F_{ai})} \prod_{j\in\partial a\setminus i} Q_{j\to a}(\mu_{j\to a}) \tag{22.10}$$

$$Q_{i\to a}(\mu_{i\to a}) \simeq \sum_{\underline{\widehat{\mu}}} \mathbb{1}_i(\mu_{i\to a}) e^{-x(F_i-F_{ai})} \prod_{b\in\partial i\setminus a} \widehat{Q}_{b\to i}(\widehat{\mu}_{b\to i}). \tag{22.11}$$

These are the *cavity equations*, an instance of sum-product equations for the level-one model. Note that the cavity equations do not make any reference to the graph Γ_1 and we can now revert to the original one. As usual, if the graph was a tree, these equations give the exact marginals of (22.2).

The x dependent exponentials are sometimes called reweighting factors. Their explicit expression will be useful later on,

$$e^{-(F_i-F_{ai})} = \sum_{x_i} \prod_{b\in\partial i\setminus a} \widehat{\mu}_{b\to i}(x_i), \qquad e^{-(F_a-F_{ai})} = \sum_{x_{\partial a}} f_a(x_{\partial a}) \prod_{\partial j\in a\setminus i} \mu_{j\to a}(x_i) \tag{22.12}$$

Note that these are in fact the normalization factors in (22.4)-(22.5).

Bethe free energy and complexity

The Bethe free energy functional of the level-one model is a functional of the cavity messages $Q_{i\to a}, \widehat{Q}_{a\to i}$. We could derive it as in Chapter **??** by first deriving the exact free energy $f_1(x)$ on a tree, and then take this expression as a defition for general graph instances. But we can also guess the fomula. It is basically given by the usual definition, but with the extra feature that it must contain the reweighting factors. Moreover its stationary points must yield (**??**). This is enough information to guess that

$$\mathcal{F}_{\text{Bethe}}(\underline{Q}, \underline{\widehat{Q}}) = \sum_{i\in V} \mathcal{F}_i + \sum_{a\in C} \mathcal{F}_a - \sum_{ai\in E} \mathcal{F}_{ai} \tag{22.13}$$

where

$$\mathcal{F}_i(\{\widehat{Q}_{b\to i}\}_{b\in\partial i}) = -\frac{1}{x}\ln\Big\{\sum_{\underline{\widehat{\mu}}} e^{-xF_i}\prod_{b\in\partial i}\widehat{Q}_{b\to i}\Big\},$$

$$\mathcal{F}_a(\{Q_{j\to a}\}_{j\in\partial a}) = -\frac{1}{x}\ln\Big\{\sum_{\underline{\mu}} e^{-xF_a}\prod_{j\in\partial a} Q_{j\to a}\Big\},$$

$$\mathcal{F}_{ai}(Q_{i\to a},\widehat{Q}_{a\to i}) = -\frac{1}{x}\ln\Big\{\sum_{\mu,\widehat{\mu}} e^{-xF_{ai}}Q_{i\to a}\widehat{Q}_{a\to i}\Big\}.$$

The complexity functional within the Bethe formalism is given by $\Sigma_{\rm Bethe} = x^2\frac{\partial}{\partial x}\mathcal{F}_{\rm Bethe}$. Explicitly,

$$\Sigma_{\rm Bethe}(\underline{Q},\underline{\widehat{Q}}) = \sum_{i\in V}\Sigma_i + \sum_{a\in C}\Sigma_a - \sum_{ai\in E}\Sigma_{ai} \tag{22.14}$$

where

$$x^{-1}\Sigma_i(\{\widehat{Q}_{b\to i}\}_{b\in\partial i}) = -\mathcal{F}_i + \frac{\sum_{\underline{\widehat{\mu}}} F_i e^{-xF_i}\prod_{b\in\partial i}\widehat{Q}_{b\to i}}{\sum_{\underline{\widehat{\mu}}} e^{-xF_i}\prod_{b\in\partial i}\widehat{Q}_{b\to i}},$$

$$x^{-1}\Sigma_a(\{Q_{j\to a}\}_{j\in\partial a}) = -\mathcal{F}_a + \frac{\sum_{\underline{\mu}} F_a e^{-xF_a}\prod_{j\in\partial a} Q_{j\to a}}{\sum_{\underline{\mu}} e^{-xF_a}\prod_{j\in\partial a} Q_{j\to a}},$$

$$x^{-1}\Sigma_{ai}(Q_{i\to a},\widehat{Q}_{a\to i}) = -\mathcal{F}_{ai} + \frac{\sum_{\mu,\widehat{\mu}} F_{ai} e^{-xF_{ai}}Q_{i\to a}\widehat{Q}_{a\to i}}{\sum_{\mu,\widehat{\mu}} e^{-xF_{ai}}Q_{i\to a}\widehat{Q}_{a\to i}}.$$

