9

FACTOR GRAPHS AND GRAPH ENSEMBLES

{ch:Graphs}

Systems involving a large number of simple variables with mutual dependencies (or constraints, or interactions) appear recurrently in several fields of science. It is often the case that such dependencies can be ‘factorized’ in a non-trivial way, and distinct variables interact only ‘locally’. In statistical physics, the fundamental origin of such a property can be traced back to the locality of physical interactions. In computer vision it is due to the two dimensional character of the retina and the locality of reconstruction rules. In coding theory it is a useful property for designing a system with fast encoding/decoding algorithms. This important structural property plays a crucial role in many interesting problems.

There exist several possibilities for expressing graphically the structure of dependencies among random variables: undirected (or directed) graphical models, Bayesian networks, dependency graphs, normal realizations, etc. We adopt here the *factor graph* language, because of its simplicity and flexibility.

As argumented in the previous Chapters, we are particularly interested in *ensembles* of probability distributions. These may emerge either from ensembles of error correcting codes, or in the study of disordered materials, or, finally, when studying random combinatorial optimization problems. Problems drawn from these ensembles are represented by factor graphs which are themselves *random*. The most common examples are random hyper-graphs, which are a simple generalization of the well known random graphs.

Section 9.1 introduces factor graphs and provides a few examples of their utility. In Sec. 9.2 we define some standard ensembles of random graphs and hyper-graphs. We summarize some of their important properties in Sec. 9.3. One of the most surprising phenomena in random graph ensembles, is the sudden appearance of a ‘giant’ connected component as the number of edges crosses a threshold. This is the subject of Sec. 9.4. Finally, in Sec. 9.5 we describe the local structure of large random factor graphs.

9.1 Factor graphs

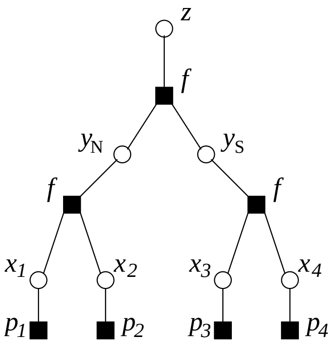
{se:FactorGeneral}

9.1.1 *Definitions and general properties*

{se:FactorDefinition}

We begin with a toy example.

Example 9.1 A country elects its president among two candidates $\{A, B\}$ according to the following peculiar system. The country is divided into four regions $\{1, 2, 3, 4\}$, grouped in two states: North (regions 1 and 2), and South (3 and 4). Each of the regions chooses its favorites candidate according to popular vote: we call him $x_i \in \{A, B\}$, with $i \in \{1, 2, 3, 4\}$. Then a North candidate $y_{\rm N}$, and a



{fig:ElectionFactor} FIG. 9.1. Factor graph representation of the electoral process described in Example 1.

South candidate y_S are decided according to the following rule. If the preferences x_1 and x_2 in regions 1 and 2 agree, then y_N takes this same value. In they don't agree y_N is decided according to a fair coin trial. The same procedure is adopted for the choice of y_S, given x_3, x_4. Finally, the president $z \in \{A, B\}$ is decided on the basis of the choices y_N and y_S in the two states using the same rule as inside each state.

A polling institute has obtained fairly good estimates of the probabilities $p_i(x_i)$ for the popular vote in each region i to favor the candidate x_i. They ask you to calculate the odds for each of the candidates to become the president.

It is clear that the electoral procedure described above has important 'factorization' properties. More precisely, the probability distribution for a given realization of the random variables $\{x_i\}$, $\{y_j\}$, z has the form:

$$P(\{x_i\}, \{y_j\}, z) = f(z, y_\mathrm{N}, y_\mathrm{S})\, f(y_\mathrm{N}, x_1, x_2)\, f(y_\mathrm{S}, x_3, x_4) \prod_{i=1}^{4} p_i(x_i)\,. \tag{9.1}$$

⋆ We invite the reader to write explicit forms for the function f. The election process, as well as the above probability distribution, can be represented graphically as in Fig. 9.1. Can this particular structure be exploited when computing the chances for each candidate to become president?

Abstracting from the previous example, let us consider a set of N variables $x_1, \dots, x_N$ taking values in a finite alphabet $\mathcal{X}$. We assume that their joint probability distribution takes the form

$$P(\underline{x}) = \frac{1}{Z} \prod_{a=1}^{M} \psi_a(\underline{x}_{\partial a})\,. \tag{9.2}$$

Here we use the shorthands $\underline{x} \equiv \{x_1, \dots, x_N\}$, and $\underline{x}_{\partial a} \equiv \{x_i \,|\, i \in \partial a\}$, where $\partial a \subseteq [N]$. The set of indices ∂a, with $a \in [M]$, has size $k_a \equiv |\partial a|$. When necessary,

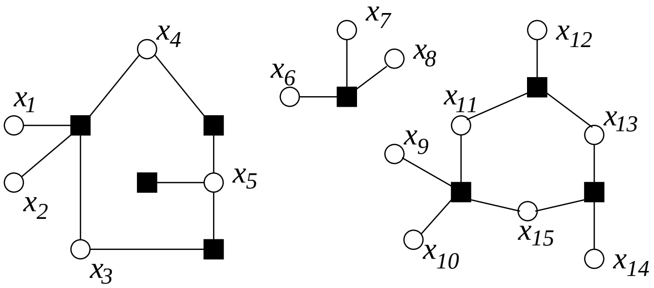


FIG. 9.2. A generic factor graph is formed by several connected components. Variables belonging to distinct components (for instance x_3 and x_{15} in the graph above) are statistically independent. {fig:DisconnectedFactor}

we shall use the notation $\{i_1^a, \dots, i_{k_a}^a\} \equiv \partial a$ to denote the variable indices which correspond to the factor a, and $\underline{x}_{i_1^a,\dots,i_{k_a}^a} \equiv \underline{x}_{\partial a}$ for the corresponding variables. The **compatibility functions** $\psi_a : \mathcal{X}^{k_a} \to \mathbb{R}$ are non-negative, and Z is a positive constant. In order to completely determine the form (9.2), we should precise both the functions $\psi_a(\cdot)$, and an ordering among the indices in ∂a. In practice this last specification will be always clear from the context.

Factor graphs provide a graphical representations of distributions of the form (9.2). The factor graph for the distribution (9.2) contains two types of nodes: N **variable nodes**, each one associated with a variable x_i (represented by circles); M **function nodes**, each one associated with a function ψ_a (squares). An edge joins the variable node i and the function node a if the variable x_i is among the arguments of $\psi_a(\underline{x}_{\partial a})$ (in other words if $i \in \partial a$). The set of function nodes that are adjacent to (share an edge with) the variable node i, is denoted as ∂i. The graph is bipartite: an edge always joins a variable node to a function nodes. The reader will easily check that the graph in Fig. 9.1 is indeed the factor graph corresponding to the factorized form (9.1). The degree of a variable node (defined as in usual graphs by the number of edges which are incident on it) is arbitrary, but the degree of a function node is always ≥ 1. ⋆

The basic property of the probability distribution (9.2) encoded in its factor graph, is that two 'well separated' variables interact uniquely through those variables which are interposed between them. A precise formulation of this intuition is given by the following observation, named the **global Markov property**: {propo:GlobalMarkov}

Proposition 9.2 *Let $A, B, S \subseteq [N]$ be three disjoint subsets of the variable nodes, and denote by $\underline{x}_A$, $\underline{x}_B$ and $\underline{x}_S$ denote the corresponding sets of variables. If S 'separates' A and B (i.e., if there is no path on the factor graph joining a node of A to a node of B without passing through S) then*

$$P(\underline{x}_A, \underline{x}_B | \underline{x}_S) = P(\underline{x}_A | \underline{x}_S)\, P(\underline{x}_B | \underline{x}_S)\,. \tag{9.3}$$

In such a case the variables $\underline{x}_A, \underline{x}_B$ are said to be conditionally independent.

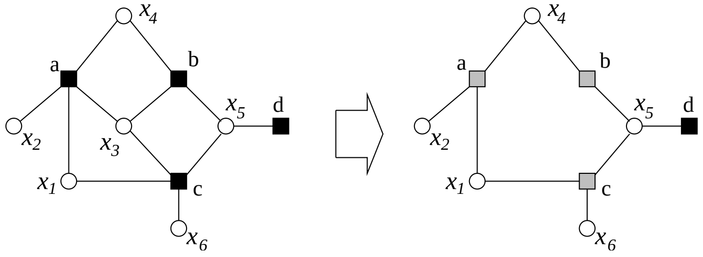


FIG. 9.3. The action of conditioning on the factor graph. The probability distribution on the left has the form $P(\underline{x}_{1\dots 6}) \propto f_a(\underline{x}_{1\dots 4})f_b(\underline{x}_{3,4,5})f_c(\underline{x}_{1,3,5,6})f_d(\underline{x}_5)$. After conditioning on x_3, we get $P(\underline{x}_{1\dots 6}|x_3 = x_*) \propto f'_a(\underline{x}_{1,2,4})f'_b(\underline{x}_{4,5})f'_c(\underline{x}_{1,5,6})f_d(\underline{x}_5)$. Notice that the functions $f'_a(\cdot)$, $f'_b(\cdot)$, $f'_c(\cdot)$ (gray nodes on the right) are distinct from $f_a(\cdot)$, $f_b(\cdot)$, $f_c(\cdot)$ and depend upon the value of x_*.

{fig:ConditionFactor}

Proof: It is easy to provide a 'graphical' proof of this statement. Notice that, if the factor graph is disconnected, then variables belonging to distinct components are independent, cf. Fig. 9.2. Conditioning upon a variable x_i is equivalent to eliminating the corresponding variable node from the graph and modifying the adjacent function nodes accordingly, cf. Fig. 9.3. Finally, when conditioning upon $\underline{x}_S$ as in Eq. (9.3), the factor graph gets split in such a way that A and B belong
⋆ to distinct components. We leave to the reader the exercise of filling the details. □

It is natural to wonder whether any probability distribution which is 'globally Markov' with respect to a given graph can be written in the form (9.2). In general, the answer is negative, as can be shown on a simple example. Consider the small factor graph in Fig. (9.4). The global Markov property has a non trivial content only for the following choice of subsets: $A = \{1\}$, $B = \{2, 3\}$, $S = \{4\}$. The most general probability distribution such that x_1 is independent from $\{x_2, x_3\}$ conditionally to x_4 is of the type $f_a(x_1, x_2)f_b(x_2, x_3, x_3)$. The probability distribution encoded by the factor graph is a special case where $f_b(x_2, x_3, x_4) = f_c(x_2, x_3)f_d(x, x_4)f_e(x_4, x_2)$.

The factor graph of our counterexample, Fig. 9.4, has a peculiar property: it contains a subgraph (the one with variables $\{x_2, x_3, x_4\}$) such that, for any pair of variable nodes, there is a function node adjacent to both of them. We call any factor subgraph possessing this property a **clique**[24]. It turns out that, once one gets rid of cliques, the converse of Proposition 9.2 can be proved. We shall 'get rid' of cliques by completing the factor graph. Given a factor graph F, its **completion** $\overline{F}$ is obtained by adding one factor node for each clique in the

[24] In usual graph theory, the word clique refers to graph (recall that a graph is defined by a set of nodes and a set of edges which join node *pairs*), rather than to factor graphs. Here we use the same word in a slightly extended sense.

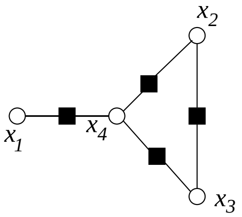


FIG. 9.4. A factor graph with four variables. $\{x_1\}$ and $\{x_2, x_3\}$ are independent conditionally to x_4. The set of variables $\{x_2, x_3, x_4\}$ and the three function nodes connecting two points in this set form a clique.

{fig:FactorClique}

graph and connecting it to each variable node in the clique and to no other node (if such a node does not already exist).

Theorem 9.3. (Hammersley-Clifford) *Let $P(\cdot)$ be a strictly positive probability distributions over the variables $\underline{x} = (x_1, \dots, x_N) \in \mathcal{X}^N$, satisfying the global Markov property (9.3) with respect to a factor graph F. Then P can be written in the factorized form (9.2), with respect to the completed graph $\overline{F}$.*

Roughly speaking: the only assumption behind the factorized form (9.2) is the rather weak notion of locality encoded by the global Markov property. This may serve as a general justification for studying probability distributions having a factorized form. Notice that the positivity hypothesis $P(x_1, \dots, x_N) > 0$ is not just a technical assumption: there exist counterexamples to the Hammersley-Clifford theorem if P is allowed to vanish.

9.1.2 *Examples*

{se:FactorExamples}

Let us look at a few examples

Example 9.4 The random variables $X_1, \dots, X_N$ taking values in the finite state space $\mathcal{X}$ form a **Markov chain of order** r (with $r < N$) if

$$P(x_1 \dots x_N) = P_0(x_1 \dots x_r) \prod_{t=r}^{N-1} w(x_{t-r+1} \dots x_t \to x_{t+1})\,, \tag{9.4}$$

for some non-negative transition probabilities $\{w(x_{-r} \dots x_{-1} \to x_0)\}$, and initial condition $P_0(x_1 \dots x_r)$, satisfying the normalization conditions

$$\sum_{x_1 \dots x_r} P_0(x_1 \dots x_r) = 1\,, \qquad \sum_{x_0} w(x_{-r} \dots x_{-1} \to x_0) = 1\,. \tag{9.5}$$

The parameter r is the ‘memory range’ of the chain. Ordinary Markov chains have $r = 1$. Higher order Markov chains allow to model more complex phenomena. For instance, in order to get a reasonable probabilistic model of the English language with the usual alphabet $\mathcal{X} = \{$a,b,... z, blank$\}$ as state space, a memory of the typical size of words ($r \geq 6$) is probably required.

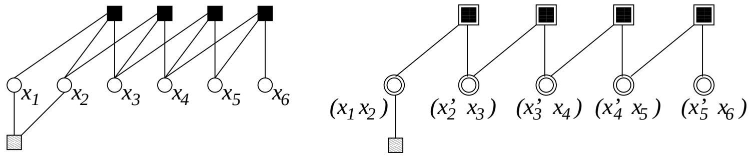


FIG. 9.5. On the left: factor graph for a Markov chain of length $N = 6$ and memory range $r = 2$. On the right: by adding auxiliary variables, the same probability distribution can be written as a Markov chain with memory range $r = 1$. {fig:FactorMarkov}

It is clear that Eq. (9.4) is a particular case of the factorized form (9.2). The corresponding factor graph includes N variable nodes, one for each variable x_i, $N - r$ function nodes for each of the factors $w(\cdot)$, and one function node for the initial condition $P_0(\cdot)$. In Fig. 9.5 we present a small example with $N = 6$ and $r = 2$.

Notice that a Markov chain with memory r and state space $\mathcal{X}$ can always be rewritten as a Markov chain with memory 1 and state space $\mathcal{X}^r$. The transition probabilities $\hat{w}$ of the new chain are given in terms of the original ones

$$\hat{w}(\vec{x} \to \vec{y}) = \begin{cases} w(x_1, \dots, x_r \to y_r) & \text{if } x_2 = y_1,\, x_3 = y_2, \dots x_r = y_{r-1}\,, \\ 0 & \text{otherwise,} \end{cases} \tag{9.6}$$

where we used the shorthands $\vec{x} \equiv (x_1, \dots, x_r)$ and $\vec{y} = (y_1, \dots, y_r)$. Figure 9.5 shows the reduction to an order 1 Markov chain in the factor graph language.

What is the content of the global Markov property for Markov chains? Let us start from the case of order 1 chains. Without loss of generality we can choose S as containing one single variable node (let's say the i-th) while A and B are, respectively the nodes on the left and on the right of i: $A = \{1, \dots, r-1\}$ and $B = \{r+1, \dots, N\}$. The global Markov property reads

$$P(x_1 \dots x_N | x_i) = P(x_1 \dots x_{i-1} | x_i)\, P(x_{i+1} \dots x_N | x_i)\,, \tag{9.7}$$

which is just a rephrasing of the usual Markov condition: $X_{i+1} \dots X_N$ depend $\star$ upon $X_1 \dots X_i$ uniquely through X_i. We invite the reader to discuss the global Markov property for order r Markov chains.

{ex:FirstLinearCode}

Example 9.5 Consider the code $\mathfrak{C}$ of block-length $N = 7$ defined by the codebook:

$$\begin{aligned} \mathfrak{C} = \{(x_1, x_2, x_3, x_4) \in \{0,1\}^4 \mid\ & x_1 \oplus x_3 \oplus x_5 \oplus x_7 = 0\,, \\ & x_2 \oplus x_3 \oplus x_6 \oplus x_7 = 0\,,\ x_4 \oplus x_5 \oplus x_6 \oplus x_7 = 0\}\,. \end{aligned} \tag{9.8}$$

Let $P_0(\underline{x})$ be the uniform probability distribution over the codewords: as discussed in Chap. 6, it is reasonable to assume that encoding produces codewords according to such a distribution. Then:

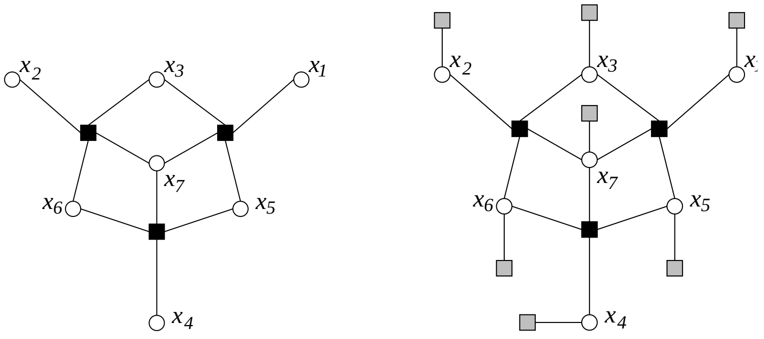


FIG. 9.6. Left: factor graph for the uniform distribution over the code defined in Eq. (9.8). Right: factor graph for the distribution of the transmitted message conditional to the channel output. Gray function nodes encode the information carried by the channel output.

{fig:FactorHamming}

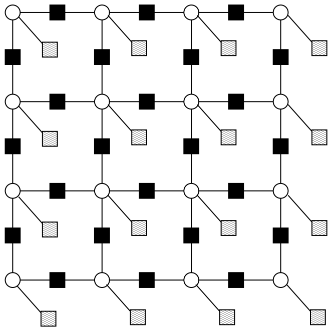


FIG. 9.7. Factor graph for an Edwards-Anderson model with size $L = 4$ in $d = 2$ dimensions. Full squares correspond to pairwise interaction terms $-J_{ij}\sigma_i\sigma_j$. Hatched squares denote magnetic field terms $-B\sigma_i$.

{fig:FactorIsing}

$$P_0(\underline{x}) = \frac{1}{Z_0}\,\mathbb{I}(x_1 \oplus x_3 \oplus x_5 \oplus x_7 = 0)\,\mathbb{I}(x_2 \oplus x_3 \oplus x_6 \oplus x_7 = 0)\,\cdot \qquad (9.9)$$
$$\cdot\,\mathbb{I}(x_4 \oplus x_5 \oplus x_6 \oplus x_7 = 0)\,,$$

where $Z_0 = 16$ is a normalization constant. This distribution has the form (9.2) and the corresponding factor graph is reproduced in Fig. 9.6.

Suppose that a codeword in $\mathfrak{C}$ is transmitted through a binary memoryless channel, and that the message $(y_1, y_2, \dots y_7)$ is received. As argued in Chap. 6, it is useful to consider the probability distribution of the transmitted message conditional to the channel output, cf. Eq. (6.3). Show that the factor graph representation for this distribution is the one given in Fig. 9.6, right-hand frame. ⋆

Example 9.6 In Sec. 2.6 we introduced the Edwards-Anderson model, a statistical mechanics model for spin glasses, whose energy function reads: $E(\underline{\sigma}) = -\sum_{(ij)} J_{ij}\sigma_i\sigma_j - B\sum_i \sigma_i$. The Boltzmann distribution can be written as

$$p_\beta(\underline{\sigma}) = \frac{1}{Z}\prod_{(ij)} e^{\beta J_{ij}\sigma_i\sigma_j}\prod_i e^{\beta B\sigma_i}\,, \tag{9.10}$$

with i runs over the sites of a d-dimensional cubic lattice of side L: $i \in [L]^d$, and (ij) over the couples of nearest neighbors in the lattice. Once again, this distribution admits a factor graph representation, as shown in Fig. 9.7. This graph includes two types of function nodes. Nodes corresponding to pairwise interaction terms $-J_{ij}\sigma_i\sigma_j$ in the energy function are connected to two neighboring variable nodes. Nodes representing magnetic field terms $-B\sigma_i$ are connected to a unique variable.

{ex:SatFactor}

Example 9.7 Satisfiability is a decision problem introduced in Chap. 3. Given N boolean variables $x_1, \dots, x_N \in \{T, F\}$ and a bunch of M logical clauses among them, one is asked to find a truth assignment verifying all of the clauses. The logical AND of the M clauses is usually called a formula. As an example, consider the following formula over $N = 7$ variables:

$$(x_1 \vee x_2 \vee \overline{x_4}) \wedge (x_2 \vee x_3 \vee x_5) \wedge (\overline{x_4} \vee \overline{x_5}) \wedge (x_5 \vee \overline{x_7} \vee \overline{x_6})\,. \tag{9.11}$$

For a given satisfiability formula, it is quite natural to consider the uniform probability distribution $P_{\rm sat}(x_1, \dots, x_N)$ over the truth assignments which satisfy (9.11)(whenever such an assignment exist). A little thought shows that such a distribution can be written in the factorized form (9.2). For instance, the formula (9.11) yields

$$\begin{aligned} P_{\rm sat}(x_1, \dots, x_7) = \frac{1}{Z_{\rm sat}}\, & \mathbb{I}(x_1 \vee x_2 \vee \overline{x_4})\, \mathbb{I}(x_2 \vee x_3 \vee x_5))\, \mathbb{I}(\overline{x_4} \vee \overline{x_5}) \cdot \\ & \cdot \mathbb{I}(x_5 \vee \overline{x_7} \vee \overline{x_6})\,, \end{aligned} \tag{9.12}$$

where $Z_{\rm sat}$ is the number of distinct truth assignment which satisfy Eq. (9.11).
⋆ We invite the reader to draw the corresponding factor graph.

Exercise 9.1 Consider the problem of coloring a graph $\mathcal{G}$ with q colors, already encountered in Sec. 3.3. Build a factor graph representation for this problem, and write the associated compatibility functions. [Hint: in the simplest such representation the number of function nodes is equal to the number of edges of $\mathcal{G}$, and every function node has degree 2.] {ex:factor_colouring}

{se:EnsemblesDefinition}

9.2 Ensembles of factor graphs: definitions

We shall be generically interested in understanding the properties of *ensembles* of probability distributions taking the factorized form (9.2). We introduce here a few useful ensembles of factor graphs. In the simple case where every function node has degree 2, factor graphs are in one to one correspondence with usual graphs, and we are just treating random graph ensembles, as first studied by Erdös and Renyi. The case of arbitrary factor graphs is in many cases a simple generalization. From the graph theoretical point of view they can be regarded either as **hyper-graphs** (by associating a vertex to each variable node and an hyper-edge to each function node), or as bipartite graphs (variable and function nodes are both associated to vertices in this case).

For any integer $k \geq 1$, the **random k-factor graph** with M function nodes and N variables nodes is denoted by $\mathbb{G}_N(k, M)$, and is defined as follows. For each function node $a \in \{1 \dots M\}$, the k-uple ∂a is chosen uniformly at random among the $\binom{N}{k}$ k-uples in $\{1 \dots N\}$.

Sometimes, one may encounter variations of this basic distribution. For instance, it can be useful to prevent any two function nodes to have the same neighborhood (in other words, to impose the condition $\partial a \neq \partial b$ for any $a \neq b$). This can be done in a natural way through the ensemble $\mathbb{G}_N(k, \alpha)$ defined as follows. For each of the $\binom{N}{k}$ k-uples of variables nodes, a function node is added to the factor graph independently with probability $\alpha / \binom{N}{k}$, and all of the variables in the k-uple are connected to it. The total number M of function nodes in the graph is a random variable, with expectation $M_{\text{av}} = \alpha N$.

In the following we shall often be interested in large graphs ($N \to \infty$) with a finite density of function nodes. In $\mathbb{G}_N(k, M)$ this means that $M \to \infty$, with the ratio M/N kept fixed. In $\mathbb{G}_N(k, \alpha)$, the large N limit is taken at α fixed. The exercises below suggests that, for some properties, the distinction between the two graph ensembles does not matter in this limit.

Exercise 9.2 Consider a factor graph from the ensemble $\mathbb{G}_N(k, M)$. What is the probability p_{dist} that for any couple of function nodes, the corresponding neighborhoods are distinct? Show that, in the limit $N \to \infty$, $M \to \infty$ with $M/N \equiv \alpha$ and k fixed

$$p_{\text{dist}} = \begin{cases} \Theta(e^{-\frac{1}{2}\alpha^2 N}) & \text{if } k = 1\,, \\ e^{-\alpha^2}[1 + \Theta(N^{-1})] & \text{if } k = 2\,, \\ 1 + \Theta(N^{-k+2}) & \text{if } k \geq 3\,. \end{cases} \qquad (9.13)$$

Exercise 9.3 Consider a random factor graph from the ensemble $\mathbb{G}_N(k,\alpha)$, in the large N limit. Show that the probability of getting a number of function nodes M different from its expectation αN by an 'extensive' number (i.e. a number of order N) is exponentially small. In mathematical terms: there exist a constant $A > 0$ such that, for any $\varepsilon > 0$,

$$\mathbb{P}\left[|M - M_{\rm av}| > N\varepsilon\right] \le e^{-AN\varepsilon^2} \, . \tag{9.14}$$

Consider the distribution of a $\mathbb{G}_N(k,\alpha)$ random graph conditioned on the number of function nodes being $\overline{M}$. Show that this is the same as the distribution of a $\mathbb{G}_N(k,\overline{M})$ random graph conditioned on all the function nodes having distinct neighborhoods.

An important local property of a factor graph is its **degree profile**. Given a graph, we denote by Λ_i (by P_i) the fraction of variable nodes (function nodes) of degree i. Notice that $\Lambda \equiv \{\Lambda_n : n \ge 0\}$ and $P \equiv \{P_n : n \ge 0\}$ are in fact two distributions over the non-negative integers (they are both non-negative and normalized). Moreover, they have non-vanishing weight only on a finite number of degrees (at most N for Λ and M for P). We shall refer to the couple (Λ, P) as to the degree profile of the graph F. A practical representation of the degree profile is provided by the generating functions $\Lambda(x) = \sum_{n\ge 0} \Lambda_n \, x^n$ and $P(x) = \sum_{n \ge 0} P_n \, x^n$. Because of the above remarks, both $\Lambda(x)$ and $P(x)$ are in fact finite polynomials with non-negative coefficients. The average variable node (resp. function node) degree is given by $\sum_{n \ge 0} \Lambda_n \, n = \Lambda'(1)$ (resp. $\sum_{n\ge 0} P_n \, n = P'(1)$)

If the graph is randomly generated, its degree profile is a random variable. For instance, in the random k-factor graph ensemble $\mathbb{G}_N(k,M)$ defined above, the variable node degree Λ depends upon the graph realization: we shall investigate some of its properties below. In contrast, its function node profile $P_n = \mathbb{I}(n=k)$ is deterministic.

It is convenient to consider *ensembles* of factor graphs with a prescribed degree profile. We therefore introduce the ensemble of **degree constrained factor graphs** $\mathbb{D}_N(\Lambda, P)$ by endowing the set of graphs with degree profile (Λ, P) with the uniform probability distribution. Notice that the number M of function nodes is fixed by the relation $MP'(1) = N\Lambda'(1)$. Moreover, the ensemble is non-empty only if $N\Lambda_n$ and MP_n are integers for any $n \ge 0$. Even if these conditions are satisfied, it is not obvious how to construct efficiently a graph in $\mathbb{D}_N(\Lambda, P)$. Since this ensemble plays a crucial role in the theory of sparse graph codes, we postpone this issue to Chap. 11. A special case which is important in this context is that of **random regular graphs** in which the degrees of variable nodes is fixed, as well as the degree of function nodes. In a (k,l) random regular graph, each variable node has degree l and each function node has degree k, corresponding to $\Lambda(x) = x^l$ and $P(x) = x^k$.

{se:EnsemblesProperties}

9.3 Random factor graphs: basic properties

In this Section and the next ones, we derive some simple properties of random factor graphs.

For the sake of simplicity, we shall study here only the ensemble $\mathbb{G}_N(k, M)$ with $k \geq 2$. Generalizations to graphs in $\mathbb{D}_N(\Lambda, P)$ will be mentioned in Sec. 9.5.1 and further developed in Chap. 11. We study the asymptotic limit of large graphs $N \to \infty$ with $M/N = \alpha$ and k fixed.

9.3.1 *Degree profile*

{subsec:DegreeRandom}

The variable node degree profile $\{\Lambda_n : n \geq 0\}$ is a random variable. By linearity of expectation $\mathbb{E}\,\Lambda_n = \mathbb{P}[\mathsf{deg}_i = n]$, where deg_i is the degree of the node i. Let p be the probability that a uniformly chosen k-uple in $\{1, \dots, N\}$ contains i. It is clear that deg_i is a binomial random variable with parameters M and p. Furthermore, since p does not depend upon the site i, it is equal to the probability that a randomly chosen site belongs to a fixed k-uple. In formulae

$$\mathbb{P}[\mathsf{deg}_i = n] = \binom{M}{n} p^n (1-p)^{M-n}\,, \qquad p = \frac{k}{N}\,. \tag{9.15}$$

If we consider the large graph limit, with n fixed, we get

$$\lim_{N\to\infty} \mathbb{P}\left[\mathsf{deg}_i = n\right] = \lim_{N\to\infty} \mathbb{E}\,\Lambda_n = e^{-k\alpha}\,\frac{(k\alpha)^n}{n!}\,. \tag{9.16}$$

The degree of site i is asymptotically a Poisson random variable.

How correlated are the degrees of the variable nodes? By a simple generalization of the above calculation, we can compute the joint probability distribution of deg_i and deg_j, with $i \neq j$. Think of constructing the graph by choosing a k-uple of variable nodes at a time and adding the corresponding function node to the graph. Each node can have one of four possible 'fates': it connects to both nodes i and j (with probability p_2); it connects only to i or only to j (each case has probability p_1); it connects neither to i nor to j (probability $p_0 \equiv 1 - 2p_1 - p_2$). A little thought shows that $p_2 = k(k-1)/N(N-1)$, $p_1 = k(N-k)/N(N-1)$ and

$$\mathbb{P}[\mathsf{deg}_i = n, \mathsf{deg}_j = m] = \sum_{l=0}^{\min(n,m)} \binom{M}{n-l,\, m-l,\, l} p_2^l p_1^{n+m-2l} p_0^{M-n-m+l} \tag{9.17}$$

where l is the number of function nodes which connect both to i and to j and we used the standard notation for multinomial coefficients (see Appendix A).

Once again, it is illuminating to look at the large graphs limit $N \to \infty$ with n and m fixed. It is clear that the $l = 0$ term dominates the sum (9.17). In fact, the multinomial coefficient is of order $\Theta(N^{n+m-l})$ and the various probabilities are of order $p_0 = \Theta(1)$, $p_1 = \Theta(N^{-1})$, $p_2 = \Theta(N^{-2})$. Therefore the l-th term of the sum is of order $\Theta(N^{-l})$. Elementary calculus then shows that

$$\mathbb{P}[\mathsf{deg}_i = n, \mathsf{deg}_j = m] = \mathbb{P}[\mathsf{deg}_i = n]\, \mathbb{P}[\mathsf{deg}_j = m] + \Theta(N^{-1})\,. \tag{9.18}$$

This shows that the nodes' degrees are (asymptotically) pairwise independent Poisson random variables. This fact can be used to show that the degree profile $\{\Lambda_n \,:\, n \geq 0\}$ is, for large graphs, close to its expectation. In fact

$$\begin{aligned}\mathbb{E}\left[(\Lambda_n - \mathbb{E}\Lambda_n)^2\right] &= \frac{1}{N^2}\sum_{i,j=1}^{N}\left\{\mathbb{P}[\mathsf{deg}_i = n, \mathsf{deg}_j = n] - \mathbb{P}[\mathsf{deg}_i = n]\mathbb{P}[\mathsf{deg}_j = n]\right\}\\ &= \Theta(N^{-1})\,,\end{aligned} \tag{9.19}$$

which implies (via Chebyshev inequality) $\mathbb{P}[|\Lambda_n - \mathbb{E}\Lambda_n| \geq \delta\, \mathbb{E}\Lambda_n] = \Theta(N^{-1})$ for any $\delta > 0$.

The pairwise independence expressed in Eq. (9.18) is essentially a consequence of the fact that, given two distinct variable nodes i and j the probability that they are connected to the same function node is of order $\Theta(N^{-1})$. It is easy to see that the same property holds when we consider any finite number of variable nodes. Suppose now that we look at a factor graph from the ensemble $\mathbb{G}_N(k, M)$ conditioned to the function node a being connected to variable nodes $i_1, \dots, i_k$. What is the distribution of the residual degrees $\mathsf{deg}'_{i_1}, \dots, \mathsf{deg}'_{i_k}$ (by residual degree deg'_i, we mean the degree of node i once the function node a has been pruned from the graph)? It is clear that the residual graph is distributed according to the ensemble $\mathbb{G}_N(k, M-1)$. Therefore the residual degrees are (in the large graph limit) independent Poisson random variables with mean $k\alpha$. We can formalize these simple observations as follows.

{PoissonPropo}

Proposition 9.8 *Let $i_1, \dots, i_n \in \{1, \dots, N\}$ be n distinct variable nodes, and G a random graph from $\mathbb{G}_N(k, M)$ conditioned to the neighborhoods of m function nodes $a_1, \dots, a_m$ being $\partial a_1, \dots, \partial a_m$. Denote by deg'_i the degree of variable node i once $a_1, \dots, a_m$ have been pruned from the graph. In the limit of large graphs $N \to \infty$ with $M/N \equiv \alpha$, k, n and m fixed, the residual degrees $\mathsf{deg}'_{i_1}, \dots, \mathsf{deg}'_{i_n}$ converge in distribution to independent Poisson random variables with mean $k\alpha$.*

This property is particularly useful when investigating the local properties of a $\mathbb{G}_N(k, N\alpha)$ random graph. In particular, it suggests that these local properties are close to the ones of the ensemble $\mathbb{D}_N(\Lambda, P)$, where $P(x) = x^k$ and $\Lambda(x) = \exp[k\alpha(x-1)]$.

A remark: in the above discussion we have focused on the probability of finding a node with some constant degree n in the asymptotic limit $N \to \infty$. One may wonder whether, in a typical graph $G \in \mathbb{G}_N(k, M)$ there may exist some variable nodes with exceptionally large degrees. The exercise below shows that this is not the case.

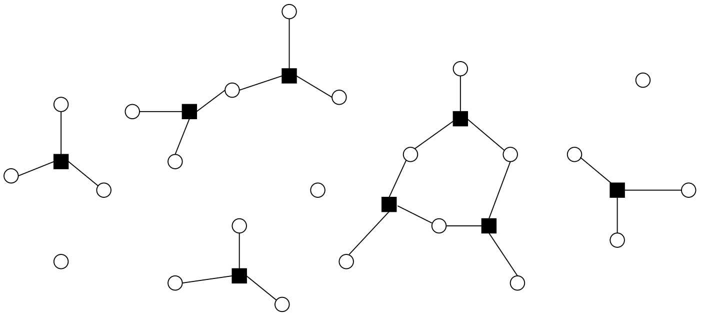


FIG. 9.8. A factor graph from the $\mathbb{G}_N(k, M)$ with $k = 3$, $N = 23$ and $M = 8$. It contains $Z_{\rm isol} = 2$ isolated function nodes, $Z_{\rm coupl} = 1$ isolated couples of function nodes and $Z_{\rm cycle,3} = 1$ cycle of length 3. The remaining 3 variable nodes have degree 0.

{fig:RandomFactor}

Exercise 9.4 We want to investigate the typical properties of the maximum variable node degree $\Delta(G)$ in a random graph G from $\mathbb{G}_N(k, M)$.

(*i*) Let $\overline{n}_{\rm max}$ be the smallest value of $n > k\alpha$ such that $N\mathbb{P}[\mathsf{deg}_i = n] \le 1$. Show that $\Delta(G) \le \overline{n}_{\rm max}$ with probability approaching one in the large graph limit. [Hints: Show that $N\mathbb{P}[\mathsf{deg}_i = \overline{n}_{\rm max} + 1] \to 0$ at large N; Apply the first moment method to Z_l, the number of nodes of degree l.]

(*ii*) Show that the following asymptotic form holds for $\overline{n}_{\rm max}$:

$$\frac{\overline{n}_{\rm max}}{k\alpha e} = \frac{z}{\log(z/\log z)}\left[1 + \Theta\left(\frac{\log\log z}{(\log z)^2}\right)\right] , \qquad (9.20)$$

where $z \equiv (\log N)/(k\alpha e)$.

(*iii*) Let $\underline{n}_{\rm max}$ be the largest value of n such that $N\mathbb{P}[\mathsf{deg}_i = n] \ge 1$. Show that $\Delta(G) \ge \underline{n}_{\rm max}$ with probability approaching one in the large graph limit. [Hints: Show that $N\mathbb{P}[\mathsf{deg}_i = \underline{n}_{\rm max} - 1] \to \infty$ at large N; Apply the second moment method to Z_l.]

(*iv*) What is the asymptotic behavior of $\underline{n}_{\rm max}$? How does it compare to $\overline{n}_{\rm max}$?

9.3.2 *Small subgraphs*

{SmallSection}

The next simplest question one may ask concerning a random graph, is the occurrence in it of a given small subgraph. We shall not give a general treatment of the problem here, but rather work out a few simple examples.

Let's begin by considering a fixed k-uple of variable nodes $i_1, \dots, i_k$ and ask for the probability p that they are connected by a function node in a graph $G \in \mathbb{G}_N(k, M)$. In fact, it is easier to compute the probability that they are *not* connected:

$$1 - p = \left[1 - \binom{N}{k}^{-1}\right]^{M} . \tag{9.21}$$

The quantity in brackets is the probability that a given function node *is not* a neighbor of $i_1, \dots, i_k$. It is raised to the power M because the M function nodes are independent in the model $\mathbb{G}_N(k, M)$. In the large graph limit, we get

$$p = \frac{\alpha\, k!}{N^{k-1}} [1 + \Theta(N^{-1})] . \tag{9.22}$$

This confirms an observation of the previous Section: for any fixed (finite) set of nodes, the probability that a function node connects any two of them vanishes in the large graph limit.

As a first example, let's ask how many isolated function nodes appear in a graph $G \in \mathbb{G}_N(k, M)$. We say that a node is isolated if all the neighboring variable nodes have degree one. Call the number of such function nodes $Z_{\rm isol}$. It is easy to compute the expectation of this quantity

$$\mathbb{E}\, Z_{\rm isol} = M \left[\binom{N}{k}^{-1} \binom{N-k}{k}\right]^{M-1} . \tag{9.23}$$

The factor M is due to the fact that each of the M function nodes can be isolated. Consider one such node a and its neighbors $i_1, \dots, i_k$. The factor in $\binom{N}{k}^{-1}\binom{N-k}{k}$ is the probability that a function node $b \neq a$ is not incident on any of the nodes $i_1, \dots, i_k$. This must be counted for any $b \neq a$, hence the exponent $M - 1$. Once again, things become more transparent in the large graph limit:

$$\mathbb{E}\, Z_{\rm isol} = N\alpha e^{-k^2\alpha} [1 + \Theta(N^{-1})] . \tag{9.24}$$

So there is a non-vanishing 'density' of isolated function nodes. This density approaches 0 at small α (because there are few function nodes at all) and at large α (because function nodes are unlikely to be isolated). A more refined analysis shows that indeed $Z_{\rm isol}$ is tightly concentrated around its expectation: the probability of an order N fluctuation vanishes exponentially as $N \to \infty$.

There is a way of getting the asymptotic behavior (9.24) without going through the exact formula (9.23). We notice that $\mathbb{E}\, Z_{\rm isol}$ is equal to the number of function nodes ($M = N\alpha$) times the probability that the neighboring variable nodes $i_1, \dots, i_k$ have degree 0 in the residual graph. Because of Proposition 9.8, the degrees $\mathsf{deg}'_{i_1}, \dots, \mathsf{deg}'_{i_k}$ are approximatively i.i.d. Poisson random variables with mean $k\alpha$. Therefore the probability for all of them to vanish is close to $(e^{-k\alpha})^k = e^{-k^2\alpha}$.

Of course this last type of argument becomes extremely convenient when considering small structures which involve more than one function node. As a second example, let us compute the number $Z_{\rm isol,2}$ of couples of function nodes which have exactly one variable node in common and are isolated from the rest of the factor graph (for instance in the graph of Fig. 9.8, we have $Z_{\rm isol,2} = 1$). One gets

$$\mathbb{E}\, Z_{\rm isol,2} = \binom{N}{2k-1} \cdot \frac{k}{2}\binom{2k-1}{k} \cdot \left(\frac{\alpha k!}{N^{k-1}}\right)^2 \cdot (e^{-k\alpha})^{2k-1} \left[1 + \Theta\left(\frac{1}{N}\right)\right] \tag{9.25}$$

The first factor counts the ways of choosing the $2k-1$ variable nodes which support the structure. Then we count the number of way of connecting two function nodes to $(2k-1)$ variable nodes in such a way that they have only one variable in common. The third factor is the probability that the two function nodes are indeed present (see Eq. (9.22)). Finally we have to require that the residual graph of all the $(2k-1)$ variable nodes is 0, which gives the factor $(e^{-k\alpha})^{2k-1}$. The above expression is easily rewritten as

$$\mathbb{E}\, Z_{\rm isol,2} = N \cdot \frac{1}{2}(k\alpha)^2\, e^{-k(2k-1)\alpha}\, [1 + \Theta(1/N)] \,. \tag{9.26}$$

With some more work one can prove again that $Z_{\rm isol,2}$ is in fact concentrated around its expected value: a random factor graph contains a finite density of isolated couples of function nodes.

Let us consider, in general, the number of small subgraphs of some definite type. Its most important property is how it scales with N in the large N limit. This is easily found. For instance let's have another look at Eq. (9.25): N enters only in counting the $(2k-1)$-uples of variable nodes which can support the chosen structure, and in the probability of having two function nodes in the desired positions. In general, if we consider a small subgraph with v variable nodes and f function nodes, the number $Z_{v,f}$ of such structures has an expectation which scales as:

$$\mathbb{E}\, Z_{v,f} \sim N^{v-(k-1)f} \,. \tag{9.27}$$

This scaling has important consequences on the nature of small structures which appear in a large random graph. For discussing such structures, it is useful to introduce the notions of 'connected (sub-)graph', of 'tree', of 'path' in a factor graphs exactly in the same way as in usual graphs, identifying both variable nodes and function nodes to vertices (see Chap. 3). We further define a **component** of the factor graph G as a subgraph C which is is connected and isolated, in the sense that there is no path between a node of C and a node of $G\backslash C$

Consider a factor graph with v variable nodes and f function nodes, all of them having degree k.This graph is a tree if and only if $v = (k-1)f + 1$. Call $\star$ $Z_{{\rm tree},v}$ the number of isolated trees over v variable nodes which are contained in a $\mathbb{G}_N(k, M)$ random graph. Because of Eq. (9.27), we have $\mathbb{E}\, Z_{{\rm tree},v} \sim N$: a random graph contains a finite density (when $N \to \infty$) of trees of any finite size. On the other hand, all the subgraphs which are not trees must have $v < (k-1)f + 1$, and Eq. (9.27) shows that their number does not grow with N. In other words, almost all *finite* components of a random factor graph are trees.

