14

BELIEF PROPAGATION

{ch:BP}

Consider the ubiquitous problem of computing marginals of a graphical model with N variables $\underline{x} = (x_1, \dots, x_N)$ taking values in a finite alphabet $\mathcal{X}$. The naive algorithm, summing over all configurations, takes a time of order $|\mathcal{X}|^N$. The complexity can be reduced dramatically when the underlying factor graph has some special structure. One extreme case is that of tree factor graphs. On trees, marginals can be computed in a number of operations which grows linearly with N. This can be done through a 'dynamic programming' procedure that recursively sums over all variables starting from the leaves and progressing towards the 'center' of the tree.

Remarkably, such a recursive procedure can be recast as a distributed 'message passing' algorithm. Message passing algorithms operate on 'messages' associated with edges of the factor graph, and update them recursively through local computations done at the vertices of the graph. The update rules that yield exact marginals on trees have been discovered independently in several different contexts: statistical physics (under the name 'Bethe Peierls approximation'), coding theory (sum-product algorithm), and artificial intelligence (belief propagation - BP). Here we will adopt the artificial intelligence terminology.

It is straightforward to prove that belief propagation exactly computes marginals on tree factor graphs. However, it was found only recently that it can be extremely effective on loopy graphs as well. One of the basic intuitions behind this success is that BP, being a local algorithm, should be successful whenever the underlying graph is 'locally' a tree. Such factor graphs appear frequently, for instance in error correcting codes, and BP turns out to be very powerful in this context. However, even in such cases, its application is limited to distributions such that far apart variables become uncorrelated. The onset of long range correlations, typical of the occurrence of a phase transition, generically leads to poor performances of BP. We shall see several applications of this idea in the next chapters.

We introduce the basic ideas in Section 14.1 by working out a couple of simple examples. The general BP equations are stated in Section 14.2, which also shows how they provide exact results on tree factor graphs. Section 14.3 describes some alternative message passing procedures, the Max-Product (equivalently, Min-Sum) algorithms, which can be used in optimization problems. In Section 14.4 we discuss the use of BP in graphs with loops. In the study of random constraint satisfaction problems, BP messages become random variables. The study of their distribution provides a large amount of information on such instances and can be used to characterize the corresponding phase diagram. The time evolution

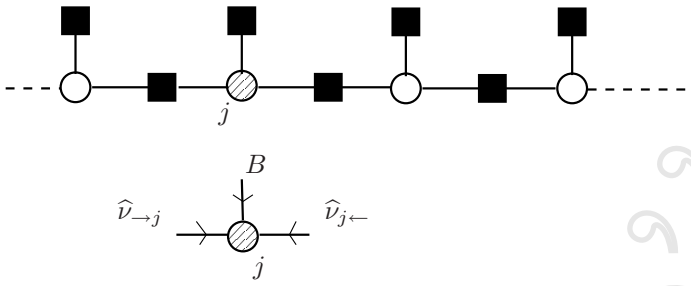


FIG. 14.1. Top: the factor graph of the one-dimensional Ising model in an external field. Bottom: the three messages arriving on site j describe the contributions to the probability distribution of σ_j, due to the left chain ($\widehat{\nu}_{\to j}$), to the right chain ($\widehat{\nu}_{j\leftarrow}$) and to the external field B. {fig:ising1dfg}

of these distributions is known under the name of density evolution, while their fixed point analysis is the replica symmetric cavity method. Both are explained in Section 14.6.

14.1 Two examples {se:2examples}

14.1.1 *Example 1: Ising chain*

Consider the ferromagnetic Ising model on a line. The variables are spins $(\sigma_1, \dots, \sigma_N) = \underline{\sigma}$, with $\sigma_i \in \{+1, -1\}$ and their joint distribution takes Boltzmann's form

$$p_\beta(\underline{\sigma}) = \frac{1}{Z}\, e^{-\beta E(\underline{\sigma})}\,, \qquad E(\underline{\sigma}) = -\sum_{i=1}^{N-1} \sigma_i \sigma_{i+1} - B \sum_{i=1}^{N} \sigma_i\,. \tag{14.1}$$

The corresponding factor graph is shown in Figure 14.1.1.