One can interpret the Bethe complexity as the difference of the Bethe free energy of the level-one model and a Bethe expression for the internal energy of the level one model,

$$x^{-1}\Sigma_{\rm Bethe} = \mathcal{F}_{\rm Bethe} - \langle F_{\rm Bethe}\rangle_{\rm cav}. \tag{22.15}$$

The bracket $\langle -\rangle_{\rm cav}$ is a natural average that can be read off from the above formulas.

Simplifications for $x = 1$

As alluded to before $x = 1$ plays a specially important role. So it is fortunate that a large portion of the formalism above can be simplified by eliminating entirely the need for reweighting factors. This makes the replica analysis much simpler and allows to make much simpler and precise numerical computations (e.g. by population dynamics) .

Let us first discuss the level-one Bethe free energy. Replacing (19.12), (19.13) and (19.14) into (22.13) one finds

$$\mathcal{F}_{\rm Bethe}(\underline{Q},\underline{\widehat{Q}})|_{x=1} = F_{\rm Bethe}(\underline{\mu}^{\rm av},\underline{\hat{\mu}}^{\rm av}) \tag{22.16}$$

which is the *usual* Bethe free energy expressed in terms of "average messages",

$$\mu^{\rm av}_{i\to a}(x_i) = \sum_{\mu_{i\to a}} \mu_{i\to a}(x_i) Q_{i\to a}(\mu_{i\to a}), \quad \hat\mu^{\rm av}_{a\to i}(x_i) = \sum_{\hat\mu_{a\to i}} \hat\mu_{a\to i}(x_i) \hat Q_{a\to i}(\hat\mu_{a\to i}).$$

Remarkably, the average messages satisfy the usual sum-product equations,

$$\mu^{\rm av}_{i\to a}(x_i) \simeq \sum_{x_i} \prod_{b\in\partial i\setminus a} \hat\mu^{\rm av}_{b\to i}(x_i), \qquad \hat\mu^{\rm av}_{i\to a}(x_i) \simeq \sum_{x_{\partial a}} f_a(x_{\partial a}) \prod_{j\in\partial a\setminus i} \mu^{\rm av}_{j\to a}(x_j).$$

One way to prove this is to notice that[5] $\delta_{Q_{i\to a}}\mathcal{F}_{\rm Bethe} = (\delta_{\mu^{\rm av}_{i\to a}} F_{\rm Bethe})\mu_{i\to a}(x_i)$ and $\delta_{\hat Q_{i\to a}}\mathcal{F}_{\rm Bethe} = (\delta_{\hat\mu^{\rm av}_{i\to a}} F_{\rm Bethe})\hat\mu_{i\to a}(x_i)$. Therefore if $(\underline{Q},\underline{\hat Q})$ is a stationary point of $\mathcal{F}_{\rm Bethe}|_{x=1}$ then $(\mu^{\rm av},\hat\mu^{\rm av})$ is a stationary point of $F_{\rm Bethe}$. Thus the cavity equations for $(\underline{Q},\underline{\hat Q})$ imply the sum-product equations for $(\mu^{\rm av},\hat\mu^{\rm av})$. This conclusion can also be reached by a direct calculation starting from the cavity equations for $x = 1$.

Conceptually $\mu^{\rm av}_{i\to a}(x_i)$ and $\hat\mu^{\rm av}_{i\to a}(x_i)$ are very natural messages to consider. Suppose for the sake of the argument that $Q(\mu_{i\to a})$ and $\hat Q(\hat\mu_{i\to a})$ are the true marginals of the level-one model. Then the average messages are the Gibbs averages of the dynamical variables of the level-one model (much like the magnetization is the Gibbs average of the spin variable). In other words if we sample among the set of solutions of the sum-product equations according to the weight $e^{-F_{\rm Bethe}}/Z_1(x=1)$ these are the expected messages that we get. From these expected messages one can reconstruct a Bethe measure which one can hope to be a good proxy for the convex superposition. However this is *not* a pure Bethe measure. As a consequence the marginals of this Bethe measure do not allow us to correctly sample from pure states $\mu^{(p)}(\underline{x})$. In particular for K-SAT they do not allow us to find solutions, and this is why BP guided decimation does not succeed above a certain density. When it does succeed this means that the the convex decomposition is essentially dominated by a unique Bethe measure (which is pure). The correct sampling procedure that suitably addresses these points is Survey Propagation guided decimation discussed in Chapter **??**.