Exercise 9.5 Consider the largest component in the graph of Fig. 9.8 (the one with three function nodes), and let $Z_{\rm cycle,3}$ be the number of times it occurs as a component of a $\mathbb{G}_N(k,M)$ random graph. Compute $\mathbb{E}\, Z_{\rm cycle,3}$ in the large graph limit.

Exercise 9.6 A factor graph is said to be **unicyclic** if it contains a unique (up to shifts) closed, non reversing path $\omega_0, \omega_1, \dots, \omega_\ell = \omega_0$ satisfying the condition $\omega_t \neq \omega_s$ for any $t, s \in \{0 \dots \ell - 1\}$, with $t \neq s$.

(i) Show that a factor graph with v variable nodes and f function nodes, all of them having degree k is unicyclic if and only if $v = (k-1)f$.

(ii) Let $Z_{{\rm cycle},v}(N)$ be the number of unicyclic components over v nodes in a $\mathbb{G}_N(k,M)$ random graph. Use Eq. (9.27) to show that $Z_{{\rm cycle},v}$ is finite with high probability in the large graph limit. More precisely, show that $\lim_{n\to\infty}\lim_{N\to\infty}\mathbb{P}_{\mathbb{G}_N}[Z_{{\rm cycle},v} \geq n] = 0$.

{GiantSection}

9.4 Random factor graphs: The giant component

While we have just argued that most components of any fixed (as $N \to \infty$) size of a $\mathbb{G}_N(k,M)$ factor graph are trees, we shall now see that there is much more than just finite size trees in a large $\mathbb{G}_N(k,M)$ factor graph. We always consider the limit $N \to \infty, M \to \infty$ taken at fixed $\alpha = M/N$. It turns out that when α becomes larger than a threshold value, a 'giant component' appears in the graph. This is a connected component containing an extensive (proportional to N) number of variable nodes, with many cycles.

9.4.1 *Nodes in finite trees*

We want to estimate which fraction of a random graph from the $\mathbb{G}_N(k,M)$ ensemble is covered by finite size trees. This fraction is defined as:

$$x_{\rm tr}(\alpha,k) \equiv \lim_{s\to\infty}\lim_{N\to\infty}\frac{1}{N}\,\mathbb{E}\, N_{{\rm trees},s}\, , \tag{9.28}$$

where $N_{{\rm trees},s}$ is the number of sites contained in trees of size not larger than s. In order to compute $\mathbb{E}\, N_{{\rm trees},s}$, we use the number of trees of size equal to s, which we denote by $Z_{{\rm trees},s}$. Using the approach discussed in the previous Section, we get

{eq:NumberOfTrees}

$$\begin{aligned}\mathbb{E}\, N_{{\rm trees},s} &= \sum_{v=0}^{s} v\cdot \mathbb{E}\, Z_{{\rm trees},v} = \\ &= \sum_{v=0}^{s} v \binom{N}{v}\cdot T_k(v)\cdot\left(\frac{\alpha k!}{N^{k-1}}\right)^{\frac{v-1}{k-1}}\cdot (e^{-k\alpha})^v\left[1+\Theta\left(\frac{1}{N}\right)\right] = \\ &= N(\alpha k!)^{-1/(k-1)}\sum_{v=0}^{s}\frac{1}{(v-1)!}T_k(v)\left[(\alpha k!)^{\frac{1}{k-1}}e^{-k\alpha}\right]^v\, ,\end{aligned} \tag{9.29}$$

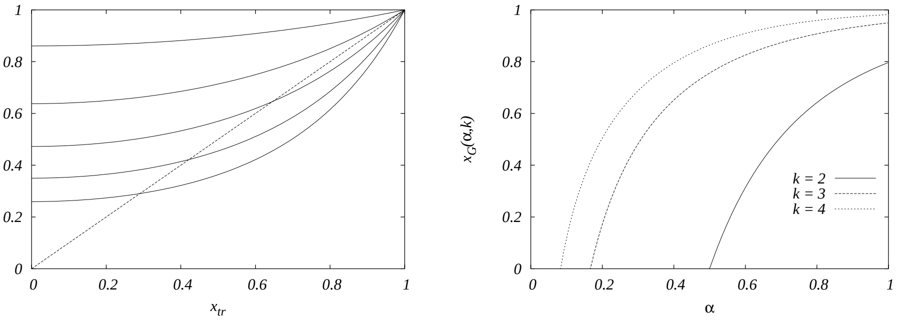


FIG. 9.9. Left: graphical representation of Eq. (9.32) for the fraction of nodes of a $\mathbb{G}_N(k,M)$ random factor graph that belong to finite-size tree components. The curves refer to $k=3$ and (from top to bottom) $\alpha = 0.05$, 0.15, 0.25, 0.35, 0.45. Right: typical size of the giant component. {fig:Giant}

where $T_k(v)$ is the number of trees which can be built out of v distinct variable nodes and $f = (v-1)/(k-1)$ function nodes of degree k. The computation of $T_k(v)$ is a classical piece of enumerative combinatorics which is developed in Sec. 9.4.3 below. The result is

$$T_k(v) = \frac{(v-1)!\, v^{f-1}}{(k-1)!^f f!}\,, \tag{9.30}$$

and the generating function $\widehat{T}_k(z) = \sum_{v=1}^{\infty} T_k(v) z^v/(v-1)!$, which we need in order to compute $\mathbb{E}N_{\rm trees,s}$ from (9.29), is found to satisfy the self consistency equation:

$$\widehat{T}_k(z) = z\, \exp\left\{ \frac{\widehat{T}_k(z)^{k-1}}{(k-1)!} \right\}\,. \tag{9.31}$$

It is a simple exercise to see that, for any $z \geq 0$, this equation has two solutions such that $\widehat{T}_k(z) \geq 0$, the relevant one being the smallest of the two (this is a consequence of the fact that $\widehat{T}_k(z)$ has a regular Taylor expansion around $z=0$). Using this characterization of $\widehat{T}_k(z)$, one can show that $x_{\rm tr}(\alpha,k)$ is the smallest positive solution of the equation ⋆

$$x_{\rm tr} = \exp\left(-k\alpha + k\alpha\, x_{\rm tr}^{k-1}\right)\,. \tag{9.32}$$

This equation is solved graphically in Fig. 9.9, left frame. In the range $\alpha \leq \alpha_{\rm p} \equiv 1/(k(k-1))$, the only non-negative solution is $x_{\rm tr} = 1$: almost all sites belong to finite size trees. When $\alpha > \alpha_{\rm p}$, the solution has $0 < x_{\rm tr} < 1$: the fraction of nodes in finite trees is strictly smaller than one.

9.4.2 *Size of the giant component*

This result is somewhat surprising. For $\alpha > \alpha_{\rm p}$, a finite fraction of variable nodes does not belong to any finite tree. On the other hand, we saw in the previous

Section that finite components with cycles contain a vanishing fraction of nodes. Where are all the other nodes (there are about $N(1-x_{\rm tr})$ of them)? It turns out that, roughly speaking, they belong to a unique connected component, the so-called giant component which is not a tree. One basic result describing this phenomenon is the following.

Theorem 9.9 *Let X_1 be the size of the largest connected component in a $\mathbb{G}_N(k,M)$ random graph with $M = N[\alpha + o_N(1)]$, and $x_{\rm G}(\alpha,k) = 1 - x_{\rm tr}(\alpha,k)$ where $x_{\rm tr}(\alpha,k)$ is defined as the smallest solution of (9.32). Then, for any positive ε,*

$$|X_1 - N x_{\rm G}(\alpha,k)| \le N\varepsilon\,, \tag{9.33}$$

with high probability.

Furthermore, the giant component contains many loops. Let us define the **cyclic number** c of a factor graph containing v vertices and f function nodes of degree k, as $c = v-(k-1)f-1$. Then the cyclic number of the giant component is $c = \Theta(N)$ with high probability.

Exercise 9.7 Convince yourself that there cannot be more than one component of size $\Theta(N)$. Here is a possible route. Consider the event of having two connected components of sizes $\lfloor Ns_1\rfloor$ and $\lfloor Ns_2\rfloor$ for two fixed positive numbers s_1 and s_2 in a $\mathbb{G}_N(k,M)$ random graph with $M = N[\alpha+o_N(1)]$ (with $\alpha \ge s_1+s_2$). In order to estimate the probability of such an event, imagine constructing the $\mathbb{G}_N(k,M)$ graph by adding one function node at a time. Which condition must hold when the number of function nodes is $M-\Delta M$? What can happen to the last ΔM nodes? Now take $\Delta M = \lfloor N^\delta\rfloor$ with $0<\delta<1$.

The appearance of a giant component is sometimes referred to as **percolation on the complete graph** and is one of the simplest instance of a phase transition. We shall now give a simple heuristic argument which predicts correctly the typical size of the giant component. This argument can be seen as the simplest example of the 'cavity method' that we will develop in the next Chapters. We first notice that, by linearity of expectation, $\mathbb{E}\,X_1 = Nx_{\rm G}$, where $x_{\rm G}$ is the probability that a given variable node i belongs to the giant component. In the large graph limit, site i is connected to $l(k-1)$ distinct variable nodes, l being a Poisson random variable of mean $k\alpha$ (see Sec. 9.3.1). The node i belongs to the giant component if any of its $l(k-1)$ neighbors does. If we assume that the $l(k-1)$ neighbors belong to the giant component independently with probability $x_{\rm G}$, then we get

$$x_{\rm G} = \mathbb{E}_l[1-(1-x_{\rm G})^{l(k-1)}]\,. \tag{9.34}$$

where l is Poisson distributed with mean $k\alpha$. Taking the expectation, we get

$$x_{\rm G} = 1 - \exp[-k\alpha + k\alpha(1-x_{\rm G})^{k-1}]\,, \tag{9.35}$$

which coincides with Eq. (9.32) if we set $x_{\rm G} = 1-x_{\rm tr}$.

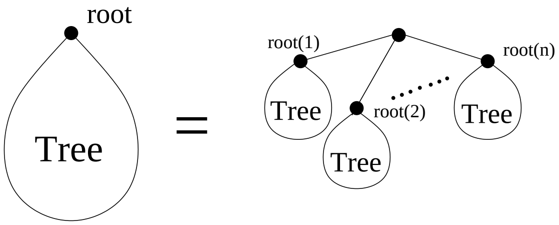


FIG. 9.10. A rooted tree G on $v+1$ vertices can be decomposed into a root and the union of n rooted trees $G_1,\dots,G_n$, respectively on $v_1,\dots,v_n$ vertices. {fig:CayleyRec}

The above argument has several flaws but only one of them is serious. In writing Eq. (9.34), we assumed that the probability that none of l randomly chosen variable nodes belongs to the giant component is just the product of the probabilities that each of them does not. In the present case it is not difficult to fix the problem, but in subsequent Chapters we shall see several examples of the same type of heuristic reasoning where the solution is less straightforward.

9.4.3 *Counting trees*

{se:tkdev}

This paragraph is a technical annex where we compute $T_k(v)$, the number of trees with v variable nodes, when function nodes have degree k. Let us begin by considering the case $k=2$. Notice that, if $k=2$, we can uniquely associate to any factor graph F an ordinary graph G obtained by replacing each function node by an edge joining the neighboring variables (for basic definitions on graphs we refer to Chap. 3). In principle G may contain multiple edges but this does not concern us as long as we stick to F being a tree. Therefore $T_2(v)$ is just the number of ordinary (non-factor) trees on v distinct vertices. Rather than computing $T_2(v)$ we shall compute the number $T_2^*(v)$ of **rooted** trees on v distinct vertices. Recall that a rooted graph is just a couple (G,i_*) where G is a graph and i_* is a distinguished node in G. Of course we have the relation $T_2^*(v)=vT_2(v)$.

Consider now a rooted tree on $v+1$ vertices, and assume that the root has degree n (of course $1\le n\le v$). Erase the root together with its edges and mark the n vertices that were connected to the root. One is left with n rooted trees of sizes $v_1,\dots,v_n$ such that $v_1+\cdots+v_n=v$. This naturally leads to the recursion

$$T_2^*(v+1)=(v+1)\sum_{n=1}^{v}\frac{1}{n!}\sum_{\substack{v_1\dots v_n>0\\ v_1+\cdots+v_n=v}}\binom{v}{v_1,\cdots,v_n}T_2^*(v_1)\cdots T_2^*(v_n)\,, \tag{9.36}$$

which holds for any $v\ge 1$. Together with the initial condition $T_2^*(1)=1$, this relation allows to determine recursively $T_2^*(v)$ for any $v>0$. This recursion is depicted in Fig. 9.10.

The recursion is most easily solved by introducing the generating function $\widehat{T}(z)=\sum_{v>0}T_2^*(v)\,z^v/v!$. Using this definition in Eq. (9.36), we get

$$\widehat{T}(z)=z\,\exp\{\widehat{T}(z)\}\,, \tag{9.37}$$

which is closely related to the definition of Lambert's W function (usually written as $W(z)\exp(W(z)) = z$). One has in fact the identity $\widehat{T}(z) = -W(-z)$. The expansion of $\widehat{T}(z)$ in powers of z can be obtained through Lagrange inversion method (see Exercise below). We get $T_2^*(v) = v^{v-1}$, and therefore $T_2(v) = v^{v-2}$. This result is known as **Cayley formula** and is one of the most famous results in enumerative combinatorics.

Exercise 9.8 Assume that the generating function $A(z) = \sum_{n>0} A_n z^n$ is solution of the equation $z = f(A(z))$, with f an analytic function such that $f(0) = 0$ and $f'(0) = 1$. Use Cauchy formula $A_n = \oint \frac{dz}{2\pi i}\, z^{-n-1}\, A(z)$ to show that

$$A_n = \mathsf{coeff}\left\{f'(x)\,(x/f(x))^{n+1};\, x^{n-1}\right\}\,. \tag{9.38}$$

Use this result, known as 'Lagrange inversion method', to compute the power expansion of $\widehat{T}(z)$ and prove Cayley formula $T_2(v) = v^{v-2}$.

Let us now return to the generic k case. The reasoning is similar to the $k = 2$ case. One finds that the generating function $\widehat{T}_k(z) \equiv \sum_{v>0} T_k^*(v) z^v/v!$ satisfies ⋆ the equation :

$$\widehat{T}_k(z) = z\,\exp\left\{\frac{\widehat{T}_k(z)^{k-1}}{(k-1)!}\right\}\,, \tag{9.39}$$

from which one deduces the number of trees with v variable nodes:

$$T_k^*(v) = \frac{v!\, v^{f-1}}{(k-1)!^f f!}\,. \tag{9.40}$$

In this expression the number of function nodes f is fixed by $v = (k-1)f + 1$.

{LocalSection}

9.5 The local tree-like structure in random graphs

{se:Neighborhood}

9.5.1 *Neighborhood of a node*

There exists a natural notion of distance between variable nodes of a factor graph. Given a path $(\omega_0, \dots, \omega_\ell)$ on the factor graph, we define its length as the number of function nodes in it. Then the **distance** between two variable nodes is defined as the length of the shortest path connecting them (by convention it is set to $+\infty$ when the nodes belong to distinct connected components). We also define the **neighborhood** of radius r of a variable node i, denoted by $\mathsf{B}_{i,r}(F)$ as the subgraph of F including all the variable nodes at distance at most r from i, and all the function nodes connected only to these variable nodes.

What does the neighborhood of a typical node look like in a random graph? It is convenient to step back for a moment from the $\mathbb{G}_N(k, M)$ ensemble and consider a degree-constrained factor graph $F \overset{\rm d}{=} \mathbb{D}_N(\Lambda, P)$. We furthermore define the **edge perspective** degree profiles as $\lambda(x) \equiv \Lambda'(x)/\Lambda'(1)$ and $\rho(x) \equiv P'(x)/P'(1)$. These are polynomials

$$\lambda(x) = \sum_{l=1}^{l_{\rm max}} \lambda_l\, x^{l-1}\,, \qquad\qquad \rho(x) = \sum_{k=1}^{k_{\rm max}} \rho_k\, x^{k-1}\,, \tag{9.41}$$

where λ_l (respectively ρ_k) is the probability that a randomly chosen edge in the graph is adjacent to a variable node (resp. function node) of degree l (degree k). The explicit formulae

$$\lambda_l = \frac{l\Lambda_l}{\sum_{l'} l'\Lambda_{l'}}\,, \qquad\qquad \rho_k = \frac{kP_k}{\sum_{k'} k'P_{k'}}\,, \tag{9.42}$$

are derived by noticing that the graph F contains $nl\Lambda_l$ (resp. mkP_k) edges adjacent to variable nodes of degree l (resp. function nodes of degree k).

Imagine constructing the neighborhoods of a node i of increasing radius r. Given $\mathsf{B}_{i,r}(F)$, let $i_1, \dots, i_L$ be the nodes at distance r from i, and $\mathsf{deg}'_{i_1}, \dots, \mathsf{deg}'_{i_L}$ their degrees in the residual graph[25]. Arguments analogous to the ones leading to Proposition 9.8 imply that $\mathsf{deg}'_{i_1}, \dots, \mathsf{deg}'_{i_L}$ are asymptotically i.i.d. random variables with $\mathsf{deg}'_{i_n} = l_n - 1$, and l_n distributed according to λ_{l_n}. An analogous result holds for function nodes (just invert the roles of variable and function nodes).

This motivates the following definition of an r-generations tree ensemble $\mathbb{T}_r(\Lambda, P)$. If $r = 0$ there is a unique element in the ensemble: a single isolated node, which is attributed the generation number 0. If $r > 0$, first generate a tree from the $\mathbb{T}_{r-1}(\Lambda, P)$ ensemble. Then for each variable-node i of generation $r-1$ draw an independent integer $l_i \geq 1$ distributed according to λ_{l_i} and add to the graph $l_i - 1$ function nodes connected to the variable i (unless $r = 1$, in which case l_i function nodes are added, with l_i distributed according to Λ_{l_i}). Next, for each of the newly added function nodes $\{a\}$, draw an independent integer $k_a \geq 1$ distributed according to ρ_k and add to the graph $k_a - 1$ variable nodes connected to the function a. Finally, the new variable nodes are attributed the generation number r. The case of uniformly chosen random graphs where function nodes have a fixed degree, k, corresponds to the tree-ensemble $\mathbb{T}_r(e^{k\alpha(x-1)}, x^k)$. (In this case, it is easy to checkthat the degrees in the residual graph have a Poisson distribution with mean $k\alpha$, in agreement with proposition 9.8) With a slight abuse of notation, we shall use the shorthand $\mathbb{T}_r(k, \alpha)$ to denote this tree ensemble. ⋆

It is not unexpected that $\mathbb{T}_r(\Lambda, P)$ constitutes a good model for r-neighborhoods in the degree-constrained ensemble. Analogously, $\mathbb{T}_r(k, \alpha)$ is a good model for r-neighborhoods in the $\mathbb{G}_N(k, M)$ ensemble when $M \simeq N\alpha$. This is made more precise below.

[25] By this we mean F minus the subgraph $\mathsf{B}_{i,r}(F)$.

Theorem 9.10 *Let F be a random factor graph in the $\mathbb{D}_N(\Lambda, P)$ ensemble (respectively in the $\mathbb{G}_N(k, M)$ ensemble), i a uniformly random variable node in F, and r a non-negative integer. Then $\mathsf{B}_{i,r}(F)$ converges in distribution to $\mathbb{T}_r(\Lambda, P)$ (resp. to $\mathbb{T}_r(k, \alpha)$) as $N \to \infty$ with Λ, P fixed (α, k fixed).*

In other words, the factor graph F looks locally like a random tree from the ensemble $\mathbb{T}_r(\Lambda, P)$.

9.5.2 *Loops*

We have seen that in the large graph limit, a factor graph $F \overset{\mathrm{d}}{=} \mathbb{G}_N(k, M)$ converges locally to a tree. Furthermore, it has been shown in Sec. 9.3.2 that the number of 'small' cycles in such a graph is only $\Theta(1)$ an $N \to \infty$. It is therefore natural to wonder at which distance from any given node loops start playing a role.

More precisely, let i be a uniformly random site in F. We would like to know what is the typical length of the shortest loop through i. Of course, this question has a trivial answer if $k(k-1)\alpha < 1$, since in this case most of the variable nodes belong to small tree components, cf. Sec. 9.4. We shall hereafter consider $k(k-1)\alpha > 1$.

A heuristic guess of the size of this loop can be obtained as follows. Assume that the neighborhood $\mathsf{B}_{i,r}(F)$ is a tree. Each function node has $k-1$ adjacent variable nodes at the successive generation. Each variable node has a Poisson number adjacent function nodes at the successive generation, with mean $k\alpha$. Therefore the average number of variable nodes at a given generation is $[k(k-1)\alpha]$ times the number at the previous generation. The total number of nodes in $\mathsf{B}_{i,r}(F)$ is about $[k(k-1)\alpha]^r$, and loops will appear when this quantity becomes comparable with the total number of nodes in the system. This yields $[k(k-1)\alpha]^r = \Theta(N)$, or $r = \log N / \log[k(k-1)\alpha]$. This is of course a very crude argument, but it is also a very robust one: one can for instance change N with $N^{1\pm\varepsilon}$ affecting uniquely the prefactor. It turns out that the result is correct, and can be generalized to the $\mathbb{D}_N(\Lambda, P)$ ensemble:

Proposition 9.11 *Let F be a random factor graph in the $\mathbb{D}_N(\Lambda, P)$ ensemble (in the $\mathbb{G}_N(k, M)$ ensemble), i a uniformly chosen random variable node in F, and ℓ_i the length of the shortest loop in F through i. Assume that $c = \lambda'(1)\rho'(1) > 1$ ($c = k(k-1)\alpha > 1$). Then, with high probability,*

$$\ell_i = \frac{\log N}{\log c}[1 + o(1)]\,. \tag{9.43}$$

We shall refer the reader to the literature for the proof, the following exercise gives a slightly more precise, but still heuristic, version of the previous argument.

Exercise 9.9 Assume that the neighborhood $\mathsf{B}_{i,r}(F)$ is a tree and that it includes n 'internal' variables nodes (i.e. nodes whose distance from i is smaller than r), $n_{\rm l}$ 'boundary' variable nodes (whose distance from i is equal to r), and m function nodes. Let F_r be the residual graph, i.e. F minus the subgraph $\mathsf{B}_{i,r}(F)$. It is clear that $F_r \overset{\rm d}{=} \mathbb{G}_{N-n}(k, M-m)$. Show that the probability, p_r, that a function node of F_r connects two of the variable nodes on the boundary of $\mathsf{B}_{i,r}(F)$ is

$$p_r = 1 - \left[(1-q)^k + k\,(1-q)^{k-1}\,q\right]^{M-m}\,, \qquad (9.44)$$

where $q \equiv n_{\rm l}/(N-n)$. As a first estimate of p_r, we can substitute in this expression $n_{\rm l}$, n, m, with their expectations (in the tree ensemble) and call $\overline{p}_r$ the corresponding estimate. Assuming that $r = \rho\frac{\log N}{\log[k(k-1)\alpha]}$, show that

$$\overline{p}_r = 1 - \exp\left\{-\frac{1}{2}k(k-1)\alpha N^{2\rho-1}\right\}\left[1 + O(N^{-2+3\rho})\right]. \qquad (9.45)$$

If $\rho > 1/2$, this indicates that, under the assumption that there is no loop of length $2r$ or smaller through i, there is, with high probability, a loop of length $2r+1$. If, on the other hand, $\rho < 1/2$, it indicates that there is no loop of length $2r+1$ *or smaller* through i. This argument suggests that the length of the shortest loop through i is about $\frac{\log N}{\log[k(k-1)\alpha]}$.

Notes

A nice introduction to factor graphs is the paper (Kschischang, Frey and Loeliger, 2001), see also (Aji and McEliece, 2000). They are also related to graphical models (Jordan, 1998), to Bayesian networks (Pearl, 1988), and to Tanner graphs in coding (Tanner, 1981). Among the alternatives to factor graphs, it is worth recalling 'normal realizations' discussed by Forney in (Forney, 2001).

The proof of the Hammersley-Clifford theorem (initially motivated by the probabilistic modeling of some physical problems) goes back to 1971. A proof, more detailed references and some historical comments can be found in (Clifford, 1990).

The theory of random graphs has been pioneered by Erdös and Renyi (Erdös and Rényi, 1960). The emergence of a giant component in a random graph is a classic result which goes back to their work. Two standard textbooks on random graphs like (Bollobás, 2001) and (Janson, Luczak and Ruciński, 2000) provide in particular a detailed study of the phase transition. Graphs with constrained degree profiles were studied in (Bender and Canfield, 1978). A convenient 'configuration mode' for analyzing them was introduced in (Bollobás, 1980) and allowed for the location of the phase transition in (Molloy and Reed, 1995). Finally, (Wormald, 1999) provides a useful survey (including short loop properties) of degree constrained ensembles.

For general background on hyper-graphs, see (Duchet, 1995). The threshold for the emergence of a giant component in a random hyper-graph with edges of fixed size k (corresponding to the factor graph ensemble $\mathbb{G}_N(k, M)$) is discussed in (Schmidt-Pruzan and Shamir, 1985). The neighborhood of the threshold is analyzed in (Karoński and Luczak, 2002) and references therein.

Ensembles with hyper-edges of different sizes were considered recently in combinatorics (Darling and Norris, 2005), as well as in coding theory (as code ensembles). Our definitions and notations for degree profiles and degree constrained ensembles follows the coding literature (Luby, Mitzenmacher, Shokrollahi, Spielman and Stemann, 1997; Richardson and Urbanke, 2001a).

The local structure of random graphs, and of more complex random objects (in particular random *labeled* graphs) is the object of the theory of *local weak convergence* (Aldous and Steele, 2003). The results in Section 9.5.1 can be phrased in this framework, cf. for instance ???.

10

SATISFIABILITY

{ch:sat}

Because of Cook's theorem, see Chapter 3, satisfiability lies at the heart of computational complexity theory: this fact has motivated an intense research activity on this problem. This Chapter will not be a comprehensive introduction to such a vast topic, but rather present some selected research directions. In particular, we shall pay special attention to the definition and analysis of ensembles of random satisfiability instances. There are various motivations for studying random instances. For testing and improving algorithms that solve satisfiability, it is highly desirable to have an automatic generator of 'hard' instances at hand. As we shall see, properly 'tuned' ensembles provide such a generator. Also, the analysis of ensembles has revealed a rich structure and induced fruitful contacts with other disciplines. We shall come back to satisfiability, using methods inspired from statistical physics, in Chapter **??**.

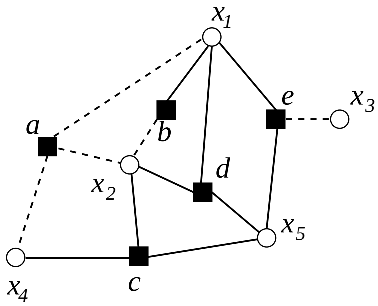
Section 10.1 recalls the definition of satisfiability and introduces some standard terminology. A basic, and widely adopted, strategy for solving decision problems consists in exploring exhaustively the tree of possible assignments of the problem's variables. In Section 10.2 we present a simple implementation of this strategy for solving satisfiability. In Section 10.3 we introduce some important ensembles of random instances. The hardness of satisfiability depends on the maximum clause length. When clauses have length 2, the decision problem is solvable in polynomial time. This is the topic of section 10.4. Finally, in Section 10.5 we discuss the existence of a phase transition for random K-satisfiability with $K \geq 3$, when the density of clauses is varied, and derive some rigorous bounds on the location of this transition.

10.1 The satisfiability problem

{se:sat_intro}

10.1.1 *SAT and UNSAT formulas*

An instance of the satisfiability problem is defined in terms of N Boolean variables, and a set of M constraints between them, where each constraint takes the special form of a clause. A clause is the logical OR of some variables or their negations. Here we shall adopt the following representation: a variable x_i, with $i \in \{1, \dots, N\}$, takes values in $\{0, 1\}$, 1 corresponding to 'true', and 0 to 'false'; the negation of x_i is $\overline{x}_i \equiv 1 - x_i$. A variable or its negation is called a literal, and we shall denote it z_i , with $i \in \{1, \dots, N\}$ (therefore z_i denotes any of x_i, $\overline{x}_i$). A clause a, with $a \in \{1, ..., M\}$, involving K_a variables is a constraint which forbids exactly one among the 2^{K_a} possible assignments to these K_a variables. It is written as the logical OR (denoted by $\vee$) function of some variables or their



{fig:facgraphsatex} FIG. 10.1. Factor graph representation of the formula $(\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_4) \wedge (x_1 \vee \overline{x}_2) \wedge (x_2 \vee x_4 \vee x_5) \wedge (x_1 \vee x_2 \vee \overline{x}_5) \wedge (x_1 \vee \overline{x}_3 \vee x_5)$.

negations. For instance the clause $x_2 \vee \overline{x}_{12} \vee x_{37} \vee \overline{x}_{41}$ is satisfied by all the variables' assignments except those where $x_2 = 0, x_{12} = 1, x_{37} = 0, x_{41} = 1$. When it is not satisfied, a clause is said to be violated.

We denote by ∂a the subset $\{i_1^a, \dots, i_{K_a}^a\} \subset \{1, \dots, N\}$ containing the indices of the $K_a = |\partial a|$ variables involved in clause a. Then clause a is written as $C_a = z_{i_1^a} \vee z_{i_2^a} \vee \cdots \vee z_{i_{K_a}^a}$. An instance of the satisfiability problem can be summarized as the logical formula (called a **conjunctive normal form (CNF)**):

$$F = C_1 \wedge C_2 \wedge \cdots \wedge C_M \, . \tag{10.1}$$

As we have seen in Chapter 9, Example 9.7, there exists [26] a simple and natural representation of a satisfiability formula as a factor graph associated with the indicator function $\mathbb{I}(\underline{x} \text{ satisfies } F)$. Actually, it is often useful to use a slightly more elaborate factor graph using two types of edges: A full edge is drawn between a variable vertex i and a clause vertex a whenever x_i appears in a, and a dashed edge is drawn whenever $\overline{x}_i$ appears in a. In this way there is a one to one correspondence between a CNF formula and its graph. An example is shown in Fig. 10.1.

Given the formula F, the question is whether there exists an assignment of the variables x_i to $\{0, 1\}$ (among the 2^N possible assignments), such that the formula F is true. An algorithm solving the satisfiability problem must be able, given a formula F, to either answer 'YES' (the formula is then said to be **SAT**), and provide such an assignment, called a **SAT-assignment**, or to answer 'NO', in which case the formula is called **UNSAT**. The restriction of the satisfiability problem obtained by requiring that all the clauses in F have the same length $K_a = K$, is called the ***K*-satisfiability** (or K-SAT) problem.

As usual, an optimization problem is naturally associated to the decision version of satisfiability: Given a formula F, one is asked to find an assignment

[26] It may happen that there does not exist any assignment satisfying F, so that one cannot use this indicator function to build a probability measure. However one can still characterize the local structure of $\mathbb{I}(\underline{x} \text{ satisfies } F)$ by the factor graph

which violates the smallest number of clauses. This is called the **MAX-SAT** problem.

{ex:2-satex1}

Exercise 10.1 Consider the 2-SAT instance defined by the formula $F_1 = (x_1 \vee \overline{x}_2) \wedge (x_2 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee x_4) \wedge (x_4 \vee \overline{x}_1) \wedge (\overline{x}_3 \vee \overline{x}_4) \wedge (\overline{x}_2 \vee x_3)$. Show that this formula is SAT and write a SAT-assignment. [Hint: assign for instance $x_1 = 1$; the clause $x_4 \vee \overline{x}_1$ is then reduced to x_4, this is a **unit clause** which fixes $x_4 = 1$; the chain of 'unit clause propagation' either leads to a SAT assignment, or to a contradiction.]

{ex:2-satex2}

Exercise 10.2 Consider the 2-SAT formula $F_2 = (x_1 \vee \overline{x}_2) \wedge (x_2 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee x_4) \wedge (x_4 \vee \overline{x}_1) \wedge (\overline{x}_3 \vee \overline{x}_4) \wedge (\overline{x}_2 \vee \overline{x}_3)$. Show that this formula is UNSAT by using the same method as in the previous Exercise.

{ex:3-satex1}

Exercise 10.3 Consider the 3-SAT formula $F_3 = (x_1 \vee x_2 \vee \overline{x}_3) \wedge (x_1 \vee x_3 \vee \overline{x}_4) \wedge (x_2 \vee x_3 \vee x_4) \wedge (\overline{x}_1 \vee x_2 \vee \overline{x}_4) \wedge (x_1 \vee \overline{x}_2 \vee x_4) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee x_4) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4) \wedge (x_2 \vee \overline{x}_3 \vee x_4) \wedge (\overline{x}_1 \vee x_3 \vee \overline{x}_4)$. Show that it is UNSAT. [Hint: try to generalize the previous method by using a decision tree, cf. Sec. 10.2.2 below, or list the 16 possible assignments and cross out which one is eliminated by each clause.]

As we already mentioned, satisfiability was the first problem to be proved NP-complete. The restriction defined by requiring $K_a \leq 2$ for each clause a, is polynomial. However, if one relaxes this condition to $K_a \leq K$, with $K = 3$ or more, the resulting problem is NP-complete. For instance 3-SAT is NP-complete while 2-SAT is polynomial. It is intuitively clear that MAX-SAT is "at least as hard" as SAT: an instance is SAT if and only if the minimum number of violated clauses (that is the output of MAX-SAT) vanishes. It is less obvious that MAX-SAT can be "much harder" than SAT. For instance, MAX-2-SAT is NP-hard, while as said above, its decision counterpart is in P.

The study of applications is not the aim of this book, but one should keep in mind that satisfiability is related to a myriad of other problems, some of which have enormous practical relevance. It is a problem of direct importance to the fields of mathematical logic, computing theory and artificial intelligence. Applications range from integrated circuit design (modeling, placement, routing, testing,...) to computer architecture design (compiler optimization, scheduling and task partitioning,...) and to computer graphics, image processing etc...

10.2 Algorithms

{se:sat_algo}

10.2.1 *A simple case: 2-SAT*

{se:2satalgo}

The reader who worked out Exercises 10.1 and 10.2 has already a feeling that 2-SAT is an easy problem. The main tool for solving it is the so-called **unit clause propagation (UCP)** procedure. If we start from a 2-clause $C = z_1 \vee z_2$ and fix the literal z_1, two things may happen:

- If we fix $z_1 = 1$ the clause is satisfied and disappears from the formula
- If we fix $z_1 = 0$ the clause is transformed into the unit clause z_2 which implies that $z_2 = 1$.

Given a 2-SAT formula, one can start from a variable x_i, $i \in \{1, \dots, N\}$ and fix, for instance $x_i = 0$. Then apply the reduction rule described above to all the clauses in which x_i or $\overline{x}_i$ appears. Finally, fix recursively in the same way all the literals which appear in unit clauses. This procedure may halt for one of the following reasons: (*i*) the formula does not contain any unit clause; (*ii*) the formula contains the unit clause z_j together with its negation $\overline{z}_j$.

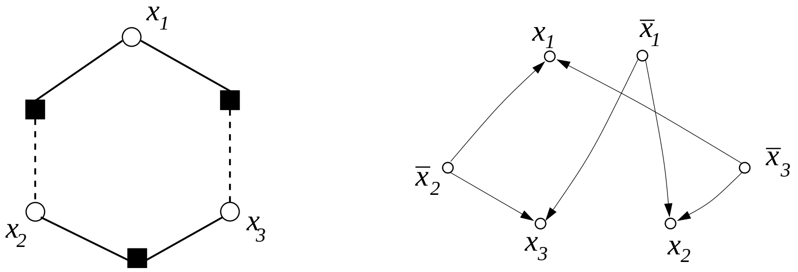
In the first case, a partial SAT assignment (i.e. an assignment of a subset of the variables such that no clause is violated) has been found. We will prove below that such a partial assignment can be extended to a complete SAT assignment if and only if the formula is SAT. One therefore repeats the procedure by fixing a not-yet-assigned variable x_j.

In the second case, the partial assignment cannot be extended to a SAT assignment. One proceeds by changing the initial choice and setting $x_i = 1$. Once again, if the procedure stops because of reason (*i*), then the formula can be effectively reduced and the already-fixed variables do not need to be reconsidered in the following. If on the other hand, also the choice $x_i = 1$ leads to a contradiction $\star$ (i.e. the procedure stops because of (*ii*)), then it is immediate to show that the formula is necessarily UNSAT.

It is clear that the algorithm defined in this way is very efficient. Its complexity can be measured by the number of variable-fixing operations that it involves. Since each variable is considered at most twice, this number is at most $2N$.

For proving the correctness of this algorithm, we still have to show the following fact: if the formula is SAT and UCP stops because of reason (*i*), then the resulting partial assignment can be extended to a global SAT assignment (The implication in the reverse direction is obvious). The key point is that the residual formula is formed by a subset $\mathcal{R}$ of the variables (the ones which have not yet been fixed) together with *a subset of the original clauses* (those which involve uniquely variables in $\mathcal{R}$). If a SAT assignment exists, its restriction to $\mathcal{R}$ satisfies the residual formula and constitutes an extension of the partial assignment generated by UCP.

Exercise 10.4 Write a code for solving 2-SAT using the algorithm described above.



{fig:DirectedGraph} FIG. 10.2. Factor graph representation of the 2SAT formula $F = (x_1 \vee \overline{x}_2) \wedge (x_1 \vee \overline{x}_3) \wedge (x_2 \vee x_3)$ (left) and corresponding directed graph $\mathcal{D}(F)$ (right).

{ex:2sat-directed}

Exercise 10.5 A nice way of understanding UCP, and why it is so effective for 2-SAT, consists in associating to the formula F a directed graph $\mathcal{D}(F)$ (not to be confused with the factor graph!) as follows. Associate a vertex to each of the $2N$ literals (for instance we have one vertex for x_1 and one vertex for $\overline{x}_1$). Whenever a clause like e.g. $\overline{x}_1 \vee x_2$ appears in the formula, we have two implications: if $x_1 = 1$ then $x_2 = 1$; if $x_2 = 0$ then $x_1 = 0$. Represent them graphically by drawing an oriented edge from the vertex x_1 toward x_2, and an oriented edge from $\overline{x}_2$ to $\overline{x}_1$. Prove that the F is UNSAT if and only if there exists a variable index $i \in \{1, \dots, N\}$ such that: $\mathcal{D}(F)$ contains a directed path from x_i to $\overline{x}_i$, *and* a directed path from $\overline{x}_i$ to x_i. [Hint: Consider the UCP procedure described above and rephrase it in terms of the directed graph $\mathcal{D}(F)$. Show that it can be regarded as an algorithm for finding a pair of paths from x_i to $\overline{x}_i$ and vice-versa in $\mathcal{D}(F)$.]

Let us finally notice that the procedure described above does not give any clue about an efficient solution of MAX-2SAT, apart from determining whether the minimum number of violated clauses vanishes or not. As already mentioned MAX-2SAT is NP-hard.

10.2.2 *A general complete algorithm*

{se:dpll}

As soon as we allow an unbounded number of clauses of length 3 or larger, satisfiability becomes an NP-complete problem. Exercise 10.3 shows how the UCP strategy fails: fixing a variable in a 3-clause may leave a 2-clause. As a consequence, UCP may halt without contradictions and produce a residual formula containing clauses which were not present in the original formula. Therefore, it can be that the partial assignment produced by UCP cannot be extended to a global SAT assignment even if the original formula is SAT. Once a contradiction is found, it may be necessary to change any of the choices made so far in order to find a SAT assignment (as opposite to 2SAT where only the last choice had to be changed). The exploration of all such possibilities is most conveniently described through a decision tree. Each time that a contradiction is found, the search algorithm backtracks to the last choice for which both possibilities were not explored.

The most widely used **complete algorithms** (i.e. algorithms which are able to either find a satisfying assignment, or prove that there is no such assignment) rely on this idea. They are known under the name **DPLL**, from the initials of their inventors, Davis, Putnam, Logemann and Loveland. The basic recursive process is best explained on an example, as in Fig. 10.3. Its structure can be summarized in few lines:

```
DPLL

Input: A CNF formula $F$.

Output: A SAT assignment, or a message '$F$ is UNSAT'.

  1. Initialize $n = 0$, and $G(0) = F$.
  2. If $G(n)$ contains no clauses, return the assignment $x_i = 0$ for each $i$ present in $G(n)$ and stop.
  3. If $G$ contains the empty clause return the message '$F$ is UNSAT' and stop.
  4. Select a variable index $i$ among those which have not yet been fixed.
  5. Let $G(n+1)$ be the formula obtained from $G(n)$ by fixing $x_i = 1$.
  6. Set $n \leftarrow n+1$ and go to 2.
  7. Set $n \leftarrow n-1$. (No SAT assignment was found such that $x_i = 1$.)
  8. Let $G(n+1)$ be the formula obtained from $G(n)$ by fixing $x_i = 0$.
  9. Set $n \leftarrow n+1$ and go to 2.
```

The algorithm keeps track of the current satisfiability formula as $G(n)$. As shown in Fig. 10.3 the algorithm state can be represented as a node in the decision tree. The index n corresponds to the current depth in this tree.

It is understood that, whenever a variable is fixed (instructions 5 and 8 above), all the clauses in which that variable appears are reduced. More precisely, suppose that the literal x_i appears in a clause: the clause is eliminated if one fixes $x_i = 1$, and it is shortened (by elimination of x_i) if one fixes $x_i = 0$. Vice-versa, if the literal $\overline{x}_i$ is present, the clause is eliminated if one fixes $x_i = 0$ and shortened in the opposite case.

In the above pseudo-code, we did not specify how to select the next variable to be fixed in step 4. Various versions of the DPLL algorithm differ in the order in which the variables are taken in consideration and the branching process is performed. Unit clause propagation can be rephrased in the present setting as the following rule: whenever the formula $G(n)$ contains clauses of length 1, x_i must be chosen among the variables appearing in such clauses. In such a case, no real branching takes place. For instance, if the literal x_i appears in a unit clause, setting $x_i = 0$ immediately leads to an empty clause and therefore to a stop of the process: one is obviously forced to set $x_i = 1$.

Apart from the case of unit clauses, deciding on which variable the next branching will be done is an art, and can result in very different performances. For instance, it is a good idea to branch on a variable which appears in many clauses, but other criteria, like the number of UCP that a branching will generate, can also be used. It is customary to characterize the performances of this class of algorithms by the number of branching points it generates. This does not count the actual number of operations executed, which may depend on the heuristic. However, for any reasonable heuristics, the actual number of operations is within a polynomial factor (in the instance size) from the number of branchings and such a factor does not affect the leading exponential behavior.

Whenever the DPLL procedure does not return a SAT assignment, the formula is UNSAT: a representation of the explored search tree provides a proof. This is sometimes also called an UNSAT **certificate**. Notice that the length of an UNSAT certificate is (in general) larger than polynomial in the input size. This is at variance with a SAT certificate, which is provided, for instance, by a particular SAT assignment.

Exercise 10.6 Resolution and DPLL.

(*i*) A powerful approach to proving that a formula is UNSAT relies on the idea of the **resolution proof**. Imagine that F contains two clauses: $x_j \vee A$, and $\overline{x}_j \vee B$, where A and B are subclauses. Show that these two clauses automatically imply the **resolvent on** x_j, that is the clause $A \vee B$.