We now turn to the Bethe complexity (22.15) for $x = 1$. For the free energy contribution we already have the simplification (22.16), so we only have to show how to eliminate the reweighting factors from the internal energy contribution.

[5] Formally $\delta_R G$ is an infinitesimal variation of G with respect to R.

Replacing (19.12) in $\langle F_i\rangle_{\rm cav}$ we find

$$\begin{aligned}\langle F_i\rangle_{\rm cav} &= \frac{\sum_{\underline{\hat{\mu}}} \ln\Big\{\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}_{b\to i}(x_i)\Big\}\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}_{b\to i}(x_i)\hat{Q}_{b\to i}}{\sum_{\underline{\hat{\mu}}}\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}_{b\to i}(x_i)\hat{Q}_{b\to i}}\\ &= \frac{\sum_{\underline{\hat{\mu}}} \ln\Big\{\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}_{b\to i}(x_i)\Big\}\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}_{b\to i}(x_i)\hat{Q}_{b\to i}}{\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}^{\rm av}_{b\to i}(x_i)}\\ &= \sum_{\underline{\hat{\mu}}}\ln\Big\{\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}_{b\to i}(x_i)\Big\}\sum_{x_i}\nu^{\rm av}_i(x_i)\prod_{b\in\partial i}\hat{R}_{b\to i}(\hat{\mu}_{b\to i}|x_i)\end{aligned}$$

In the last equality we have defined the probability distributions

$$\nu^{\rm av}_i(x_i) = \frac{\prod_{b\in\partial i}\hat{\mu}^{\rm av}_{b\to i}(x_i)}{\sum_{x_i}\prod_{b\in\partial i}\hat{\mu}^{\rm av}_{b\to i}(x_i)}, \qquad \hat{R}_{b\to i}(\hat{\mu}_{b\to i}|x_i) = \frac{\hat{\mu}_{b\to i}(x_i)\hat{Q}_{b\to i}}{\hat{\mu}^{\rm av}_{b\to i}(x_i)}$$

Replacing (19.13) in $\langle F_a\rangle_{\rm cav}$ we find

$$\begin{aligned}\langle F_a\rangle_{\rm cav} &= \frac{\sum_{\underline{\mu}} \ln\Big\{\sum_{x_{\partial a}}\prod_{i\in\partial a}\mu_{i\to a}(x_i)\Big\}\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu_{i\to a}(x_i)Q_{i\to a}}{\sum_{\underline{\mu}}\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu_{i\to a}(x_i)\hat{Q}_{i\to a}}\\ &= \frac{\sum_{\underline{\mu}} \ln\Big\{\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu_{i\to a}(x_i)\Big\}\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu_{i\to a}(x_i)Q_{i\to a}}{\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu^{\rm av}_{i\to a}(x_i)}\\ &= \sum_{\underline{\mu}}\ln\Big\{\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu_{i\to a}(x_i)\Big\}\sum_{x_{\partial a}}\nu^{\rm av}_a(x_{\partial a})\prod_{i\in\partial a}R_{i\to a}(\mu_{i\to a}|x_i)\end{aligned}$$

with the distributions

$$\nu^{\rm av}_a(x_{\partial a}) = \frac{f_a(x_{\partial a})\prod_{i\in\partial a}\mu^{\rm av}_{i\to a}(x_i)}{\sum_{x_{\partial a}} f_a(x_{\partial a})\prod_{i\in\partial a}\mu^{\rm av}_{i\to a}(x_i)}, \qquad R_{i\to a}(\mu_{i\to a}|x_i) = \frac{\mu_{i\to a}(x_i)Q_{i\to a}}{\mu^{\rm av}_{i\to a}(x_i)}$$

Replacing (19.14) in $\langle F_{ai}\rangle_{\rm cav}$ we find

$$\begin{aligned}\langle F_{ai}\rangle_{\rm cav} &= \frac{\sum_{\underline{\mu},\underline{\hat{\mu}}} \ln\Big\{\sum_{x_i}\hat{\mu}_{a\to i}(x_i)\mu_{i\to a}(x_i)\Big\}\sum_{x_i}\hat{\mu}_{a\to i}(x_i)\hat{Q}_{a\to i}\mu_{i\to a}(x_i)Q_{i\to a}}{\sum_{\underline{\mu},\underline{\hat{\mu}}}\sum_{x_i}\hat{\mu}_{a\to i}(x_i)\hat{Q}_{a\to i}\mu_{i\to a}(x_i)Q_{i\to a}}\\ &= \frac{\sum_{\underline{\mu},\underline{\hat{\mu}}} \ln\Big\{\sum_{x_i}\hat{\mu}_{a\to i}(x_i)\mu_{i\to a}(x_i)\Big\}\sum_{x_i}\hat{\mu}_{a\to i}(x_i)\hat{Q}_{a\to i}\mu_{i\to a}(x_i)Q_{i\to a}}{\sum_{x_i}\hat{\mu}^{\rm av}_{a\to i}(x_i)\mu^{\rm av}_{i\to a}(x_i)}\\ &= \sum_{\underline{\mu},\underline{\hat{\mu}}}\ln\Big\{\sum_{x_i}\hat{\mu}_{a\to i}(x_i)\mu_{i\to a}(x_i)\Big\}\sum_{x_i}\nu_{ai}(x_i)\hat{R}_{a\to i}(\hat{\mu}_{a\to i}|x_i)R_{i\to a}(\mu_{i\to a}|x_i)\end{aligned}$$

where

$$\nu^{\rm av}_{ai}(x_i) = \frac{\hat{\mu}^{\rm av}_{a\to i}(x_i)\mu^{\rm av}_{i\to a}(x_i)}{\sum_{x_i}\hat{\mu}^{\rm av}_{a\to i}(x_i)\mu^{\rm av}_{i\to a}(x_i)}$$

So far we have shown that the Bethe complexity can be expressed in terms of the average messages $\hat{\mu}^{\rm av}_{a\to i}$ and $\mu^{\rm av}_{i\to a}$ and the conditional distributions $\hat{R}_{a\to i}(\hat{\mu}_{a\to i}|x_i)$ and $R_{i\to a}(\mu_{i\to a}|x_i)$. We have already seen that the average messages satisfy the usual sum-product equations. We will now show that the conditional distributions satisfy similar equations.

Multiplying the cavity equations (22.10)-(22.11) by $\mu_{a\to i}(x_i)$ and $\hat{\mu}_{a\to i}(x_i)$, and using the expressions of the reweighting factor (22.12) we get for $x = 1$

$$\mu_{i\to a}(x_i)Q_{i\to a}(\mu_{i\to a}) \simeq \sum_{\underline{\widehat{\mu}}} \mathbb{1}_i(\mu_{i\to a}) \prod_{b\in\partial i\setminus a} \widehat{\mu}_{b\to i}(x_i)\widehat{Q}_{b\to i}(\widehat{\mu}_{b\to i})$$

$$\hat{\mu}_{a\to i}(x_i)\widehat{Q}_{a\to i}(\widehat{\mu}_{a\to i}) \simeq \sum_{\sim x_i} f_a(x_{\partial a}) \sum_{\underline{\mu}} \hat{\mathbb{1}}_a(\widehat{\mu}_{a\to i}) \prod_{j\in\partial a\setminus i} \mu_{j\to a}(x_j)Q_{j\to a}(\mu_{j\to a})$$

If we normalize each member of these equalities the proportionality relations become equalities. Here normalizing means dividing by the sums of the numerators over $\mu_{i\to a}$ and $\hat{\mu}_{i\to a}$. One finds a closed set of equations linking the conditional distributions,

$$R_{i\to a}(\mu_{i\to a}|x_i) = \sum_{\underline{\widehat{\mu}}} \mathbb{1}_i(\mu_{i\to a}) \prod_{b\in\partial i\setminus a} \hat{R}_{b\to i}(\hat{\mu}_{b\to i}|x_i) \tag{22.17}$$

$$\hat{R}_{a\to i}(\hat{\mu}_{a\to i}|x_i) = \sum_{\sim x_i} \pi_{a,i}(x_{\partial a\setminus i}|x_i) \sum_{\underline{\mu}} \hat{\mathbb{1}}_a(\widehat{\mu}_{a\to i}) \prod_{j\in\partial a\setminus i} R_{j\to a}(\mu_{j\to a}|x_j) \tag{22.18}$$

where

$$\pi_{a,i}(x_{\partial a\setminus i}|x_i) = \frac{f_a(x_{\partial a})\prod_{j\in\partial a\setminus i}\mu^{\rm av}_{j\to a}(x_j)}{\sum_{\sim x_i} f_a(x_{\partial a})\prod_{j\in\partial a\setminus i}\mu^{\rm av}_{j\to a}(x_j)}$$