(*ii*) A resolution proof is constructed by adding resolvent clauses to F. Show that, if this process produces an empty clause, then the original formula is necessarily UNSAT. An UNSAT certificate is simply given by the sequence of resolvents leading to the empty clause.

(*iii*) Although this may look different from DPLL, any DPLL tree is an example of resolution proof. To see this proceed as follows. Label each 'UNSAT' leave of the DPLL tree by the resolution of a pair of clauses of the original formula which are shown to be contradictory on this branch (e.g. the leftmost such leaf in Fig. 10.3 corresponds to the pair of initial clauses $x_1 \vee x_2 \vee \overline{x}_3$ and $x_1 \vee x_2 \vee x_3$, so that it can be labeled by the resolvent of these two clauses on x_3, namely $x_1 \vee x_2$). Show that each branching point of the DPLL tree can be labeled by a clause which is a resolvent of the two clauses labeling its children, and that this process, when carried on an UNSAT formula, produces a root (the top node of the tree) which is an empty clause.

10.2.3 *Incomplete search*

{se:Schoning}

As we have seen above, proving that a formula is SAT is much easier than proving that it is UNSAT: one 'just' needs to exhibit an assignment that satisfies all the clauses. One can therefore relax the initial objective, and look for an algorithm that only tries to deal with the first task. This is often referred to

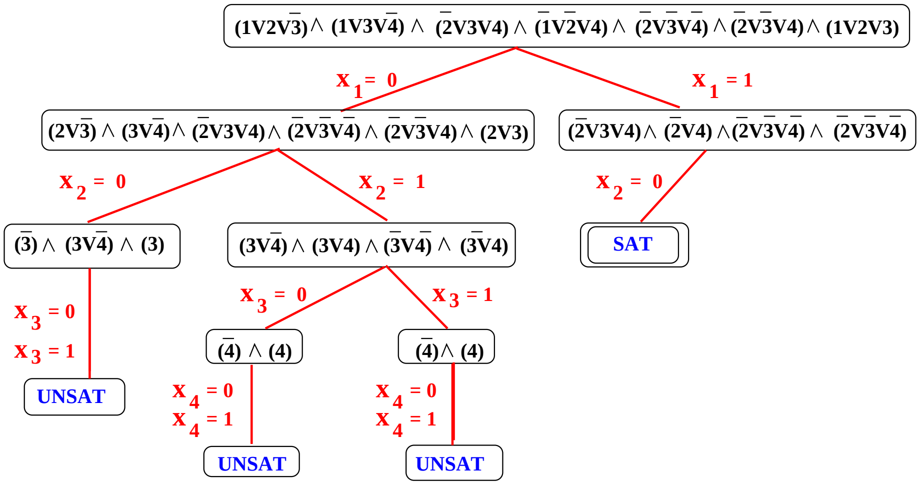

(1V2V3̄)∧ (1V3V4̄) ∧ (2̄V3V4) ∧ (1̄V2V4) ∧ (2̄V3̄V4̄) ∧ (2̄V3̄V4) ∧ (1V2V3)
x1 = 0
x1 = 1
(2V3̄) ∧ (3V4̄) ∧ (2̄V3V4) ∧ (2̄V3̄V4̄) ∧ (2̄V3̄V4) ∧ (2V3)
(2̄V3V4) ∧ (2V4) ∧ (2̄V3̄V4̄) ∧ (2̄V3̄V4)
x2 = 0
x2 = 1
x2 = 0
(3̄) ∧ (3V4̄) ∧ (3)
(3V4̄) ∧ (3V4) ∧ (3̄V4̄) ∧ (3̄V4)
SAT
x3 = 0
x3 = 1
x3 = 0
x3 = 1
UNSAT
(4̄) ∧ (4)
(4̄) ∧ (4)
x4 = 0
x4 = 1
x4 = 0
x4 = 1
UNSAT
UNSAT

FIG. 10.3. A sketch of the DPLL algorithm, acting on the formula $(x_1 \vee x_2 \vee \overline{x}_3) \wedge (x_1 \vee x_3 \vee \overline{x}_4) \wedge (\overline{x}_2 \vee x_3 \vee x_4) \wedge (\overline{x}_1 \vee x_2 \vee x_4) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee x_4) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee \overline{x}_4)$. In order to get a more readable figure, the notation has been simplified: a clause like $(\overline{x}_1 \vee x_2 \vee x_4)$ is denoted here as $(\overline{1}\; 2\; 4)$. One fixes a first variable, here $x_1 = 0$. The problem is then reduced: clauses containing x_1 are eliminated, and clauses containing $\overline{x}_1$ are shortened by eliminating the literal $\overline{x}_1$. Then one proceeds by fixing a second variable, etc... At each step, if a unit clause is present, the next variable to be fixed is chosen among the those appearing in unit clauses. This corresponds to the unit clause propagation (UCP) rule. When the algorithm finds a contradiction (two unit clauses fixing a variable simultaneously to 0 and to 1), it backtracks to the last not-yet-completed branching point and explores another choice for the corresponding variable. In this case for instance, the algorithm first fixes $x_1 = 0$, then it fixes $x_2 = 0$, which implies through UCP that $x_3 = 0$ and $x_3 = 1$. This is a contradiction, and therefore the algorithm backtracks to the last choice, which was $x_2 = 0$, and tries instead the other choice: $x_2 = 1$, etc... Here we have taken the naive rule of branching in the fixed order given by the clause index.

{fig:DPL_example}

as an **incomplete search** algorithm. Such an algorithm can either return a satisfying assignment or just say 'I do not know' whenever it is unable to find one (or to prove that the formula is UNSAT).

A simple incomplete algorithm, due to Schöning, is based on the simple random walk routine:

```
Walk( $F$ )

Input: A CNF formula $F$.

Output: A SAT assignment, or a message 'I do not know'.
```

```
1. Assign to each variable a random value $0$ or $1$ with probability
   $1/2$.
2. Repeat $3N$ times:
   3. If the current assignment satisfies $F$ return it and stop.
   4. Choose an unsatisfied clause uniformly at random.
   5. Choose a variable $x_i$ uniformly at random among the ones belonging
      to this clause.
   6. Flip it (i.e. set it to $0$ if it was $1$ and vice-versa).
```

For this algorithm one can obtain a guarantee of performance:

Proposition 10.1 *Denote by $p(F)$ the probability that this routine, when executed on a formula F, returns a satisfying assignment. If F is SAT, then $p(F) \geq p_N$ where*

$$p_N = \frac{2}{3}\left(\frac{K}{2(K-1)}\right)^N . \tag{10.2}$$

One can therefore run the routine many times (with independent random numbers each time) in order to increase the probability of finding a solution. Suppose that the formula is SAT. If the routine is run $20/p_N$ times, the probability of not finding any solution is $(1-p_N)^{20/p_N} \leq e^{-20}$. While this is of course not a proof of unsatisfiability, it is very close to it. In general, the time required for this procedure to reduce the error probability below any fixed ε grows as

$$\tau_N \doteq \left(\frac{2(K-1)}{K}\right)^N . \tag{10.3}$$

This simple randomized algorithm achieves an exponential improvement over the naive exhaustive search which takes about 2^N operations.

Proof: Let us now prove the lower bound (10.2) on the probability of finding a satisfying assignment during a single run of the routine `Walk(`$\cdot$`)`. Since, by assumption, F is SAT, we can consider a particular SAT assignment, let us say $\underline{x}_*$. Let $\underline{x}_t$ be the assignment produced by `Walk(`$\cdot$`)` after t spin flips, and d_t be the Hamming distance between $\underline{x}_*$ and $\underline{x}_t$. Obviously, at time 0 we have

$$\mathbb{P}\{d_0 = d\} = \frac{1}{2^N}\binom{N}{d} . \tag{10.4}$$

Since $\underline{x}_*$ satisfies F, each clause is satisfied by at least one variable as assigned in $\underline{x}_*$. Mark *exactly* one such variable per clause. Each time `Walk(`$\cdot$`)` chooses a violated clause, it flips a marked variable with probability $1/K$, reducing the Hamming distance by one. Of course, the Hamming distance can decrease also when another variable is flipped (if more than one variable satisfies that clauses in $\underline{x}_*$). In order to get a bound we introduce an auxiliary integer variable $\hat{d}_t$ which decreases by one each time a marked variable is selected, and increases by one (the maximum possible increase in Hamming distance due to a single flip) otherwise. If we choose the initial condition $\hat{d}_0 = d_0$, it follows from the previous observations that $d_t \le \hat{d}_t$ for any $t \ge 0$. We can therefore upper bound the probability that `Walk( · )` finds a solution by the probability that $\hat{d}_t = 0$ for some $0 \le t \le 3N$. But the random process $\hat{d}_t = 0$ is simply a biased random walk on the half-line with initial condition (10.4): at each time step it moves to the right with probability $1/K$ and to the right with probability $1 - 1/K$. The probability of hitting the origin can then be estimated as in Eq. (10.2), as shown in the following exercise.

Exercise 10.7 Analysis of the biased random walk $\hat{d}_t$.

(i) Show that the probability for $\hat{d}_t$ to start at position d at $t = 0$ and be at the origin at time t is

$$\mathbb{P}\big\{\hat{d}_0 = d\,;\ \hat{d}_t = 0\big\} = \frac{1}{2^N}\binom{N}{d}\frac{1}{K^t}\binom{t}{\frac{t-d}{2}}(K-1)^{\frac{t-d}{2}} \tag{10.5}$$

for $t + d$ even, and vanishes otherwise.

(ii) Use Stirling's formula to derive an approximation of this probability to the leading exponential order: $\mathbb{P}\big\{\hat{d}_0 = d\,;\ \hat{d}_t = 0\big\} \doteq \exp\{-N\Psi(\theta, \delta)\}$, where $\theta = t/N$ and $\delta = d/N$.

(iii) Minimize $\Psi(\theta, \delta)$ with respect to $\theta \in [0, 3]$ and $\delta \in [0, 1]$, and show that the minimum value is $\Psi_* = \log[2(K - 1)/K]$. Argue that $p_N \doteq \exp\{-N\Psi_*\}$ to the leading exponential order.

□

Notice that the above algorithm applies a very noisy strategy. While 'focusing' on unsatisfied clauses, it makes essentially random steps. The opposite philosophy would be that of making greedy steps. An example of 'greedy' step is the following: flip a variable which will lead to the largest positive increase in the number of satisfied clause.

There exist several refinements of the simple random walk algorithm. One of the greatest improvement consists in applying a mixed strategy: With probability p, pick an unsatisfied clause, and flip a randomly chosen variable in this clause (as in `Walk`); With probability $1 - p$, perform a 'greedy' step as defined above.

This strategy works reasonably well if p is properly optimized. The greedy steps drive the assignment toward 'quasi-solutions', while the noise term allows to escape from local minima.

{se:sat_random_intro}

10.3 Random K-satisfiability ensembles

Satisfiability is NP-complete. One thus expects a complete algorithm to take exponential time in the worst case. However empirical studies have shown that many formulas are very easy to solve. A natural research direction is therefore to characterize ensembles of problems which are easy, separating them from those that are hard. Such ensembles can be defined by introducing a probability measure over the space of instances.

One of the most interesting family of ensembles is **random K-SAT**. An instance of random K-SAT contains only clauses of length K. The ensemble is further characterized by the number of variables N, and the number of clauses M, and denoted as $\mathsf{SAT}_N(K, M)$. A formula in $\mathsf{SAT}_N(K, M)$ is generated by selecting M clauses of size K uniformly at random among the $\binom{N}{K}2^K$ such clauses. Notice that the factor graph associated to a random K-SAT formula from the $\mathsf{SAT}_N(K, M)$ ensemble is in fact a random $\mathbb{G}_N(K, M)$ factor graph.

It turns out that a crucial parameter characterizing the random K-SAT ensemble is the **clause density** $\alpha \equiv M/N$. We shall define the 'thermodynamic' limit as $M \to \infty$, $N \to \infty$, with fixed density α. In this limit, several important properties of random formulas concentrate in probability around their typical values.

As in the case of random graphs, it is sometimes useful to consider slight variants of the above definition. One such variant is the $\mathsf{SAT}_N(K, \alpha)$ ensemble. A random instance from this ensemble is generated by including in the formula each of the $\binom{N}{K}2^K$ possible clauses independently with probability $\alpha N 2^{-K}/\binom{N}{K}$. Once again, the corresponding factor graph will be distributed according to the $\mathbb{G}_N(K, \alpha)$ ensemble introduced in Chapter 9. For many properties, differences between such variants vanish in the thermodynamic limit (this is analogous to the equivalence of different factor graph ensembles).

10.3.1 *Numerical experiments*

Using the DPLL algorithm, one can investigate the properties of typical instances of the random K-SAT ensemble $\mathsf{SAT}_N(K, M)$. Figure 10.4 shows the probability $P_N(K, \alpha)$ that a randomly generated formula is satisfiable, for $K = 2$ and $K = 3$. For fixed K and N, this is a decreasing function of α, which goes to 1 in the $\alpha \to 0$ limit and goes to 0 in the $\alpha \to \infty$ limit. One interesting feature in these simulations is the fact that the crossover from high to low probability becomes sharper and sharper when N increases. This numerical result points at the existence of a phase transition at a finite value $\alpha_c(K)$: for $\alpha < \alpha_c(K)$ ($\alpha > \alpha_c(K)$) a random K-SAT formula is SAT (respectively, UNSAT) with probability approaching 1 as $N \to \infty$.

The conjectured phase transition in random satisfiability problems with $K \geq 3$ has drawn considerable attention. One important reason comes from the study of the computational effort needed to solve the problem. Figure 10.5 shows the typical number of branching nodes in the DPLL tree required to solve a typical random 3-SAT formula. One may notice two important features: For a given value of the number of variables N, the computational effort has a peak in the region of clause density where a phase transition seems to occur (compare to Fig. 10.4). In this region it also increases rapidly with N. Looking carefully at the datas one can distinguish qualitatively three different regions: at low α the solution is 'easily' found and the computer time grows polynomially; at intermediate α, in

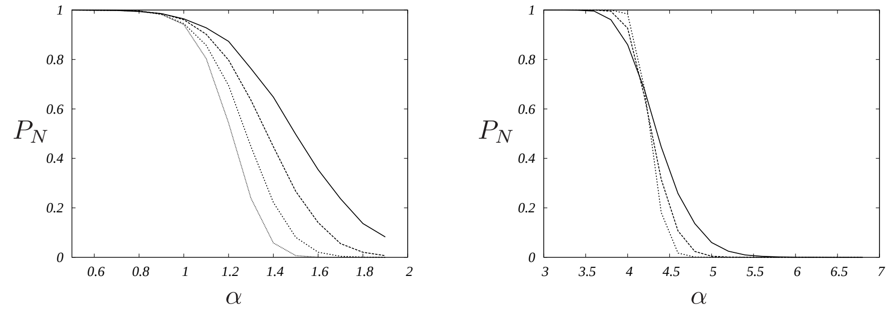


FIG. 10.4. Probability that a formula generated from the random K-SAT ensemble is satisfied, plotted versus the clause density α. Left: $K = 2$, right: $K = 3$. The curves have been generated using a DPLL algorithm. Each point is the result of averaging over 10^4 random formulas. The curves for $K = 2$ correspond to formulas of size $N = 50$, 100, 200, 400 (from right to left). In the case $K = 3$ the curves correspond to $N = 50$ (full line), $N = 100$ (dashed), $N = 200$ (dotted). The transition between satisfiable and unsatisfiable formulas becomes sharper as N increases.

{fig:alphac_SAT_num}

the phase transition region, the problem becomes typically very hard and the computer time grows exponentially. At larger α, in the region where a random formula is almost always UNSAT, the problem becomes easier, although the size of the DPLL tree still grows exponentially with N.

The hypothetical phase transition region is therefore the one where the hardest instances of random 3-SAT are located. This makes such a region particularly interesting, both from the point of view of computational complexity and from that of statistical physics.

{se:2sat}

10.4 Random 2-SAT

From the point of view of computational complexity, 2-SAT is polynomial while K-SAT is NP-complete for $K \geq 3$. It turns out that random 2-SAT is also much simpler to analyze than the other cases. One important reason is the existence of the polynomial decision algorithm described in Sec. 10.2.1 (see in particular Exercise 10.5). This can be analyzed in details using the representation of a 2-SAT formula as a directed graph whose vertices are associated to literals. One can then use the mathematical theory of random directed graphs. In particular, the existence of a phase transition at critical clause density $\alpha_{\rm c}(2) = 1$ can be established.

Theorem 10.2 *Let $P_N(K = 2, \alpha)$ the probability for a $\mathsf{SAT}_N(K = 2, M)$ random formula to be SAT. Then*

$$\lim_{N\to\infty} P_N(K = 2, \alpha) = \begin{cases} 1 & \textit{if } \alpha < 1 \;, \\ 0 & \textit{if } \alpha > 1 \;. \end{cases} \tag{10.6}$$

{thm:2sat_threshold}

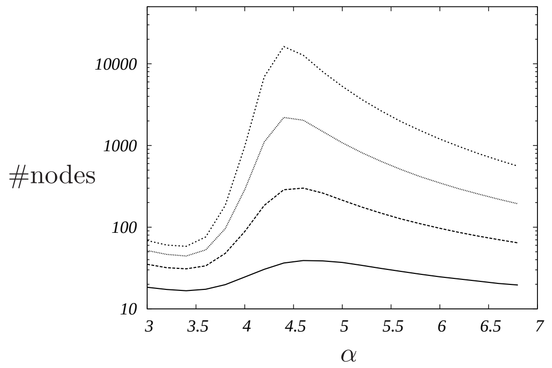


FIG. 10.5. Computational effort of our DPLL algorithm applied to random 3-SAT formulas. Plotted is the average (over 10^4 instances) of the logarithm of the number of branching nodes in the search tree, versus the clause density α. From bottom to top: $N = 50$, 100, 150, 200. {fig:algoperf_3SAT_num}

Proof: Here we shall prove that a formula is almost surely SAT for $\alpha < 1$. The result for $\alpha > 1$ is a consequence of theorem 10.5 below. We use the directed graph representation defined in Ex. 10.5. In this graph, define a bicycle of length s as a path $(u, w_1, w_2, \dots, w_s, v)$, where the w_i are literals on s distinct variables, and $u, v \in \{w_1, \dots, w_s, \overline{w}_1, \dots, \overline{w}_s\}$. As we saw in Ex. 10.5, if a formula F is UNSAT, its directed graph $\mathcal{D}(F)$ has a cycle containing the two literals x_i and $\overline{x}_i$ for some i. From such a cycle one easily builds a bicycle. Therefore:

$$\mathbb{P}(\text{F is UNSAT}) \leq \mathbb{P}(\mathcal{D}(F)\text{has a bicycle}) \leq \sum_{s=2}^{N} N^s 2^s (2s)^2 M^{s+1} \left(\frac{1}{4\binom{N}{2}} \right)^{s+1} . \tag{10.7}$$

{eq:proof2sat1}

The sum is over the size s of the bicycle; N^s is an upper bound to $\binom{N}{s}$, the number of ways one can choose the s variables; 2^s is the choice of literals, given the variables; $(2s)^2$ is the choice of u, v; M^{s+1} is an upper bound to $\binom{M}{s+1}$, the choices of the clauses involved in the bicycle; the last factor is the probability that each of the chosen clauses of the bicycle appears in the random formula. A direct summation of the series in 10.7 shows that, in the large N limit, the result behaves as C/N with a fixed C whenever $M/(N-1) < 1$. □

10.5 Phase transition in random $K(\geq 3)$-SAT

{se:Ksat_intro}

10.5.1 *Satisfiability threshold conjecture*

As noticed above, numerical studies suggest that random K-SAT undergoes a phase transition between a SAT phase and an UNSAT phase, for any $K \geq 2$. There is a widespread belief that this is indeed true, as formalized by the following conjecture:

Conjecture 10.3 *For any $K \geq 2$, there exists a threshold $\alpha_c(K)$ such that:*

$$\lim_{N\to\infty} P_N(K,\alpha) = \begin{cases} 1 & \textit{if } \ \alpha < \alpha_{\rm c}(K) \ , \\ 0 & \textit{if } \ \alpha > \alpha_{\rm c}(K) \ . \end{cases} \tag{10.8}$$

{conj:sat_threshold}

As discussed in the previous Section, this Conjecture is proved in the case $K = 2$. The existence of a phase transition is still an open mathematical problem for larger K, although the following theorem gives some strong support:

{thm:Friedgut}

Theorem 10.4 *(Friedgut) Let $P_N(K,\alpha)$ the probability for a random formula from the $\mathsf{SAT}_N(K,M)$ ensemble to be satisfiable, and assume $K \geq 2$. Then there exists a sequence of $\alpha_{\rm c}^{(N)}(K)$ such that, for any $\varepsilon > 0$,*

$$\lim_{N\to\infty} P_N(K,\alpha) = \begin{cases} 1 & \textit{if } \ \alpha < \alpha_{\rm c}^{(N)}(K) - \varepsilon \ , \\ 0 & \textit{if } \ \alpha > \alpha_{\rm c}^{(N)}(K) + \varepsilon \ , \end{cases} \tag{10.9}$$

In other words, the crossover from SAT to UNSAT becomes sharper and sharper as N increases. For N large enough, it takes place in a window smaller than any fixed width ε. The 'only' missing piece to prove the satisfiability threshold conjecture is the convergence of $\alpha_{\rm c}^{(N)}(K)$ to some value $\alpha_{\rm c}(K)$ as $N \to \infty$.

10.5.2 *Upper bounds*

{sec:UpperBoundSat}

Rigorous studies have allowed to establish bounds on the satisfiability threshold $\alpha_{\rm c}^{(N)}(K)$ in the large N limit. Upper bounds are obtained by using the first moment method. The general strategy is to introduce a function $U(F)$ acting on formulas, with values in $\mathbb{N}$, such that:

{eq:satUBcond}

$$U(F) = \begin{cases} 0 & \text{if } F \text{ is UNSAT,} \\ \geq 1 & \text{otherwise.} \end{cases} \tag{10.10}$$

Therefore, if F is a random K-SAT formula

{eq:sat1mom}

$$\mathbb{P}\{F \text{ is SAT}\} \ \leq \ \mathbb{E}\, U(F) \ . \tag{10.11}$$

The inequality becomes an equality if $U(F) = \mathbb{I}(F \text{ is SAT})$. Of course, we do not know how to compute the expectation in this case. The idea is to find some function $U(F)$ which is simple enough that $\mathbb{E}\, U(F)$ can be computed, and with an expectation value that goes to zero as $N \to \infty$, for large enough α.

The simplest such choice is $U(F) = Z(F)$, the number of SAT assignments (this is the analogous of a "zero-temperature" partition function). The expectation $\mathbb{E}\, Z(F)$ is equal to the number of assignments, 2^N, times the probability that an assignment is SAT (which does not depend on the assignment). Consider for instance the all zeros assignment $x_i = 0$, $i = 1, \dots, N$. The probability that it is SAT is equal to the product of the probabilities that is satisfies each of the M clauses. The probability that the all zeros assignment satisfies a clause is $(1-2^{-K})$ because a K-clause excludes one among the 2^K assignments of variables which appear in it. Therefore

{eq:satZann}
$$\mathbb{E}\, Z(F) = 2^N (1-2^{-K})^M = \exp\left[N\left(\log 2 + \alpha \log(1-2^{-K})\right)\right] . \qquad (10.12)$$

This result shows that, for $\alpha > \alpha_{\rm UB,1}(K)$, where

{eq:alphaub1sat}
$$\alpha_{\rm UB,1}(K) \equiv -\log 2 / \log(1-2^{-K}) \; , \qquad (10.13)$$

$\mathbb{E}\, Z(F)$ is exponentially small at large N. Equation (10.11) implies that the probability of a formula being SAT also vanishes at large N for such an α:

{thm:satupb1}

Theorem 10.5 *If $\alpha > \alpha_{\rm UB,1}(K)$, then $\lim_{N\to\infty} \mathbb{P}\{F \text{ is SAT}\} = 0$. Therefore $\alpha_{\rm c}^{(N)}(K) < \alpha_{\rm UB,1}(K) + \delta$ for any $\delta > 0$ and N large enough.*

One should not expect this bound to be tight. The reason is that, in the SAT phase, $Z(F)$ takes exponentially large values, and its fluctuations tend to be exponential in the number of variables.

Example 10.6 As a simple illustration consider a toy example: the random 1-SAT ensemble $\mathsf{SAT}_N(1,\alpha)$. A formula is generated by including each of the $2N$ literals as a clause independently with probability $\alpha/2$ (we assume of course $\alpha \le 2$). In order for the formula to be SAT, for each of the N variables, at most 1 of the corresponding literals must be included. We have therefore

$$P_N(K=1,\alpha) = (1-\alpha^2/4)^N\,. \tag{10.14}$$

In other words, the probability for a random formula to be SAT goes exponentially fast to 0 for any $\alpha > 0$: $\alpha_{\rm c}(K=1)=0$ (while $\alpha_{\rm UB,1}(K)=1$). Consider now the distribution of $Z(F)$. If F is SAT, then $Z(F)=2^n$, where n is the number of clauses such that none of the corresponding literals is included in F. One has:

$$\mathbb{P}\left\{Z(F)=2^n\right\} = \binom{N}{n}\left(1-\frac{\alpha}{2}\right)^{2n}\left[\alpha\left(1-\frac{\alpha}{2}\right)\right]^{N-n}\,, \tag{10.15}$$

for any $n \ge 0$. We shall now use this expression to compute $\mathbb{E}\,Z(F)$ in a slightly indirect but instructive fashion. First, notice that Eq. (10.15) implies the following large deviation principle for $n > 0$:

$$\mathbb{P}\left\{Z(F)=2^{N\nu}\right\} \doteq \exp\{-N\,I_\alpha(\nu)\} \tag{10.16}$$

$$I_\alpha(\nu) \equiv -\mathcal{H}(\nu) - (1+\nu)\log(1-\alpha/2) - (1-\nu)\log\alpha \tag{10.17}$$

We now compute the expectation of $Z(F)$ via the saddle point approximation

$$\mathbb{E}\,Z(F) \doteq \int e^{-NI_\alpha(\nu)+N\nu\log 2}{\rm d}\nu \doteq \exp\left\{N\max_\nu[-I_\alpha(\nu)+\nu\log 2]\right\}. \tag{10.18}$$

The maximum is achieved at $\nu^* = 1-\alpha/2$. One finds $I_\alpha(\nu^*) = \log(1-\alpha/2) + (\alpha/2)\log 2 > 0$: the probability of having $Z(F) \doteq 2^{N\nu^*}$ is exponentially small. On the other hand $-I_\alpha(\nu^*)+\nu^*\log 2 = \log(2-\alpha) > 0$ for $\alpha < 1$, the factor $2^{N\nu^*}$ overcomes the exponentially small probability of having such a large $Z(F)$, resulting in an exponentially large $\mathbb{E}\,Z(F)$.

Exercise 10.8 Repeat the derivation of Theorem 10.5 for the $\mathsf{SAT}_N(K,\alpha)$ ensemble (i.e. compute $\mathbb{E}\,Z(F)$ for this ensemble and find for which values of α this expectation is exponentially small). Show that the upper bound obtained in this case is $\alpha = 2^K\log 2$. This is worse than the previous upper bound $\alpha_{\rm UB,1}(K)$, although one expects the threshold to be the same. Why? [Hint: The number of clauses M in a $\mathsf{SAT}_N(K,\alpha)$ formula has binomial distribution with parameters N, and α. What values of M provide the dominant contribution to $\mathbb{E}\,Z(F)$?]

In order to improve upon Theorem 10.5 using the first moment method, one needs a better (but still simple) choice of the function $U(F)$. A possible strategy consists in defining some small subclass of ‘special’ SAT assignments, such that if a SAT assignment exists, then a special SAT assignment exists too. If the subclass is small enough, one can hope to reduce the fluctuations in $U(F)$ and sharpen the bound.

One choice of such a subclass consists in ‘locally maximal’ SAT assignments. Given a formula F, an assignment $\underline{x}$ for this formula is said to be a locally maximal SAT assignment if and only if: (1) It is a SAT assignment, (2) for any i such that $x_i = 0$, the assignment obtained by flipping the i-th variable from 0 to 1 is UNSAT. Define $U(F)$ as the number of locally maximal SAT assignments and apply the first moment method to this function. This gives:

{thm:satupb2}

Theorem 10.7 *For any $K \geq 2$, let $\alpha_{\rm UB,2}(K)$ be the unique positive solution of the equation:*

$$\alpha \log(1-2^{-K}) + \log\left[2 - \exp\left(-\frac{K\alpha}{2^K-1}\right)\right] = 0\,. \qquad (10.19)$$

{eq:alphaub2sat}

Then $\alpha_{\rm c}^{(N)}(K) \leq \alpha_{\rm UB,2}(K)$ for large enough N.

The proof is left as the following exercise:

Exercise 10.9 Consider an assignment $\underline{x}$ where exactly L variables are set to 0, the remaining $N-L$ ones being set to 1. Without loss of generality, assume $x_1, \dots, x_L$ to be the variables set to zero.

(*i*) Let p be the probability that a clause constrains the variable x_1, *given that* the clause is satisfied by the assignment $\underline{x}$ (By a clause constraining x_1, we mean that the clause becomes unsatisfied if x_1 is flipped from 0 to 1). Show that $p = \binom{N-1}{K-1}[(2^K-1)\binom{N}{K}]^{-1}$.

(*ii*) Show that the probability that variable x_1 is constrained by at least one of the M clauses, given that all these clauses are satisfied by the assignment $\underline{x}$, is equal to $q = 1-(1-p)^M$

(*iii*) Let $\mathcal{C}_i$ be the event that x_i is constrained by at least one of the M clauses. If $\mathcal{C}_1, \dots, \mathcal{C}_L$ were independent events, under the condition that $\underline{x}$ satisfies F, the probability that $x_1, \dots x_L$ are constrained would be equal q^L. Of course $\mathcal{C}_1, \dots, \mathcal{C}_L$ are not independent. Find an heuristic argument to show that they are anti-correlated and their joint probability is *at most* q^L (consider for instance the case $L=2$).

(*iv*) Show that $\mathbb{E}\,[U(F)] = (1-2^{-K})^M \sum_{L=0}^{N} \binom{N}{L} q^L = (1-2^{-K})^M\,[1+q]^N$ and finish the proof by working out the large N asymptotics of this formula (with $\alpha = M/N$ fixed).

In Table 10.1 we report the numerical values of the upper bounds $\alpha_{\rm UB,1}(K)$ and $\alpha_{\rm UB,2}(K)$ for a few values of K. These results can be slightly improved upon by pursuing the same strategy. For instance, one may strengthen the condition of maximality to flipping 2 or more variables. However the quantitative improvement in the bound is rather small.

10.5.3 *Lower bounds*

Two main strategies have been used to derive lower bounds of $\alpha_{\rm c}^{(N)}(K)$ in the large N limit. In both cases one takes advantage of Theorem 10.4: In order to show that $\alpha_{\rm c}^{(N)}(K) \geq \alpha^*$, it is sufficient to prove that a random $\mathsf{SAT}_N(K,M)$ formula, with $M = \alpha N$, is SAT with non vanishing probability in the $N \to \infty$ limit.

The first approach consists in analyzing explicit heuristic algorithms for finding SAT assignments. The idea is to prove that a particular algorithm finds a SAT assignment with finite probability as $N \to \infty$ when α is smaller than some value.

One of the simplest such bounds is obtained by considering unit clause propagation. Whenever there exist unit clauses, assign one of the variables appearing in these clauses in order to satisfy it, and proceed recursively. Otherwise, chose a variable uniformly at random among those which are not yet fixed assign it to 0 or 1 with probability 1/2. The algorithm halts if it finds a contradiction (i.e. a couple of opposite unit clauses) or if all the variables have been assigned. In the latter case, the found assignment satisfies the formula.

This algorithm is then applied to a random K-SAT formula with clause density α. It can be shown that a SAT assignment is found with positive probability for α small enough: this gives the lower bound $\alpha_{\rm c}^{(N)}(K) \geq \frac{1}{2}\left(\frac{K-1}{K-2}\right)^{K-2}\frac{2^K}{K}$ in the $N \to \infty$ limit. In the Exercise below we give the main steps of the reasoning for the case $K = 3$, referring to the literature for more detailed proofs.

{ex:UCPAnalysis}

Exercise 10.10 After T iterations, the formula will contain 3-clauses, as well as 2-clauses and 1-clauses. Denote by $\mathcal{C}_s(T)$ the set of s-clauses, $s = 1, 2, 3$, and by $C_s(T) \equiv |\mathcal{C}_s(T)|$ its size. Let $\mathcal{V}(T)$ be the set of variables which have not yet been fixed, and $\mathcal{L}(T)$ the set of literals on the variables of $\mathcal{V}(T)$ (obviously we have $|\mathcal{L}(T)| = 2|\mathcal{V}(T)| = 2(N-T)$). Finally, if a contradiction is encountered after $T_{\rm halt}$ steps, we adopt the convention that the formula remains unchanged for all $T \in \{T_{\rm halt}, \dots, N\}$.

(*i*) Show that, for any $T \in \{1, \dots, N\}$, each clause in $\mathcal{C}_s(T)$ is uniformly distributed among the s-clauses over the literals in $\mathcal{L}(T)$.

(*ii*) Show that the expected change in the number of 3- and 2-clauses is given by $\mathbb{E}\left[C_3(T+1) - C_3(T)\right] = -\frac{3C_3(T)}{N-T}$ and $\mathbb{E}\left[C_2(T+1) - C_2(T)\right] = \frac{3C_3(T)}{2(N-T)} - \frac{2C_2(T)}{N-T}$.

(*iii*) Show that, conditional on $C_1(T)$, $C_2(T)$, and $C_3(T)$, the change in the number of 1-clauses is distributed as follows: $C_1(T+1) - C_1(T) \overset{\rm d}{=} -\mathbb{I}(C_1(T) > 0) + B\left(C_2(T), \frac{1}{N-T}\right)$. (We recall that $B(n,p)$ denotes a binomial random variable of parameters n, and p (cf. App. A)).

(*iv*) It can be shown that, as $N \to \infty$ at fixed $t = T/N$, the variables $C_{2/3}(T)/N$ concentrate around their expectation values, and these converge to smooth functions $c_s(t)$. Argue that these functions must solve the ordinary differential equations: $\frac{dc_3}{dt} = -\frac{3}{1-t}c_3(t)$; $\frac{dc_2}{dt} = \frac{3}{2(1-t)}c_3(t) - \frac{2}{1-t}c_2(t)$. Check that the solutions of these equations are: $c_3(t) = \alpha(1-t)^3$, $c_2(t) = (3\alpha/2)t(1-t)^2$.

(*v*) Show that the number of unit clauses is a Markov process described by $C_1(0) = 0$, $C_1(T+1) - C_1(T) \overset{\rm d}{=} -\mathbb{I}(C_1(T) > 0) + \eta(T)$, where $\eta(T)$ is a Poisson distributed random variable with mean $c_2(t)/(1-t)$, where $t = T/N$. Given C_1 and a time T, show that the probability that there is no contradiction generated by the unit clause algorithm up to time T is $\prod_{\tau=1}^{T} \left(1 - 1/(2(N-\tau))\right)^{[C_1(\tau)-1]\mathbb{I}(C_1(\tau \geq 1)}$.

(*vi*) Let $\rho(T)$ be the probability that there is no contradiction up to time T. Consider $T = N(1-\epsilon)$; show that $\rho(N(1-\epsilon)) \geq (1 - 1/(2N\epsilon))^{AN+B}\ \mathbb{P}(\sum_{\tau=1}^{N(1-\epsilon)} C_1(\tau) \leq AN + B)$. Assume that α is such that, $\forall t \in [0, 1-\epsilon]:\ \ c_2(t)/(1-t) < 1$. Show that there exists A, B such that $\lim_{N\to\infty} \mathbb{P}(\sum_{\tau=1}^{N(1-\epsilon)} C_1(\tau) \leq AN + B)$ is finite. Deduce that in the large N limit, there is a finite probability that, at time $N(1-\epsilon)$, the unit clause algorithm has not produced any contradiction so far, and $C_1(N(1-\epsilon)) = 0$.

(*vii*) Conditionnaly to the fact that the algorithm has not produced any contradiction and $C_1(N(1-\epsilon)) = 0$, consider the problem that remains at time $T = N(1-\epsilon)$. Transform each 3-clause into a 2-clause by removing from it a uniformly random variable. Show that one obtains, for ϵ small enough, a random 2-SAT problem with a small clause density $\leq 3\epsilon^2/2$, so that this is a satisfiable instance.

(*viii*) Deduce that, for $\alpha < 8/3$, the unit clause propagation algorithm finds a solution with a finite probability

More refined heuristics have been analyzed using this type of method and lead to better lower bounds on $\alpha_{\rm c}^{(N)}(K)$. We shall not elaborate on this here, but rather present a second strategy, based on a structural analysis of the problem. The idea is to use the second moment method. More precisely, we consider a function $U(F)$ of the SAT formula F, such that $U(F)=0$ whenever F is UNSAT and $U(F)>0$ otherwise. We then make use of the following inequality:

{eq:sat2mom}

$$\mathbb{P}\{F \text{ is SAT}\} \;= \mathbb{P}\{U(F)>0\} \;\geq\; \frac{[\mathbb{E}\, U(F)]^2}{\mathbb{E}[U(F)^2]}\ . \tag{10.20}$$

The present strategy is more delicate to implement than the first moment method, used in Sec. 10.5.2 to derive upper bounds on $\alpha_{\rm c}^{(N)}(K)$. For instance, the simple choice $U(F)=Z(F)$ does not give any result: It turns out that the ratio $[\mathbb{E}\, Z(F)]^2/\mathbb{E}[Z(F)^2]$ is exponentially small in N for any non vanishing value of α, so that the inequality (10.20) is useless. Again one needs to find a function $U(F)$ whose fluctuations are smaller than the number $Z(F)$ of SAT assignments. More precisely, one needs the ratio $[\mathbb{E}\, U(F)]^2/\mathbb{E}[U(F)^2]$ to be non vanishing in the $N\to\infty$ limit. ⋆

A successful idea uses a weighted sum of SAT assignments:

$$U(F) = \sum_{\underline{x}} \prod_{a=1}^{M} W(\underline{x},a)\ . \tag{10.21}$$

Here the sum is over all the 2^N assignments, and $W(\underline{x},a)$ is a weight associated with clause a. This weight must be such that $W(\underline{x},a)=0$ when the assignment $\underline{x}$ does not satisfy clause a, and $W(\underline{x},a)>0$ otherwise. Let us choose a weight which depends on the number $r(\underline{x},a)$ of variables which satisfy clause a in the assignment $\underline{x}$:

$$W(\underline{x},a) = \begin{cases} \varphi(r(\underline{x},a)) & \text{if } r(\underline{x},a)\geq 1,\\ 0 & \text{otherwise.}\end{cases} \tag{10.22}$$

It is then easy to compute the first two moments of $U(F)$:

$$\mathbb{E}\, U(F) = 2^N \left[2^{-K} \sum_{r=1}^{K} \binom{K}{r} \varphi(r)\right]^M\ , \tag{10.23}$$

$$\mathbb{E}\left[U(F)^2\right] = 2^N \sum_{L=0}^{N} \binom{N}{L}\ [g_{\varphi}(N,L)]^M\ . \tag{10.24}$$

Here $g_\varphi(N,L)$ is the expectation value of the product $W(\underline{x},a)W(\underline{y},a)$ when a clause a is chosen uniformly at random, given that $\underline{x}$ and $\underline{y}$ are two assignments of N variables which agree on *exactly* L of them.

In order to compute $g_\varphi(N,L)$, it is convenient to introduce two binary vectors $\vec{u},\vec{v}\in\{0,1\}^K$. They encode the following information: Consider a clause a, fix

$u_s = 1$ if in the assignment $\underline{x}$ the s-th variable of clause a satisfies the clause, and $u_s = 0$ otherwise. The components of $\vec{v}$ are defined similarly but with the assignment $\underline{y}$. Furthermore, we denote by $d(\vec{u}, \vec{v})$ the Hamming distance between these vectors, and by $w(\vec{u})$, $w(\vec{v})$ their Hamming weights (number of non zero components). Then

$$g_\varphi(N, L) = 2^{-K} {\sum_{\vec{u},\vec{v}}}' \varphi\left(w(\vec{u})\right) \varphi\left(w(\vec{v})\right) \left(\frac{L}{N}\right)^{d(\vec{u},\vec{v})} \left(1 - \frac{L}{N}\right)^{K - d(\vec{u},\vec{v})} . \tag{10.25}$$

Here the sum $\sum'$ runs over K-component vectors $\vec{u}$, $\vec{v}$ with at least one non zero component. A particularly simple case is $\varphi(r) = \lambda^r$. Denoting $z = L/N$, one finds:

$$g_w(N, L) = 2^{-K} \left(\left[(\lambda^2 + 1)z + 2\lambda(1 - z)\right]^K - 2\left[z + \lambda(1 - z)\right]^K + z^k \right) . \tag{10.26}$$

The first two moments can be evaluated from Eqs. (10.23), (10.24):

$$\mathbb{E}\, U(F) \doteq \exp\{N h_1(\lambda, \alpha)\} , \quad \mathbb{E}\left[U(F)^2\right] \doteq \exp\{N \max_z h_2(\lambda, \alpha, z)\} , \tag{10.27}$$

where the maximum is taken over $z \in [0, 1]$ and

$$h_1(\lambda, \alpha) \equiv \log 2 - \alpha K \log 2 + \alpha \log\left[(1 + \lambda)^K - 1\right] , \tag{10.28}$$

$$\begin{aligned} h_2(\lambda, \alpha, z) \equiv {} & \log 2 - z \log z - (1 - z)\log(1 - z) - \alpha K \log 2 + \\ & + \alpha \log\left(\left[(\lambda^2 + 1)z + 2\lambda(1 - z)\right]^K - 2\left[z + \lambda(1 - z)\right]^K + z^k \right) . \end{aligned} \tag{10.29}$$

Evaluating the above expression for $z = 1/2$ one finds $h_2(\lambda, \alpha, 1/2) = 2h_1(\lambda, \alpha)$. The interpretation is as follows. Setting $z = 1/2$ amounts to assuming that the second moment of $U(F)$ is dominated by completely uncorrelated assignments (two uniformly random assignments agree on about half of the variables). This results in the factorization of the expectation $\mathbb{E}\left[U(F)^2\right] \approx [\mathbb{E}\, U(F)]^2$.

Two cases are possible: either the maximum of $h_2(\lambda, \alpha, z)$ over $z \in [0, 1]$ is achieved only at $z = 1/2$ or not.

(*i*) In the latter case $\max_z h_2(\lambda, \alpha, z) > 2h_1(\lambda, \alpha)$ strictly, and therefore the ratio $[\mathbb{E}\, U(F)]^2/\mathbb{E}[U(F)^2]$ is exponentially small in N, the second moment inequality (10.20) is useless.

(*ii*) If on the other hand the maximum of $h_2(\lambda, \alpha, z)$ is achieved only at $z = 1/2$, then the ratio $[\mathbb{E}\, U(F)]^2/\mathbb{E}[U(F)^2]$ is 1 to the leading exponential order. It is not difficult to work out the precise asymptotic behavior (i.e. to compute the prefactor of the exponential). One finds that $[\mathbb{E}\, U(F)]^2/\mathbb{E}[U(F)^2]$ remains finite when $N \to \infty$. As a consequence $\alpha \leq \alpha_{\rm c}^{(N)}(K)$ for N large enough.