These equations are quite similar to standard sum-product equations and are much easier to solve than the original cavity equations.

22.4 Application to K-SAT

We work at finite temperature for reasons that will become clear below. It is straightforward to apply the general theory to K-SAT using the parametrization of messages (15.8). With this parametrization the sum-product equations become (15.11)-(15.15) (with the necessary modification for finite temperatures) so to write the cavity equations (22.10)-(22.11) we make the replacements

$$\mathbb{1}_i \to \delta\Big(h_{i\to a} - \sum_{b\in S_{ia}} \hat{h}_{b\to i} + \sum_{b\in U_{ia}} \hat{h}_{b\to i})$$

and

$$\hat{\mathbb{1}}_a \to \delta\Big(\hat{h}_{a\to i} + \frac{1}{2}\ln\Big\{1 - (1-e^{-\beta}) \prod_{j\in\partial a\setminus i} \frac{1-\tanh_{j\to a}}{2}\Big\}\Big).$$

Furthermore all sums become integrals (dropping subscripts) $\sum_\mu Q(\mu)\cdots \to \int dh Q(h)\ldots$ and $\sum_{\hat\mu} \hat Q(\hat\mu)\cdots \to \int d\hat h \hat Q(\hat h)\ldots$.

To get the general expressions for the level-one Bethe free energy and complexity (22.13), (22.14) one uses F_i, F_a and F_{ai} given in (19.23)-(19.25) and replaces sums by integrals as just indicated.

For the simplified formulas when $x = 1$ we introduce averaged messages

$$\tanh h^{\rm av}_{i\to a} = \int Q(h_{i\to a}) \tanh h_{i\to a}, \qquad \tanh \hat h^{\rm av}_{i\to a} = \int \hat Q(h_{i\to a}) \tanh \hat h_{i\to a}$$

which satisfy the finite temperature version of message passing equations (15.11)-(15.15). With these average messages the level-one Bethe free energy is the same than (19.21), i.e. it is given by the RS expression. The other set of message passing equations (22.17), (22.18) are obtained by replacing indicator functions by Dirac functions as above, $x_i \to s_i$, and (dropping subscripts) $\sum_\mu R(\mu|x_i)\cdots \to \int dh R(h|x_i)\ldots$, $\sum_\mu \hat R(\hat\mu|x_i)\cdots \to \int d\hat h \hat R(\hat h|x_i)\ldots$. With all these ingredients one also writes down the Bethe complexity for $x = 1$. This is left as an exercise.

22.5 Replica Symmetry Broken Analysis for K-SAT

General analysis

The phase diagram of K-SAT is derived from the cavity equations and the Bethe formulas through a "density evolution type" analysis done at the level of the cavity messages $Q_{i\to a}(.)$, $\hat Q_{i\to a}(.)$. One can write down formal equations linking probability distributions of the cavity messages $\mathcal{Q}(Q(.))$ and $\hat{\mathcal{Q}}(\hat Q(.))$ which are often called replica symmetry broken (1-RSB) equations. The associated average level-one free energy functional is the 1-RSB free energy.[6] Let us illustrate the 1RSB replica formula for the free energy in more detail.