Table 10.1 *Satisfiability thresholds for random K-SAT. We report the lower bound from Theorem (10.8) and the upper bounds from Eqs. (10.13) and (10.19).*

K	3	4	5	6	7	8	9	10
$\alpha_{\rm LB}(K)$	2.548	7.314	17.62	39.03	82.63	170.6	347.4	701.5
$\alpha_{\rm UB,1}(K)$	5.191	10.74	21.83	44.01	88.38	177.1	354.5	709.4
$\alpha_{\rm UB,2}(K)$	4.666	10.22	21.32	43.51	87.87	176.6	354.0	708.9

{tab:alphabounds}

A necessary condition for the second case to occur is that $z = 1/2$ is a local maximum of $h_2(\lambda, \alpha, z)$. This implies that λ must be the (unique) strictly positive root of:

$$(1+\lambda)^{K-1} = \frac{1}{1-\lambda}\,. \qquad (10.30)$$

{eq:lambdadef}

{thm:KSAT_lowerbound} We have thus proved that:

Theorem 10.8 *Let λ be the positive root of Eq. (10.30), and the function $h_2(\,\cdot\,)$ be defined as in Eq. (10.29). Assume that $h_2(\lambda, \alpha, z)$ achieves its maximum, as a function of $z \in [0,1]$ only at $z = 1/2$. Then a random $\mathsf{SAT}_N(K, \alpha)$ is SAT with probability approaching one as $N \to \infty$.*

Let $\alpha_{\rm LB}(K)$ be the largest value of α such that the hypotheses of this Theorem are satisfied. The Theorem implies an explicit lower bound on the satisfiability threshold: $\alpha_{\rm c}^{(N)}(K) \geq \alpha_{\rm LB}(K)$ in the $N \to \infty$ limit. Table 10.1 summarizes some of the values of the upper and lower bounds found in this Section for a few values of K. In the large K limit the following asymptotic behaviors can be shown to hold:

$$\alpha_{\rm LB}(K) = 2^K \log 2 - 2(K+1)\log 2 - 1 + o(1)\,, \qquad (10.31)$$

$$\alpha_{\rm UB,1}(K) = 2^K \log 2 - \frac{1}{2}\log 2 + o(1)\,. \qquad (10.32)$$

In other words, the simple methods exposed in this Chapter allow to determine the satisfiability threshold with a relative error behaving as 2^{-K} in the large K limit. More sophisticated tools, to be discussed in the next Chapters, are necessary for obtaining sharp results at finite K.

{ex:SecondMoment}

Exercise 10.11 [Research problem] Show that the choice of weight $\varphi(r) = \lambda^r$ is optimal: all other choices for $\varphi(r)$ give a worse lower bound. What strategy could be followed to improve the bound further?

Notes

The review paper (Gu, Purdom, Franco and Wah, 2000) is a rather comprehensive source of information on the algorithmic aspects of satisfiability. The reader interested in applications will also find there a detailed and referenced list.

Davis and Putnam first studied an algorithm for satisfiability in (Davis and Putnam, 1960). This was based on a systematic application of the resolution rule. The backtracking algorithm discussed in the main text was introduced in (Davis, Logemann and Loveland, 1962).

Other ensembles of random CNF formulas have been studied, but it turns out it is not so easy to find hard formulas. For instance take N variables, and generate M clauses independently according to the following rule. In a clause a, each of the variables appears as x_i or $\overline{x}_i$ with the same probability $p \leq 1/2$, and does not appear with probability $1 - 2p$. The reader is invited to study this ensemble; an introduction and guide to the corresponding literature can be found in (Franco, 2000). Another useful ensemble is the "$2 + p$" SAT problem which interpolates between $K = 2$ and $K = 3$ by picking pM 3-clauses and $(1 - p)M$ 2-clauses, see (Monasson, Zecchina, Kirkpatrick, Selman and Troyansky, 1999) ⋆

The polynomial nature of 2-SAT is discussed in (Cook, 1971). MAX-2SAT was shown to be NP-complete in (Garey, Johnson and Stockmeyer, 1976).

Schöning's algorithm was introduced in (Schöning, 1999) and further discussed in (Schöning, 2002). More general random walk strategies for SAT are treated in (Papadimitriou, 1991; Selman and Kautz, 1993; Selman, Kautz and Cohen, 1994).

The threshold $\alpha_{\rm c} = 1$ for random 2-SAT was proved in (Chvátal and Reed, 1992), (Goerdt, 1996) and (de la Vega, 1992), but see also (de la Vega, 2001). The scaling behavior near to the threshold has been analyzed through graph theoretical methods in (Bollobas, Borgs, Chayes, Kim and Wilson, 2001).

The numerical identification of the phase transition in random 3-SAT, and the observation that difficult formulas are found near to the phase transition, are due to Kikpatrick and Selman (Kirkpatrick and Selman, 1994; Selman and Kirkpatrick, 1996). See also (Selman, Mitchell and Levesque, 1996).

Friedgut's theorem is proved in (Friedgut, 1999).

Upper bounds on the threshold are discussed in (Dubois and Boufkhad, 1997; Kirousis, Kranakis, Krizanc and Stamatiou, 1998). Lower bounds for the threshold in random K-SAT based on the analysis of some algorithms were pioneered by Chao and Franco. The paper (Chao and Franco, 1986) corresponds to Exercise 10.10, and a generalization can be found in (Chao and Franco, 1990). A review of this type of methods is provided by (Achlioptas, 2001). (Cocco, Monasson, Montanari and Semerjian, 2003) gives a survey of the analysis of algorithms based on physical methods. The idea of deriving a lower bound with the weighted second moment method was discussed in (Achlioptas and Moore, 2005). The lower bound which we discuss here is derived in (Achlioptas and Peres, 2004); this paper also solves the first question of Exercise 10.11. A simple introduction to the second moment method in various constraint satisfaction problems is (Achlioptas, Naor and Peres, 2005), see also (Gomes and Selman, 2005).

11

LOW-DENSITY PARITY-CHECK CODES

{ch:LDPC}

Low-density parity-check (LDPC) error correcting codes were introduced in 1963 by Robert Gallager in his Ph.D. thesis. The basic motivation came from the observation that random linear codes, cf. Section **??**, had excellent theoretical performances but were unpractical. In particular, no efficient algorithm was known for decoding. In retrospect, this is not surprising, since it was later shown that decoding for linear codes is an NP-hard problem.

The idea was then to restrict the RLC ensemble. If the resulting codes had enough structure, one could exploit it for constructing some efficient decoding algorithm. This came of course with a price: restricting the ensemble could spoil its performances. Gallager's proposal was simple and successful (but ahead of times): LDPC codes are among the most efficient codes around.

In this Chapter we introduce one of the most important families of LDPC ensembles and derive some of their basic properties. As for any code, one can take two quite different points of view. The first is to study the code performances[27] under *optimal* decoding. In particular, no constraint is imposed on the computational complexity of decoding procedure (for instance decoding through a scan of the whole, exponentially large, codebook is allowed). The second approach consists in analyzing the code performance under some specific, efficient, decoding algorithm. Depending on the specific application, one can be interested in algorithms of polynomial complexity, or even require the complexity to be linear in the block-length.

Here we will focus on performances under optimal decoding. We will derive rigorous bounds, showing that appropriately chosen LDPC ensembles allow to communicate reliably at rates close to Shannon's capacity. However, the main interest of LDPC codes is that they can be decoded efficiently, and we will discuss a simple example of decoding algorithm running in linear time. The full-fledged study of LDPC codes under optimal decoding is deferred to Chapters **??**. A more sophisticated decoding algorithm will be presented and analyzed in Chapter **??**.

After defining LDPC codes and LDPC code ensembles in Section 11.1, we discuss some geometric properties of their codebooks in Section 11.2. In Section 11.3 we use these properties to a lower bound on the threshold for reliable communication. An upper bound follows from information-theoretic considera-

[27] Several performance parameters (e.g. the bit or block error rates, the information capacity, etc.) can be of interest. Correspondingly, the 'optimal' decoding strategy can vary (for instance symbol MAP, word MAP, etc.). To a first approximation, the choice of the performance criterion is not crucial, and we will keep the discussion general as far as possible.

tions. Section 11.4 discusses a simple-minded decoding algorithm, which is shown to correct a finite fraction of errors.

11.1 Definitions

{se:DefLDPC}

11.1.1 *Boolean linear algebra*

Remember that a code is characterized by its codebook $\mathfrak{C}$, which is a subset of $\{\mathtt{0},\mathtt{1}\}^N$. LDPC codes are **linear codes**, which means that the codebook is a linear subspace of $\{\mathtt{0},\mathtt{1}\}^N$. In practice such a subspace can be specified through an $M\times N$ matrix $\mathbb{H}$, with binary entries $\mathbb{H}_{ij}\in\{\mathtt{0},\mathtt{1}\}$, and $M<N$. The codebook is defined as the kernel of $\mathbb{H}$:

$$\mathfrak{C}=\{\,\underline{x}\in\{\mathtt{0},\mathtt{1}\}^N\ :\ \mathbb{H}\underline{x}=\underline{0}\,\}\,. \tag{11.1}$$

Here and in all this chapter, the multiplications and sums involved in $\mathbb{H}\underline{x}$ are understood as being computed modulo 2. The matrix $\mathbb{H}$ is called the **parity check matrix** of the code. The size of the codebook is $2^{N-\mathrm{rank}(\mathbb{H})}$, where $\mathrm{rank}(\mathbb{H})$ denotes the rank of the matrix $\mathbb{H}$ (number of linearly independent rows). As $\mathrm{rank}(\mathbb{H})\le M$, the size of the codebook is $|\mathfrak{C}|\ge 2^{N-M}$. With a slight modification with respect to the notation of Chapter 1, we let $L\equiv N-M$. The rate R of the code verifies therefore $R\ge L/N$, equality being obtained when all the rows of $\mathbb{H}$ are linearly independent.

Given such a code, encoding can always be implemented as a linear operation. There exists a $N\times L$ binary matrix $\mathbb{G}$ (the generating matrix) such that the codebook is the image of $\mathbb{G}$: $\mathfrak{C}=\{\underline{x}=\mathbb{G}\underline{z}\,,\text{ where }\underline{z}\in\{\mathtt{0},\mathtt{1}\}^L\}$. Encoding is therefore realized as the mapping $\underline{z}\mapsto\underline{x}=\mathbb{G}\underline{z}$. (Notice that the product $\mathbb{H}\,\mathbb{G}$ is a $M\times L$ 'null' matrix with all entries equal to zero).

11.1.2 *Factor graph*

In Example 9.5 we described the factor graph associated with one particular linear code (a Hamming code). The recipe to build the factor graph, knowing $\mathbb{H}$, is as follows. Let us denote by $i_1^a,\dots,i_{k(a)}^a\in\{1,\dots,N\}$ the column indices such that $\mathbb{H}$ has a matrix element equal to $\mathtt{1}$ at row a and column i_j^a. Then the a-th coordinate of the vector $\mathbb{H}\underline{x}$ is equal to $x_{i_1^a}\oplus\cdots\oplus x_{i_{k(a)}^a}$. Let $P_{\mathbb{H}}(\underline{x})$ be the uniform distribution over all codewords of the code $\mathbb{H}$ (hereafter we shall often identify a code with its parity check matrix). It is given by:

$$P_{\mathbb{H}}(\underline{x})=\frac{1}{Z}\prod_{a=1}^{M}\mathbb{I}(x_{i_1^a}\oplus\cdots\oplus x_{i_k^a}=\mathtt{0})\,. \tag{11.2}$$

Therefore, the factor graph associated with $P_{\mathbb{H}}(\underline{x})$ (or with $\mathbb{H}$) includes N variable nodes, one for each column of $\mathbb{H}$, and M function nodes (also called, in this context, **check nodes**), one for each row. A factor node and a variable node are joined by an edge if the corresponding entry in $\mathbb{H}$ is non-vanishing. Clearly this procedure can be inverted: to any factor graph with N variable nodes and M

function nodes, we can associate an $M \times N$ binary matrix $\mathbb{H}$, the **adjacency matrix** of the graph, whose non-zero entries correspond to the edges of the graph.

{se:LDPCegdp}

11.1.3 *Ensembles with given degree profiles*

In Chapter 9 we introduced the ensembles of factor graphs $\mathbb{D}_N(\Lambda, P)$. These have N variable nodes, and the two polynomials $\Lambda(x) = \sum_{n=0}^{\infty} \Lambda_n x^n$, $P(x) = \sum_{n=0}^{\infty} P_n x^n$ define the degree profiles: Λ_n is the probability that a randomly chosen variable node has degree n, P_n is the probability that a randomly chosen function node has degree n. We always assume that variable nodes have degrees ≥ 1, and function nodes have degrees ≥ 2, in order to eliminate trivial cases. The numbers of parity check and variable nodes satisfy the relation $M = N\Lambda'(1)/P'(1)$.

We define the ensemble $\text{LDPC}_N(\Lambda, P)$ to be the ensemble of LDPC codes whose parity check matrix is the adjacency matrix of a random graph from the $\mathbb{D}_N(\Lambda, P)$ ensemble. (We will be interested in the limit $N \to \infty$ while keeping the degree profiles fixed. Therefore each vertex typically connects to a vanishingly small fraction of other vertices, hence the qualification 'low density'). The ratio $L/N = (N - M)/N = 1 - \Lambda'(1)/P'(1)$, which is a lower bound to the actual rate R, is called the **design rate** R_{des} of the code (or, of the ensemble). The actual rate of a code from the $\text{LDPC}_N(\Lambda, P)$ ensemble is of course a random variable, but we will see below that it is in general sharply concentrated 'near' R_{des}.

A special case which is often considered is the one of 'regular' graphs with fixed degrees: all variable nodes have degree l and all functions nodes have degree k, (i.e. $P(x) = x^k$ and $\Lambda(x) = x^l$). The corresponding code ensemble is usually simply denoted as $\text{LDPC}_N(l, k)$, or, more synthetically as (l, k). It has design rate $R_{\text{des}} = 1 - \frac{l}{k}$.

Generating a uniformly random graph from the $\mathbb{D}_N(\Lambda, P)$ ensemble is not a trivial task. The simplest way to by-pass such a problem consists in substituting the uniformly random ensemble with a slightly different one which has a simple algorithmic description. One can proceed for instance as follows. First separate the set of variable nodes uniformly at random into subsets of sizes $N\Lambda_0$, $N\Lambda_1$, $\dots$, $N\Lambda_{l_{\max}}$, and attribute 0 'sockets' to the nodes in the first subset, one socket to each of the nodes in the second, and so on. Analogously, separate the set of check nodes into subsets of size MP_0, MP_1, $\dots$, $MP_{k_{\max}}$ and attribute to nodes in each subset $0, 1, \dots, k_{\max}$ socket. At this point the variable nodes have $N\Lambda'(1)$ sockets, and so have the check nodes. Draw a uniformly random permutation over $N\Lambda'(1)$ objects and connect the sockets on the two sides accordingly.

Exercise 11.1 In order to sample a graph as described above, one needs two routines. The first one separates a set of N objects uniformly into subsets of prescribed sizes. The second one samples a random permutation over a $N\Lambda'(1)$. Show that both of these tasks can be accomplished with $O(N)$ operations (having at our disposal a random number generator).

This procedure has two flaws: (i) it does not sample uniformly $\mathbb{D}_N(\Lambda, P)$, because two distinct factor graphs may correspond to a different number of permutations. (ii) it may generate multiple edges joining the same couple of nodes in the graph.

In order to cure the last problem, we shall agree that each time n edges join any two nodes, they must be erased if n is even, and they must be replaced by a single edge if n is odd. Of course the resulting graph does not necessarily have the prescribed degree profile (Λ, P), and even if we condition on this to be the case, its distribution is not uniform. We shall nevertheless insist in denoting the ensemble as $\text{LDPC}_N(\Lambda, P)$. The intuition is that, for large N, the degree profile is 'close' to the prescribed one and the distribution is 'almost uniform', for all our purposes. Moreover, what is really important is the ensemble that is implemented in practice.

Exercise 11.2 This exercise aims at proving that, for large N, the degree profile produced by the explicit construction is close to the prescribed one.

(i) Let m be the number of multiple edges appearing in the graph and compute its expectation. Show that $\mathbb{E}\, m = O(1)$ as $N \to \infty$ with Λ and P fixed.

(ii) Let (Λ', P') be the degree profile produced by the above procedure. Denote by

$$d \equiv \sum_l |\Lambda_l - \Lambda'_l| + \sum_k |P_k - P'_k| \, , \tag{11.3}$$

the 'distance' between the prescribed and the actual degree profiles. Derive an upper bound on d in terms of m and show that it implies $\mathbb{E}\, d = O(1/N)$.

11.2 Geometry of the codebook

{se:WELDPC}

As we saw in Sec. 6.2, a classical approach to the analysis of error correcting codes consists in studying the 'geometric' properties of the corresponding codebooks. An important example of such properties is the distance enumerator $\mathcal{N}_{\underline{x}_0}(d)$, giving the number of codewords at Hamming distance d from $\underline{x}_0$. In the case of linear codes, the distance enumerator does not depend upon the reference codeword $\underline{x}_0$ (the reader is invited to prove this simple statement).It is therefore customary to take the all-zeros codeword as the reference, and to use the denomination **weight enumerator**: $\mathcal{N}(w) = \mathcal{N}_{\underline{x}_0}(d = w)$ is the number of codewords having **weight** (the number of ones in the codeword) equal to w. ⋆

In this section we want to estimate the expected weight enumerator $\overline{\mathcal{N}}(w) \equiv \mathbb{E}\mathcal{N}(w)$, for a random code in the $\text{LDPC}_N(\Lambda, P)$ ensemble. In general one expects, as for the random code ensemble of Sec. 6.2, that $\overline{\mathcal{N}}(w)$ grows exponentially in the block-length N, and that most of the codewords have a weight

$w = N\omega$ growing linearly with N. We will in fact compute the exponential growth rate $\phi(\omega)$ defined by

$$\overline{\mathcal{N}}(w = N\omega) \doteq e^{N\phi(\omega)} \; . \tag{11.4}$$ {eq:weightphidef}

Notice that this number is an 'annealed average', in the terminology of disordered systems: in other words, it can be dominated by rare instances in the ensemble. On the other hand, one expects $\log \mathcal{N}(w)$ to be tightly concentrated around its typical value $N\phi_{\rm q}(\omega)$. The typical exponent $\phi_{\rm q}(\omega)$ can be computed through a quenched calculation, for instance considering $\lim_{N\to\infty} N^{-1}\mathbb{E}\log\left[1+\mathcal{N}(w)\right]$. Of course $\phi_{\rm q}(\omega) \le \phi(\omega)$ because of the concavity of the logarithm. In this Chapter we keep to the annealed calculation, which is much easier and gives an upper bound. Quenched calculations will be the object of Chapter **???**.

Let $\underline{x} \in \{0,1\}^N$ be a binary word of length N and weight w. Notice that $\mathbb{H}\underline{x} = 0 \mod 2$ if and only if the corresponding factor graph has the following property. Consider all variable nodes i such that $x_i = 1$, and color in red all edges incident on these nodes. Color in blue all the other edges. Then all the check nodes must have an even number of incident red edges. A little thought shows that $\overline{\mathcal{N}}(w)$ is the number of 'colored' factor graphs having this property, divided by the total number of factor graphs in the ensemble. We shall compute this number first for a graph with fixed degrees, associated with a code in the $\text{LDPC}_N(l,k)$ ensemble, and then we shall generalize to arbitrary degree profiles.

11.2.1 *Weight enumerator: fixed degrees*

In the fixed degree case we have N variables nodes of degree l, M function nodes of degree k. We denote by $F = Mk = Nl$ the total number of edges. A valid colored graph must have $E = wl$ red edges. It can be constructed as follows. First choose w variable nodes, which can be done in $\binom{N}{w}$ ways. Assign to each node in this set l red sockets, and to each node outside the set l blue sockets. Then, for each of the M function nodes, color in red an even subset of its sockets in such a way that the total number of red sockets is $E = wl$. Let m_r be the number of function nodes with r red sockets. The numbers m_r can be non-zero only when r is even, and they are constrained by $\sum_{r=0}^k m_r = M$ and $\sum_{r=0}^k r m_r = lw$. The number of ways one can color the sockets of the function nodes is thus:

{eq:colsock}

$$\begin{aligned}\mathcal{C}(k,M,w) = \sum_{m_0,\dots,m_k}^{(e)} \binom{M}{m_0,\dots,m_k} \prod_r \binom{k}{r}^{m_r} \\ \mathbb{I}\Big(\sum_{r=0}^k m_r = M\Big)\; \mathbb{I}\Big(\sum_{r=0}^k r m_r = lw\Big) \; ,\end{aligned} \tag{11.5}$$

where the sum $\sum^{(e)}$ means that non-zero m_r appear only for r even. Finally we join the variable node and check node sockets in such a way that colors are matched. There are $(lw)!(F-lw)!$ such matchings out of the total number of $F!$

corresponding to different element in the ensemble. Putting everything together, we get the final formula:

$$\overline{\mathcal{N}}(w) = \frac{(lw)!(F-lw)!}{F!}\binom{N}{w}\,\mathcal{C}(k,M,w)\ . \tag{11.6}$$

In order to compute the function $\phi(\omega)$ in (11.4), one needs to work out the asymptotic behavior of this formula when $N\to\infty$ at fixed $\omega = w/N$. Assuming that $m_r = x_r M = x_r Nl/k$, one can expand the multinomial factors using Stirling's formula. This gives:

$$\phi(\omega) = \max_{\{x_r\}}{}^{*}\left[(1-l)\mathcal{H}(\omega) + \frac{l}{k}\sum_r\left(-x_r\log x_r + x_r\log\binom{k}{r}\right)\right]\ , \tag{11.7}$$ {eq:weightphires1}

where the max* is taken over all choices of $x_0, x_2, x_4, \dots$ in $[0,1]$, subject to the two constraints $\sum_r x_r = 1$ and $\sum_r r x_r = k\omega$. The maximization can be done by imposing these constraints via two Lagrange multipliers. One gets $x_r = Cz^r\binom{k}{r}\mathbb{I}(r \text{ even})$, where C and z are two constants fixed by the constraints:

$$C = \frac{2}{(1+z)^k + (1-z)^k} \tag{11.8}$$

$$\omega = z\frac{(1+z)^{k-1} - (1-z)^{k-1}}{(1+z)^k + (1-z)^k} \tag{11.9}$$

Plugging back the resulting x_r into the expression (11.10) of ϕ, this gives finally:

$$\phi(\omega) = (1-l)\mathcal{H}(\omega) + \frac{l}{k}\log\frac{(1+z)^k + (1-z)^k}{2} - \omega l\log z\ , \tag{11.10}$$ {eq:weightphires1}

where z is the function of ω defined in (11.9).

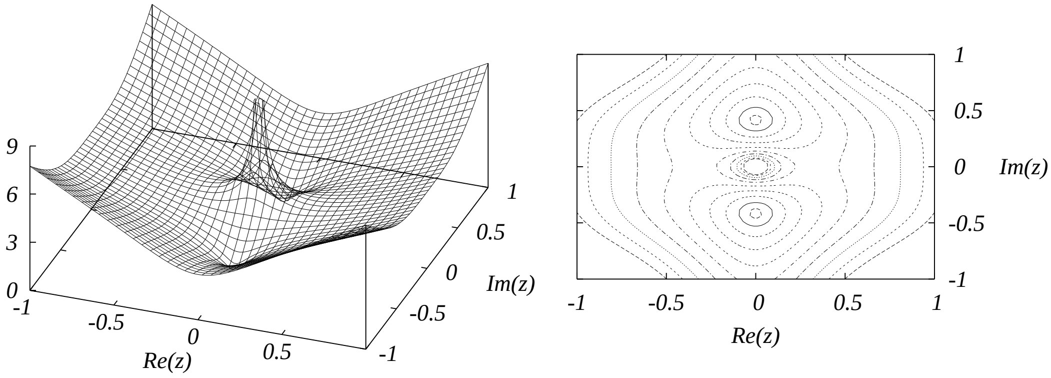
We shall see in the next sections how to use this result, but let us first explain how it can be generalized.

11.2.2 *Weight enumerator: general case*

We shall compute the leading exponential behavior $\overline{\mathcal{N}}(w) \doteq \exp[N\phi(\omega)]$ of the expected weight enumerator for a general LDPC$_N(\Lambda, P)$ code. The idea of the approach is the same as the one we have just used for the case of regular ensembles, but the computation becomes somewhat heavier. It is therefore useful to adopt more compact notations. Altogether this section is more technical than the others: the reader who is not interested in the details can skip it and go to the results.

We want to build a valid colored graph, let us denote by E its number of red edges (which is no longer fixed by w). There are $\mathsf{coeff}[\prod_l (1+xy^l)^{N\Lambda_l}, x^w y^E]$ ways of choosing the w variable nodes in such a way that their degrees add up to E [28]. As before, for each of the M function nodes, we color in red an even subset

[28] We denote by $\mathsf{coeff}[f(x), x^n]$ the coefficient of x^n in the formal power series $f(x)$.



{fig:SaddleWE}

FIG. 11.1. Modulus of the function $z^{-3\xi}\, q_4(z)^{3/4}$ for $\xi = 1/3$.

of its sockets in such a way that the total number of red sockets is E. This can be done in $\mathsf{coeff}[\prod_k q_k(z)^{MP_k}, z^E]$ ways, where $q_k(z) \equiv \frac{1}{2}(1+z)^k + \frac{1}{2}(1-z)^k$. The numbers of ways one can match the red sockets in variable and function nodes is still $E!(F-E)!$, where $F = N\Lambda'(1) = MP'(1)$ is the total number of edges in the graph. This gives the exact result

{eq:WELeading1}

$$\overline{\mathcal{N}}(w) = \sum_{E=0}^{F} \frac{E!(F-E)!}{F!} \mathsf{coeff}\left[\prod_{l=1}^{l_{\max}} (1+xy^l)^{N\Lambda_l}, x^w y^E\right] \mathsf{coeff}\left[\prod_{k=2}^{k_{\max}} q_k(z)^{MP_k}, z^E\right] . \qquad (11.11)$$

In order to estimate the leading exponential behavior at large N, when $w = N\omega$, we set $E = F\xi = N\Lambda'(1)\xi$. The asymptotic behaviors of the $\mathsf{coeff}[\dots,\dots]$ terms can be estimated using the saddle point method. Here we sketch the idea for the second of these terms. By Cauchy theorem

$$\mathsf{coeff}\left[\prod_{k=2}^{k_{\max}} q_k(z)^{MP_k}, z^E\right] = \oint \frac{1}{z^{N\Lambda'(1)\xi+1}} \prod_{k=2}^{k_{\max}} q_k(z)^{MP_k} \frac{\mathrm{d}z}{2\pi i} \equiv \oint \frac{f(z)^N}{z} \frac{\mathrm{d}z}{2\pi i} , \qquad (11.12)$$

where the integral runs over any path encircling the origin in the complex z plane, and

$$f(z) \equiv \frac{1}{z^{\Lambda'(1)\xi}} \prod_{k=2}^{k_{\max}} q_k(z)^{\Lambda'(1)P_k/P'(1)} . \qquad (11.13)$$

In Fig. 11.1 we plot the modulus of the function $f(z)$ for degree distributions $\Lambda(x) = x^3$, $P(x) = x^4$ and $\xi = 1/3$. The function has a saddle point, whose location $z_* = z_*(\xi) \in \mathbb{R}_+$ solves the equation $f'(z) = 0$, which can also be written as

$$\xi = \sum_{k=2}^{k_{\max}} \rho_k \, z \frac{(1+z)^{k-1} - (1-z)^{k-1}}{(1+z)^k + (1-z)^k} \, , \tag{11.14}$$

where we used the notation $\rho_k \equiv kP_k/P'(1)$ already introduced in Sec. 9.5 (analogously, we shall write $\lambda_l \equiv l\Lambda_l/\Lambda'(1)$). This equation generalizes (11.9). If we take the integration contour in Eq. (11.12) to be the circle of radius z_*, the integral is dominated by the saddle point at z_* (together with the symmetric point $-z_*$). We get therefore

$$\mathsf{coeff}\left[\prod_{k=2}^{k_{\max}} q_k(z)^{MP_k}, z^E\right] \doteq \exp\left\{N\left[-\Lambda'(1)\xi\log z_* + \frac{\Lambda'(1)}{P'(1)}\sum_{k=2}^{k_{\max}} P_k \log q_k(z_*)\right]\right\} .$$

Proceeding analogously with the second $\mathsf{coeff}[\dots,\dots]$ term in Eq. (11.11), we get $\overline{\mathcal{N}}(w = N\omega) \doteq \exp\{N\phi(\omega)\}$. The function ϕ is given by

$$\begin{aligned}\phi(\omega) = \sup_{\xi} \inf_{x,y,z} \Big\{ &-\Lambda'(1)\mathcal{H}(\xi) - \omega\log x - \Lambda'(1)\xi\log(yz) + \\ &+ \sum_{l=2}^{l_{\max}} \Lambda_l \log(1+xy^l) + \frac{\Lambda'(1)}{P'(1)} \sum_{k=2}^{k_{\max}} P_k \log q_k(z) \Big\} \, ,\end{aligned} \tag{11.15}$$

where the minimization over x, y, z is understood to be taken over the positive real axis while $\xi \in [0,1]$. The stationarity condition with respect to variations of z is given by Eq. (11.14). Stationarity with respect to ξ, x, y yields, respectively

$$\xi = \frac{yz}{1+yz} \, , \qquad \omega = \sum_{l=1}^{l_{\max}} \Lambda_l \frac{xy^l}{1+xy^l} \, , \qquad \xi = \sum_{l=1}^{l_{\max}} \lambda_l \, \frac{xy^l}{1+xy^l} \, . \tag{11.16}$$

If we use the first of these equations to eliminate ξ, we obtain the final parametric representation (in the parameter $x \in [0, \infty[$) of $\phi(\omega)$:

$$\begin{aligned}\phi(\omega) = -\omega\log x - \Lambda'(1)\log(1+yz) + \sum_{l=1}^{l_{\max}} \Lambda_l \log(1+xy^l) + \qquad (11.17)\\ + \frac{\Lambda'(1)}{P'(1)} \sum_{k=2}^{k_{\max}} P_k \log q_k(z) \, ,\end{aligned}$$

$$\omega = \sum_{l=1}^{l_{\max}} \Lambda_l \frac{xy^l}{1+xy^l} \, , \tag{11.18}$$

with $y = y(x)$ and $z = z(x)$ solutions of the coupled equations

$$y = \frac{\sum_{k=2}^{k_{\max}} \rho_k \; p_k^-(z)}{\sum_{k=2}^{k_{\max}} \rho_k \; p_k^+(z)}\,, \qquad z = \frac{\sum_{l=1}^{l_{\max}} \lambda_l x y^{l-1}/(1+xy^l)}{\sum_{l=1}^{l_{\max}} \lambda_l/(1+xy^l)\,,} \tag{11.19}$$

where we defined $p_k^{\pm}(z) \equiv \frac{(1+z)^{k-1} \pm (1-z)^{k-1}}{(1+z)^k + (1-z)^k}$.

Exercise 11.3 The numerical solution of Eqs. (11.18) and (11.19) can be quite tricky. Here is a simple iterative procedure which seems to work reasonably well (at least, in all the cases explored by the authors). The reader is invited to try it with her favorite degree distributions Λ, P.

First, solve Eq. (11.18) for x at given $y \in [0, \infty[$ and $\omega \in [0,1]$, using a bisection method. Next, substitute this value of x in Eq. (11.19), and write the resulting equations as $y = f(z)$ and $z = g(y, \omega)$. Define $F_\omega(y) \equiv f(g(y,\omega))$. Solve the equation $y = F_\omega(y)$ by iteration of the map $y_{n+1} = F_\omega(y_n)$ Once the fixed point y_* is found, the other parameters are computed as $z_* = g(y_*, \omega)$ and x_* is the solution of Eq. (11.18) for $y = y_*$. Finally x_*, y_*, z_* are substituted in Eq. (11.17) to obtain $\phi(\omega)$.

Examples of functions $\phi(\omega)$ are shown in Figures 11.2, 11.3, 11.4. We shall discuss these results in the next section, paying special attention to the region of small ω.

11.2.3 *Short distance properties*

In the low noise limit, the performance of a code depends a lot on the existence of codewords at short distance from the transmitted one. For linear codes and symmetric communication channels, we can assume without loss of generality that the all zeros codeword has been transmitted. Here we will work out the short distance (i.e. small weight ω) behavior of $\phi(\omega)$ for several LDPC ensembles. These properties will be used to characterize the code performances in Section 11.3.

As $\omega \to 0$, solving Eqs. (11.18) and (11.19) yields $y, z \to 0$. By Taylor expansion of these equations, we get

$$y \simeq \rho'(1) z\,, \qquad z \simeq \lambda_{l_{\min}} x y^{l_{\min}-1}\,, \qquad \omega \simeq \Lambda_{l_{\min}} x y^{l_{\min}}\,, \tag{11.20}$$

where we neglected higher order terms in y, z. At this point we must distinguish whether $l_{\min} = 1$, $l_{\min} = 2$ or $l_{\min} \geq 3$.

We start with the case $l_{\min} = 1$. Then x, y, z all scale like $\sqrt{\omega}$, and a short computation shows that

$$\phi(\omega) = -\frac{1}{2}\,\omega \, \log\left(\omega/\Lambda_1^2\right) + O(\omega)\,. \tag{11.21}$$

In particular $\phi(\omega)$ is strictly positive for ω sufficiently small. The expected number of codewords within a small (but $\Theta(N)$) Hamming distance from a given

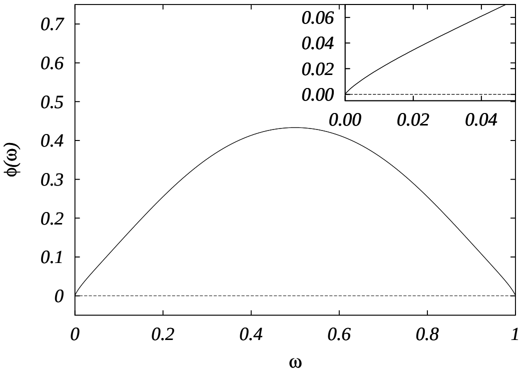


FIG. 11.2. Logarithm of the expected weight enumerator, $\phi(\omega)$, plotted versus the reduced weight $\omega = w/N$, for the ensemble $\text{LDPC}_N(\frac{1}{4}x + \frac{1}{4}x^2 + \frac{1}{2}x^3, x^6)$. Inset: small weight region. $\phi(\omega)$ is positive near to the origin, and in fact its derivative diverges as $\omega \to 0$: each codeword is surrounded by a large number of very close other codewords. This makes it a very bad error correcting code.

{fig:WEIRR1}

codeword is exponential in N. Furthermore, Eq. (11.21) is reminiscent of the behavior in absence of any parity check. In this case $\phi(\omega) = \mathcal{H}(\omega) \simeq -\omega \log \omega$.

Exercise 11.4 In order to check Eq. (11.21), compute the weight enumerator for the regular $\text{LDPC}_N(l = 1, k)$ ensemble. Notice that, in this case the weight enumerator does not depend on the code realization and admits the simple representation $\mathcal{N}(w) = \mathsf{coeff}[q_k(z)^{N/k}, z^w]$.

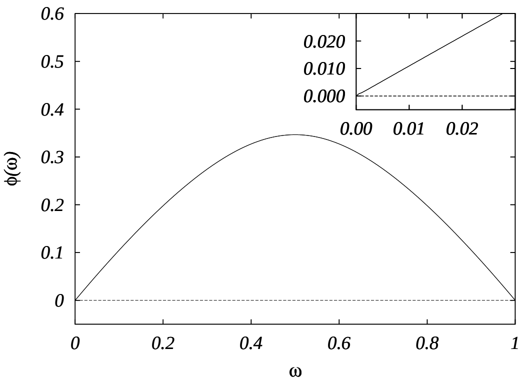
An example of weight enumerator for an irregular code with $l_{\min} = 1$ is shown in Fig. 11.2. The behavior (11.21) is quite bad for an error correcting code. In order to understand why, let us for a moment forget that this result was obtained by taking $\omega \to 0$ *after* $N \to \infty$, and apply it in the regime $N \to \infty$ at $w = N\omega$ fixed. We get

$$\overline{\mathcal{N}}(w) \sim \left(\frac{N}{w}\right)^{\frac{1}{2}w} . \tag{11.22}$$

It turns out that this result holds not only in average but for most codes in the ensemble. In other words, already at Hamming distance 2 from any given codeword there are $\Theta(N)$ other codewords. It is intuitively clear that discriminating between two codewords at $\Theta(1)$ Hamming distance, given a noisy observation, is in most of the cases impossible. Because of these remarks, one usually discards $l_{\min} = 1$ ensembles for error correcting purposes.

Consider now the case $l_{\min} = 2$. From Eq. (11.20), we get

$$\phi(\omega) \simeq A\omega\,, \qquad A \equiv \log\left[\frac{P''(1)}{P'(1)}\,\frac{2\Lambda_2}{\Lambda'(1)}\right] = \log\left[\rho'(1)\lambda'(0)\right] . \tag{11.23}$$



{fig:WE24} FIG. 11.3. Logarithm of the expected weight enumerator for the $\text{LDPC}_N(2,4)$ ensemble: $\Lambda(x) = x^2$, meaning that all variable nodes have degree 2, and $P(x) = 4$, meaning that all function nodes have degree 4. Inset: small weight region. The constant A is positive, so there exist codewords at short distances

The code ensemble has significantly different properties depending on the sign of A. If $A > 0$, the expected number of codewords within a small (but $\Theta(N)$) Hamming distance from any given codeword is exponential in the block-length. The situation seems similar to the $l_{\min} = 1$ case. Notice however that $\phi(\omega)$ goes much more quickly to 0 as $\omega \to 0$ in the present case. Assuming again that (11.23) holds beyond the asymptotic regime in which it was derived, we get

$$\overline{\mathcal{N}}(w) \sim e^{Aw} \,. \tag{11.24}$$

In other words the number of codewords around any particular one is $o(N)$ until we reach a Hamming distance $d_* \simeq \log N/A$. For many purposes d_* plays the role of an 'effective' minimum distance. The example of the regular code $\text{LDPC}_N(2,4)$, for which $A = \log 3$, is shown in Fig. 11.3

If on the other hand $A < 0$, then $\phi(\omega) < 0$ in some interval $\omega \in]0, \omega_*[$. The first moment method then shows that there are no codewords of weight 'close to' $N\omega$ for any ω in this range.

A similar conclusion is reached if $l_{\min} \geq 3$, where one finds:

$$\phi(\omega) \simeq \left(\frac{l_{\min} - 2}{2}\right) \omega \, \log\left(\frac{\omega}{\Lambda_{l_{\min}}}\right) \,, \tag{11.25}$$

An example of weight enumerator exponent for a code with good short distance properties, the $\text{LDPC}_N(3,6)$ code, is given in Fig. 11.4.

This discussion can be summarized as:

Proposition 11.1 *Consider a random linear code from the* $\text{LDPC}_N(\Lambda, P)$ *ensemble with* $l_{\min} \geq 2$ *and assume* $\frac{P''(1)}{P'(1)} \frac{2\Lambda_2}{\Lambda'(1)} < 1$. *Let* $\omega_* \in]0, 1/2[$ *be the first non-trivial zero of* $\phi(\omega)$, *and consider any interval* $[\omega_1, \omega_2] \subset]0, \omega_*[$. *With high*

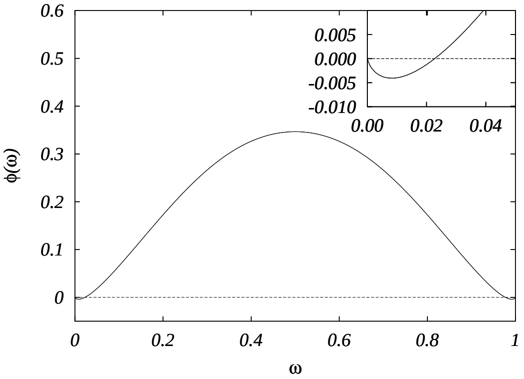


FIG. 11.4. Logarithm of the expected weight enumerator for the $\mathrm{LDPC}_N(3,6)$ ensemble. Inset: small weight region. $\phi(\omega) < 0$ for $\omega < \omega_* \sim .02$. There are no codewords except from the 'all-zeros' one in the region $\omega < \omega_*$.

{fig:WE36}

probability, there does not exist any pair of codewords with distance belonging to this interval.

Notice that our study only deals with weights $w = \omega N$ which grow linearly with N. The proposition excludes the existence of codewords of arbitrarily small ω, but it does not tell anything about possible codewords of sub-linear weight: $w = o(N)$ (for instance, with w finite as $N \to \infty$). It turns out that, if $l_{\min} \geq 3$, the code has with high probability no such codewords, and its minimum distance is at least $N\omega_*$. If on the other hand $l_{\min} = 2$, the code has typically codewords of finite weight. However (if $A < 0$), they can be eliminated without changing the code rate by an 'expurgation' procedure.

11.2.4 *Rate*

The weight enumerator can also be used to obtain a precise characterization of the rate of a $\mathrm{LDPC}_N(\Lambda, P)$ code. For $\omega = 1/2$, $x = y = z = 1$ satisfy Eqs. (11.18) and (11.19); this gives:

$$\phi(\omega = 1/2) = \left(1 - \frac{\Lambda'(1)}{P'(1)}\right) \log 2 = R_{\rm des} \log 2 \,. \qquad (11.26)$$

It turns out that, in most[29] of the cases of practical interest, the curve $\phi(\omega)$ has its maximum at $\omega = 1/2$ (see for instance the figures 11.2, 11.3, 11.4). In such cases the result (11.26) shows that the rate equals the design rate:

{prop:Rate}

Proposition 11.2 *Let R be the rate of a code from the* $\mathrm{LDPC}_N(\Lambda, P)$*ensemble, $R_{\rm des} = 1 - \Lambda'(1)/P'(1)$ the associated design rate and $\phi(\omega)$ the function defined in Eqs. (11.17) to (11.19). Assume that $\phi(\omega)$ achieves its absolute maximum*

[29] There exist exceptions though (see the Notes section for references).

over the interval $[0,1]$ *at* $\omega = 1/2$. *Then, for any* $\delta > 0$, *there exists a positive* N*-independent constant* $C_1(\delta)$ *such that*

$$\mathbb{P}\{|R - R_{\rm des}| \geq \delta\} \leq C_1(\delta)\, 2^{-N\delta/2}\,. \tag{11.27}$$

Proof: Since we already established that $R \geq R_{\rm des}$, we only need to prove an upper bound on R. The rate is defined as $R \equiv (\log_2 \mathcal{N})/N$, where $\mathcal{N}$ is the total number of codewords. Markov's inequality gives:

$$\mathbb{P}\{R \geq R_{\rm des} + \delta\} = \mathbb{P}\{\mathcal{N} \geq 2^{N(R_{\rm des}+\delta)}\} \leq 2^{-N(R_{\rm des}+\delta)}\, \mathbb{E}\mathcal{N}\,. \tag{11.28}$$

The expectation of the number of codewords is $\mathbb{E}\mathcal{N}(w) \doteq \exp\{N\phi(w/N)\}$, and there are only $N+1$ possible values of the weight w, therefore:

$$\mathbb{E}\mathcal{N} \doteq \exp\{N \sup_{\omega\in[0,1]} \phi(\omega)\}\,, \tag{11.29}$$

As $\sup\phi(\omega) = \phi(1/2) = R_{\rm des}\log 2$ by hypothesis, there exists a constant $C_1(\delta)$ such that, for any N, $\mathbb{E}\mathcal{N} \leq C_1(\delta) 2^{N(R_{\rm des}+\delta/2)}$ for any N. Plugging this into Eq. (11.28), we get

$$\mathbb{P}\{R \geq R_{\rm des} + \delta\} \leq C_1(\delta)\, 2^{N\delta/2}\,. \tag{11.30}$$

□

{se:BoundsLDPC}

11.3 Capacity of LDPC codes for the binary symmetric channel

Our study of the weight enumerator has shown that codes from the $\text{LDPC}_N(\Lambda, P)$ ensemble with $l_{\rm min} \geq 3$ have a good short distance behavior. The absence of codewords within an extensive distance $N\omega_*$ from the transmitted one, guarantees that any error (even introduced by an adversarial channel) changing a fraction of the bits smaller than $\omega_*/2$ can be corrected. Here we want to study the performance of these codes in correcting *typical* errors introduced from a given (probabilistic) channel. We will focus on the BSC(p) which flips each bit independently with probability $p < 1/2$. Supposing as usual that the all-zero codeword $\underline{x}^{(0)} = \underline{0}$ has been transmitted, let us call $\underline{y} = (y_1 \dots y_N)$ the received message. Its components are iid random variables taking value 0 with probability $1-p$, value 1 with probability p. The decoding strategy which minimizes the block error rate is word MAP decoding[30], which outputs the codeword closest to the channel output $\underline{y}$. As already mentioned, we don't bother about the practical implementation of this strategy and its computational complexity.