Fix a trial distribution $\mathcal{Q}(Q(.))$. Take $K-1$ iid copies of the random distribution $Q(.)$ and define the random variable $\hat Q(.)$ [compute reweighting factor in here]

$$\hat Q(\hat\xi) \stackrel{\rm distr}{=} \int \prod_{k=1}^{K-1} d\xi_k Q(h_k) \left(2 - \prod_{k=1}^{K-1} \frac{1-\tanh h_k}{2}\right)^x \tag{22.19}$$

$$\times \frac{\delta\left(\hat h + \frac{1}{2}\ln\left\{1 - (1-e^{-\beta})\prod_{k=1}^{K-1}\frac{1-\tanh h_k}{2}\right\}\right)}{\int \prod_{k=1}^{K-1} d\xi_k Q(h_k)\left(2 - \prod_{k=1}^{K-1}\frac{1-\tanh h_k}{2}\right)^x} \tag{22.20}$$

This random distribution is distributed according to $\hat{\mathcal{Q}}(\hat Q(.))$. Pick two Poisson integers p and q of mean $\alpha K/2$ and $p+q$ iid copies of the random distribution

[6] Historically these equations were first derived in the context of the replica method and involve breaking the symmetry between replicas of the original system, hence the name.

$\hat{Q}(.)$. Let

$$
\begin{aligned}
& f(Q(.),\hat{Q}(.),p,q) \\
& = x^{-1}\ln\Bigg\{\int\prod_{k=1}^{p+q}d\hat{h}_k\hat{Q}_k(\hat{h}_k)\left(\prod_{k=1}^{p}(1-\tanh\hat{h}_k)\prod_{k=p+1}^{p+q}(1+\tanh\hat{h}_k)\right. \\
& \qquad \left.+\prod_{k=1}^{p}(1+\tanh\hat{h}_k)\prod_{k=p+1}^{p+q}(1-\tanh\hat{h}_k)\right)^{x}\Bigg\} \\
& + x^{-1}\ln\Bigg\{\int\prod_{k=1}^{K}dh_kQ_k(h_k)\left(1-(1-e^{-beta})\prod_{k=1}^{K}\frac{1-\tanh h_k}{2}\right)^{x}\Bigg\} \\
& - x^{-1}\ln\Bigg\{\int dhQ(h)d\hat{h}\hat{Q}(\hat{h})\left(1+\tanh h\tanh\hat{h}\right)^{x}\Bigg\}
\end{aligned}
$$

The 1-RSB free energy functional is defined as

$$f_{\text{1RSB}}(\mathcal{Q}(.);x)=\mathbb{E}[f(Q(.),\hat{Q}(.),p,q)]$$

where the expectation is with respect to Q, $\hat{Q}$, p, q. The stationnary point equation of the 1RSB functional yield the 1RSB fixed point equations for the distributions $\mathcal{Q}(.),\hat{\mathcal{Q}}(.)$. These are the DE equations corresponding to the cavity message passing equations: one of them is precisely (22.19). The derivation of the second one is left as an exercise to the reader.

The interpolation method allows to prove the following theorem,

THEOREM 22.1 *For any trial distribution $\mathcal{Q}(.)$ and any $0<x<1$, the thermodynamic limit of the free energy of SAT exists, and moreover is lower bounded by the 1RSB formula*

$$\lim_{n\to+\infty}\frac{1}{n}\mathbb{E}[\ln Z]\leq f_{\text{1RSB}}(\mathcal{Q}(.);x)$$

The 1RSB conjecture states that taking the supremum over $\mathcal{Q}(.)$ and x on the right hand side yields an equality. We point out that this conjecture is surprizing from the standpoint of deterministic mean field models because for such models the variational expression for the free energy always involves a minimization (e.g. in the CW model). Here the free energy of K-SAT is given by a variational principle involving a maximization over trial parameters, rather than a minimization. This feature is in fact generic for replica formulas was already encountered in the early days of of the replica method. Note that it has nothing to do with the fact that the solution is RS or RSB. Now, for coding the RS variational expression for the free energy involves a minimization: this is surprizing from the standpoint of replica formulas! A look at the derivation of the bounds in the interpolation method (Chapter 21) shows that this can be traced to the channel or Nishimori symmetry.

Accepting the 1RSB conjecture teaches us something about the correct choice of the Parisi parameter x. Indeed recall that the complexity is the Gibbs-Shannon

K	α_d	$\alpha_{d,80,3}$	α_c	$\alpha_{c,80,3}$	α_s	$\alpha_{s,80,3}$
3	3.86	3.86	3.86	3.86	4.267	4.268
4	9.38	9.55	9.55	9.56	9.93	10.06

Table 22.1 Thresholds of individual and coupled K-SAT model for $L = 80$ and $w = 3$. Note that for 3-SAT teh dynamical and condensation thresholds are the same. The condensation and SAT-UNSAT thresholds correspond to non analyticities of the entropy and ground state energy and remain unchanged (for $L \to +\infty$). Already for $w = 3$ the dynamical threshold saturates very close to α_c and α_s.