The block error probability for a code $\mathfrak{C}$, denoted by $\text{P}_\text{B}(\mathfrak{C})$, is the probability that there exists a 'wrong' codeword, distinct from $\underline{0}$, whose distance to $\underline{y}$ is smaller than $d(\underline{0}, \underline{y})$. Its expectation value over the code ensemble, $\text{P}_\text{B} = \mathbb{E}\,\text{P}_\text{B}(\mathfrak{C})$,

[30] Since all the codewords are *a priori* equiprobable, this coincides with maximum likelihood decoding.

is an important indicator of ensemble performances. We will show that in the large N limit, codes with $l_{\rm min} \geq 3$ undergo a phase transition, separating a low noise phase, $p < p_{\rm ML}$, in which the limit of $\mathrm{P_B}$ is zero, from a high noise phase, $p > p_{\rm ML}$, where it is not. While the computation of $p_{\rm ML}$ is deferred to Chapter ??, we derive here some rigorous bounds which indicate that some LDPC codes have very good (i.e. close to Shannon's bound) performances under ML decoding.

11.3.1 *Lower bound*

{se:LBLDPC}

We start by deriving a general bound on the block error probability $\mathrm{P_B}(\mathfrak{C})$ on the BSC(p) channel, valid for any linear code. Let $\mathcal{N} = 2^{NR}$ be the size of the codebook $\mathfrak{C}$. By union bound:

$$\begin{aligned} \mathrm{P_B}(\mathfrak{C}) &= \mathbb{P}\left\{\exists\, \alpha \neq 0 \quad \text{s.t.} \quad d(\underline{x}^{(\alpha)},\underline{y}) \leq d(\underline{\mathtt{0}},\underline{y})\right\} \\ &\leq \sum_{\alpha=1}^{\mathcal{N}-1} \mathbb{P}\left\{d(\underline{x}^{(\alpha)},\underline{y}) \leq d(\underline{\mathtt{0}},\underline{y})\right\}\,. \end{aligned} \tag{11.31}$$

As the components of $\underline{y}$ are iid Bernoulli variables, the probability $\mathbb{P}\{d(\underline{x}^{(\alpha)},\underline{y}) \leq d(\underline{\mathtt{0}},\underline{y})\}$ depends on $\underline{x}^{(\alpha)}$ only through its weight. Let $\underline{x}(w)$ be the vector formed by w ones followed by $N-w$ zeroes, and denote by $\mathcal{N}(w)$ the weight enumerator of the code $\mathfrak{C}$. Then

$$\mathrm{P_B}(\mathfrak{C}) \leq \sum_{w=1}^{N} \mathcal{N}(w)\; \mathbb{P}\left\{d(\underline{x}(w),\underline{y}) \leq d(\underline{\mathtt{0}},\underline{y})\right\}\,. \tag{11.32}$$

The probability $\mathbb{P}\left\{d(\underline{x}(w),\underline{y}) \leq d(\underline{\mathtt{0}},\underline{y})\right\}$ can be written as $\sum_u \binom{w}{u} p^u (1-p)^{w-u}\mathbb{I}(u \geq w/2)$, where u is the number of $y_i = \mathtt{1}$ in the first w components. A good bound is provided by a standard Chernov estimate. For any $\lambda > 0$:

$$\mathbb{P}\left\{d(\underline{x}(w),\underline{y}) \leq d(\underline{\mathtt{0}},\underline{y})\right\} \leq \mathbb{E} e^{\lambda[d(\underline{\mathtt{0}},\underline{y}) - d(\underline{x}(w),\underline{y})]} = [(1-p)\,e^{-\lambda} + p\,e^{\lambda}]^w\,.$$

The best bound is obtained for $\lambda = \frac{1}{2}\log(\frac{1-p}{p}) > 0$, and gives

$$\mathrm{P_B}(\mathfrak{C}) \leq \sum_{w=1}^{N} \mathcal{N}(w)\; e^{-\gamma w}\,. \tag{11.33}$$

where $\gamma \equiv -\log\sqrt{4p(1-p)} \geq 0$. The quantity $\sqrt{4p(1-p)}$ is sometimes referred to as **Bhattacharya parameter**.

Exercise 11.5 Consider the case of a general binary memoryless symmetric channel with transition probability $Q(y|x)$, $x \in \{\mathtt{0},\mathtt{1}\}$ $y \in \mathcal{Y} \subseteq \mathbb{R}$. First show that Eq. (11.31) remains valid if the Hamming distance $d(\underline{x},\underline{y})$ is replaced by the log-likelihood

$$d_Q(\underline{x}|\underline{y}) = -\sum_{i=1}^{N} \log Q(y_i|x_i)\,. \tag{11.34}$$

[Hint: remember the general expressions (6.3), (6.4) for the probability $P(\underline{x}|\underline{y})$ that the transmitted codeword was $\underline{x}$, given that the received message is $\underline{y}$]. Then repeat the derivation from Eq. (11.31) to Eq. (11.33). The final expression involves $\gamma = -\log B_Q$, where the Bhattacharya parameter is defined as $B_Q = \sum_y \sqrt{Q(y|\mathtt{1})Q(y|\mathtt{0})}$.

Equation (11.33) shows that the block error probability depends on two factors: one is the weight enumerator, the second one, $\exp(-\gamma w)$ is a channel-dependent term: as the weight of the codewords increases, their contribution is scaled down by an exponential factor because it is less likely that the received message $\underline{y}$ will be closer to a codeword of large weight than to the all-zero codeword.

So far the discussion is valid for any given code. Let us now consider the average over $\text{LDPC}_N(\Lambda, P)$ code ensembles. A direct averaging gives the bound:

$$\mathrm{P_B} \equiv \mathbb{E}_{\mathfrak{C}}\mathrm{P_B}(\mathfrak{C}) \le \sum_{w=1}^{N} \overline{\mathcal{N}}(w)\; e^{-\gamma w} \doteq \exp\left\{ N \sup_{\omega\in]0,1]} [\phi(\omega) - \gamma\omega] \right\}. \tag{11.35}$$

As such, this expression is useless, because the $\sup_\omega[\phi(\omega) - \gamma\omega]$, being larger or equal than the value at $\omega = 0$, is positive. However, if we restrict to codes with $l_{\min} \ge 3$, we know that, with probability going to one in the large N limit, there exists no wrong codeword in the ω interval $]0, \omega_*[$. In such cases, the maximization over ω in (11.35) can be performed in the interval $[\omega_*, 1]$ instead of $]0, 1]$. (By Markov inequality, this can be proved whenever $N \sum_{w=1}^{N\omega_*-1} \overline{\mathcal{N}}(w) \to 0$ as $N \to \infty$). The bound becomes useful whenever the supremum $\sup_{\omega\in[\omega_*,1]}[\phi(\omega)-\gamma\omega] < 0$: then $\mathrm{P_B}$ vanishes in the large N limit. We have thus obtained:

{propo:LDPCUnionBound}

Proposition 11.3 *Consider the average block error rate* $\mathrm{P_B}$ *for a random code in the* $\text{LDPC}_N(\Lambda, P)$ *ensemble, with* $l_{\min} \ge 3$*, used over a BSC*(p) *channel, with* $p < 1/2$. *Let* $\gamma \equiv -\log\sqrt{4p(1-p)}$ *and let* $\phi(\omega)$ *be the the weight enumerator exponent, defined in (11.4) [*$\phi(\omega)$ *can be computed using Eqs. (11.17), (11.18), and (11.19)]. If* $\phi(\omega) < \gamma\omega$ *for any* $\omega \in (0, 1]$ *such that* $\phi(\omega) \ge 0$*, then* $\mathrm{P_B} \to 0$ *in the large block-length limit.*

This result has a pleasing geometric interpretation which is illustrated in Fig. 11.5 for the $(3, 6)$ regular ensemble. As p increases from 0 to $1/2$, γ decreases

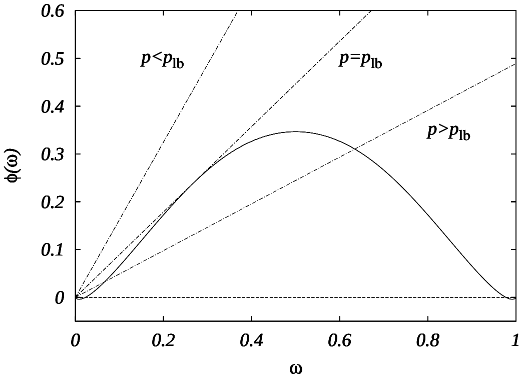


FIG. 11.5. Geometric construction yielding the lower bound on the threshold for reliable communication for the $\text{LDPC}_N(3,6)$ ensemble used over the binary symmetric channel. In this case $p_{\text{LB}} \approx 0.0438737$. The other two lines refer to $p = 0.01 < p_{\text{LB}}$ and $p = 0.10 > p_{\text{LB}}$. {fig:UnionBound36}

from $+\infty$ to 0. The condition $\phi(\omega) < \gamma\omega$ can be rephrased by saying that the weight enumerator exponent $\phi(\omega)$ must lie below the straight line of slope γ through the origin. Let us call p_{LB} the smallest value of p such that the line $\gamma\omega$ touches $\phi(\omega)$.

The geometric construction implies $p_{\text{LB}} > 0$. Furthermore, for p large enough Shannon's Theorem implies that P_{B} is bounded away from 0 for any non-vanishing rate $R > 0$. The **ML threshold** p_{ML} for the ensemble $\text{LDPC}_N(\Lambda, P)$ can be defined as the largest (or, more precisely, the supremum) value of p such that $\lim_{N\to\infty} \text{P}_{\text{B}} = 0$. This definition has a very concrete practical meaning: for any $p < p_{\text{ML}}$ one can communicate with an arbitrarily small error probability, by using a code from the $\text{LDPC}_N(\Lambda, P)$ ensemble provided N is large enough. Proposition 11.3 then implies:

$$p_{\text{ML}} \geq p_{\text{LB}} \,. \tag{11.36}$$

In general one expects $\lim_{N\to\infty} \text{P}_{\text{B}}$ to exist (and to be strictly positive) for $p > p_{\text{ML}}$. However, there exists no proof of this statement.

It is interesting to notice that, at $p = p_{\text{LB}}$, our upper bound on P_{B} is dominated by codewords of weight $w \approx N\tilde{\omega}$, where $\tilde{\omega} > 0$ is the value where $\phi(\omega) - \gamma\omega$ is maximum (which is larger than ω_*). This suggests that, each time an error occurs, a finite fraction of the bits are decoded incorrectly and this fraction fluctuates little from transmission to transmission (or, from code to code in the ensemble). The geometric construction also suggests the less obvious (but essentially correct) guess that this fraction jumps discontinuously from 0 to a finite value when p crosses the critical value p_{ML}.

Exercise 11.6 Let us study the case $l_{\min} = 2$. Proposition 11.3 is no longer valid, but we can still apply Eq. (11.35). (i) Consider the $(2,4)$ ensemble whose weight enumerator exponent is plotted in Fig. 11.3, the small weight behavior being given by Eq. (11.24). At small enough p, it is reasonable to assume that the block error rate is dominated by small weight codewords. Estimate $\mathrm{P_B}$ using Eq. (11.35) under this assumption. (ii) Show that the assumption breaks down for $p \geq p_{\rm loc}$, where $p_{\rm loc} \leq 1/2$ solves the equation $3\sqrt{4p(1-p)} = 1$. (iii) Discuss the case of a general code ensemble with $l_{\min} = 2$, and $\phi(\omega)$ concave for $\omega \in [0,1]$. (iv) Draw a weight enumerator exponent $\phi(\omega)$ such that the assumption of low-weight codewords dominance breaks down before $p_{\rm loc}$. (v) What do you expect of the average bit error rate $\mathrm{P_b}$ for $p < p_{\rm loc}$? And for $p > p_{\rm loc}$?

Exercise 11.7 Discuss the qualitative behavior of the block error rate for the cases where $l_{\min} = 1$.

{se:UBLDPC}

11.3.2 *Upper bound*

Let us consider as before the communication over a BSC(p), but restrict for simplicity to regular codes $\mathrm{LDPC}_N(l,k)$. Gallager has proved the following upper bound:

{thm:GallUB}

Theorem 11.4 *Let $p_{\rm ML}$ be the threshold for reliable communication over the binary symmetric channel using codes from the $\mathrm{LDPC}_N(l,k)$, with design rate $R_{\rm des} = 1 - k/l$. Then $p_{\rm ML} \leq p_{\rm UB}$, where $p_{\rm UB} \leq 1/2$ is the solution of*

$$\mathcal{H}(p) = (1 - R_{\rm des})\mathcal{H}\left(\frac{1-(1-2p)^k}{2}\right) , \tag{11.37}$$

We shall not give a full proof of this result, but we show in this section a sequence of heuristic arguments which can be turned into a proof. The details can be found in the original literature.

Assume that the all-zero codeword $\underline{0}$ has been transmitted and that a noisy vector $\underline{y}$ has been received. The receiver will look for a vector $\underline{x}$ at Hamming distance about Np from $\underline{y}$, and satisfying all the parity check equations. In other words, let us denote by $\underline{z} = \mathbb{H}\underline{x}$, $\underline{z} \in \{0,1\}^M$, (here $\mathbb{H}$ is the parity check matrix and multiplication is performed modulo 2), the **syndrome**. This is a vector with M components. If $\underline{x}$ is a codeword, all parity checks are satisfied, and we have $\underline{z} = \underline{0}$. There is at least one vector $\underline{x}$ fulfilling these conditions (namely $d(\underline{x},\underline{y}) \approx Np$, and $\underline{z} = \underline{0}$): the transmitted codeword $\underline{0}$. Decoding is successful only if it is the unique such vector.

The number of vectors $\underline{x}$ whose Hamming distance from $\underline{y}$ is close to Np is approximatively $2^{N\mathcal{H}(p)}$. Let us now estimate the number of distinct syndromes $\underline{z} = \mathbb{H}\underline{x}$, when $\underline{x}$ is on the sphere $d(\underline{x},\underline{y}) \approx Np$. Writing $\underline{x} = \underline{y} \oplus \underline{x}'$, this is equivalent to counting the number of distinct vectors $\underline{z}' = \mathbb{H}\underline{x}'$ when the weight

Table 11.1 *Bounds on the threshold for reliable communication over the BSC(p) channel using* $\mathrm{LDPC}_N(l,k)$ *ensembles. The third column is the rate of the code, the fourth and fifth columns are, respectively, the lower bound of Proposition 11.3 and the upper bound of Theorem 11.4. The sixth column is an improved lower bound by Gallager, and the last one is the Shannon limit.*

l	k	$R_{\rm des}$	LB of Sec. 11.3.1	Gallager UB	Gallager LB	Shannon limit
3	4	1/4	0.1333161	0.2109164	0.2050273	0.2145018
3	5	2/5	0.0704762	0.1397479	0.1298318	0.1461024
3	6	1/2	0.0438737	0.1024544	0.0914755	0.1100279
4	6	1/3	0.1642459	0.1726268	0.1709876	0.1739524
5	10	1/2	0.0448857	0.1091612	0.1081884	0.1100279

{TableLDPCBSC}

of $\underline{x}'$ is about Np. It is convenient to think of $\underline{x}'$ as a vector of N iid Bernoulli variables of mean p: we are then interested in the number of distinct *typical* vectors $\underline{z}'$. Notice that, since the code is regular, each entry z'_i is a Bernoulli variable of parameter

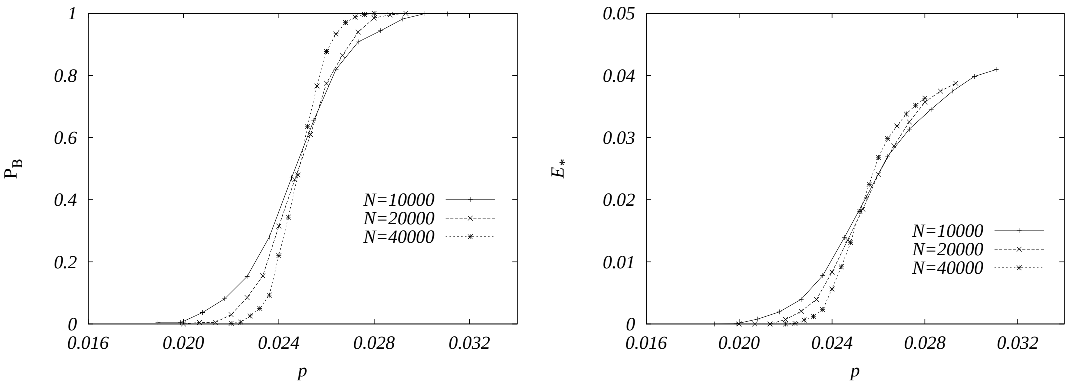
$$p_k = \sum_{n \text{ odd}}^{k} \binom{k}{n} p^n (1-p)^{k-n} = \frac{1-(1-2p)^k}{2} \, . \tag{11.38}$$

If the bits of $\underline{z}'$ were independent, the number of typical vectors $\underline{z}'$ would be $2^{N(1-R_{\rm des})\mathcal{H}(p_k)}$ (the dimension of $\underline{z}'$ being $M = N(1-R_{\rm des})$). It turns out that correlations between the bits decrease this number, so we can use the iid estimate to get an upper bound.

Let us now assume that for each $\underline{z}$ in this set, the number of reciprocal images (i.e. of vectors $\underline{x}$ such that $\underline{z} = \mathbb{H}\underline{x}$) is approximatively the same. If $2^{N\mathcal{H}(p)} \gg 2^{N(1-R_{\rm des})\mathcal{H}(p_k)}$, for each $\underline{z}$ there is an exponential number of vectors $\underline{x}$, such that $\underline{z} = \mathbb{H}\underline{x}$. This will be true, in particular, for $\underline{z} = \underline{0}$: the received message is therefore not uniquely decodable. In the alternative situation most of the vectors $\underline{z}$ correspond to (at most) a single $\underline{x}$. This will be the case for $\underline{z} = \underline{0}$: decoding can be successful.

11.3.3 *Summary of the bounds*

In Table 11.1 we consider a few regular $\mathrm{LDPC}_N(\Lambda, P)$ ensembles over the BSC(p) channel. We show the window of possible values of the noise threshold $p_{\rm ML}$, using the lower bound of Proposition 11.3 and the upper bound of Theorem 11.4. In most cases, the comparison is not satisfactory (the gap from capacity is close to a factor 2). A much smaller uncertainty is achieved using an improved lower bound again derived by Gallager, based on a refinement of the arguments in the previous Section. However, as we shall see in next Chapters, neither of the bounds is tight. Note that these codes get rather close to Shannon's limit, especially when k, l increase.



{fig:Flip510} Fig. 11.6. Performances of the bit-flipping decoding algorithm on random codes from the $(5, 10)$ regular LDPC ensemble, used over the BCS(p) channel. On the left: block error rate. On the right residual number of unsatisfied parity checks after the algorithm halted. Statistical error bars are smaller than symbols.

Exercise 11.8 Let $p_{\rm Sh}$ be the upper bound on $p_{\rm ML}$ provided by Shannon channel coding Theorem. Explicitly $p_{\rm Sh} \le 1/2$ is the solution of $\mathcal{H}(p) = 1 - R$. Prove that, if $R = R_{\rm des}$ (as is the case with high probability for $\text{LDPC}_N(l, k)$ ensembles) $p_{\rm UB} < p_{\rm Sh}$.

{se:BitFlippingLDPC}

11.4 A simple decoder: bit flipping

So far we have analyzed the behavior of LDPC ensembles under the optimal (ML) decoding strategy. However there is no known way of implementing this decoding with a fast algorithm. The naive algorithm goes through each codeword $\underline{x}^{(\alpha)}$, $\alpha = 0, \dots 2^{NR} - 1$ and finds the one of greatest likelihood $Q(\underline{y}|\underline{x}^{(\alpha)})$ (since all the codeword are *a priori* equiprobable, this is in fact the same as word MAP decoding). However this approach takes a time which grows exponentially with the block-length N. For large N (which is the regime where the error rate becomes close to optimal), this is unpractical.

LDPC codes are interesting because there exist fast sub-optimal decoding algorithms with performances close to the theoretical optimal performance, and therefore close to Shannon's limit. Here we show one example of a very simple decoding method, called the **bit flipping** algorithm. We have received the message $\underline{y}$ and try to find the sent codeword $\underline{x}$ by:

```
    Bit-flipping decoder
0.  Set $\underline{x}(0) = \underline{y}$.
1.  Find a bit belonging to more unsatisfied than satisfied parity checks.
2.  If such a bit exists, flip it: $x_i(t+1) = x_i(t) \oplus 1$. Keep the other bits:
    $x_j(t+1) = x_j(t)$ for all $j \neq i$. If there is no such bit, return $\underline{x}(t)$ and
    halt.
```

```
3. Repeat steps 2 and 3.
```

The bit to be flipped is usually chosen uniformly at random among the ones satisfying the condition at step 1. However this is irrelevant in the analysis below.

Exercise 11.9 Consider a code from the (l, k) regular LDPC ensemble (with $l \geq 3$). Assume that the received message differs from the transmitted one only in one position. Show that the bit-flipping algorithm always corrects such an error.

Exercise 11.10 Assume now that the channel has introduced two errors. Draw the factor graph of a regular (l, k) code for which the bit-flipping algorithm is unable to recover such an error event. What can you say of the probability of this type of graphs in the ensemble?

In order to monitor the bit-flipping algorithm, it is useful to introduce the 'energy':

$$E(t) \equiv \text{ Number of parity check equations not satisfied by } \underline{x}(t)\,. \quad (11.39)$$

This is a non-negative integer, and if $E(t) = 0$ the algorithm is halted and its output is $\underline{x}(t)$. Furthermore $E(t)$ cannot be larger than the number of parity checks M and decreases (by at least one) at each cycle. Therefore, the algorithm complexity is $O(N)$ (this is a commonly regarded as the ultimate goal for many communication problems).

It remains to be seen if the output of the bit-flipping algorithm is related to the transmitted codeword. In Fig. 11.6 we present the results of a numerical experiment. We considered the $(5, 10)$ regular ensemble and generated about 1000 random code and channel realizations for each value of the noise in some mesh. Then, we applied the above algorithm and traced the fraction of successfully decoded blocks, as well as the residual energy $E_* = E(t_*)$, where t_* is the total number of iterations of the algorithm. The data suggests that bit-flipping is able to overcome a finite noise level: it recovers the original message with high probability when less than about 2.5% of the bits are corrupted by the channel. Furthermore, the curves for $\mathrm{P}_\mathrm{B}^\mathrm{bf}$ under bit-flipping decoding become steeper and steeper as the system size is increased. It is natural to conjecture that asymptotically, a phase transition takes place at a well defined noise level p_bf: $\mathrm{P}_\mathrm{B}^\mathrm{bf} \to 0$ for $p < p_\mathrm{bf}$ and $\mathrm{P}_\mathrm{B}^\mathrm{bf} \to 1$ for $p > p_\mathrm{bf}$. Numerically $p_\mathrm{bf} = 0.025 \pm 0.005$.

This threshold can be compared with the one for ML decoding: The results in Table 11.1 imply $0.108188 \leq p_\mathrm{ML} \leq 0.109161$ for the $(5, 10)$ ensemble. Bit-flipping is significantly sub-optimal, but is still surprisingly good, given the extreme simplicity of the algorithm.

Can we provide any *guarantee* on the performances of the bit-flipping decoder? One possible approach consists in using the expansion properties of the underlying factor graph. Consider a graph from the (l, k) ensemble. We say that it is an (ε, δ)-**expander** if, for any set U of variable nodes such that $|U| \leq N\varepsilon$, the set $|D|$ of neighboring check nodes has size $|D| \geq \delta |U|$. Roughly speaking, if the factor graph is an expander with a large **expansion constant** δ, any small set of corrupted bits induces a large number of unsatisfied parity checks. The bit-flipping algorithm can exploit these checks to successfully correct the errors.

It turns out that random graphs are very good expanders. This can be understood as follows. Consider a fixed subset U. As long as U is small, the subgraph induced by U and the neighboring factor nodes D is a tree with high probability. If this is the case, elementary counting shows that $|D| = (l-1)|U| + 1$. This would suggest that one can achieve an expansion factor (close to) $l-1$, for small enough ε. Of course this argument have several flaws. First of all, the subgraph induced by U is a tree only if U has sub-linear size, but we are interested in all subsets U with $|U| \leq \varepsilon N$ for some fixed N. Then, while most of the small subsets U are trees, we need to be sure that *all* subsets expand well. Nevertheless, one can prove that the heuristic expansion factor is essentially correct:

Proposition 11.5 *Consider a random factor graph $\mathcal{F}$ from the (l,k) ensemble. Then, for any $\delta < l-1$, there exists a constant $\varepsilon = \varepsilon(\delta; l, k) > 0$, such that $\mathcal{F}$ is a (ε, δ) expander with probability approaching 1 as $N \to \infty$.*

In particular, this implies that, for $l \geq 5$, a random (l,k) regular factor graph is, with high probability a $(\varepsilon, \frac{3}{4}\, l)$ expander. In fact, this is enough to assure that the code will perform well at low noise level:

Theorem 11.6 *Consider a regular (l,k) LDPC code $\mathfrak{C}$, and assume that the corresponding factor graph is an $(\varepsilon, \frac{3}{4}\, l)$ expander. Then, the bit-flipping algorithm is able to correct any pattern of less then $N\varepsilon/2$ errors produced by a binary symmetric channel. In particular $\mathrm{P}_{\mathrm{B}}(\mathfrak{C}) \to 0$ for communication over a BSC(p) with $p < \varepsilon/2$.*

Proof: As usual, we assume the channel input to be the all-zeros codeword $\underline{\mathtt{0}}$. We denote by $w = w(t)$ the weight of $\underline{x}(t)$ (the current configuration of the bit-flipping algorithm), and by $E = E(t)$ the number of unsatisfied parity checks, as in Eq. (11.39). Finally, we call F the number of *satisfied* parity checks among the ones which are neighbors of at least one corrupted bit in $\underline{x}(t)$ (a bit is 'corrupted' if it takes value $\mathtt{1}$).

Assume first that $0 < w(t) \leq N\varepsilon$ at some time t. Because of the expansion property of the factor graph, we have $E + F > \frac{3}{4}\, l\, w$. On the other hand, every unsatisfied parity check is the neighbor of at least one corrupted bit, and every satisfied check which is the neighbor of some corrupted bit must involve at least two of them. Therefore $E + 2F \leq \, l\, w$. Eliminating F from the above inequalities, we deduce that $E(t) > \frac{1}{2}\, l\, w(t)$. Let $E_i(t)$ be the number of unsatisfied checks involving bit x_i. Then:

$$\sum_{i: x_i(t) = \mathtt{1}} E_i(t) \geq E(t) > \frac{1}{2}\, l\, w(t)\,. \tag{11.40}$$

Therefore, there must be at least one bit having more unsatisfied than satisfied neighbors, and the algorithm does not halt.

Let us now start the algorithm with $w(0) \leq N\varepsilon/2$. It must halt at some time t_*, either with $E(t_*) = w(t_*) = 0$ (and therefore decoding is successful), or with $w(t_*) \geq N\varepsilon$. In this second case, as the weight of $\underline{x}(t)$ changes by one at each step, we have $w(t_*) = N\varepsilon$. The above inequalities imply $E(t_*) > Nl\varepsilon/2$ and $E(0) \leq lw(0) \leq Nl\varepsilon/2$. This contradicts the fact that $E(t)$ is a strictly decreasing function of t. Therefore the algorithm, started with $w(0) \leq N\varepsilon/2$ ends up in the $w = 0$, $E = 0$ state. □

The approach based on expansion of the graph has the virtue of pointing out one important mechanism for the good performance of LDPC codes, namely the local tree-like structure of the factor graph. It also provides explicit lower bounds on the critical noise level $p_{\rm bf}$ for bit-flipping. However, these bounds turn out to be quite pessimistic. For instance, in the case of the $(5, 10)$ ensemble, it has been proved that a typical factor graph is an $(\varepsilon, \frac{3}{4}l) = (\varepsilon, \frac{15}{4})$ expander for $\varepsilon < \varepsilon_* \approx 10^{-12}$. On the other hand, numerical simulations, cf. Fig. 11.6, show that the bit flipping algorithm performs well up noise levels much larger than $\varepsilon_*/2$.

Notes

Modern (post-Cook Theorem) complexity theory was first applied to coding by (Berlekamp, McEliecee and van Tilborg, 1978) who showed that maximum likelihood decoding of linear codes is NP-hard.

LDPC codes were first introduced by Gallager in his Ph.D. thesis (Gallager, 1963; Gallager, 1962), which is indeed older than these complexity results. See also (Gallager, 1968) for an extensive account of earlier results. An excellent detailed account of modern developments is provided by (Richardson and Urbanke, 2006).

Gallager proposal did not receive enough consideration at the time. One possible explanation is the lack of computational power for simulating large codes in the sixties. The rediscovery of LDPC codes in the nineties (MacKay, 1999), was (at least in part) a consequence of the invention of Turbo codes by (Berrou and Glavieux, 1996). Both these classes of codes were soon recognized to be prototypes of a larger family: codes on graphs.

The major technical advance after this rediscovery has been the introduction of irregular ensembles (Luby, Mitzenmacher, Shokrollahi, Spielman and Stemann, 1997; Luby, Mitzenmacher, Shokrollahi and Spielman, 1998). There exist no formal proof of the 'equivalence' (whatever this means) of the various ensembles in the large block-length limit. But as we will see in Chapter **??**, the main property that enters in the analysis of LDPC ensembles is the local tree-like structure of the factor graph as described in Sec. 9.5.1; and this property is rather robust with respect to a change of the ensemble.

Gallager (Gallager, 1963) was the first to compute the expected weight enumerator for regular ensembles, and to use it in order to bound the threshold for reliable communication. The general case ensembles was considered in (Litsyn and Shevelev, 2003; Burshtein and Miller, 2004; Di, Richardson and Urbanke, 2004). It turns out that the expected weight enumerator coincides with the typical one to leading exponential order for regular ensembles (in statistical physics jargon: the annealed computation coincides with the quenched one). This is not the case for irregular ensembles, as pointed out in (Di, Montanari and Urbanke, 2004).

Proposition 11.2 is essentially known since (Gallager, 1963). The formulation quoted here is from (Méasson, Montanari and Urbanke, 2005*a*). This paper contains some examples of 'exotic' LDPC ensembles such that the maximum of the expected weight enumerator is at weight $w = N\omega_*$, with $\omega_* \neq 1/2$.

A proof of the upper bound 11.4 can be found in (Gallager, 1963). For some recent refinements, see (Burshtein, Krivelevich, Litsyn and Miller, 2002).

Bit-flipping algorithms played an important role in the revival of LDPC codes, especially following the work of Sipser and Spielman (Sipser and Spielman, 1996). These authors focused on explicit code construction based on expander graph. They also provide bounds on the expansion of random $\text{LDPC}_N(l, k)$ codes. The lower bound on the expansion mentioned in Sec. 11.4 is taken from (Richardson and Urbanke, 2006).

12

SPIN GLASSES

{chap:MagneticSystems}

We have already encountered several examples of spin glasses in Chapters 2 and 8. Like most problems in equilibrium statistical physics, they can be formulated in the general framework of factor graphs. Spin glasses are disordered systems, whose magnetic properties are dominated by randomly placed impurities. The theory aims at describing the behavior of a typical sample of such materials. This motivates the definition and study of spin glass ensembles.

In this chapter we shall explore the glass phase of these models. It is not easy to define this phase and its distinctive properties, especially in terms of purely static quantities. We provide here some criteria which have proved effective so far. We also present a classification of the two types of spin glass transitions that have been encountered in exactly soluble 'mean field models'. In contrast to these soluble cases, it must be stressed that very little is known (let alone proven) for realistic models. Even the existence of a spin glass phase is not established rigorously in the last case.

We first discuss in Section 12.1 how Ising models and their generalizations can be formulated in terms of factor graphs, and introduce several ensembles of these models. Frustration is a crucial feature of spin glasses. In Section 12.2 we discuss it in conjunction with gauge transformations. This section also explains how to derive some exact results with the sole use of gauge transformations. Section 12.3 describes the spin glass phase and the main approaches to its characterization. Finally, the phase diagram of a spin glass model with several glassy phases is traced in Section 12.4.

12.1 Spin glasses and factor graphs

{se:magFG}

12.1.1 *Generalized Ising models*

Let us recall the main ingredients of magnetic systems with interacting Ising spins. The variables are N Ising spins $\underline{\sigma} = \{\sigma_1, \dots, \sigma_N\}$ taking values in $\{+1, -1\}$. These are jointly distributed according to Boltzmann law for the energy function:

$$E(\underline{\sigma}) = -\sum_{p=1}^{p_{\max}} \sum_{i_1 < \cdots < i_p} J_{i_1 \dots i_p} \sigma_{i_1} \cdots \sigma_{i_p} \ . \tag{12.1}$$

{eq:GeneralMagnetic}

The index p gives the order of the interaction. One body terms ($p = 1$) are also referred to as external field interactions, and will be sometimes written as $-B_i \sigma_i$. If $J_{i_1 \dots i_p} \geq 0$, for any $i_1 \dots i_p$, and $p \geq 2$, the model is said to be a ferromagnet. If $J_{i_1 \dots i_p} \leq 0$, it is an **anti-ferromagnet**. Finally, if both positive and negative couplings are present for $p \geq 2$, the model is a spin glass.

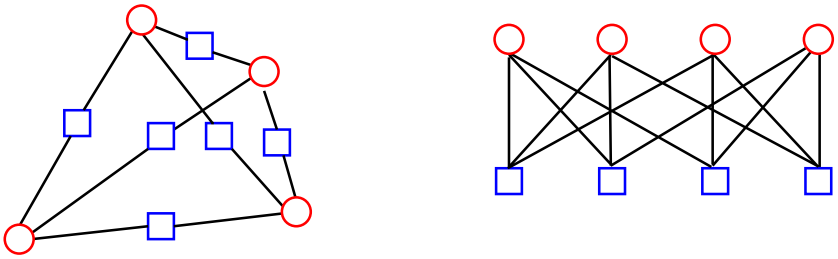


FIG. 12.1. Factor graph representation of the SK model with $N = 4$ (left), and the fully-connected 3-spin model with $N = 4$ (right). The squares denote the interactions between the spins.

{Fig:ising_fg}

The energy function can be rewritten as $E(\underline{\sigma}) = \sum_a E_a(\underline{\sigma}_{\partial a})$, where $E_a(\underline{\sigma}_{\partial a}) \equiv -J_a \sigma_{i_1^a} \cdots \sigma_{i_{p_a}^a}$. Each interaction term a involves the spins contained in a subset $\underline{\sigma}_{\partial a} = \{\sigma_{i_1^a}, \dots, \sigma_{i_{p_a}^a}\}$, of size p_a. We then introduce a factor graph in which each interaction term is represented by a square vertex and each spin is represented by a circular vertex. Edges are drawn between the interaction vertex a and the variable vertex i whenever the spin σ_i appears in $\underline{\sigma}_{\partial a}$. We have already seen in Fig. 9.7 the factor graph of a 'usual' two-dimensional spin glass, where the energy contains terms with $p = 1$ and $p = 2$. Figure 12.1.1 shows the factor graphs of some small samples of the SK model in zero magnetic field ($p = 2$ only) and the 3-spin model.

The energy function (12.1) can be straightforwardly interpreted as a model for a magnetic system. We used so far the language inherited from this application: the spins $\{\sigma_i\}$ are 'rotational' degrees of freedom associated to magnetic particle, their average is the magnetization etc. In this context, the most relevant interaction between distinct degrees of freedom is pairwise: $-J_{ij}\sigma_i\sigma_j$.

Higher order terms naturally arise in other applications, one of the simplest one being lattice particle systems. These are used to model the liquid-to-gas, liquid-to-solid, and similar phase transitions. One normally starts by considering some base graph $\mathcal{G}$ over N vertices, which is often taken to be a portion of $\mathbb{Z}^d$ (to model a real physical system the dimension of choice is of course $d = 3$). Each vertex in the graph can be either occupied by a particle, which we shall assume indistinguishable from the others, or empty. The particles are assumed indistinguishable from each other, and a configuration is characterized by occupation variables $n_i = \{0, 1\}$. The energy is a function $E(\underline{n})$ of the occupancies $\underline{n} = \{n_1, \dots, n_N\}$, which takes into account local interaction among neighboring particles. Usually it can be rewritten in the form (12.1), with an N independent $p_{\max}$ using the mapping $\sigma_i = 1 - 2n_i$. We give a few examples in the exercises below.

Exercise 12.1 Consider an empty box which is free to exchange particles with a reservoir, and assume that particles do not interact with each other (except for the fact that they cannot superimpose). This can be modeled by taking $\mathcal{G}$ to be a cube of side L in $\mathbb{Z}^d$, and establishing that each particle in the system contributes by a constant amount $-\mu$ to the energy: $E(\underline{n}) = -\mu \sum_i n_i$. This is a model for what is usually called an **ideal gas**.

Compute the partition function. Rewrite the energy function in terms of spin variables and draw the corresponding factor graph.

Exercise 12.2 In the same problem, imagine that particles attract each other at short distance: whenever two neighboring vertices i and j are occupied, the system gains an energy $-\epsilon$. This is a model for the liquid-gas phase transition.

Write the corresponding energy function both in terms of occupancy variables $\{n_i\}$ and spin variables $\{\sigma_i\}$. Draw the corresponding factor graph. Based on the phase diagram of the Ising model, cf. Sec. 2.5, discuss the behavior of this particle system. What physical quantity corresponds to the magnetization of the Ising model?

Exercise 12.3 In some system molecules cannot be packed in a regular lattice at high density, and this may result in amorphous solid materials. In order to model this phenomenon, one may modify the energy function of the previous Exercises as follows. Each time that a particle (i.e. an occupied vertex) is surrounded by more than k other particles in the neighboring vertices, a penalty $+\delta$ is added to the energy.

Write the corresponding energy function (both in terms of $\{n_i\}$ and $\{\sigma_i\}$) and draw the factor graph associated with it.

12.1.2 *Spin glass ensembles*

{se:SGensembles}

A sample (or an instance) of a spin glass is defined by:

- Its factor graph, which specifies the subsets of spins which interact;
- The value of the coupling constant $J_a \in \mathbb{R}$ for each function node in the factor graph.

An ensemble is defined by a probability distribution over the space of samples. In all cases which we shall consider here, the couplings are assumed to be iid random variables, independent of the factor graph. The most studied cases are Gaussian J_a's, or J_a taking values $\{+1, -1\}$ with equal probability (in jargon this is called the $\pm J$ model). More generally, we shall denote by $\mathcal{P}(J)$ the pdf of J_a.

One can distinguish two large families of spin glass ensembles which have attracted the attention of physicists: 'realistic' and 'mean field' ones. While in the first case the focus is on modeling actual physical systems, one hopes that mean field models can be treated analytically, and that this understanding offers some clues of the physical behavior of real materials.

Physical spin glasses are real three-dimensional (or, in some cases, two-dimensional) systems. The main feature of realistic ensembles is that they retain this geometric structure: a position x in d dimensions can be associated with each spin. The interaction strength (the absolute value of the coupling J) decays rapidly with the distance among the positions of the associated spins. The Edwards-Anderson model is a prototype (and arguably the most studied example) of this family. The spins are located on the vertices of a d-dimensional hyper-cubic lattice. Neighboring spins interact, through two-body interactions (i.e. $p_{\max} = 2$ in Eq. (12.1)). The corresponding factor graph is therefore non-random: we refer to Fig. 9.7 for an example with $d = 2$. The only source of disorder are the random couplings J_{ij} distributed according to $\mathcal{P}(J)$. It is customary to add a uniform magnetic field (i.e. a $p = 1$ term with J_i non-random). Very little is known about these models when $d \geq 2$, and most of our knowledge comes from numerical simulations. They suggest the existence of a glass phase when $d \geq 3$ but this is not proven yet.

There exists no general mathematical definition of mean field models. Fundamentally, they are models in which one expects to be able obtain exact expressions for the asymptotic ($N \to \infty$) free energy density, by optimizing some sort of large deviation rate function (in N). The distinctive feature allowing for a solution in this form, is the lack of any finite-dimensional geometrical structure.

The p-spin glass model discussed in Sec. 8.2 (and in particular the $p = 2$ case, which is the SK model) is a mean field model. Also in this case the factor graph is non-random, and the disorder enters only in the random couplings. The factor graph is a regular bipartite graph. It contains $\binom{N}{p}$ function nodes, one for each p-uple of spins; for this reason it is called **fully connected**. Each function node has degree p, each variable node has degree $\binom{N-1}{p-1}$. Since the degree diverges with N, the coupling distribution $\mathcal{P}(J)$ must be scaled appropriately with N, cf. Eq. (8.25).

Fully connected models are among the best understood in the mean field family. They can be studied either via the replica method, as in Chapter 8, or via the cavity method that we shall develop in the next Chapters. Some of the predictions from these two heuristic approaches have been confirmed rigorously.

One unrealistic feature of fully connected models is that each spin interacts with a diverging number of other spins (the degree of a spin variable in the factor graph diverges in the thermodynamic limit). In order to eliminate this feature, one can study spin glass models on Erdös-Rényi random graphs with finite average degree. Spins are associated with vertices in the graph and $p = 2$ interactions (with couplings that are iid random variables drawn from $\mathcal{P}(J)$) are associated with edges in the graph. The generalization to p-spin interactions is immediate. The corresponding spin glass models will be named **diluted spin glasses (DSG)**. We define the ensemble $\mathsf{DSG}_N(p, M, \mathcal{P})$ as follows:

- Generate a factor graph from the $\mathbb{G}_N(p, M)$ ensemble;

- For every function node a in the graph, connecting spins $i_1^a, \dots, i_p^a$, draw a random coupling $J_{i_1^a,\dots,i_p^a}$ from the distribution $\mathcal{P}(J)$, and introduce an energy term;
$$E_a(\underline{\sigma}_{\partial a}) = -J_{i_1^a,\dots,i_p^a}\sigma_{i_1^a}\cdots\sigma_{i_p^a} \;; \tag{12.2}$$
- The final energy is $E(\underline{\sigma}) = \sum_{a=1}^{M} E_a(\underline{\sigma}_{\partial a})$.

The thermodynamic limit is taken by letting $N \to \infty$ at fixed $\alpha = M/N$.

As in the case of random graphs, one can introduce some variants of this definition. In the ensemble $\mathsf{DSG}(p, \alpha, \mathcal{P})$, the factor graph is drawn from $\mathbb{G}_N(p, \alpha)$: each p-uple of variable nodes is connected by a function node independently with probability $\alpha/\binom{N}{p}$. As we shall see, the ensembles $\mathsf{DSG}_N(p, M, \mathcal{P})$ and $\mathsf{DSG}_N(p, \alpha, P)$ have the same free energy per spin in the thermodynamic limit (as well as several other thermodynamic properties in common). One basic reason of this phenomenon is that any finite neighborhood (in the sense of Sec. 9.5.1) of a random site i has the same asymptotic distribution in the two ensembles.

Obviously, any ensemble of random graphs can be turned into an ensemble of spin glasses by the same procedure. Some of these ensembles have been considered in the literature. Mimicking the notation defined in Section 9.2, we shall introduce general diluted spin glasses with constrained degree profiles, to be denoted by $\mathsf{DSG}_N(\Lambda, P, \mathcal{P})$, as the ensemble derived from the random graphs in $\mathbb{D}_N(\Lambda, P)$.

Diluted spin glasses are a very interesting class of systems, which are intimately related to sparse graph codes and to random satisfiability problems, among others. Our understanding of DSGs is intermediate between fully connected models and realistic ones. It is believed that both the replica and cavity methods allow to compute exactly many thermodynamic properties for most of these models. However the number of these exact results is still rather small, and only a fraction of these have been proved rigorously.

12.2 Spin glasses: Constraints and frustration

{se:SGgauge}

Spin glasses at zero temperature can be seen as constraint satisfaction problems. Consider for instance a model with two-body interactions

$$E(\underline{\sigma}) = -\sum_{(i,j)\in\mathcal{E}} J_{ij}\sigma_i\sigma_j \;, \tag{12.3}$$

{eq:ESGdef}

where the sum is over the edge set $\mathcal{E}$ of a graph $\mathcal{G}$ (the corresponding factor graph is obtained by associating a function node a to each edge $(ij) \in \mathcal{E}$). At zero temperature the Boltzmann distribution is concentrated on those configurations which minimize the energy. Each edge (i,j) induces therefore a constraint between the spins σ_i and σ_j: they should be aligned if $J_{ij} > 0$, or anti-aligned if $J_{ij} < 0$. If there exists a spin configuration which satisfies all the constraint, the ground state energy is $E_{\rm gs} = -\sum_{(i,j)\in\mathcal{E}} |J_{ij}|$ and the sample is said to be **unfrustrated** (see Chapter 2.6). Otherwise it is frustrated: a ground state is a spin configuration which violates the minimum possible number of constraints.

As shown in the Exercise below, there are several methods to check whether an energy function of the form (12.3) is frustrated.

Exercise 12.4 Define a 'plaquette' of the graph as a circuit $i_1, i_2, \dots, i_L, i_1$ such that no shortcut exists: $\forall r, s \in \{1, \dots, L\}$, the edge (i_r, i_s) is absent from the graph whenever $r \neq s \pm 1 \pmod{L}$. Show that a spin glass sample is unfrustrated if and only if the product of the couplings along every plaquette of the graph is positive.

Exercise 12.5 Consider a spin glass of the form (12.3), and define the Boolean variables $x_i = (1 - \sigma_i)/2$. Show that the spin glass constraint satisfaction problem can be transformed into an instance of the 2-satisfiability problem. [Hint: Write the constraint $J_{ij}\sigma_i\sigma_j > 0$ in Boolean form using x_i and x_j.]

Since 2-SAT is in P, and because of the equivalence explained in the last exercise, one can check in polynomial time whether the energy function (12.3) is frustrated or not. This approach becomes inefficient to $p \geq 3$ because K-SAT is NP-complete for $K \geq 3$. However, as we shall see in Chapter **??**, checking whether a spin glass energy function is frustrated remains a polynomial problem for any p.

{se:gauge_sg}

12.2.1 *Gauge transformation*

When a spin glass sample has some negative couplings but is unfrustrated, one is in fact dealing with a 'disguised ferromagnet'. By this we mean that, through a change of variables, the problem of computing the partition function for such a system can be reduced to the one of computing the partition function of a ferromagnet. Indeed, by assumption, there exists a ground state spin configuration σ_i^* such that $\forall (i,j) \in \mathcal{E} \;\; J_{ij}\sigma_i^*\sigma_j^* > 0$. Given a configuration $\underline{\sigma}$, define $\tau_i = \sigma_i\sigma_i^*$, and notice that $\tau_i \in \{+1, -1\}$. Then the energy of the configuration is $E(\underline{\sigma}) = E_*(\underline{\tau}) \equiv -\sum_{(i,j)\in\mathcal{E}} |J_{ij}|\tau_i\tau_j$. Obviously the partition function for the system with energy function $E_*(\,\cdot\,)$ (which is a ferromagnet since $|J_{ij}| > 0$) is the same as for the original system.

Such a change of variables is an example of a **gauge transformation**. In general, such a transformation amounts to changing all spins and simultaneously all couplings according to:

{eq:gauge_sg}

$$\sigma_i \mapsto \sigma_i^{(\underline{s})} = \sigma_i s_i \;\;,\;\;\; J_{ij} \mapsto J_{ij}^{(\underline{s})} = J_{ij} s_i s_j \;, \tag{12.4}$$

where $\underline{s} = \{s_1, \dots, s_N\}$ is an arbitrary configuration in $\{-1, 1\}^N$. If we regard the partition function as a function of the coupling constants $\underline{J} = \{J_{ij} \;:\; (ij) \in \mathcal{E}\}$:

{eq:gaugeZdef}

$$Z[\underline{J}] = \sum_{\{\sigma_i\}} \exp\left(\beta \sum_{(ij)\in\mathcal{E}} J_{ij}\sigma_i\sigma_j\right) \;, \tag{12.5}$$

then we have

$$Z[\underline{J}] = Z[\underline{J}^{(\underline{s})}] \,. \tag{12.6}$$

The system with coupling constants $\underline{J}^{(\underline{s})}$ is sometimes called the 'gauge transformed system'.

Exercise 12.6 Consider adding a uniform magnetic field (i.e. a linear term of the form $-B\sum_i \sigma_i$) to the energy function (12.3), and apply a generic gauge transformation to such a system. How must the uniform magnetic field be changed in order to keep the partition function unchanged? Is the new magnetic field term still uniform?

Exercise 12.7 Generalize the above discussion of frustration and gauge transformations to the $\pm J$ 3-spin glass (i.e. a model of the type (12.1) involving only terms with $p = 3$).

12.2.2 *The Nishimori temperature...*

{se:Nishimori}

In many spin glass ensembles, there exists a special temperature (called the **Nishimori temperature**) at which some thermodynamic quantities, such as the internal energy, can be computed exactly. This nice property is particularly useful in the study of inference problems (a particular instance being symbol MAP decoding of error correcting codes), since the Nishimori temperature naturally arises in these context. There are in fact two ways of deriving it: either as an application of gauge transformations (this is how it was discovered in physics), or by mapping the system onto an inference problem.

Let us begin by taking the first point of view. Consider, for the sake of simplicity, the model (12.3). The underlying graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be arbitrary, but we assume that the couplings J_{ij} on all the edges $(ij) \in \mathcal{E}$ are iid random variables taking values $J_{ij} = +1$ with probability $1 - p$ and $J_{ij} = -1$ with probability p. We denote by $\mathbb{E}$ the expectation with respect to this distribution.

The Nishimori temperature for this system is given by $T_{\rm N} = 1/\beta_{\rm N}$, where $\beta_{\rm N} = \frac{1}{2}\log\frac{(1-p)}{p}$. It is chosen in such a way that the coupling constant distribution $\mathcal{P}(J)$ satisfies the condition:

$$\mathcal{P}(J) = e^{-2\beta_N J}\,\mathcal{P}(-J) \,. \tag{12.7}$$

{eq:NishimoriCondition}

An equivalent way of stating the same condition consists in writing

$$\mathcal{P}(J) = \frac{e^{\beta_N J}}{2\cosh(\beta_N J)}\;\mathcal{Q}(|J|) \,. \tag{12.8}$$

{eq:gasgsym}

where $\mathcal{Q}(|J|)$ denotes the distribution of the absolute values of the couplings (in the present example, this is a Dirac's delta on $|J| = 1$).

Let us now turn to the computation of the average internal energy[31] $U \equiv \mathbb{E}\langle E(\underline{\sigma})\rangle$. More explicitly

$$U = \mathbb{E}\left\{ \frac{1}{Z[\underline{J}]} \sum_{\underline{\sigma}} \Big(-\sum_{(kl)} J_{kl}\sigma_k\sigma_l \Big)\, e^{\beta \sum_{(ij)} J_{ij}\sigma_i\sigma_j} \right\}, \tag{12.9}$$

{eq:gasgU}

In general, it is very difficult to compute U. It turns out that at the Nishimori temperature, the gauge invariance allows for an easy computation. The average internal energy U can be expressed as $U = \mathbb{E}\{Z_U[\underline{J}]/Z[\underline{J}]\}$, where $Z_U[\underline{J}] = -\sum_{\underline{\sigma}} \sum_{(kl)} J_{kl}\sigma_k\sigma_l \prod_{(ij)} e^{\beta_{\rm N} J_{ij}\sigma_i\sigma_j}$.

Let $\underline{s} \in \{-1,1\}^N$. By an obvious generalization of the principle (12.6), we have $Z_U[\underline{J}^{(\underline{s})}] = Z_U[\underline{J}]$, and therefore

q:InternalEnergyAvGauge}

$$U = 2^{-N} \sum_{\underline{s}} \mathbb{E}\{Z_U[\underline{J}^{(\underline{s})}]/Z[\underline{J}^{(\underline{s})}]\}\,. \tag{12.10}$$

If the coupling constants J_{ij} are iid with distribution (12.8), then the gauge transformed constants $J'_{ij} = J_{ij}^{(\underline{s})}$ are equally independent but with distribution

{eq:ChangeOfMeasure}

$$\mathcal{P}_{\underline{s}}(J_{ij}) = \frac{e^{\beta_{\rm N} J_{ij} s_i s_j}}{2\cosh\beta_{\rm N}}\,. \tag{12.11}$$

Equation (12.10) can therefore be written as $U = 2^{-N}\sum_{\underline{s}} \mathbb{E}_{\underline{s}}\{Z_U[\underline{J}]/Z[\underline{J}]\}$, where $\mathbb{E}_{\underline{s}}$ denotes expectation with respect to the modified measure $\mathcal{P}_{\underline{s}}(J_{ij})$. Using Eq. (12.11), and denoting by $\mathbb{E}_0$ the expectation with respect to the uniform measure over $J_{ij} \in \{\pm 1\}$, we get

$$U = 2^{-N} \sum_{\underline{s}} \mathbb{E}_0 \left\{ \prod_{(ij)} \frac{e^{\beta_{\rm N} J_{ij} s_i s_j}}{\cosh\beta_{\rm N}}\, \frac{Z_U[\underline{J}]}{Z[\underline{J}]} \right\} = \tag{12.12}$$

$$= 2^{-N} (\cosh\beta_{\rm N})^{-|\mathcal{E}|} \mathbb{E}_0 \left\{ \sum_{\underline{s}} e^{\beta_{\rm N}\sum_{(ij)} J_{ij} s_i s_j}\, \frac{Z_U[\underline{J}]}{Z[\underline{J}]} \right\} = \tag{12.13}$$

$$= 2^{-N} (\cosh\beta_{\rm N})^{-|\mathcal{E}|} \mathbb{E}_0 \left\{ Z_U[\underline{J}] \right\}\,. \tag{12.14}$$

It is easy to compute $\mathbb{E}_0 Z_U[\underline{J}] = -2^N (\cosh\beta_{\rm N})^{|\mathcal{E}|-1}\ \sinh\beta_{\rm N}$. This implies our final result for the average energy at the Nishimori temperature:

$$U = -|\mathcal{E}|\tanh(\beta_{\rm N})\,. \tag{12.15}$$

Notice that this simple result holds for any choice of the underlying graph. Furthermore, it is easy to generalize it to other choices of the coupling distribution satisfying Eq. (12.8) and to models with multi-spin interactions of the form (12.1). An even wider generalization is treated below.

[31] The same symbol U was used in Chapter 2 to denote the internal energy $\langle E(\underline{\sigma})\rangle$ (instead of its average). There should be no confusion with the present use.

12.2.3 *. . . and its relation with probability*

The calculation of the internal energy in the previous Section is straightforward but somehow mysterious. It is hard to grasp what is the fundamental reason that make things simpler at the Nishimori temperature. Here we discuss a more general derivation, in a slightly more abstract setting, which is related to the connection with inference mentioned above.

Consider the following process. A configuration $\underline{\sigma} \in \{\pm 1\}$ is chosen uniformly at random, we call $\mathbb{P}_0(\underline{\sigma})$ the corresponding distribution. Next a set of coupling constants $\underline{J} = \{J_a\}$ is chosen according to the conditional distribution

$$\mathbb{P}(\underline{J}|\underline{\sigma}) = e^{-\beta E_{\underline{J}}(\underline{\sigma})}\, \mathbb{Q}_0(\underline{J})\, . \tag{12.16}$$

Here $E_{\underline{J}}(\underline{\sigma})$ is an energy function with coupling constants $\underline{J}$, and $\mathbb{Q}_0(\underline{J})$ is some reference measure (that can be chosen in such a way that the resulting $\mathbb{P}(\underline{J}|\underline{\sigma})$ is normalized). This can be interpreted as a communication process. The information source produces the message $\underline{\sigma}$ uniformly at random, and the receiver observes the couplings $\underline{J}$.

The joint distribution of $\underline{J}$ and $\underline{\sigma}$ is $\mathbb{P}(\underline{J},\underline{\sigma}) = e^{-\beta E_{\underline{J}}(\underline{\sigma})}\, \mathbb{Q}_0(\underline{J})\mathbb{P}_0(\underline{\sigma})$ We shall denote expectation with respect to the joint distribution by Av in order to distinguish it from the thermal and quenched averages.

We assume that this process enjoys a gauge symmetry (this defines the Nishimori temperature in general). By this we mean that, given $\underline{s} \in \{\pm 1\}^N$, there exists an invertible mapping $\underline{J} \to \underline{J}^{(\underline{s})}$ such that $\mathbb{Q}_0(\underline{J}^{(\underline{s})}) = \mathbb{Q}_0(\underline{J})$ and $E_{\underline{J}^{(\underline{s})}}(\underline{\sigma}^{(\underline{s})}) = E_{\underline{J}}(\underline{\sigma})$. Then it is clear that the joint probability distribution of the coupling and the spins, and the conditional one, enjoy the same symmetry

$$\mathbb{P}(\underline{\sigma}^{(\underline{s})},\underline{J}^{(\underline{s})}) = \mathbb{P}(\underline{\sigma},\underline{J}) \;\; ; \;\; \mathbb{P}(\underline{J}^{(\underline{s})}|\underline{\sigma}^{(\underline{s})}) = \mathbb{P}(\underline{J}|\underline{\sigma})\, . \tag{12.17}$$

Let us introduce the quantity

$$\mathcal{U}(\underline{J}) = \mathrm{Av}(E_{\underline{J}}(\underline{\sigma})|\underline{J}) = \sum_{\underline{\sigma}} \mathbb{P}(\underline{\sigma}|\underline{J}) E_{\underline{J}}(\underline{\sigma})\, . \tag{12.18}$$

and denote by $U(\underline{\sigma}_0) = \sum_{\underline{J}} \mathbb{P}(\underline{J}|\underline{\sigma}_0)\mathcal{U}(\underline{J})$. This is nothing but the average internal energy for a disordered system with energy function $E_{\underline{J}}(\underline{\sigma})$ and coupling distribution $\mathbb{P}(\underline{J}|\underline{\sigma}_0)$. For instance, if we take $\underline{\sigma}_0$ as the 'all-plus' configuration, $\mathbb{Q}_0(\underline{J})$ proportional to the uniform measure over $\{\pm 1\}^{\mathcal{E}}$, and $E_{\underline{J}}(\underline{\sigma})$ as given by Eq. (12.3), then $U(\underline{\sigma}_0)$ is exactly the quantity U that we computed in the previous Section.

Gauge invariance implies that $\mathcal{U}(\underline{J}) = \mathcal{U}(\underline{J}^{(\underline{s})})$ for any $\underline{s}$, and $U(\underline{\sigma}_0)$ does not depend upon $\underline{\sigma}_0$. We can therefore compute $U = U(\underline{\sigma}_0)$ by averaging over $\underline{\sigma}_0$. We obtain

$$\begin{aligned} U &= \sum_{\underline{\sigma}_0} \mathbb{P}_0(\underline{\sigma}_0) \sum_{\underline{J}} \mathbb{P}(\underline{J}|\underline{\sigma}_0) \sum_{\underline{\sigma}} \mathbb{P}(\underline{\sigma}|\underline{J}) E_{\underline{J}}(\underline{\sigma}) \\ &= \sum_{\underline{\sigma},\underline{J}} \mathbb{P}(\underline{\sigma},\underline{J}) E_{\underline{J}}(\underline{\sigma}) = \sum_{\underline{J}} \mathbb{P}(\underline{J}|\underline{\sigma}_0) E_{\underline{J}}(\underline{\sigma})\, , \end{aligned} \tag{12.19}$$

where we used gauge invariance, once more, in the last step. The final expression is generally easy to evaluate since the coublings J_a are generically independent under $\mathbb{P}(\underline{J}|\underline{\sigma}_0)$ In particular, it is straightforward to recover Eq. (12.15) for the case treated in the last Section.

{ex:Nishimori_gen}

Exercise 12.8 Consider a spin glass model on an arbitrary graph, with energy given by (12.3), and iid random couplings on the edges, drawn from the distribution $\mathcal{P}(J) = \mathcal{P}_0(|J|)e^{aJ}$. Show that the Nishimori inverse temperature is $\beta_N = a$, and that the internal energy at this point is given by: $U = -|\mathcal{E}| \sum_J \mathcal{P}_0(|J|)\ J\ \sinh(\beta_N J)$. In the case where $\mathcal{P}$ is a Gaussian distribution of mean J_0, show that $U = -|\mathcal{E}|J_0$.

{se:SGphasedef}

12.3 What is a glass phase?

{sec:LocalMagnetization}

12.3.1 *Spontaneous local magnetizations*

In physics, a 'glass' is defined through its dynamical properties. For classical spin models such as the ones we are considering here, one can define several types of physically meaningful dynamics. For definiteness we use the single spin flip Glauber dynamics defined in Section 4.5, but the main features of our discussion should be robust with respect to this choice. Consider a system at equilibrium at time 0 (i.e., assume $\underline{\sigma}(0)$ to be distributed according to the Boltzmann distribution) and denote by $\langle \cdot \rangle_{\underline{\sigma}(0)}$ the expectation with respect to Glauber dynamics *conditional* to the initial configuration. Within a 'solid' [32] phase, spins are correlated with their initial value on long time scales:

$$\lim_{t\to\infty} \lim_{N\to\infty} \langle \sigma_i(t) \rangle_{\underline{\sigma}(0)} \equiv m_{i,\underline{\sigma}(0)} \neq \langle \sigma_i \rangle \,. \tag{12.20}$$

In other words, on arbitrary long but finite (in the system size) time scales, the system converges to a 'quasi-equilibrium' state (for brevity 'quasi-state') with local magnetizations $m_{i,\underline{\sigma}(0)}$ depending on the initial condition.

The condition (12.20) is for instance satisfied by a $d \geq 2$ Ising ferromagnet in zero external field, at temperatures below the ferromagnetic phase transition. In this case we have either $m_{i,\underline{\sigma}(0)} = M(\beta)$, or $m_{i,\underline{\sigma}(0)} = -M(\beta)$ depending on the initial condition (here $M(\beta)$ is the spontaneous magnetization of the system). There are two quasi-states, invariant by translation and related by a simple symmetry transformation. If the different quasi-states are not periodic, nor related by any such transformation, one may speak of a glass phase.

We shall discuss in greater detail the dynamical definition of quasi-states in Chapter **??**. It is however very important to characterize the glass phase at the level of equilibrium statistical mechanics, without introducing a specific dynamics. For the case of ferromagnets we have already seen the solution of this problem in Chapter 2. Let $\langle\,.\,\rangle_B$ denote expectation with respect to the

[32] The name comes from the fact that in a solid the preferred position of the atoms are time independent, for instance in a crystal they are the vertices of a periodic lattice

Boltzmann measure for the energy function (12.1), after a uniform magnetic field has been added. One then defines the two quasi-states by:

$$m_{i,\pm} \equiv \lim_{B\to 0\pm} \lim_{N\to\infty} \langle \sigma_i \rangle_B \, . \tag{12.21}$$

A natural generalization to glasses consists in adding a small magnetic field which is not uniform. Let us add to the energy function (12.1) a term of the form $-\epsilon \sum_i s_i \sigma_i$ where $\underline{s} \in \{\pm 1\}^N$ is an arbitrary configuration. Denote by $\langle \cdot \rangle_{\epsilon,\underline{s}}$ the expectation with respect to the corresponding Boltzmann distribution and let

$$m_{i,\underline{s}} \equiv \lim_{\epsilon\to 0\pm} \lim_{N\to\infty} \langle \sigma_i \rangle_{\epsilon,\underline{s}} \, . \tag{12.22}$$

The **Edwards-Anderson order parameter**, defined as

$$q_{\rm EA} \equiv \lim_{\epsilon\to 0\pm} \lim_{N\to\infty} \frac{1}{N} \sum_i \langle \sigma_i \rangle_{\epsilon,\underline{s}}^2 \, , \tag{12.23}$$

where $\underline{s}$ is an equilibrium configuration, then signals the onset of the spin glass phase.

The careful reader will notice that the Eq. (12.20) is not really completely defined: How should we take the $N \to \infty$ limit? Do the limits exist, how does the result depend on $\underline{\sigma}$? These are subtle questions. They underly the problem of defining properly the pure states (extremal Gibbs states) in disordered systems. In spite of many interesting efforts, there is no completely satisfactory definition of pure states in spin glasses.

Instead, all the operational definitions of the glass phase rely on the idea of comparing several equilibrated (i.e. drawn from the Boltzmann distribution) configurations of the system: one can then use one configuration as defining the direction of the polarizing field. This is probably the main idea underlying the success of the replica method. We shall explain below two distinct criteria, based on this idea, which can be used to define a glass phase. But we will first discuss a criterion of stability of the high temperature phase.

12.3.2 *Spin glass susceptibility*

{se:SGsusceptibility}

Take a spin glass sample, with energy (12.1), and add to it a local magnetic field on site i, B_i. The magnetic susceptibility of spin j with respect to the field B_i is defined as the rate of change of $m_j = \langle \sigma_j \rangle_{B_i}$ with respect to B_i:

$$\chi_{ji} \equiv \left. \frac{{\rm d} m_j}{{\rm d} B_i} \right|_{B_i=0} = \beta(\langle \sigma_i \sigma_j \rangle - \langle \sigma_i \rangle \langle \sigma_j \rangle) \, , \tag{12.24}$$

where we used the fluctuation dissipation relation (2.44).

The uniform (ferromagnetic) susceptibility defined in Sec. 2.5.1 gives the rate of change of the average magnetization with respect to an infinitesimal global uniform field: $\chi = \frac{1}{N} \sum_{i,j} \chi_{ji}$. Consider a ferromagnetic Ising model as introduced in Sec. 2.5. Within the ferromagnetic phase (i.e. at zero external field and below the critical temperature) χ diverges with the system size N. One way to understand this divergence is the following. If we denote by $m(B)$ the infinite volume magnetization in a magnetic field B, then

$$\chi = \lim_{B\to 0} \frac{1}{2B}[m(B) - m(-B)] = \lim_{B\to 0+} M/B = \infty\, , \tag{12.25}$$

within the ferromagnetic phase.

The above argument relates the susceptibility divergence with the existence of two distinct pure states of the system ('plus' and 'minus'). What is the appropriate susceptibility to detect a spin glass ordering? Following our previous discussion, we should consider the addition of a small non-uniform field $B_i = s_i\epsilon$. The local magnetizations are given by

$$\langle\sigma_i\rangle_{\epsilon,\underline{s}} = \langle\sigma_i\rangle_0 + \epsilon\sum_j \chi_{ij}s_j + O(\epsilon^2)\, . \tag{12.26}$$

As suggested by Eq. (12.25) we compare the local magnetization obtained by perturbing the system in two different directions $\underline{s}$ and $\underline{s}'$

$$\langle\sigma_i\rangle_{\epsilon,\underline{s}} - \langle\sigma_i\rangle_{\epsilon,\underline{s}'} = \epsilon\sum_j \chi_{ij}(s_j - s_j') + O(\epsilon^2)\, . \tag{12.27}$$

How should we choose $\underline{s}$ and $\underline{s}'$? A simple choice takes them independent and uniformly random in $\{\pm 1\}^N$; let us denote by $\mathbb{E}_s$ the expectation with respect to this distribution. The above difference becomes therefore a random variable with zero mean. Its second moment allows to define **spin glass susceptibility** (sometimes called **non-linear susceptibility**):

$$\chi_{\rm SG} \equiv \lim_{\epsilon\to 0}\frac{1}{2N\epsilon^2}\sum_i \mathbb{E}_s\left(\langle\sigma_i\rangle_{\epsilon,\underline{s}} - \langle\sigma_i\rangle_{\epsilon,\underline{s}'}\right)^2 \tag{12.28}$$

This is somehow the equivalent of Eq. (12.25) for the spin glass case. Using Eq. (12.27) one gets the expression $\chi_{\rm SG} = \frac{1}{N}\sum_{ij}(\chi_{ij})^2$, that is, thanks to the fluctuation dissipation relation

{eq:chiSGdef}

$$\chi_{\rm SG} = \frac{\beta^2}{N}\sum_{i,j}\left[\langle\sigma_i\sigma_j\rangle - \langle\sigma_i\rangle\langle\sigma_j\rangle\right]^2\, . \tag{12.29}$$

A necessary condition for the system to be in a 'normal' paramagnetic phase [33] is that $\chi_{\rm SG}$ remain finite when $N\to\infty$. We shall see below that this necessary condition of local stability is not always sufficient.

[33] One could construct models with 'exotic' paramagnetic phases, and a divergent spin glass susceptibility if (for instance) coupling distribution has infinite second moment. We disregard such situations.

Exercise 12.9 Another natural choice would consist in choosing $\underline{s}$ and $\underline{s}'$ as independent configurations drawn from Boltzmann's distribution. Show that with such a choice one would get $\chi_{\rm SG} = (1/N)\sum_{i,j,k} \chi_{ij}\chi_{jk}\chi_{ki}$. This susceptibility has not been studied in the literature, but it is reasonable to expect that it will lead generically to the same criterion of stability as the usual one (12.29).

12.3.3 *The overlap distribution function $P(q)$*

One of the main indicators of a glass phase is the overlap distribution, which we defined in Section 8.2.2, and discussed on some specific examples. Given a general magnetic model of the type (12.1), one generates two independent configurations $\underline{\sigma}$ and $\underline{\sigma}'$ from the associated Boltzmann distribution and consider their overlap $q(\underline{\sigma},\underline{\sigma}') = N^{-1}\sum_i \sigma_i\sigma_i'$. The overlap distribution $P(q)$ is the distribution of $q(\underline{\sigma},\underline{\sigma}')$ when the couplings and the underlying factor graph are taken randomly from their ensemble. Its moments are given by[34]:

$$\int P(q) q^r \, {\rm d}q = \mathbb{E}\Big\{\frac{1}{N^r}\sum_{i_1,\dots,i_r}\langle\sigma_{i_1}\dots\sigma_{i_r}\rangle^2\Big\} \, . \qquad (12.30)$$

In particular, the first moment $\int P(q)\, q \, {\rm d}q = N^{-1}\sum_i m_i^2$ is the expected overlap and the variance ${\rm Var}(q) \equiv \int P(q)\, q^2 \, {\rm d}q - \big[\int P(q)\, q\, {\rm d}q\big]^2$ is related to the spin glass susceptibility:

$${\rm Var}(q) = \mathbb{E}\Big\{\frac{1}{N^2}\sum_{i,j}\left[\langle\sigma_i\sigma_j\rangle - \langle\sigma_i\rangle\langle\sigma_j\rangle\right]^2\Big\} = \frac{1}{N}\chi_{\rm SG} \, . \qquad (12.31)$$

{eq:Pdeq2ndmom}

How is a glass phase detected through the behavior of the overlap distribution $P(q)$? We will discuss here some of the features emerging from the solution of mean field models. In the next Section we will see that the overlap distribution is in fact related to the idea, discussed in Section 12.3.1, of perturbing the system in order to explore its quasi-states.

Generically[35], at small β, a system of the type (12.1) is found in a 'paramagnetic' or 'liquid' phase. In this regime $P(q)$ concentrates as $N \to \infty$ on a single (deterministic) value $q(\beta)$. With high probability, two independent configurations $\underline{\sigma}$ and $\underline{\sigma}'$ have overlap $q(\beta)$. In fact, in such a phase, the spin glass $\chi_{\rm SG}$ susceptibility is finite, and the variance of $P(q)$ vanishes therefore as $1/N$.

For β larger than a critical value $\beta_{\rm c}$, the distribution $P(q)$ may acquire some structure, in the sense that several values of the overlap have non-zero probability

[34] Notice that, unlike in Section 8.2.2, we denote here by $P(q)$ the overlap distribution for a *finite* system of size N, instead of its $N \to \infty$ limit.

[35] This expression should be interpreted as 'in most model of interest studied until now' and subsumes a series of hypotheses. We assume, for instance, that the coupling distribution $\mathcal{P}(J)$ has finite second moment.

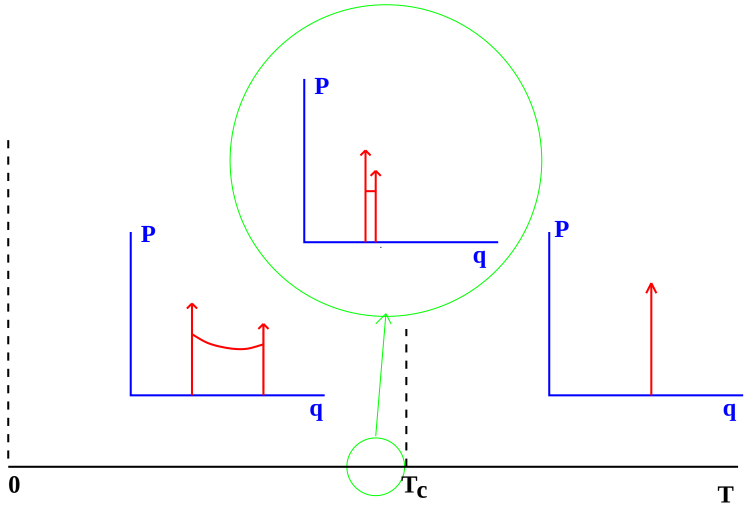


FIG. 12.2. Typical behavior of the order parameter $P(q)$ (overlap distribution at a continuous-FRSB glass transition. Vertical arrows denote Dirac's delta function.

{fig:pdeq_continu}

in the $N \to \infty$ limit. The temperature $T_c = 1/\beta_c$ is called the **static (or equilibrium) glass transition temperature**. For $\beta > \beta_c$ the system is in an equilibrium glass phase.

How does $P(q)$ look like at $\beta > \beta_c$? Let us focus here on its asymptotic ($N \to \infty$) limit. Generically, the transition falls into one of the following two categories, the names of which come from the corresponding replica symmetry breaking pattern found in the replica approach:

(*i*) **Continuous** ("Full replica symmetry breaking -FRSB") glass transition. In Fig. 12.2 we sketch the behavior of the thermodynamic limit of $P(q)$ in this case. The delta function present at $\beta < \beta_c$ 'broadens' for $\beta > \beta_c$, giving rise to a distribution with support in some interval $[q_0(\beta), q_1(\beta)]$. The width $q_1(\beta) - q_0(\beta)$ vanishes continuously when $\beta \downarrow \beta_c$. Furthermore, the asymptotic distribution has a continuous density which is strictly positive in $(q_0(\beta), q_1(\beta))$ and two discrete (delta) contributions at $q_0(\beta)$ and $q_1(\beta)$. This type of transition has a 'precursor'. If we consider the $N \to \infty$ limit of the spin glass susceptibility, this diverges as $\beta \uparrow \beta_c$. This phenomenon is quite important for identifying the critical temperature experimentally, numerically and analytically.

(*ii*) **Discontinuous** ("1RSB") glass transition. Again, the asymptotic limit of $P(q)$ acquires a non trivial structure in the glass phase, but the scenario is different. When β increases above β_c, the δ-peak at $q(\beta)$, which had unit mass at $\beta \leq \beta_c$, becomes a peak at $q_0(\beta)$, with a mass $1 - x(\beta) < 1$. Simultaneously, a second δ-peak appears at a value of the overlap $q_1(\beta) >$

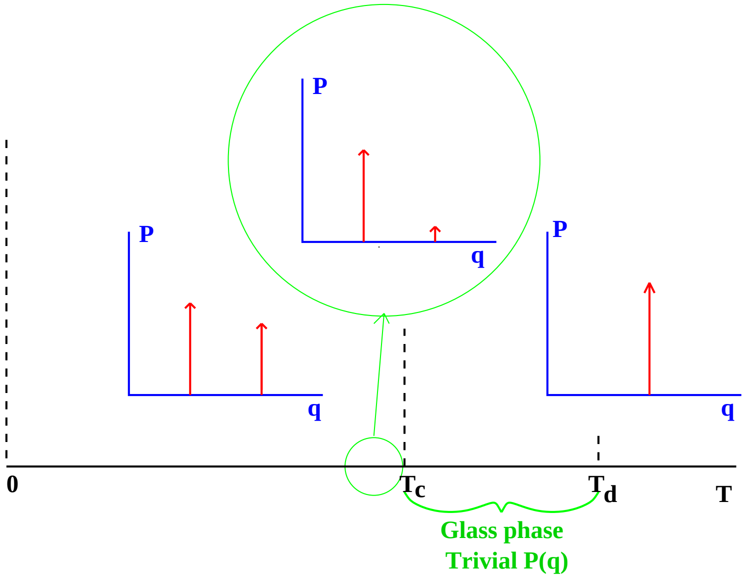


FIG. 12.3. Typical behavior of the order parameter $P(q)$ (overlap distribution) in a discontinuous-1RSB glass transition. Vertical arrows denote Dirac's delta function. {fig:pdeq_1step}

$q_0(\beta)$ with mass $x(\beta)$. As $\beta \downarrow \beta_c$, $q_0(\beta) \to q(\beta_c)$ and $x(\beta) \to 0$. Unlike in a continuous transition, the width $q_1(\beta) - q_0(\beta)$ does not vanish as $\beta \downarrow \beta_c$ and the open interval $]q_0(\beta), q_1(\beta)[$ has vanishing probability in the $N \to \infty$ limit. Furthermore, the thermodynamic limit of the spin glass susceptibility, χ_{SG} has a finite limit as $\beta \uparrow \beta_c$. This type of transition has no 'simple' precursor (but we shall describe below a more subtle indicator).

The two-peaks structure of $P(q)$ in a discontinuous transition has a particularly simple geometrical interpretation. When two configurations $\underline{\sigma}$ and $\underline{\sigma}'$ are chosen independently with the Boltzmann measure, their overlap is (with high probability) either approximately equal to q_0 or to q_1. In other words, their Hamming distance is either $N(1-q_1)/2$ or $N(1-q_0)/2$. This means that the Boltzmann measure $p(\underline{\sigma})$ is concentrated in some regions of the Hamming space (**clusters**). With high probability, two independent random configurations in the same cluster have distance (close to) $N(1-q_1)/2$, and two configurations in distinct clusters have distance (close to) $N(1-q_0)/2$. In other words, while the overlap does not concentrate in probability when $\underline{\sigma}$ and $\underline{\sigma}'$ are drawn from the Boltzmann measure, it does when this measure is restricted to one cluster. In a more formal (but still imprecise) way, we might write

$$p(\underline{\sigma}) \approx \sum_{\alpha} W_\alpha p_\alpha(\underline{\sigma}) \,, \tag{12.32}$$

where the $p_\alpha(\,\cdot\,)$ are probability distributions concentrated onto a single cluster, and W_α are the weights attributed by the Boltzmann distribution to each cluster.

According to this interpretation, $x(\beta) = \mathbb{E}\sum_\alpha W_\alpha^2$. Notice that, since $x(\beta) > 0$ for $\beta > \beta_{\rm c}$, the weights are sizeable only for a finite number of clusters (if there were R clusters, all with the same weight $W_\alpha = 1/R$, one would have $x(\beta) = 1/R$). This is what we found already in the REM, as well as in the replica solution of the completely connected p-spin model, cf. Sec. 8.2.

Generically, clusters exist already in some region of temperatures above $T_{\rm c}$, but the measure is not yet condensed on a finite number of them. In order to detect the existence of clusters in this intermediate temperature region, one needs some of the other tools described below.

There is no clear criterion that allows to distinguish *a priori* between systems undergoing one or the other type of transition. The experience gained on models solved via the replica or cavity methods indicated that a continuous transition typically occurs in standard spin glasses with $p = 2$-body interactions, but also, for instance, in the vertex-cover problem. A discontinuous transition is instead found in structural glasses, generalized spin glasses with $p \geq 3$, random satisfiability and coloring. To complicate things, both types of transitions may occur in the same system at different temperatures (or varying some other parameter). This may lead to a rich phase diagram with several glass phases of different nature.

It is natural to wonder whether gauge transformations may give some information on $P(q)$. Unfortunately, it turns out that the Nishimori temperature never enters a spin glass phase: the overlap distribution at $T_{\rm N}$ is concentrated on a single value, as suggested in the next exercise.

{ex:pdeqNishim}

Exercise 12.10 Using the gauge transformation of Sec. 12.2.1, show that, at the Nishimori temperature, the overlap distribution $P(q)$ is equal to the distribution of the magnetization per spin $m(\underline{\sigma}) \equiv N^{-1}\sum_i \sigma_i$. (In many spin glass models one expects that this distribution of magnetization per spin obeys a large deviation principle, and that it concentrates onto a single value as $N \to \infty$.)

12.3.4 *From the overlap distribution to the ϵ-coupling method*

The overlap distribution is in fact related to the idea of quasi-states introduced in Sec. 12.3.1. Let us again use a perturbation of the Boltzmann distribution which adds to the energy a magnetic field term $-\epsilon\sum_i s_i\sigma_i$, where $\underline{s} = (s_1, \dots, s_N)$ is a generic configuration. We introduce the ϵ-perturbed energy of a configuration $\underline{\sigma}$ as

{eq:PerturbedEnergy}

$$E_{\epsilon,\underline{s}}(\underline{\sigma}) = E(\underline{\sigma}) - \epsilon\sum_{i=1}^{N} s_i\sigma_i \ . \tag{12.33}$$

Is is important to realize that both the original energy $E(\underline{\sigma})$ and the new term $-\epsilon\sum_i s_i\sigma_i$ are extensive, i.e. they grow proportionally to N as $N \to \infty$. Therefore in this limit the presence of the perturbation can be relevant. The ϵ-perturbed Boltzmann measure is

$$p_{\epsilon,\underline{s}}(\underline{\sigma}) = \frac{1}{Z_{\epsilon,\underline{s}}}\, e^{-\beta E_{\epsilon,\underline{s}}(\underline{\sigma})}\ . \tag{12.34}$$

In order to quantify the effect of the perturbation, let us measure the expected distance between $\underline{\sigma}$ and $\underline{s}$

$$d(\underline{s},\epsilon) \equiv \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(1 - s_i\langle\sigma_i\rangle_{\underline{s},\epsilon}) \tag{12.35}$$

(notice that $\sum_i (1 - s_i\sigma_i)/2$ is just the number of positions in which $\underline{\sigma}$ and $\underline{s}$ differ). For $\epsilon > 0$ the coupling between $\underline{\sigma}$ and $\underline{s}$ is attractive, for $\epsilon < 0$ it is repulsive. In fact it is easy to show that $d(\underline{s},\epsilon)$ is a decreasing function of ϵ. ⋆

In the **ϵ-coupling method**, $\underline{s}$ is taken as a random variable, drawn from the (unperturbed) Boltzmann distribution. The rationale for this choice is that in this way $\underline{s}$ will point in the directions corresponding to quasi-states. The average distance induced by the ϵ-perturbation is then obtained, after averaging over $\underline{s}$ and over the choice of sample:

$$d(\epsilon) \equiv \mathbb{E}\Big\{\sum_{\underline{s}}\frac{1}{Z}\, e^{-\beta E(\underline{s})}\ d(\underline{s},\epsilon)\Big\}\,. \tag{12.36}$$

There are two important differences between the ϵ-coupling method computation of the overlap distribution $P(q)$: (i) When computing $P(q)$, the two copies of the system are treated on equal footing: they are independent and distributed according to the Boltzmann law. In the ϵ-coupling method, one of the copies is distributed according to Boltzmann law, while the other follows a perturbed distribution depending on the first one. (ii) In the ϵ-coupling method the $N \to \infty$ limit is taken *at fixed* ϵ. Therefore, the sum in Eq. (12.36) can be dominaded by values of the overlap $q(\underline{s},\underline{\sigma})$ which would have been exponentially unlikely for the original (unperturbed) measure. In the $N \to \infty$ limit of $P(q)$, such values of the overlap are given a vanishing weight. The two approaches provide complementary informations.

Within a paramagnetic phase $d(\epsilon)$ remains a smooth function of ϵ in the $N \to \infty$ limit: perturbing the system does not have any dramatic effect. But in a glass phase $d(\epsilon)$ becomes singular. Of particular interest are discontinuities at $\epsilon = 0$, that can be detected by defining

$$\Delta = \lim_{\epsilon\to 0+}\lim_{N\to\infty} d(\epsilon) - \lim_{\epsilon\to 0-}\lim_{N\to\infty} d(\epsilon)\,. \tag{12.37}$$

Notice that the limit $N \to \infty$ is taken first: for finite N there cannot be any discontinuity.

One expects Δ to be non-zero if and only if the system is in a 'solid' phase. One can think the process of adding a positive ϵ coupling and then letting it to

0 as a physical process. The system is first forced in an energetically favorable configuration (given by $\underline{s}$). The forcing is then gradually removed and one checks whether any memory of the preparation is retained ($\Delta > 0$), or, vice-versa, the system 'liquefies' ($\Delta = 0$).

The advantage of the ϵ-coupling method with respect to the overlap distribution $P(q)$ is twofold:

- In some cases the dominant contribution to the Boltzmann measure comes from several distinct clusters, but a single one dominates over the others. More precisely, it may happen that the weights for sub-dominant clusters scales as $W_\alpha = \exp[-\Theta(N^\theta)]$, with $\theta \in]0,1[$. In this case, the thermodynamic limit of $P(q)$ is a delta function and does not allow to distinguish from a purely paramagnetic phase. However, the ϵ-coupling method identifies the phase transition through a singularity of $d(\epsilon)$ at $\epsilon = 0$.
- One can use it to analyze a system undergoing a discontinuous transition, when it is in a glass phase but in the $T > T_\text{c}$ regime. In this case, the existence of clusters cannot be detected from $P(q)$ because the Boltzmann measure is spread among an exponential number of them. This situation will be the object of the next Section.