entropy of the level-one model $\Sigma(x) = x^2 \frac{\partial}{\partial x^2} f_1(x)$. In place of $f_1(x)$ we use the 1RSB free energy formula (for the optimal $\mathcal{Q}(.)$), a function of x that can be computed by population dynamics. As long as $\Sigma(x) \geq 0$ for $0 < x < 1$ the optimal x is given by $x = 1$. We will see that this happens as long as $\alpha < \alpha_c$, where α_c is called the condensation threshold. When $\alpha > \alpha_c$ we get $\Sigma(x) \geq 0, 0 < x < x_*(\alpha)$, and $\Sigma(x) \leq 0$, $x_*(\alpha) < x < 1$, so that the optimal value of the parisi parameter is $x = x_*(\alpha)$. As we will see in the next chapter at the SAT-UNSAT density we have $x_*(\alpha_s) = 0$; for this value of the Parisi parameter the 1RSB formulas also simplify and yield the *survey propagation formulas*. This discussion shows that the condensation threshold can be obtained from the 1RSB complexity computed for $x = 1$. The same quantity will also give us the dynamical threshold $\alpha_d = \inf\{\alpha | \Sigma(x = 1) > 0\}$. This is sufficient motivation for giving the simplified 1RSB formulas for $x = 1$.

Analysis for $x = 1$

explain that free energy is RS free energy. Give the complexity and the fixed point equations without reweighting factor. Give population dynamic pseudo code.

22.6 Dynamical and Condensation Thresholds

The most important feature of the convex decomposition ansatz is the number of pure Bethe states involved. The RSB analysis of the level-one model predicts the existence of two sharply defined thresholds α_d and α_c at which the nature of the convex decomposition (22.1) changes drastically. The values of these thresholds are given in Table 22.1 and compared to the SAT-UNSAT threshold for a few values of K. Note that $K = 3$ is not generic because $\alpha_d = \alpha_c$. Figure 22.3 gives a pictorial view of the transitions associated with the decomposition (22.1). The goal of this paragraph is to explain this picture.

As already explained for $\alpha < \alpha_c$ we have $\Sigma(x) \geq 0$ for all $x \in [0, 1]$ and the correct value of the Parisi paramter is $x = 1$. The entropy is given by the RS

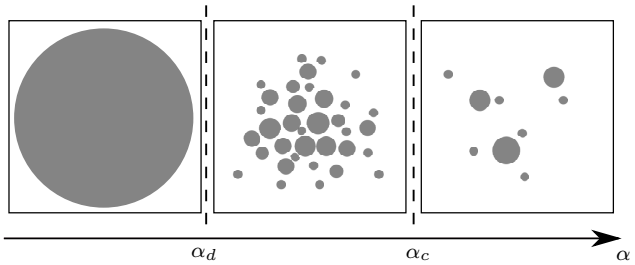


Figure 22.3 Pictorial representation of the decomposition of the Gibbs distribution into a convex superposition of extremal states. Balls represent extremal states (their size represents their internal entropy). For $\alpha < \alpha_d$ there is one extremal state. For $\alpha_d < \alpha < \alpha_c$ there are exponentially many extremal states (with the same internal free enetropy) that dominate to the convex superposition. For $\alpha > \alpha_c$ there is a finite number of extremal states that dominate the convex superposition.

formula. In particular this function is analytic for $\alpha < \alpha_c$ and therefore there is no thermodynamic static phase transition in this range. Above the condensation threshold the correct choice of the Parisi parameter $x = x_*(\alpha)$ forces the complexity to vanish. The Gibbs measure is supported by a finite number of pure Bethe states. Because of the change in x the entropy is not given by the same analytic function below and above α_c. therefore the condensation threshold is a thermodynamic static phase transition.