{se:1rsbqualit}

12.3.5 *Clustered phase of 1RSB systems and the potential*

The 1RSB equilibrium glass phase corresponds to a condensation of the measure on a small number of clusters of configurations. However, the most striking phenomenon is the appearance of clusters themselves. In the next Chapters we will argue that this has important consequences on Monte Carlo dynamics as well as on other algorithmic approaches to these systems. It turns out that the Boltzmann measure splits into clusters at a distinct temperature $T_\text{d} > T_\text{c}$. In the region of temperatures $[T_\text{c}, T_\text{d}]$ we will say that the system is in a **clustered phase** (or, sometimes, **dynamical glass phase**). The phase transition at T_d will be referred to as **clustering** or **dynamical transition**. In this regime, an exponential number of clusters $\mathcal{N} \doteq e^{N\Sigma}$ carry a roughly equal weight. The rate of growth Σ is called **complexity**[36] or **configurational entropy**.

The thermodynamic limit of the overlap distribution $P(q)$ does not show any signature of the clustered phase. In order to understand this point, it is useful to work out an toy example. Assume that the Boltzmann measure is entirely supported onto *exactly* $e^{N\Sigma}$ sets of configurations in $\{\pm 1\}^N$ (each set is a clusters), denoted by $\alpha = 1, \dots, e^{N\Sigma}$ and that the Boltzmann probability of each of these sets is $w = e^{-N\Sigma}$. Assume furthermore that, for any two configurations belonging to the same cluster $\underline{\sigma}, \underline{\sigma}' \in \alpha$, their overlap is $q(\underline{\sigma}, \underline{\sigma}') = q_1$, while if they belong to different clusters $\underline{\sigma} \in \alpha$, $\underline{\sigma}' \in \alpha'$, $\alpha \neq \alpha'$ their overlap is $q(\underline{\sigma}, \underline{\sigma}') = q_0 < q_1$. Although it might be actually difficult to construct such a measure, we shall neglect this for a moment, and compute the overlap distribution. The probability

[36] This use of the term 'complexity', which is customary in statistical physics, should not be confused with its use in theoretical computer science.

that two independent configurations fall in the same cluster is $e^{N\Sigma}w^2 = e^{-N\Sigma}$. Therefore, we have

$$P(q) = (1 - e^{-N\Sigma})\,\delta(q - q_0) + e^{-N\Sigma}\,\delta(q - q_1)\,, \tag{12.38}$$

which converges to $\delta(q - q_0)$ as $N \to \infty$: a single delta function as in the paramagnetic phase.

A first signature of the clustered phase is provided by the ϵ-coupling method described in the previous Section. The reason is very clear if we look at Eq. (12.33): the epsilon coupling 'tilts' the Boltzmann distribution in such a way that unlikely values of the overlap acquire a finite probability. It is easy to compute the thermodynamic limit $d_*(\epsilon) \equiv \lim_{N\to\infty} d(\epsilon)$. We get

$$d_*(\epsilon) = \begin{cases} (1 - q_0)/2 & \text{for } \epsilon < \epsilon_{\rm c}, \\ (1 - q_1)/2 & \text{for } \epsilon > \epsilon_{\rm c}, \end{cases} \tag{12.39}$$

where $\epsilon_{\rm c} = \Sigma/\beta(q_1 - q_0)$. As $T \downarrow T_{\rm c}$, clusters becomes less and less numerous and $\Sigma \to 0$. Correspondingly, $\epsilon_{\rm c} \downarrow 0$ as the equilibrium glass transition is approached.

The picture provided by this toy example is essentially correct, with the caveats that the properties of clusters will hold only within some accuracy and with high probability. Nevertheless, one expects $d_*(\epsilon)$ to have a discontinuity at some $\epsilon_{\rm c} > 0$ for all temperatures in an interval $]T_{\rm c}, T'_{\rm d}]$. Furthermore $\epsilon_{\rm c} \downarrow 0$ as $T \downarrow T_{\rm c}$.

In general, the temperature $T'_{\rm d}$ computed through the ϵ-coupling method does not coincide with the clustering transition. The reason is easily understood. As illustrated by the above example, we are estimating the exponentially small probability $\mathbb{P}(q|\underline{s}, \underline{J})$ that an equilibrated configuration $\underline{\sigma}$ has overlap q with the reference configuration $\underline{s}$, in a sample $\underline{J}$. In order to do this we compute the distance $d(\epsilon)$ which can be expressed by taking the expectation with respect to $\underline{s}$ and $\underline{J}$ of a rational function of $\mathbb{P}(q|\underline{s}, \underline{J})$. As shown several times since Chapter 5, exponentially small (or large) quantities, usually do not concentrate in probability, and $d(\epsilon)$ may be dominated by exponentially rare samples. We also learnt the cure for this problem: take logarithms! We therefore define[37] the **potential**

$$V(q) = -\lim_{N\to\infty} \frac{1}{N\beta} \mathbb{E}_{\underline{s},\underline{J}} \left\{\log \mathbb{P}(q|\underline{s}, \underline{J})\right\}\,. \tag{12.40}$$

Here (as in the ϵ-coupling method) the reference configuration is drawn from the Boltzmann distribution. In other words

$$\mathbb{E}_{\underline{s},\underline{J}}(\cdots) = \mathbb{E}_{\underline{J}}\Big\{\frac{1}{Z_{\underline{J}}} \sum_{\underline{s}} e^{-\beta E_{\underline{J}}(\underline{s})}(\cdots)\Big\}\,. \tag{12.41}$$

If, as expected, $\log \mathbb{P}(q|\underline{s}, \underline{J})$ concentrates in probability, one has $\mathbb{P}(q|\underline{s}, \underline{J}) \doteq e^{-NV(q)}$

[37] One should introduce a resolution, so that the overlap is actually constrained in some window around q. The width of this window can be let to 0 *after* $N \to \infty$.

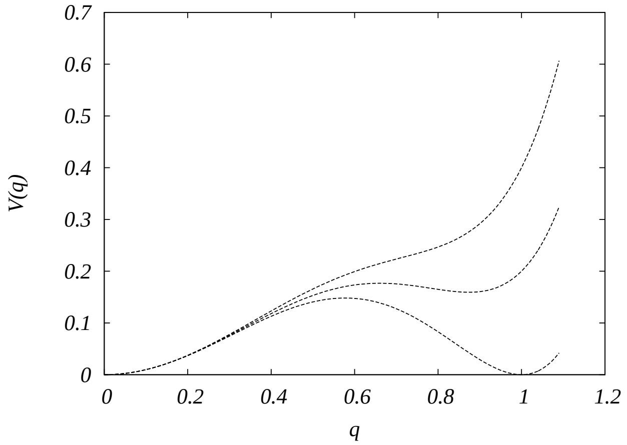


FIG. 12.4. Qualitative shapes of the potential $V(q)$ at various temperatures. When the temperature is very high (not shown) $V(q)$ is convex. Below $T = T_{\rm d}$, it develops a secondary minimum. The height difference between the two minima is $V(q_1) - V(q_0) = T\Sigma$. In the case shown here $q_0 = 0$ is independent of the temperature.

{fig:pot_qualit}

{exercise:RandomSigma}

Exercise 12.11 Consider the following refined version of the toy model (12.38): $\mathbb{P}(q|\underline{s},\underline{J}) = (1 - e^{-N\Sigma(\underline{s},\underline{J})})G_{q_0(\underline{s},\underline{J});b_0/N\beta}(q) + e^{-N\Sigma(\underline{s},\underline{J})})G_{q_1(\underline{s},\underline{J});b_1/N\beta}(q)$, where $G_{a,b}$ is a Gaussian distribution of mean a and variance b. We suppose that b_0, b_1 are constants, but $\Sigma(\underline{s},\underline{J}), q_0(\underline{s},\underline{J}), q_1(\underline{s},\underline{J})$ fluctuate as follows: when $\underline{J}$ and $\underline{s}$ are distributed according to the correct joint distribution (12.41), then $\Sigma(\underline{s},\underline{J}), q_0(\underline{s},\underline{J}), q_1(\underline{s},\underline{J})$ are independent Gaussian random variable of means respectively $\overline{\Sigma}, \overline{q}_0, \overline{q}_1$ and variances $\delta\Sigma^2/N, \delta q_0^2/N, \delta q_1^2/N$.

Assuming for simplicity that $\delta\Sigma^2 < 2\overline{\Sigma}$, compute $P(q)$ and $d(\epsilon)$ for this model. Show that the potential $V(q)$ is given by two arcs of parabolas:

$$V(q) = \min\left\{ \frac{(q-\overline{q}_0)^2}{2b_0} \;,\; \frac{(q-\overline{q}_1)^2}{2b_1} + \frac{1}{\beta}\,\overline{\Sigma} \right\} \qquad (12.42)$$

The potential $V(q)$ has been computed exactly, using the replica method, only in a small number of cases, mainly fully connected p-spin glasses. Here we shall just mention the qualitative behavior that is expected on the basis of these computations. The result is summarized in Fig. 12.4. At small enough β the potential is convex. Increasing β one first encounters a value β_* where $V(q)$ stops to be convex. When $\beta > \beta_{\rm d} = 1/T_{\rm d}$, $V(q)$ develops a secondary minimum, at $q = q_1(\beta) > q_0(\beta)$. This secondary minimum is in fact an indication of the existence of an exponential number of clusters, such that two configurations in the same cluster typically have overlap q_1, while two configurations in distinct clusters have overlap q_0. A little thought shows that the difference between the value of the potential at the two minima gives the complexity: $V(q_1) - V(q_0) = T\Sigma$.

In models in which the potential has been computed exactly, the temperature $T_{\rm d}$ computed in this way has been shown to coincide with a dramatic slowing down of the dynamics. More precisely, a properly defined relaxation time for Glauber-type dynamics is finite for $T > T_{\rm d}$ and diverges exponentially in the system size for $T < T_{\rm d}$.

12.3.6 *Cloning and the complexity function*

When the various clusters don't have all the same weight, the system is most appropriately described through a **complexity function**. Consider a cluster of configurations, called α. Its free energy F_α can be defined by restricting the partition function to configurations in cluster α. One way of imposing this restriction is to chose a reference configuration $\underline{\sigma}_0 \in \alpha$, and restricting the Boltzmann sum to those configurations $\underline{\sigma}$ whose distance from $\underline{\sigma}_0$ is smaller than $N\delta$. In order to correctly identify clusters, one has to take $(1-q_1)/2 < \delta < (1-q_0)/2$.

Let $\mathcal{N}_\beta(f)$ be the number of clusters such that $F_\alpha = Nf$ (more precisely, this is an un-normalized measure attributing unit weight to the points F_α/N). We expect it to satisfy a large deviations principle of the form

$$\mathcal{N}_\beta(f) \doteq \exp\{N\Sigma(\beta, f)\}\,. \tag{12.43}$$

The rate function $\Sigma(\beta, f)$ is the complexity function. If clusters are defined as above, with the cut-off δ in the appropriate interval, they are expected to be disjoint up to a subset of configurations of exponentially small Boltzmann weight. Therefore the total partition function is given by:

$$Z = \sum_\alpha e^{-\beta F_\alpha} \doteq \int e^{N[\Sigma(\beta,f)-\beta f]}\,{\rm d}f \doteq e^{N[\Sigma(\beta,f_*)-\beta f_*]}\,, \tag{12.44}$$

where we applied the saddle point method as in standard statistical mechanics calculations, cf. Sec. 2.4. Here $f_* = f_*(\beta)$ solves the saddle point equation $\partial\Sigma/\partial f = \beta$.

For several reasons, it is interesting to determine the full complexity function $\Sigma(\beta, f)$, as a function of f for a given inverse temperature β. The **cloning method** is a particularly efficient (although non-rigorous) way to do this computation. Here we sketch the basic idea: several applications will be discussed in the next Chapters. One begins by introducing m identical 'clones' of the initial system. These are non-interacting except for the fact that they are constrained to be in the same cluster. In practice one can constrain all their pairwise Hamming distances to be smaller than $N\delta$, where $(1-q_1)/2 < \delta < (1-q_0)/2$. The partition function for the m clones systems is therefore

$$Z_m = \sum_{\underline{\sigma}^{(1)},\dots,\underline{\sigma}^{(m)}}{}' \exp\Big\{ -\beta E(\underline{\sigma}^{(1)}) \cdots - \beta E(\underline{\sigma}^{(m)})\Big\} . \qquad (12.45)$$

where the prime reminds us that $\underline{\sigma}^{(1)}$, $\dots \underline{\sigma}^{(m)}$ stay in the same cluster. By splitting the sum over the various clusters we have

$$Z_m = \sum_{\alpha} \sum_{\underline{\sigma}^{(1)} \dots \underline{\sigma}^{(m)} \in \alpha} e^{-\beta E(\underline{\sigma}^{(1)}) \cdots - \beta E(\underline{\sigma}^{(m)})} = \sum_{\alpha} \Big(\sum_{\underline{\sigma}\in\alpha} e^{-\beta E(\underline{\sigma})}\Big)^m . \, (12.46)$$

At this point we can proceed as for the calculation of the usual partition function and obtain

{eq:SaddlePointCloned}

$$Z_m = \sum_{\alpha} e^{-\beta m F_\alpha} \doteq \int e^{N[\Sigma(\beta,f)-\beta m f]} \, \mathrm{d}f \doteq e^{N[\Sigma(\beta,\hat{f})-\beta m \hat{f}]} \, , \qquad (12.47)$$

where $\hat{f} = \hat{f}(\beta, m)$ solves the saddle point equation $\partial\Sigma/\partial f = \beta m$.

The free energy density per clone of the cloned system is defined as

$$\Phi(\beta, m) = - \lim_{N\to\infty} \frac{1}{\beta m N} \log Z_m \, . \qquad (12.48)$$

The saddle point estimate (12.47) implies that $\Phi(\beta, m)$ is related to $\Sigma(\beta, f)$ through a Legendre transform:

$$\Phi(\beta, m) = f - \frac{1}{\beta m}\Sigma(\beta, f) \;\; ; \;\; \frac{\partial \Sigma}{\partial f} = \beta m \, . \qquad (12.49)$$

If we forget that m is an integer, and admit that $\Phi(\beta, m)$ can be 'continued' to non-integer m, the complexity $\Sigma(\beta, f)$ can be computed from $\Phi(\beta, m)$ by inverting this Legendre transform[38].

[38] The similarity to the procedure used in the replica method is not fortuitous. Notice however that replicas are introduced to deal with quenched disorder, while cloning is more general

Exercise 12.12 In the REM, the natural definition of overlap between two configurations $i, j \in \{1, \dots, 2^N\}$ is $Q(i,j) = \delta_{ij}$. Taking a configuration j_0 as reference, the ϵ-perturbed energy of a configuration j is $E'(\epsilon, j) = E_j - N\epsilon\delta_{j,j_0}$. (Note the extra N multiplying ϵ, introduced in order to ensure that the new ϵ-coupling term is typically extensive).

(i) Consider the high temperature phase where $\beta < \beta_{\rm c} = 2\sqrt{\log 2}$. Show that the ϵ-perturbed system has a phase transition at $\epsilon = \frac{\log 2}{\beta} - \frac{\beta}{4}$.

(ii) In the low temperature phase $\beta > \beta_{\rm c}$, show that the phase transition takes place at $\epsilon = 0$.

Therefore in the REM the clusters exist at any β, and every cluster is reduced to one single configuration: one must have $\Sigma(\beta, f) = \log 2 - f^2$ independently of β. Show that this is compatible with the cloning approach, through a computation of the potential $\Phi(\beta, m)$:

$$\Phi(\beta, m) = \begin{cases} -\frac{\log 2}{\beta m} - \frac{\beta m}{4} & \text{for } m < \frac{\beta_c}{\beta} \\ -\sqrt{\log 2} & \text{for } m > \frac{\beta_c}{\beta} \end{cases} \tag{12.50}$$

12.4 An example: the phase diagram of the SK model

{sec:PhaseDiag}

Several mean field models have been solved using the replica method. Sometimes a model may present two or more glass phases with different properties. Determining the phase diagram can be particularly challenging in these cases.

A classical example is provided by the SK model with ferromagnetically biased couplings. As in the other examples of this Chapter, this is a model for N Ising spins $\underline{\sigma} = (\sigma_1, \dots, \sigma_N)$. The energy function is

$$E(\underline{\sigma}) = -\sum_{(i,j)} J_{ij}\sigma_i\sigma_j \ , \tag{12.51}$$

where (i,j) are un-ordered couples, and the couplings J_{ij} are iid Gaussian random variables with mean J_0/N and variance $1/N$. The model somehow interpolates between the Curie-Weiss model treated in Sec. 2.5.2, corresponding to $J_0 \to \infty$, and the unbiased Sherrington-Kirkpatrick model, considered in Chapter 8, for $J_0 = 0$.

The phase diagram is plotted in terms of two parameters: the ferromagnetic bias J_0, and the temperature T. Depending on their values, the system is found in one of four phases, cf. Fig. 12.5: paramagnetic (P), ferromagnetic (F), symmetric spin glass (SG) and mixed ferromagnetic spin glass (F-SG). A simple characterization of these four phases is obtained in terms of two quantities: the average magnetization and overlap. In order to define them, we must first observe that, since $E(\underline{\sigma}) = E(-\underline{\sigma})$, in the present model $\langle\sigma_i\rangle = 0$ identically for all values of J_0, and T. In order to break this symmetry, we may add a magnetic field term $-B\sum_i \sigma_i$ and let $B \to 0$ after the thermodynamic limit. We then define

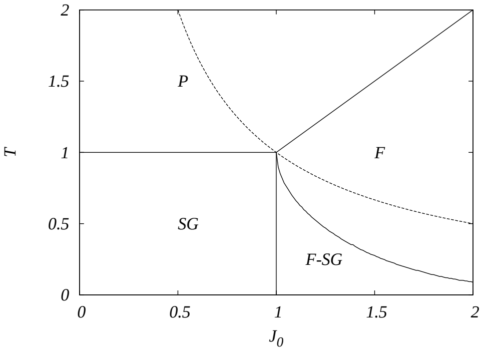


FIG. 12.5. Phase diagram of the SK model in zero magnetic field. When the temperature T and the ferromagnetic bias J_0 are varied, there exist four possible phases: paramagnetic (P), ferromagnetic (F), spin glass (SG) and mixed ferromagnetic-spin glass (F-SG). The full lines separate these various phases. The dashed line is the location of the Nishimori temperature.

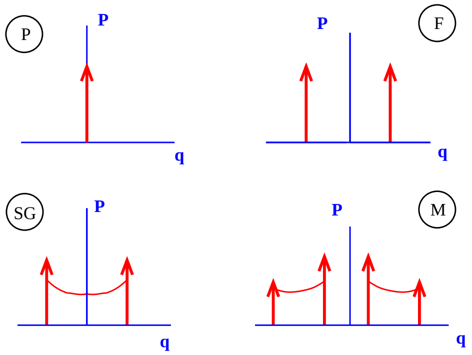
{fig:sk_phasediag}

$$m = \lim_{B\to 0+} \lim_{N\to\infty} \mathbb{E}\langle\sigma_i\rangle_B \,, \qquad \overline{q} = \lim_{B\to 0+} \lim_{N\to\infty} \mathbb{E}(\langle\sigma_i\rangle_B^2) \,, \tag{12.52}$$

(which don't depend on i because the coupling distribution is invariant under a permutation of the sites). In the P phase one has $m = 0, \overline{q} = 0$; in the SG phase $m = 0, \overline{q} > 0$, and in the F and F-SG phases one has $m > 0, \overline{q} > 0$.

A more complete description is obtained in terms of the overlap distribution $P(q)$. Because of the symmetry under spin inversion mentioned above, $P(q) = P(-q)$ identically. The qualitative shape of $P(q)$ in the thermodynamic limit is shown in Fig. 12.6. In the P phase it consists of a single δ function with unit weight at $q = 0$: two independent configurations drawn from the Boltzmann distribution have, with high probability, overlap close to 0. In the F phase, it is concentrated on two symmetric values $q(J_0, T) > 0$ and $-q(J_0, T) < 0$, each carrying weight one half. We can summarize this behavior by saying that a random configuration drawn from the Boltzmann distribution is found, with equal probability, in one of two different states. In the first one the local magnetizations are $\{m_i\}$, in the second one they are $\{-m_i\}$. If one draws two independent configurations, they fall in the same state (corresponding to the overlap value $q(J_0, T) = N^{-1}\sum_i m_i^2$) or in opposite states (overlap $-q(J_0, T)$) with probability 1/2. In the SG phase the support of $P(q)$ is a symmetric interval $[-q_{\rm max}, q_{\rm max}]$, with $q_{\rm max} = q_{\rm max}(J_0, T)$. Finally, in the F-SG phase the support is the union of two intervals $[-q_{\rm max}, -q_{\rm min}]$ and $[q_{\rm min}, q_{\rm max}]$. Both in the SG and F-SG phases, the presence of a whole range of overlap values carrying non-vanishing probability, suggests the existence of a multitude of quasi-states (in the sense discussed in the previous Section).

In order to remove the degeneracy due to the symmetry under spin inversion, one sometimes define an asymmetric overlap distribution by adding a magnetic



{fig:pdeq_SK} FIG. 12.6. The typical shape of the $P(q)$ function in each of the four phases of the SK model ferromagnetically biased couplings.

field terms, and taking the thermodynamic limit as in Eq. (12.52):

$$P_+(q) = \lim_{B\to 0+} \lim_{N\to\infty} P_B(q)\,. \tag{12.53}$$

Somewhat surprisingly, it turns out that $P_+(q) = 0$ for $q < 0$, while $P_+(q) = 2P(q)$ for $q > 0$. In other words $P_+(q)$ is equal to the distribution of the *absolute value* of the overlap.

Exercise 12.13 Consider the Curie-Weiss model in a magnetic field, cf. Sec. 2.5.2. Draw the phase diagram and compute the asymptotic overlap distribution. Discuss its qualitative features for different values of the temperature and magnetic field.

A few words for the reader interested in how one derives this diagram: Some of the phase boundaries were already derived using the replica method in Exercise 8.12. The boundary P-F is obtained by solving the RS equation (8.68) for q, μ, m. The P-SG and F-M lines are obtained by the AT stability condition (8.69). Deriving the phase boundary between the SG and F-SG phases is much more challenging, because it separates glassy phases, therefore it cannot be derived within the RS solution. It is known to be approximately vertical, but there is no simple expression for it. The Nishimori temperature is deduced from the condition (12.7): $T_{\rm N} = 1/J_0$, and the line $T = 1/J_0$ is usually called 'Nishimori line'. The internal energy per spin on this line is $U/N = -J_0/2$. Notice that the line does not enter any of the glass phases, as we know from general arguments.

An important aspect of the SK model is that the appearance of the glass phase on the lines separating P from SG on the one hand, and F from F-SG on the other hand is a continuous transition. Therefore it is associated with the divergence of the non-linear susceptibility $\chi_{\rm SG}$. The following exercise, reserved to the replica aficionados, sketches the main lines of the argument showing this.

Exercise 12.14 Let us see how to compute the non-linear susceptibility of the SK model, $\chi_{\rm SG} = \frac{\beta^2}{N}\sum_{i\neq j}\left(\langle\sigma_i\sigma_j\rangle - \langle\sigma_i\rangle\langle\sigma_j\rangle\right)^2$, with the replica method Show that:

$$\begin{aligned}\chi_{\rm SG} = \lim_{n\to 0}\frac{\beta^2}{N}\sum_{i\neq j}&\left(\binom{n}{2}^{-1}\sum_{(ab)}\langle\sigma_i^a\sigma_i^b\sigma_j^a\sigma_j^b\rangle - \binom{n}{3}^{-1}\sum_{(abc)}\langle\sigma_i^a\sigma_i^b\sigma_j^a\sigma_j^c\rangle\right.\\ &\left. + \binom{n}{4}^{-1}\sum_{(abcd)}\langle\sigma_i^a\sigma_i^b\sigma_j^c\sigma_j^d\rangle\right)\\ = N\lim_{n\to 0}&\int\prod_{(ab)}(dQ_{ab}d\lambda_{ab})e^{-NG(Q,\lambda)}A(Q)\ , \qquad (12.54)\end{aligned}$$

where we follow the notations of (8.30), the sum over $(a_1a_2\dots a_k)$ is understood to run over all the k-uples of distinct replica indices, and

$$A(Q)\equiv\binom{n}{2}^{-1}\sum_{(ab)}Q_{ab}^2 - \binom{n}{3}^{-1}\sum_{(abc)}Q_{ab}Q_{ac} + \binom{n}{4}^{-1}\sum_{(abcd)}Q_{ab}Q_{cd} \qquad (12.55)$$

Analyze the divergence of $\chi_{\rm SG}$ along the following lines: The leading contribution to (12.54) should come from the saddle point and be given, in the high temperature phase, by $A(Q_{ab}=q)$ where $Q_{ab}=q$ is the RS saddle point. However this contribution clearly vanishes when one takes the $n\to 0$ limit. One must thus consider the fluctuations around the saddle point. Each of the term like $Q_{ab}Q_{cd}$ in $A(Q)$ gives a factor $\frac{1}{N}$ time the appropriate matrix element of the inverse of the Hessian matrix. When this Hessian matrix is non-singular, these elements are all finite and one obtains a finite result (The $1/N$ cancels the factor N in (12.54)). But when one reaches the AT instability line, the elements of the inverse of the Hessian matrix diverge, and therefore $\chi_{\rm SG}$ also diverges.

Notes

A review on the simulations of the Edwards Anderson model can be found in (Marinari, Parisi and Ruiz-Lorenzo, 1997).

Mathematical results on mean field spin glasses are found in the book (Talagrand, 2003). A short recent survey is provided by (Guerra, 2005).

Diluted spin glasses were introduced in (Viana and Bray, 1988).

The implications of the gauge transformation were derived by Hidetoshi Nishimori and his coworkers, and are explained in details in his book (Nishimori, 2001).

The notion of pure states in phase transitions, and the decomposition of Gibbs measures into superposition of pure states, is discussed in the book (Georgii, 1988).

The divergence of the spin glass susceptibility is specially relevant because this susceptibility can be measured in zero field. The experiments of (Monod and Bouchiat, 1982) present evidence of a divergence, which support the existence of a finite spin glass transition in real (three dimensional) spin glasses in zero magnetic field.

The existence of two transition temperatures $T_c < T_d$ was first discussed by Kirkpatrick, Thirumalai and Wolynes (Kirkpatrick and Wolynes, 1987; Kirkpatrick and Thirumalai, 1987), who pointed out the relevance to the theory of structural glasses. In particular, (Kirkpatrick and Thirumalai, 1987) discusses the case of the p-spin glass. A review of this line of approach to structural glasses, and particularly its relevance to dynamical effects, is (Bouchaud, Cugliandolo, Kurchan and Mézard, 1997).

The ϵ-coupling method was introduced in (Caracciolo, Parisi, Patarnello and Sourlas, 1990). The idea of cloning in order to study the complexity function is due to Monasson (Monasson, 1995). The potential method was introduced in (Franz and Parisi, 1995).

13

BRIDGES

{ch:inference}

We have seen in the last three Chapters how some problems with very different origins can be cast into the unifying framework of factor graph representations. The underlying mathematical structure, namely the locality of probabilistic dependencies between variables, is also present in many problems of probabilistic inference, which provides another unifying view of the field. On the other hand, locality is an important ingredient that allows sampling from complex distributions using the Monte Carlo technique.

In Section 13.1 we present some basic terminology and simple examples of statistical inference problems. Statistical inference is an interesting field in itself with many important applications (ranging from artificial intelligence, to modeling and statistics). Here we emphasize the possibility of considering coding theory, statistical mechanics and combinatorial optimization, as inference problems.

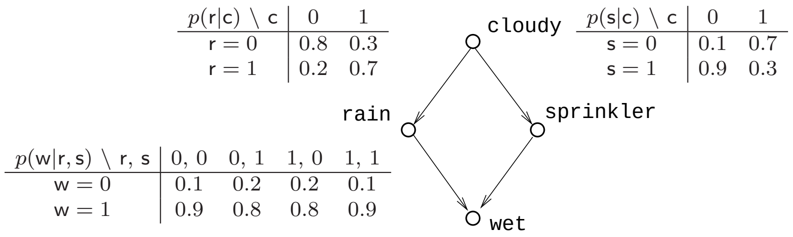
Section 13.2 develops a very general tool in all these problems, the Monte Carlo Markov Chain (MCMC) technique, already introduced in Sec. 4.5. This is often a very powerful approach. Furthermore, Monte Carlo sampling can be regarded as a statistical inference method, and the Monte Carlo dynamics is a simple prototype of the local search strategies introduced in Secs. 10.2.3 and 11.4. Many of the difficulties encountered in decoding, in constraint satisfaction problems, or in glassy phases, are connected to a dramatic slowing down of the MCMC dynamics. We present the results of simple numerical experiments on some examples, and identify regions in the phase diagram where the MCMC slowdown implies poor performances as a sampling/inference algorithm. Finally, in Section 13.3 we explain a rather general argument to estimate the amount of time MCMC has to be run in order to produce roughly independent samples with the desired distribution.

{sec:Inference}

13.1 Statistical inference

13.1.1 *Bayesian networks*

It is common practice in artificial intelligence and statistics, to formulate inference problems in terms of Bayesian networks. Although any such problem can also be represented in terms of a factor graph, it is worth to briefly introduce this alternative language. A famous toy example is the ‘rain–sprinkler’ network.



$p(\mathsf{r}\vert\mathsf{c}) \setminus \mathsf{c}$	0	1
$\mathsf{r}=0$	0.8	0.3
$\mathsf{r}=1$	0.2	0.7

$p(\mathsf{s}\vert\mathsf{c}) \setminus \mathsf{c}$	0	1
$\mathsf{s}=0$	0.1	0.7
$\mathsf{s}=1$	0.9	0.3

$p(\mathsf{w}\vert\mathsf{r},\mathsf{s}) \setminus \mathsf{r},\mathsf{s}$	0, 0	0, 1	1, 0	1, 1
$\mathsf{w}=0$	0.1	0.2	0.2	0.1
$\mathsf{w}=1$	0.9	0.8	0.8	0.9

{fig:SprinklerRain}

FIG. 13.1. The rain-sprinkler Bayesian network.

Example 13.1 During a walk to the park, a statistician notices that the grass is wet. There are two possible reasons for that: either it rained during the night, or the sprinkler was activated in the morning to irrigate the lawn. Both events are in turn correlated with the weather condition in the last 24 hours.

After a little thought, the statistician formalizes these considerations as the probabilistic model depicted in Fig. 13.1. The model includes four random variables: cloudy, rain, sprinkler, wet, taking values in $\{0, 1\}$ (respectively, false or true). The variables are organized as the vertices of an oriented graph. A directed edge corresponds intuitively to a relation of causality. The joint probability distribution of the four variables is given in terms of conditional probabilities associated to the edges. Explicitly (variables are indicated by their initials):

$$p(\mathsf{c},\mathsf{s},\mathsf{r},\mathsf{w}) = p(\mathsf{c})\, p(\mathsf{s}|\mathsf{c})\, p(\mathsf{r}|\mathsf{c})\, p(\mathsf{w}|\mathsf{s},\mathsf{r})\,. \tag{13.1}$$

The three conditional probabilities in this formula are given by the Tables in Fig. 13.1. A 'uniform prior' is assumed on the event that the day was cloudy: $p(\mathsf{c}=0) = p(\mathsf{c}=1) = 1/2$.

Assuming that wet grass was observed, we may want to know whether the most likely cause was the rain or the sprinkler. This amount to computing the marginal probabilities

$$p(\mathsf{s}|\mathsf{w}=1) = \frac{\sum_{\mathsf{c},\mathsf{r}} p(\mathsf{c},\mathsf{s},\mathsf{r},\mathsf{w}=1)}{\sum_{\mathsf{c},\mathsf{r},\mathsf{s}'} p(\mathsf{c},\mathsf{s}',\mathsf{r},\mathsf{w}=1)}\,, \tag{13.2}$$

$$p(\mathsf{r}|\mathsf{w}=1) = \frac{\sum_{\mathsf{c},\mathsf{s}} p(\mathsf{c},\mathsf{s},\mathsf{r},\mathsf{w}=1)}{\sum_{\mathsf{c},\mathsf{r},\mathsf{s}'} p(\mathsf{c},\mathsf{s}',\mathsf{r},\mathsf{w}=1)}\,. \tag{13.3}$$

Using the numbers in Fig. 13.1, we get $p(\mathsf{s}=1|\mathsf{w}=1) \approx 0.40$ and $p(\mathsf{r}=1|\mathsf{w}=1) \approx 0.54$: the most likely cause of the wet grass is rain.

In Fig. 13.2 we show the factor graph representation of (13.1), and the one corresponding to the conditional distribution $p(\mathsf{c},\mathsf{s},\mathsf{r}|\mathsf{w}=1)$. As is clear from the factor graph representation, the observation $\mathsf{w}=1$ induces some further dependency among the variables s and r, beyond the one induced by their relation with c. The reader is invited to draw the factor graph associated to the *marginal* distribution $p(\mathsf{c},\mathsf{s},\mathsf{r})$.

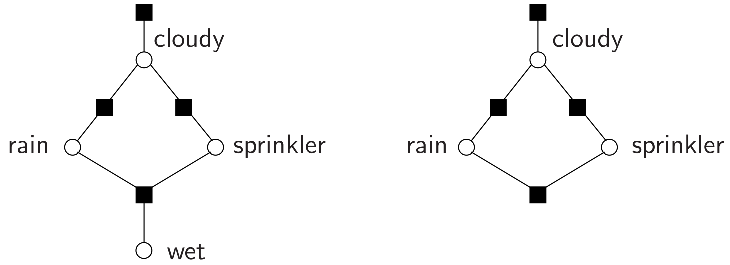


FIG. 13.2. Left: Factor graph corresponding to the sprinkler-rain Bayesian network, represented in Fig. 13.1. Right: factor graph for the same network under the observation of the variable w.

ig:FactorSprinklerRain}

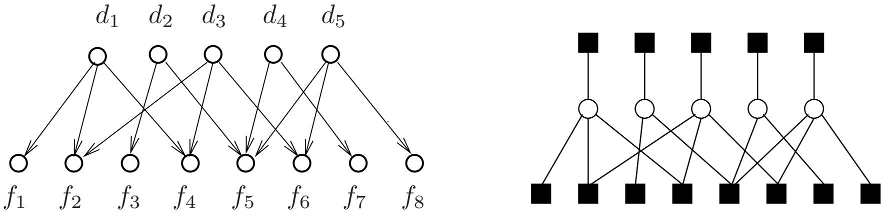


FIG. 13.3. Left: toy example of QMR-DT Bayesian network. Right: factor graph representation of the conditional distribution of the diseases $d_1, \dots d_5$, given the findings $f_1, \dots f_8$.

{fig:BayesFactor}

In general, a **Bayesian network** is an acyclic directed graph $G = (V, E)$ defining a probability distribution for variables at the vertices of the graph. A directed graph is an ordinary graph with a direction (i.e. an ordering of the adjacent vertices) chosen on each of its edges, and no cycle. In such a graph, we say that a vertex $u \in V$ is a **parent** of v, and write $u \in \pi(v)$, if (u, v) is a (directed) edge of G. A random variable X_v is associated with each vertex v of the graph (for simplicity we assume all the variables to take values in the same finite set $\mathcal{X}$). The joint distribution of $\{X_v,\ v \in V\}$ is determined by the conditional probability distributions $\{p(x_v|\underline{x}_{\pi(v)})\}$, where $\pi(v)$ denotes the set of parents of vertex v, and $\underline{x}_{\pi(v)} = \{x_u\ :\ u \in \pi(v)\}$:

$$p(\underline{x}) = \prod_{v \in \pi(G)} p(x_v) \prod_{v \in G \setminus \pi(G)} p(x_v|\underline{x}_{\pi(v)})\,, \tag{13.4}$$

where $\pi(G)$ denotes the set of vertices that have no parent in G.

A general class of statistical inference problems is formulated as follows. One is given a Bayesian network, i.e. a directed graph G plus the associated conditional probability distributions, $\{p(x_v|\underline{x}_{\pi(v)})\}$. A subset $O \subseteq V$ of the variables is observed and takes values $\underline{x}_O$. The problem is to compute marginals of the conditional distribution $p(\underline{x}_{V \setminus O}|\underline{x}_O)$.

Given a Bayesian network G and a set of observed variable O, it is easy to obtain a factor graph representation of the conditional distribution $p(\underline{x}_{V\setminus O}|\underline{x}_O)$, by a generalization of the procedure that we applied in Fig. 13.2. The general rule is as follows: (i) associate a variable node with each non-observed variable (i.e. each variable in $\underline{x}_{V\setminus O}$); (ii) for each variable in $\pi(G)\setminus O$, add a degree 1 function node connected uniquely to that variable; (iii) for each non observed vertex v which is not in $\pi(G)$, add a function node and connect it to v and to all the parents of v; (iv) finally, for each observed variable u, add a function node and connect it to all the parents of u.

Here is an example showing the practical utility of Bayesian networks.

Example 13.2 The Quick Medical Reference–Decision Theoretic (QMR-DT) network is a two level Bayesian network developed for automatic medical diagnostic. A schematic example is shown in Fig. 13.3. Variables in the top level, denoted by $d_1,\dots,d_N$, are associated with *diseases*. Variables in the bottom level, denoted by $f_1,\dots,f_M$, are associated with symptoms or *findings*. Both diseases and findings are described by binary variables. An edge connects the disease d_i to the finding f_a whenever such a disease may be a cause for that finding. Such networks of implications are constructed on the basis of accumulated medical experience.

The network is completed with two types of probability distributions. For each disease d_i we are given an *a priori* occurrence probability $P(d_i)$. Furthermore, for each finding we have a conditional probability distribution for that finding given a certain disease pattern. This usually takes the so called 'noisy-OR' form:

$$P(f_a = 0|d) = \frac{1}{z_a}\exp\left\{-\sum_{i=1}^{N}\theta_{ia}d_i\right\}\,. \tag{13.5}$$

This network is to be used for diagnostic purposes. The findings are set to values determined by the observation of a patient. Given this pattern of symptoms, one would like to compute the marginal probability that any given disease is indeed present.

13.1.2 *Inference in coding, statistical physics and combinatorial optimization*

Several of the problems encountered so far in this book can be recast in an inference language.

Let us start with the decoding of error correcting codes. As discussed in Chapter 6, in order to implement symbol-MAP decoding, one has to compute the marginal distribution of input symbols, given the channel output. In the case of LDPC (and related) code ensembles, dependencies between input symbols are induced by the parity check constraints. The joint probability distribution to be marginalized has a natural graphical representation (although we used factor graphs rather than Bayesian networks). Also, the introduction of finite–temperature decoding, allows to view word MAP decoding as the zero temperature limit case of a one-parameter family of inference problems.

In statistical mechanics models one is mainly interested in the expectations and covariances of local observables with respect to the Boltzmann measure. For instance, the paramagnetic to ferromagnetic transition in an Ising ferromagnet, cf. Sec. 2.5, can be located using the magnetization $M_N(\beta, B) = \langle \sigma_i \rangle_{\beta,B}$. The computation of covariances, such as the correlation function $C_{ij}(\beta, B) = \langle \sigma_i; \sigma_j \rangle_{\beta,B}$, is a natural generalization of the simple inference problem discussed so far.

Let us finally consider the case of combinatorial optimization. Assume, for the sake of definiteness, that a feasible solution is an assignment of the variables $\underline{x} = (x_1, x_2, \dots, x_N) \in \mathcal{X}^N$ and that its cost $E(\underline{x})$ can be written as the sum of 'local' terms:

$$E(\underline{x}) = \sum_a E_a(\underline{x}_a)\,. \tag{13.6}$$

Here $\underline{x}_a$ denotes a subset of the variables $(x_1, x_2, \dots, x_N)$. Let $p_*(\underline{x})$ denote the uniform distribution over optimal solutions. The minimum energy can be computed as a sum of expectation with respect to this distribution: $E_* = \sum_a [\sum_{\underline{x}} p_*(\underline{x}) E_a(\underline{x}_a)]$. Of course the distribution $p_*(\underline{x})$ does not necessarily have a simple representation, and therefore the computation of E_* can be significantly harder than simple inference[39].

This problem can be overcome by 'softening' the distribution $p_*(\underline{x})$. One possibility is to introduce a finite temperature and define $p_\beta(\underline{x}) = \exp[-\beta E(\underline{x})]/Z$ as already done in Sec. 4.6: if β is large enough, this distribution concentrates on optimal solutions. At the same time it has an explicit representation (apart from the value of the normalization constant Z) at any value of β.

How large should β be in order to get a good estimate of E_*? The Exercise below, gives the answer under some rather general assumptions.

Exercise 13.1 Assume that the cost function $E(\underline{x})$ takes integer values and let $U(\beta) = \langle E(\underline{x}) \rangle_\beta$. Due to the form (13.6) the computation of $U(\beta)$ is essentially equivalent to statistical inference. Assume, furthermore that $\Delta_{\max} = \max[E(\underline{x}) - E_*]$ is bounded by a polynomial in N. Show that

$$0 \le \frac{\partial U}{\partial T} \le \frac{1}{T^2}\, \Delta_{\max}^2\, |\mathcal{X}|^N e^{-1/T}\,. \tag{13.7}$$

where $T = 1/\beta$. Deduce that, by taking $T = \Theta(1/N)$, one can obtain $|U(\beta) - E_*| \le \varepsilon$ for any fixed $\varepsilon > 0$.

[39] Consider, for instance, the MAX-SAT problem, and let $E(\underline{x})$ be the number of unsatisfied clauses under the assignment $\underline{x}$. If the formula under study is satisfiable, then $p_*(\underline{x})$ is proportional to the product of characteristic functions associated to the clauses, cf. Example 9.7. In the opposite case, no explicit representation is known.

In fact a much larger temperature (smaller β) can be used in many important cases. We refer to Chapter 2 for examples in which $U(\beta) = E_* + E_1(N)\, e^{-\beta} + O(e^{-2\beta})$ with $E_1(N)$ growing polynomially in N. In such cases one expects $\beta = \Theta(\log N)$ to be large enough.

13.2 Monte Carlo method: inference via sampling

{sec:MonteCarloInference}

Consider the statistical inference problem of computing the marginal probability $p(x_i = x)$ from a joint distribution $p(\underline{x})$, $\underline{x} = (x_1, x_2, \dots, x_N) \in \mathcal{X}^N$. Given L i.i.d. samples $\{\underline{x}^{(1)}, \dots, \underline{x}^{(L)}\}$ drawn from the distribution $p(\underline{x})$, the desired marginal $p(x_i = x)$ can be estimated as the the fraction of such samples for which $x_i = x$.

'Almost i.i.d.' samples from $p(\underline{x})$ can be produced, in principle, using the Monte Carlo Markov Chain (MCMC) method of Sec. 4.5. Therefore MCMC can be viewed as a general-purpose inference strategy which can be applied in a variety of contexts.

Notice that the locality of the interactions, expressed by the factor graph, is very useful since it allows to generate easily 'local' changes in $\underline{x}$ (e.g. changing only one x_i, or a small number of them). This will[40] in fact typically change the value of just a few compatibility functions and hence produce only a small change in $p(\underline{x})$ (i.e., in physical terms, in the energy of $\underline{x}$). The possibility of generating, given $\underline{x}$, a new configuration close in energy is in fact important for MCMC to work. In fact, moves increasing the system energy by a large amount are typically rejected within MCMC rules .

One should also be aware that sampling, for instance by MCMC, only allows to estimate marginals or expectations which involve a small subset of variables. It would be very hard for instance to estimate the probability of a particular configuration $\underline{x}$ through the number $L(\underline{x})$ of its occurrences in the samples. The reason is that at least $1/p(\underline{x})$ samples would be required to have any accuracy, and this is typically a number exponentially large in N.

13.2.1 *LDPC codes*

Consider a code $\mathfrak{C}$ from one of the LDPC ensembles introduced in Chapter 11, and assume it has been used to communicate over a binary input memoryless symmetric channel with transition probability $Q(y|x)$. As shown in Chapter 6, cf. Eq. (6.3), the conditional distribution of the channel input $\underline{x}$, given the output $\underline{y}$, reads

$$P(\underline{x}|\underline{y}) = \frac{1}{Z(\underline{y})}\, \mathbb{I}(\underline{x} \in \mathfrak{C}) \prod_{i=1}^{N} Q(y_i|x_i)\,. \tag{13.8}$$

We can use the explicit representation of the code membership function to write

[40] We do not claim here that this is the case always, but just in many examples of interest.

$$P(\underline{x}|\underline{y}) = \frac{1}{Z(\underline{y})} \prod_{a=1}^{M} \mathbb{I}(x_{i_1^a} \oplus \cdots \oplus x_{i_k^a} = \mathtt{0}) \prod_{i=1}^{N} Q(y_i|x_i)\,. \tag{13.9}$$

in order to implement symbol MAP decoding, we must compute the marginals $P^{(i)}(x_i|\underline{y})$ of this distribution. Let us see how this can be done in an approximate way via MCMC sampling.

Unfortunately, the simple MCMC algorithms introduced in Sec. 4.5 (single bit flip with acceptance test satisfying detailed balance) cannot be applied in the present case. In any reasonable LDPC code, each variable x_i is involved into at least one parity check constraint. Suppose that we start the MCMC algorithm from a random configuration $\underline{x}$ distributed according to Eq. (13.9). Since $\underline{x}$ has non-vanishing probability, it satisfies all the parity check constraints. If we propose a new configuration where bit x_i is flipped, this configuration will violate all the parity check constraints involving x_i. As a consequence, such a move will be rejected by any rule satisfying detailed balance. The Markov chain is therefore reducible (each codeword forms a separate ergodic component), and useless for sampling purposes.

In good codes, this problem is not easily cured by allowing for moves that flip more than a single bit. As we saw in Sec. 11.2, if $\mathfrak{C}$ is drawn from an LDPC ensemble with minimum variable degree equal to 2 (respectively, at least 3), its minimum distance diverges logarithmically (respectively, linearly) with the block-length. In order to avoid the problem described above, a number of bits equal or larger than the minimum distance must be flipped simultaneously. On the other hand, large moves of this type are likely to be rejected, because they imply a large and uncontrolled variation in the likelihood $\prod_{i=1}^{N} Q(y_i|x_i)$.

A way out of this dilemma consists in 'softening' the parity check constraint by introducing a 'parity check temperature' γ and the associated distribution

$$P_\gamma(\underline{x}|\underline{y}) = \frac{1}{Z(\underline{y},\gamma)} \prod_{a=1}^{M} e^{-\gamma E_a(x_{i_1^a}\dots x_{i_k^a})} \prod_{i=1}^{N} Q(y_i|x_i)\,. \tag{13.10}$$

Here the energy term $E_a(x_{i_1^a}\dots x_{i_k^a})$ takes values 0 if $x_{i_1^a} \oplus \cdots \oplus x_{i_k^a} = \mathtt{0}$ and 2 otherwise. In the limit $\gamma \to \infty$, the distribution (13.10) reduces to (13.9). The idea is now to estimate the marginals of (13.10), $P_\gamma^{(i)}(x_i|\underline{y})$ via MCMC sampling and then to use the decoding rule

$$x_i^{(\gamma)} \equiv \arg\max_{x_i} P_\gamma^{(i)}(x_i|\underline{y})\,. \tag{13.11}$$

For any finite γ, this prescription is surely sub-optimal with respect to symbol MAP decoding. In particular, the distribution (13.10) gives non-zero weight to words $\underline{x}$ which do not belong to the codebook $\mathfrak{C}$. On the other hand, one may hope that for γ large enough, the above prescription achieves a close-to-optimal bit error rate.

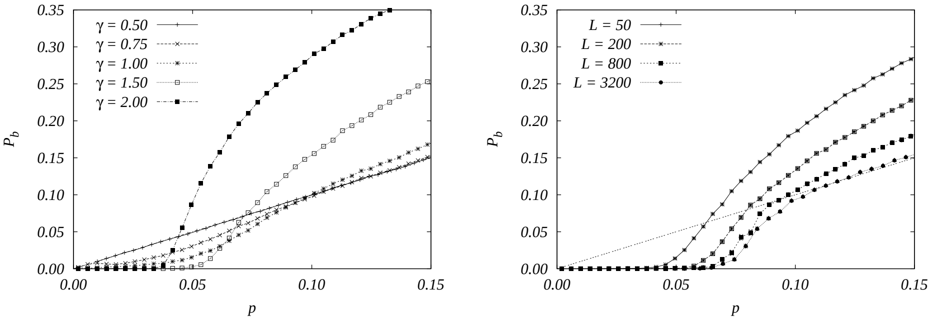


FIG. 13.4. Decoding LDPC codes from the $(3, 6)$ ensemble, used over the BSC channel with flip probability p, using MCMC sampling. The bit error rate is plotted versus p. The block-length is fixed to $N = 2000$, the number of sweeps is $2L$. Left: For $L = 100$, several values of the effective inverse temperature γ. Right: improvement of the performance as the number of sweeps increases at fixed $\gamma = 1.5$. {fig:LDPCMC}

We can simplify further the above strategy by giving up the objective of approximating the marginal $P_\gamma^{(i)}(x_i|\underline{y})$ within any prescribed accuracy. We shall rather run the Glauber single bit flip MCMC algorithm for a fixed computer time and extract an estimate of $P_\gamma^{(i)}(x_i|\underline{y})$ from this run. Fig 13.4 shows the results of Glauber dynamics executed for $2LN$ steps starting from a uniformly random configuration. At each step a bit is chosen uniformly at random and flipped with probability (here $\underline{x}^{(i)}$ is the configuration obtained from $\underline{x}$, by flipping the i-th bit)

$$w_i(\underline{x}) = \frac{P_\gamma(\underline{x}^{(i)}|\underline{y})}{P_\gamma(\underline{x}^{(i)}|\underline{y}) + P_\gamma(\underline{x}|\underline{y})} . \qquad (13.12)$$

The reader is invited to derive an explicit expression for $w_i(\underline{x})$, and to show that ⋆ this probability can be computed with a number of operations independent of the block-length. In this context, one often refer to a sequence of N successive updates, as a **sweep** (on average, one flip is proposed at each bit in a sweep). The value of x_i is recorded at each of the last L sweeps, and the decoder output is $x_i = \mathtt{0}$ or $x_i = \mathtt{1}$ depending on which value occurs more often in this record.

The data in Fig. 13.4 refers to communication over a binary symmetric channel (BSC) with flip probability p. In the left frame, we fix $L = 100$ and use several values of γ. At small γ, the resulting bit error rate is almost indistinguishable from the one in absence of coding, namely $\mathrm{P_b} = p$. As γ increases, parity checks are enforced more and more strictly and the error correcting capabilities improve at low noise. The behavior is qualitatively different for larger noise levels: for $p \gtrsim 0.05$, the bit error rate increases with γ. The reason of this change is essentially dynamical. The Markov chain used for sampling from the distribution (13.10) decorrelates more and more slowly from its initial condition. Since the initial condition is uniformly random, thus yielding $\mathrm{P_b} = 1/2$, the bit error rate obtained through our algorithm approaches $1/2$ at large γ (and above a certain threshold in p). This interpretation is confirmed by the data in the right frame of the same figure.

We shall see in Chapter **??** that in the large blocklength limit, the threshold for error-less bit MAP decoding in this case is predicted to be $p_{\rm c} \approx 0.101$. Unfortunately, because of its slow dynamics, our MCMC decoder cannot be used in practice if the channel noise is close to this threshold.

The sluggish dynamics of our single spin-flip MCMC for the distribution (13.10) is partially related to its reducibility for the model with hard constraints (13.9). A first intuitive picture is as follows. At large γ, codewords correspond to isolated 'lumps' of probability with $P_\gamma(\underline{x}|\underline{y}) = \Theta(1)$, separated by unprobable regions such that $P_\gamma(\underline{x}|\underline{y}) = \Theta(e^{-2\gamma})$ or smaller. In order to decorrelate, the Markov chain must spend a long time (at least of the order of the code minimum distance) in an unprobable region, and this happens only very rarely. This rough explanation is neither complete nor entirely correct, but we shall refine it in the next Chapters.

13.2.2 *Ising model*

Some of the basic mechanisms responsible for the slowing down of Glauber dynamics can be understood on simple statistical mechanics models. In this Section we consider the ferromagnetic Ising model with energy function

$$E(\sigma) = -\sum_{(ij)\in G} \sigma_i\sigma_j\, . \tag{13.13}$$

Here G is an ordinary graph on N vertices, whose precise structure will depend on the particular example. The Monte Carlo method is applied to the problem of sampling from the Boltzmann distribution $p_\beta(\sigma)$ at inverse temperature β.

As in the previous Section, we focus on Glauber (or heath bath) dynamics, but rescale time: in an infinitesimal interval $\mathrm{d}t$ a flip is proposed with probability $N\mathrm{d}t$ at a uniformly random site i. The flip is accepted with the usual heath bath probability (here σ is the current configuration and $\sigma^{(i)}$ is the configuration obtained by flipping the spin σ_i):

$$w_i(\sigma) = \frac{p_\beta(\sigma^{(i)})}{p_\beta(\sigma) + p_\beta(\sigma^{(i)})}\, . \tag{13.14}$$

Let us consider first equilibrium dynamics. We assume therefore that the initial configuration $\sigma(0)$ is sampled from the equilibrium distribution $p_\beta(\,\cdot\,)$ and ask how many Monte Carlo steps must be performed (in other words, how much time must be waited) in order to obtain an effectively independent random configuration. A convenient way of monitoring the equilibrium dynamics, consists in computing the time correlation function

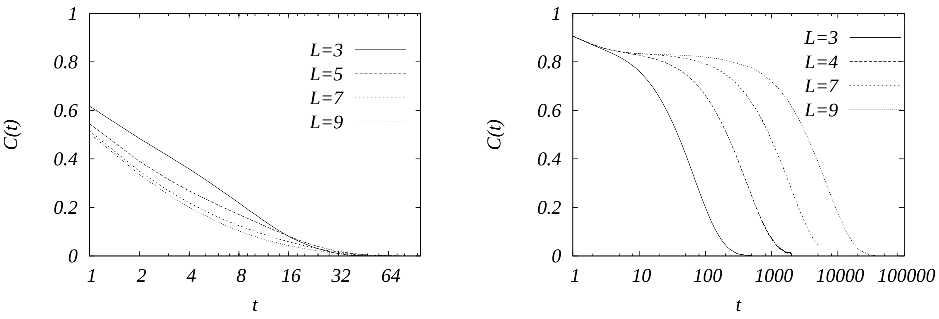


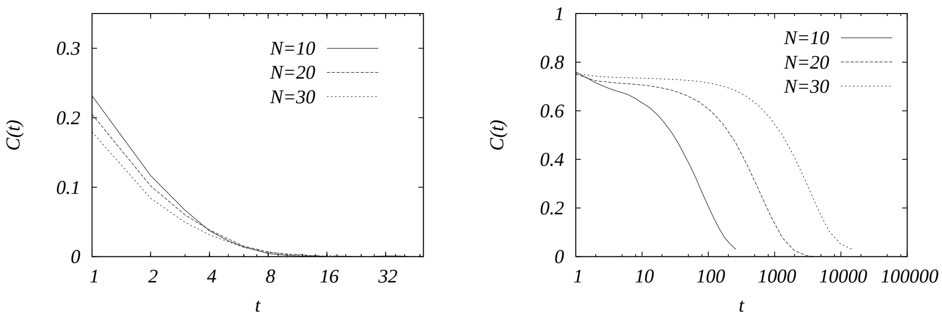
FIG. 13.5. Equilibrium correlation function for the Ising model on the two dimensional grid of side L. Left: high temperature, $T = 3$. Right: low temperature, $T = 2$. {fig:2dMC}

$$C_N(t) \equiv \frac{1}{N} \sum_{i=1}^{N} \langle \sigma_i(0)\sigma_i(t) \rangle \, . \tag{13.15}$$

Here the average $\langle \cdot \rangle$ is taken with respect to the realization of the Monte Carlo dynamics, as well as the initial state $\sigma(0)$. Notice that $(1-C(t))/2$ is the average fraction of spins with differ in the configurations $\sigma(0)$ and $\sigma(t)$. One expects therefore $C(t)$ to decrease with t, asymptotically reaching 0 when $\sigma(0)$ and $\sigma(t)$ are well decorrelated[41].

The reader may wonder how can one sample $\sigma(0)$ from the equilibrium (Boltzmann) distribution? As already suggested in Sec. 4.5, within the Monte Carlo approach one can obtain an 'almost' equilibrium configuration by starting from an arbitrary one and running the Markov chain for sufficiently many steps. In practice we initialize our chain from a uniformly random configuration (i.e. an infinite temperature equilibrium configuration) and run the dynamics for t_w sweeps. We call $\sigma(0)$ the configuration obtained after this process and run for t more sweeps in order to measure $C(t)$. The choice of t_w is of course crucial: in general the above procedure will produce a configuration $\sigma(0)$, whose distribution is not the equilibrium one, and depends on t_w. The measured correlation function will also depend on t_w. Determining how large t_w should be in order to obtain a good enough approximation of $C(t)$ is a subject of intense theoretical work. A simple empirical rule consists in measuring $C(t)$ for a given large t_w, then double it and check that nothing has changed. With these instructions, the reader is invited to write a code of MCMC for the Ising model on a general graph $\star$ and reproduce the following data.

[41] Notice that each spin is equally likely to take values $+1$ or -1 under the Boltzmann distribution with energy function (13.13.)



{fig:RGraphMC} FIG. 13.6. Equilibrium correlation function for the Ising model on random graphs from the $\mathbb{G}_N(2, M)$ ensemble, with $M = 2N$. Left: high temperature, $T = 5$. Right: low temperature, $T = 2$.

{ex:2dSimul}

Example 13.3 We begin by considering the Ising model on a two-dimensional grid of side L, with periodic boundary conditions. The vertex set is $\{(x_1, x_2) : 1 \leq x_a \leq L\}$. Edges join any two vertices at (Euclidean) distance one, plus the vertices (L, x_2) to $(1, x_2)$, and (x_1, L) to $(x_1, 1)$. We denote by $C_L(t)$ the correlation function for such a graph.

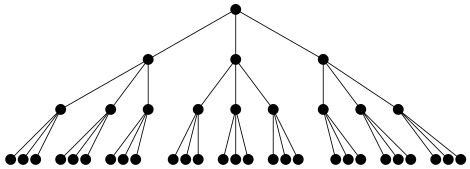
In Chapter 2 we saw that this model undergoes a phase transition at the critical temperature $T_{\rm c} = 2/\log(1+\sqrt{2}) \approx 2.269185$. The correlation functions plotted in Fig. 13.5 are representative of the qualitative behavior in the high temperature (left) and low temperature (right) phases. At high temperature $C_L(t)$ depends only mildly on the linear size of the system L. As L increases, the correlation functions approaches rapidly a limit curve $C(t)$ which decreases from 1 to 0 in a finite time scale[42].

At low temperature, there exists no limiting curve $C(t)$ decreasing from 1 to 0, such that $C_L(t) \to C(t)$ as $L \to \infty$. The time required for the correlation function $C_L(t)$ to get close to 0 is much larger than in the high-temperature phase. More importantly, it depends strongly on the system size. This suggests that strong cooperative effects are responsible for the slowing down of the dynamics.

{ex:RGraphSimul}

Example 13.4 Take G as a random graph from the $\mathbb{G}_N(2, M)$ ensemble, with $M = N\alpha$. As we shall see in Chapter ???, this model undergoes a phase transition when $N \to \infty$ at a critical temperature $\beta_{\rm c}$, satisfying the equation $2\alpha \tanh \beta = 1$. In Fig. 13.6 we present numerical data for a few values of N, and $\alpha = 2$ (corresponding to a critical temperature $T_{\rm c} \approx 3.915230$).

The curves presented here are representative of the high temperature and low temperature phases. As in the previous example, the relaxation time scale strongly depends on the system size at low temperature.



{fig:TernaryTree} FIG. 13.7. A rooted ternary tree with $n = 4$ generations and $N = 40$ vertices.

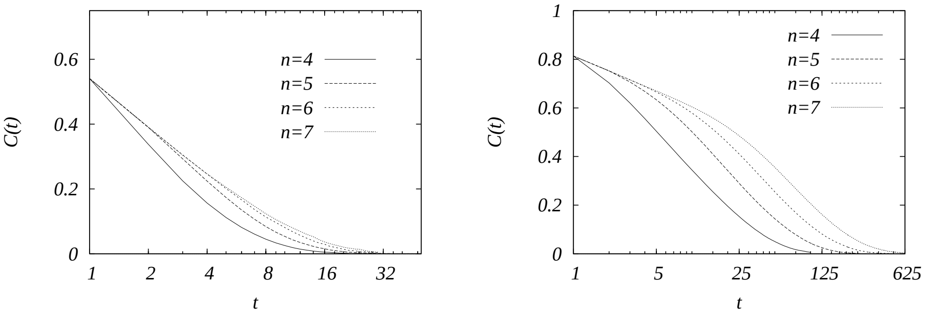


FIG. 13.8. Equilibrium correlation function for the ferromagnetic Ising model on a regular ternary tree. Left: high temperature, $T = 2$. Right: low temperature, $T = 1.25$. {fig:TreeMC}

{ex:TreeSimul}

Example 13.5 Take G as a rooted ternary tree, with n generations, cf. Fig. 13.7. Of course G contains $N = (3^n - 1)/(3 - 1)$ vertices and $N - 1$ edges. As we will see in Chapter ???, this model undergoes a phase transition at a critical temperature $\beta_{\rm c}$, which satisfies the equation $3(\tanh\beta)^2 = 1$. We get therefore $T_{\rm c} \approx 1.528651$. In this case the dynamics of spin depends strongly upon its distance to the root. In particular leaf spins are much less constrained than the others. In order to single out the 'bulk' behavior, we modify the definition of the correlation function (13.15) by averaging only over the spins σ_i in the first $\underline{n} = 3$ generations. We keep $\underline{n}$ fixed as $n \to \infty$.

As in the previous examples, $C_N(t)$ has a well defined $n \to \infty$ limit in the high temperature phase, and is strongly size-dependent at low temperature.

We summarize the last three examples by comparing the size dependence of the relaxation time scale in the respective low temperature phases. A simple way to define such a time scale consists in looking for the smallest time such that $C(t)$ decreases below some given threshold:

$$\tau(\delta; N) = \min\{\, t > 0 \text{ s.t. } C_N(t) \le \delta\} \,. \tag{13.16}$$

In Fig. 13.9 we plot the estimates obtained from the data presented in the previous examples, using $\delta = 0.2$, and keeping to the data in the low-temperature (ferromagnetic) phase. The size dependence of $\tau(\delta; N)$ is very clear. However,

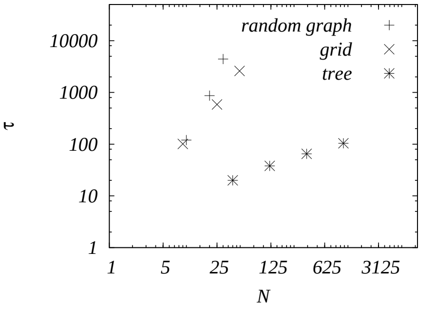
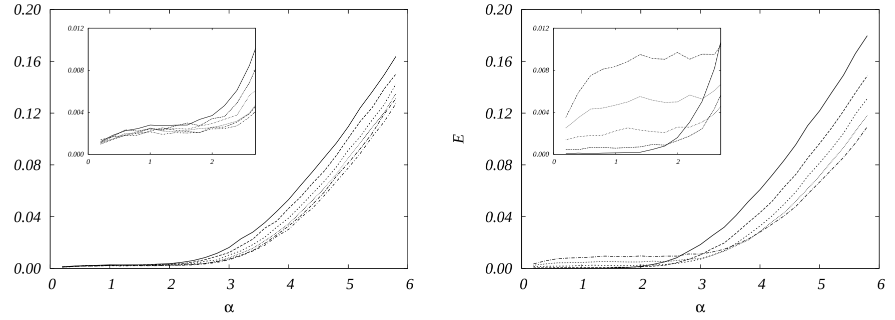


FIG. 13.9. Size dependence of the relaxation time in the ferromagnetic Ising model in its low temperature phase. Different symbols refer to the different families of graphs considered in Examples 13.3 to 13.5. {fig:Time}



{fig:MCKSAT} FIG. 13.10. Minimization of the number of unsatisfied clauses in random 3-SAT formulae via Glauber dynamics. Here the number of variables $N = 1000$ is kept fixed. Left: $T = 0.25$ and, from top to bottom $L = 2.5 \cdot 10^3$, $5 \cdot 10^3$, 10^4, $2 \cdot 10^4$, $4 \cdot 10^4$, $8 \cdot 10^4$ iterations. Right: $L = 4 \cdot 10^4$ and (from top to bottom at large α) $T = 0.15$, 0.20, 0.25, 0.30, 0.35. The insets show the small α regime in greater detail.

it is much stronger for the random graph and square grid cases (and, in particular, in the former) than on the tree. In fact, it can be shown that, in the ferromagnetic phase:

$$\tau(\delta; N) = \begin{cases} \exp\{\Theta(N)\} & \text{random graph,} \\ \exp\{\Theta(\sqrt{N})\} & \text{square lattice,} \\ \exp\{\Theta(\log N)\} & \text{tree.} \end{cases} \tag{13.17}$$

Section 13.3 will explain the origins of these different behaviors.

13.2.3 *MAX-SAT*

Given a satisfiability formula over N boolean variables $(x_1, \dots, x_N) = \underline{x}$, $x_i \in \{0,1\}$, the MAX-SAT optimization problem requires to find a truth assignment which satisfies the largest number of clauses. We denote by $\underline{x}_a$ the set of variables involved in the a-th clause and by $E_a(\underline{x}_a)$ a function of the truth assignment taking value 0, if the clause is satisfied, and 2 otherwise. With this definitions, the MAX-SAT problem can be rephrased as the problem of minimizing an energy function of the form $E(\underline{x}) = \sum_a E_a(\underline{x}_a)$, and we can therefore apply the general approach discussed after Eq. (13.6).

We thus consider the Boltzmann distribution $p_\beta(\underline{x}) = \exp[-\beta E(\underline{x})]/Z$ and try to sample a configuration from $p_\beta(\underline{x})$ at large enough β using MCMC. The assignment $\underline{x} \in \{0,1\}^N$ is initialized uniformly at random. At each time step a variable index i is chosen uniformly at random and the corresponding variable is flipped according to the heath bath rule

$$w_i(\underline{x}) = \frac{p_\beta(\underline{x}^{(i)})}{p_\beta(\underline{x}) + p_\beta(\underline{x}^{(i)})} \, . \tag{13.18}$$

As above $\underline{x}^{(i)}$ denotes the assignment obtained from $\underline{x}$ by flipping the i-th variable. The algorithm is stopped after LN steps (i.e. L sweeps), and one puts in memory the current assignment $\underline{x}_*$ (and the corresponding cost $E_* = E(\underline{x}_*)$).

In Fig. 13.10 we present the outcomes of such an algorithm, when applied to random 3-SAT formulae from the ensemble $\mathsf{SAT}_N(3, M)$ with $\alpha = M/N$. Here we focus on the mean cost $\langle E_* \rangle$ of the returned assignment. One expects that, as $N \to \infty$ with fixed L, the cost scales as $\langle E_* \rangle = \Theta(N)$, and order N fluctuations of E_* away from the mean are exponentially unlikely. At low enough temperature, the behavior depends dramatically on the value of α. For small α, E_*/N is small and grows rather slowly with α. Furthermore, it seems to decrease to 0 ad β increases. Our strategy is essentially successful and finds an (almost) satisfying assignment. Above $\alpha \approx 2 \div 3$, E_*/N starts to grow more rapidly with α, and doesn't show signs of vanishing as $\beta \to \infty$. Even more striking is the behavior as the number of sweeps L increases. In the small α regime, E_*/N rapidly decreases to some, roughly L independent saturation value, already reached after about 10^3 sweeps. At large α there seems also to be an asymptotic value but this is reached much more slowly, and even after 10^5 sweeps there is still space from improvement.

13.3 Free energy barriers

{se:arrhenius}

These examples show that the time scale required for a Monte Carlo algorithm to produce (approximately) statistically independent configurations may vary wildly depending on the particular problem at hand. The same is true if we consider the time required to generating a configuration (approximately) distributed according to the equilibrium distribution, starting from an arbitrary initial condition.

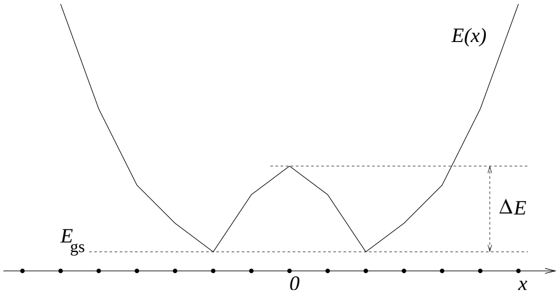


FIG. 13.11. Random walk in a double-well energy landscape. After how many steps the walker is (approximatively) distributed according to the equilibrium distribution? {fig:WellWalk}

There exist various sophisticated techniques for estimating these time scales analytically, at least in the case of unfrustrated problems. In this Section we discuss a simple argument which is widely used in statistical physics as well as in probability theory, that of free-energy barriers. The basic intuition can be conveyed by simple examples.

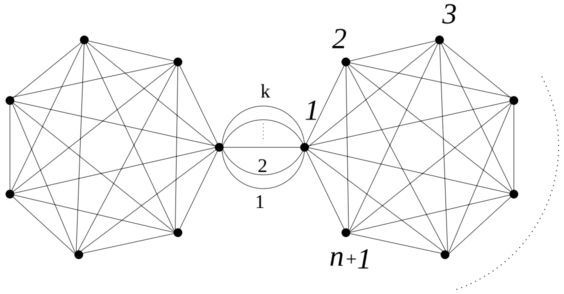


FIG. 13.12. How much time does a random walk need to explore this graph? {fig:DoubleGraph}

{ex:WalkWell}

Example 13.6 Consider a particle moving on the integer line, and denote its position as $x \in \mathbb{Z}$. Each point x on the line has an energy $E(x) \geq E_{\rm gs}$ associated to it, as depicted in Fig. 13.11. At each time step, the particle attempts to move to one of the adjacent positions (either to the right or to the left) with probability 1/2. If we denote by x' the position the particle is trying to move to, the move is accepted according to Metropolis rule

$$w(x \to x') = \min\left\{e^{-\beta[E(x')-E(x)]}, 1\right\} . \qquad (13.19)$$

The equilibrium distribution is of course given by Boltzmann law $P_\beta(x) = \exp[-\beta E(x)]/Z(\beta)$.

Suppose we start with, say $x = 10$. How many steps should we wait for x to be distributed according to $P_\beta(x)$? It is intuitively clear that, in order to equilibrate, the particle must spend some amount of time *both* in the right and in the left well, and therefore it must visit the $x = 0$ site. At equilibrium this is visited on average a fraction $P_\beta(0)$ of the times. Therefore, in order to see a jump, we must wait about

$$\tau \approx \frac{1}{P_\beta(0)} , \qquad (13.20)$$

steps.

One is often interested in the low temperature limit of τ. Assuming $E(x)$ diverges fast enough as $|x| \to \infty$, the leading exponential behavior of Z is $Z(\beta) \doteq e^{-\beta E_{\rm gs}}$, and therefore $\tau \doteq \exp\{\beta \Delta E\}$, where $\Delta E = E(0) - E_{\rm gs}$ is the energy barrier to be crossed in order to pass from one well to the others. A low temperature asymptotics of the type $\tau \doteq \exp\{\beta \Delta E\}$ is referred to as **Arrhenius law**.

{ex:WalkGraph}

Exercise 13.2 Consider a random walk on the graph of Fig. 13.12 (two cliques with $n+1$ vertices, joined by a k-fold degenerate edge). At each time step, the walker chooses one of the adjacent edges uniformly at random and moves through it to the next node. What is the stationary distribution $P_{\rm eq}(x)$, $x \in \{1, \dots 2n\}$? Show that the probability to be at node 1 is $\frac{1}{2}\frac{k+n-1}{n^2+k-n}$.

Suppose we start with a walker distributed according to $P_{\rm eq}(x)$. Using an argument similar to that in the previous example, estimate the number of time steps τ that one should wait in order to obtain an approximatively independent value of x. Show that $\tau \simeq 2n$ when $n \gg k$ and interpret this result. In this case the k-fold degenerate edge joining the two cliques is called a bottleneck, and one speaks of an entropy barrier.

In order to obtain a precise mathematical formulation of the intuition gained in the last examples, we must define what we mean by 'relaxation time'. We will focus here on ergodic continuous-time Markov chains on a finite state space $\mathcal{X}$. Such a chain is described by its transition rates $w(x \to y)$. If at time t, the chain is in state $x(t) = x \in \mathcal{X}$, then, for any $y \neq x$, the probability that the chain is in state y, 'just after' time t is

$$\mathbb{P}\left\{x(t+{\rm d}t) = y \mid x(t) = x\right\} = w(x \to y){\rm d}t\,. \tag{13.21}$$

Exercise 13.3 Consider a discrete time Markov chain and modify it as follows. Instead of waiting a unit time Δt between successive steps, wait an exponentially distributed random time (i.e. Δt is a random variable with pdf $p(\Delta t) = \exp(-\Delta t)$). Show that the resulting process is a continuous time Markov chain. What are the corresponding transition rates?

Let $x \mapsto \mathcal{O}(x)$ an observable (a function of the state), define the shorthand $\mathcal{O}(t) = \mathcal{O}(x(t))$, and assume $x(0)$ to be drawn from the stationary distribution. If ⋆ the chain satisfies the detailed balance[43] condition, one can show that the correlation function $C_{\mathcal{O}}(t) = \langle \mathcal{O}(0)\mathcal{O}(t)\rangle - \langle \mathcal{O}(0)\rangle\langle \mathcal{O}(t)\rangle$ is non negative, monotonously decreasing and that $C_{\mathcal{O}}(t) \to 0$ as $t \to \infty$. The exponential autocorrelation time for the observable $\mathcal{O}$, $\tau_{\mathcal{O},\rm exp}$, is defined by

$$\frac{1}{\tau_{\mathcal{O},\rm exp}} = -\lim_{t\to\infty}\frac{1}{t}\log C_{\mathcal{O}}(t)\,. \tag{13.22}$$

The time $\tau_{\mathcal{O},\rm exp}$ depends on the observable and tells how fast its autocorrelation function decays to 0: $C_{\mathcal{O}}(t) \sim \exp(-t/\tau_{\mathcal{O},\rm exp})$. It is meaningful to look for the 'slowest' observable and define the **exponential autocorrelation time**

[43] A continuous time Markov chains satisfies the detailed balance condition (is 'reversible') with respect to the stationary distribution $P(x)$, if, for any $x \neq y$, $P(x)w(x \to y) = P(y)w(y \to x)$.

(also called **inverse spectral gap**, or, for brevity **relaxation time**) of the Markov chain as

$$\tau_{\rm exp} = \sup_{\mathcal{O}} \{ \tau_{\mathcal{O},{\rm exp}} \} . \tag{13.23}$$

The idea of a bottleneck, and its relationship to the relaxation time, is clarified by the following theorem:

{thm:Cut}

Theorem 13.7 *Consider an ergodic continuous time Markov chain with state space $\mathcal{X}$, and transition rates $\{w(x \to y)\}$ satisfying detailed balance with respect to the stationary distribution $P(x)$. Given any two disjoint sets of states $\mathcal{A},\mathcal{B} \subset \mathcal{X}$, define the probability flux between them as $W(\mathcal{A} \to \mathcal{B}) = \sum_{x\in\mathcal{A},\, y\in\mathcal{B}} P(x)\, w(x \to y)$. Then*

$$\tau_{\rm exp} \ge \frac{P(x\in\mathcal{A})\, P(x \notin \mathcal{A})}{W(\mathcal{A} \to \mathcal{X}\backslash\mathcal{A})} . \tag{13.24}$$

In other words, a lower bound on the correlation time can be constructed by looking for 'bottlenecks' in the Markov chain, i.e. partitions of the configuration space into two subsets. The lower bound will be good (and the Markov chain will be slow) if each of the subsets carries a reasonably large probability at equilibrium, but jumping from one to the other is unlikely.

Example 13.8 Consider the random walk in the double well energy landscape of Fig. 13.11, where we confine the random walk to some big interval $[-M : M]$ in order to have a finite state space. Let us apply Theorem 13.7, by taking $\mathcal{A} = \{x \ge 0\}$. We have $W(\mathcal{A} \to \mathcal{X}\backslash\mathcal{A}) = P_\beta(0)/2$ and, by symmetry $P_\beta(x \in \mathcal{A}) = \frac{1}{2}(1 + P_\beta(0))$. The inequality (13.24) yields

$$\tau_{\rm exp} \ge \frac{1 - P_\beta(0)^2}{2P_\beta(0)} . \tag{13.25}$$

Expanding the right hand side in the low temperature limit, we get $\tau_{\rm exp} \ge 2\, e^{\beta\Delta E}\,(1 + \Theta(e^{-c\beta}))$.

Exercise 13.4 Apply Theorem 13.7 to a random walk in the asymmetric double well energy landscape of Fig. 13.13. Does Arrhenius law $\tau_{\rm exp} \sim \exp(\beta\Delta E)$ apply to this case? What is the relevant energy barrier ΔE?

Exercise 13.5 Apply Theorem 13.7 to estimate the relaxation time of the random walk on the graph in Exercise (13.2).

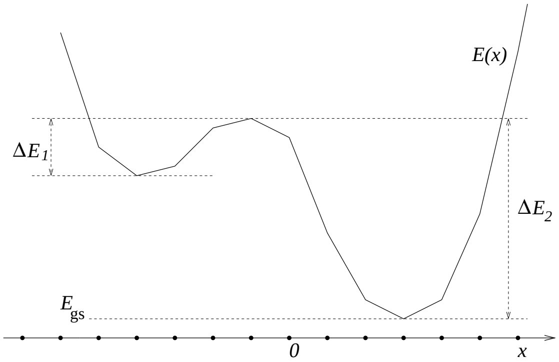


FIG. 13.13. Random walk in an asymmetric double well.

{fig:AsWell}

Example 13.9 Consider Glauber dynamics for the Ising model on a two dimensional $L\times L$ grid, with periodic boundary conditions, already discussed in Example 13.3. In the ferromagnetic phase, the distribution of the total magnetization $\mathcal{M}(\sigma)\equiv\sum_i \sigma_i$, $N=L^2$ is concentrated around the values $\pm N\, M_+(\beta)$, where $M_+(\beta)$ is the spontaneous magnetization. It is natural to expect that the bottleneck will correspond to the global magnetization changing sign. Assuming for instance that L is odd, let us define

$$\mathcal{A}=\{\sigma\,:\,\mathcal{M}(\sigma)\ge 1\}\quad;\quad \bar{\mathcal{A}}=\mathcal{X}\backslash A=\{\sigma\,:\,\mathcal{M}(\sigma)\le -1\} \tag{13.26}$$

Using the symmetry under spin reversal, Theorem 13.7 yields

$$\tau_{\rm exp}\;\ge\;4\sum_{\sigma\,:\mathcal{M}(\sigma)=1}\sum_{i\,:\sigma_i=1}P_\beta(\sigma)\;w(\sigma\to\sigma^{(i)})\,. \tag{13.27}$$

A good estimate of this sum can be obtained by noticing that, for any σ, $w(\sigma\to\sigma^{(i)})\ge w(\beta)\equiv\frac{1}{2}(1-\tanh 4\beta)$. Moreover, for any σ entering the sum, there are exactly $(L^2+1)/2$ sites i such that $\sigma_i=+1$. We get therefore $\tau_{\rm exp}\ge 2L^2 w(\beta)\sum_{\sigma\,:\mathcal{M}(\sigma)=1}P_\beta(\sigma)$ One suggestive way of writing this lower bound, consists in defining a constrained free energy as follows

$$F_L(m;\beta)\equiv-\frac{1}{\beta}\log\left\{\sum_{\sigma\,:\,\mathcal{M}(\sigma)=m}\exp[-\beta E(\sigma)]\right\}\,, \tag{13.28}$$

If we denote by $F_L(\beta)$ the usual (unconstrained) free energy, our lower bound can be written as

$$\tau_{\rm exp}\ge 2L^2 w(\beta)\;\exp\{\beta[F_L(1;\beta)-F_L(\beta)]\}\,. \tag{13.29}$$

Apart from the pre-exponential factors, this expression has the same form as Arrhenius law, the energy barrier ΔE, being replaced by a ‘free energy barrier’ $\Delta F_L(\beta)\equiv F_L(1;\beta)-F_L(\beta)$.

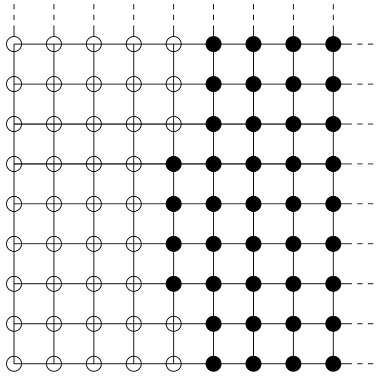


FIG. 13.14. Ferromagnetic Ising model on a 9×9 grid with periodic boundary conditions. Open circles correspond to $\sigma_i = +1$, and filled circles to $\sigma_i = -1$. The configuration shown here has energy $E(\sigma) = -122$ and magnetization $\mathcal{M}(\sigma) = +1$.

{fig:IsingZeroMagn}

We are left with the task of estimating $\Delta F_L(\beta)$. Let us start by considering the $\beta \to \infty$ limit. In this regime, $F_L(\beta)$ is dominated by the all plus and all minus configurations, with energy $E_{\text{gs}} = -2L^2$. Analogously, $F_L(1;\beta)$ is dominated by the lowest energy configurations satisfying the constraint $\mathcal{M}(\sigma) = 1$. An example of such configurations is the one in Fig. 13.14, whose energy is $E(\sigma) = -2L^2 + 2(2L+2)$. Of course, all configurations obtained from the one in Fig. 13.14, through a translation, rotation or spin inversion have the same energy. We find therefore $\Delta F_L(\beta) = 2(2L+2) + \Theta(1/\beta)$

It is reasonable to guess (and it can be proved rigorously) that the size dependence of $\Delta F_L(\beta)$ remains unchanged through the whole low temperature phase:

$$\Delta F_L(\beta) \simeq 2\gamma(\beta) L \,, \tag{13.30}$$

where the **surface tension** $\gamma(\beta)$ is strictly positive at any $\beta > \beta_{\text{c}}$, and vanishes as $\beta \downarrow \beta_{\text{c}}$. This in turns implies the following lower bound on the correlation time

$$\tau_{\text{exp}} \geq \exp\{2\beta\gamma(\beta)L + o(L)\} \,. \tag{13.31}$$

This bound matches the numerical simulations in the previous Section and can be proved to give the correct asymptotic size-dependence.

Exercise 13.6 Consider the ferromagnetic Ising model on a random graph from $\mathbb{G}_N(2, M)$ that we studied in Example 13.4, and assume, for definiteness, N even. Arguing as above, show that

$$\tau_{\rm exp} \geq C_N(\beta) \exp\{\beta[F_N(0; \beta) - F_N(\beta)]\} \,. \tag{13.32}$$

Here $C_N(\beta)$ is a constants which grows (with high probability) slower than exponentially with N; $F_N(m; \beta)$ is the free energy of the model constrained to $\mathcal{M}(\sigma) = m$, and $F_N(\beta)$ is the unconstrained partition function.

For a graph G, let $\delta(G)$ be the minimum number of bicolored edges if we color half of the vertices red, and half blue. Show that

$$F_N(0; \beta) - F_N(\beta) = 2\delta(G_N) + \Theta(1/\beta) \,. \tag{13.33}$$

The problem of computing $\delta(G)$ for a given graph G is referred to as **balanced minimum cut** (or **graph partitioning**) problem, and is known to be NP-complete. For a random graph in $\mathbb{G}_N(2, M)$, it is known that $\delta(G_N) = \Theta(N)$ with high probability in the limit $N \to \infty, M \to \infty$, with $\alpha = M/N$ fixed and $\alpha > 1/2$ (Notice that, if $\alpha < 1/2$ the graph does not contain a giant component and obviously $\delta(G) = o(N)$).

This claim can be substantiated through the following calculation. Given a spin configuration $\sigma = (\sigma_1, \dots, \sigma_N)$ with $\sum_i \sigma_i = 0$ let $\Delta_G(\sigma)$ be the number of edges in (i, j) in G such that $\sigma_i \neq \sigma_j$. Then

$$\mathbb{P}\,\{\delta(G) \leq n\} = \mathbb{P}\,\{\exists\,\sigma \text{ such that } \Delta_G(\sigma) \leq n\} \leq \sum_{m=0}^{n} \mathbb{E}\mathcal{N}_{G,m} \,, \tag{13.34}$$

where $\mathcal{N}_{G,m}$ denotes the number of spin configurations with $\Delta_G(\sigma) = m$. Show that

$$\mathbb{E}\mathcal{N}_{G,m} = \binom{N}{N/2} \binom{N}{2}^{-M} \binom{M}{m} \left(\frac{N^2}{4}\right)^m \left[\binom{N/2}{2} - \frac{N^2}{4}\right]^{M-m} . \tag{13.35}$$

Estimate this expression for large N, M with $\alpha = M/N$ fixed and show that it implies $\delta(G) \geq c(\alpha)N+$ with high probability, where $c(\alpha) > 0$ for $\alpha > 1/2$.

In Chapter ???, we will argue that the $F_N(0; \beta) - F_N(\beta) = \Theta(N)$ for all β's large enough.

{ex:TreeBarrier}

Exercise 13.7 Repeat the same arguments as above for the case of a regular ternary tree described in example 13.5, and derive a bound of the form (13.32). Show that, at low temperature, the Arrhenius law holds, i.e. $\tau_{\rm exp} \geq \exp\{\beta\Delta E_N + o(\beta)\}$. How does ΔE_N behave for large N?

[Hint: an upper bound can be obtained by constructing a sequence of configurations from the all plus to the all minus ground state, such that any two consecutive configurations differ in a single spin flip.]

Notes

For introductions to Bayesian networks, see (Jordan, 1998; Jensen, 1996). Bayesian inference was proved to be NP-hard by Cooper. Dagun and Luby showed that approximate Bayesian inference remains NP-hard. On the other hand, it becomes polynomial if the number of observed variables is fixed.

Decoding of LDPC codes via Glauber dynamics was considered in (Franz, Leone, Montanari and Ricci-Tersenghi, 2002). Satisfiability problems were considered in (Svenson and Nordahl, 1999).

Arrhenius law and the concept of energy barrier (or 'activation energy') were discovered by the Swedish chemist Svante Arrhenius in 1889, in his study of chemical kinetics. An introduction to the analysis of Monte Carlo Markov Chain methods (with special emphasis on enumeration problems), and their equilibration/convergence rate can be found in (Jerrum and Sinclair, 1996; Sinclair, 1997). The book in preparation by Aldous and Fill (Aldous and Fill, 2005) provides a complete exposition of the subject from a probabilistic point of view. For a mathematical physics perspective, we refer to the lectures of Martinelli (Martinelli, 1999).

For an early treatment of the Glauber dynamics of the Ising model on a tree, see (Henley, 1986). This paper contains a partial answer to Exercise 13.7.