The complexity $\Sigma(x = 1)$ has a non trivial behavior below the condensation threshold. It vanishes for $\alpha < \alpha_d$, jumps to a positive value at α_d and is concave decreasing with increasing α till it becomes negative just above α_c. What is the interpretation of this result? Recall that the complexity is the growth rate for the number of pure Bethe states in the convex decomposition of the Gibbs measure, and the weights of this decomposition are given by the entropies of the pure states. For densities below the dynamical threshold the Gibbs measure is supported by one pure Bethe state. It is not excluded that there exist other ones of exponentially smaller weights. For densities between the dynamical and condensation thresholds an exponential number of pure Bethe states of identical entropy contribute to the convex sum. On the other hand beyond teh condensation threshold the measure is supported by only a finite number of pure Bethe states with equal entropy. All other states have exponentially smaller weights (the cavity method also predicts that the statistics of these weights is a Poisson-Dirichlet process). As already stressed the entropy is insensitive to the dynamical threshold, and thsi is not a static phase transition threshold. Rather, as its name indicates one expects that the proliferation of pure states affects the dynamics of algorithms local algorithms. In this course we have seen indications that this indeed occurs for BP guided decimation. In fact BP decimation fails slightly below α_d. This is not believed to be an inconsistency of the theory, but rather a consequence of the fcat that during teh decimation process the graph ensemble changes and therefore the threshold for BP guided decimation is set

by a different graph ensemble. It is believd that for Markov Chain Monte Carlo algorithms such as Glauber dynamics the equilibration time diverges exactly at α_d. This has been checked in simpler models.

It is interesting to consider the spatially coupled version of the K-SAT model. The same cavity theory can be applied and the RSB equations solved with the appropriate boundary conditions. this allows to determine the dynamical and condensation thresholds of the spatially coupled model (see table 22.1). The numerical observations suggest that the condensation threshold remains invariant in the limit of an infinite chain. This is consistent with its interpretation as a singularity of the entropy. In fact one can prove by the interpolation method that the entropy of the infinite coupled chain and underlying uncoupled model are the same, and therefore α_c is the same for both models, namely $\lim_{L\to+\infty} \alpha_c(w, L) = \alpha_c$. On the other hand it is observed that the dynamical threshold saturates towards the condensation threshold in the limit of an infinite chain and a large coupling range, namely $\lim_{w\to+\infty} \lim_{L\to+\infty} \alpha_d(w, L) = \alpha_c$. These results are conssistent with the interpretation of the dynamical threshold as an algorithmic barrier and the condensation threshold as a static phase transition threshold.

In section **??** we indicated that in Ising models there is an intimate connection between the decay of correlations and the extremality of the Gibbs measure. This is also true for constarint satisfaction models defined on random graph ensembles. However the correct correlation functions have to be used. In the present context two type of correlation functions have been discovered. Point-to-set correlations defined as

$$C(i, B) = \sum_{\underline{x}_{\partial B}} \nu(\underline{x}_{\partial B}(\nu(x_i|\underline{x}_{\partial B}) - \nu(x_i))^2$$

where B is the set $\{x_j|\{\text{dist}(x_i, x_j) \geq d\}$. Within the cavity method one can compute $\lim_{d\to+\infty} \lim_{n\to+\infty} C(i, B$ and finds that the limit vanishes $\alpha < \alpha_d$, while it remains strictly positive for $\alpha > \alpha_d$. Moreover for all $\alpha < \alpha_c$ and all randomly chosen bounded set of variables

$$\mathbb{E}[(\nu(x_{i_1}, \dots x_{i_k}) - \nu(x_{i_1}) \dots \nu(x_{i_k}))^2] = O(\frac{1}{n})$$

This is similar to the decoupling property we discussed for the CW model. At α_c this decoupling property breaks down.

23 Cavity Method: Survey Propagation

We have seen BP guided decimation does not find solutions beyond α_d. This chapter is an application of the cavity theory to find solutions of K-sat for densities beyond dynamical threshold. With level one model we learned about α_d and α_c. But have not yet computed α_s. We will aplly level one model with $x = 0$. RSB analysis with $x = 0$ leads to SP equations. Allows to compute α_s. Older point of view this was called "energetic cavity method". With decimation process we find solutions up densities close to α_s.

23.1 Survey propagation equations

Simplifify equations of previous chapter for $x = 0$. Derive equations.

23.2 Connection with the energetic cavity method

Briefly explain min sum point of view. Different level one model. Notion of SP complexity.

23.3 RSB analysis and sat-unsat threshold

Compute internal entropy and SP complexity. They both yield the sat-unsat threshold.

23.4 Survey propagation guided decimation

Algorithm. Experiments.

24 *Summary of Part III*

blabla

Notes

References

[1] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge Univ. Press, 2003.

[2] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. of the 29th annual ACM Symposium on Theory of Computing*, 1997, pp. 150–159.