Let us now compute the marginal probability distribution $p(\sigma_j)$ of spin σ_j. We shall introduce three 'messages' arriving on spin j as the contributions to $p(\sigma_j)$ coming from each of the function nodes which are connected to i. More precisely, let us define

$$\widehat{\nu}_{\to j}(\sigma_j) = \frac{1}{Z_{\to j}} \sum_{\sigma_1 \dots \sigma_{j-1}} \exp\left\{ \beta \sum_{i=1}^{j-1} \sigma_i \sigma_{i+1} + \beta B \sum_{i=1}^{j-1} \sigma_i \right\},$$
$$\widehat{\nu}_{j\leftarrow}(\sigma_j) = \frac{1}{Z_{j\leftarrow}} \sum_{\sigma_{j+1} \dots \sigma_N} \exp\left\{ \beta \sum_{i=j+1}^{N-1} \sigma_i \sigma_{i+1} + \beta B \sum_{i=j+1}^{N} \sigma_i \right\}. \tag{14.2}$$

Messages are understood to be probability distributions and thus normalized. In the present case, the constants $Z_{\to j}$, $Z_{j\leftarrow}$ are set by the conditions $\widehat{\nu}_{\to j}(+1) + \widehat{\nu}_{\to j}(-1) = 1$, and $\widehat{\nu}_{j\leftarrow}(+1) + \widehat{\nu}_{j\leftarrow}(-1) = 1$. In the following, when dealing with normalized distributions, we shall avoid writing explicitly the normalization

constants and use the symbol $\cong$ to denote 'equality up to a normalization'. With this notation, the first of the above equations can be rewritten as

$$\widehat{\nu}_{\rightarrow j}(\sigma_j) \cong \sum_{\sigma_1 \dots \sigma_{j-1}} \exp\left\{ \beta \sum_{i=1}^{j-1} \sigma_i \sigma_{i+1} + \beta B \sum_{i=1}^{j-1} \sigma_i \right\} . \tag{14.3}$$

By rearranging the summation over spins σ_i, $i \neq j$, the marginal $p(\sigma_j)$ can be written as:

{eq:1dIsingMarginal}
$$p(\sigma_j) \cong \widehat{\nu}_{\rightarrow j}(\sigma_j)\, e^{\beta B \sigma_j}\, \widehat{\nu}_{j\leftarrow}(\sigma_j) \ . \tag{14.4}$$

In this expression we can interpret each of the three factors as a 'message' sent to j from each of the three function nodes connected to the variable j. Each message coincides with the marginal distribution of σ_j in a modified graphical model. For instance, $\widehat{\nu}_{\rightarrow j}(\sigma_j)$ is the distribution of σ_j in the graphical model obtained by removing all the factor nodes adjacent to j, except the one on its left (cf. Fig. 14.1.1).

This decomposition is interesting because the various messages can be computed iteratively. Consider for instance $\widehat{\nu}_{\rightarrow i+1}$. It is expressed in terms of $\widehat{\nu}_{\rightarrow i}$ as:

{eq:mespas1dising}
$$\widehat{\nu}_{\rightarrow i+1}(\sigma) \cong \sum_{\sigma'} \widehat{\nu}_{\rightarrow i}(\sigma')\ e^{\beta\sigma'\sigma + \beta B \sigma'} \ . \tag{14.5}$$

Furthermore, $\widehat{\nu}_{\rightarrow 1}$ is the uniform distribution over $\{+1, -1\}$: $\widehat{\nu}_{\rightarrow 1}(\sigma) = \frac{1}{2}$ for $\sigma = \pm 1$. Equation (14.5) allows to compute all the messages $\widehat{\nu}_{\rightarrow i}$, $i \in \{1, \dots, N\}$, in $O(N)$ operations. A similar procedure yields $\widehat{\nu}_{i\leftarrow}$ starting from the uniform distribution $\widehat{\nu}_{N\leftarrow}$ and computing recursively $\widehat{\nu}_{i-1\leftarrow}$ from $\widehat{\nu}_{i\leftarrow}$. Finally, Eq. (14.4) can be used to compute all the marginals $p(\sigma_j)$ in linear time.

All the messages are distributions over binary variables and can thus be parameterized by a single real number. One popular choice for such a parameterization is to use the log-likelihood ratio[44]

$$u_{\rightarrow i} \equiv \frac{1}{2\beta} \log \frac{\widehat{\nu}_{\rightarrow i}(+1)}{\widehat{\nu}_{\rightarrow i}(-1)} . \tag{14.6}$$

In statistical physics terms $u_{\rightarrow i}$ is an 'effective (or local) magnetic field': $\widehat{\nu}_{\rightarrow i}(\sigma) \cong e^{\beta u_{\rightarrow i}\sigma}$. Using this definition (and noticing that it implies $\widehat{\nu}_{\rightarrow i}(\sigma) = \frac{1}{2}(1 + \sigma \tanh(\beta u_{\rightarrow i}))$), Eq. (14.5) becomes:

{eq:mespas1dising2}
$$u_{\rightarrow i+1} = f\left(u_{\rightarrow i} + B\right) \ , \tag{14.7}$$

where the function $f(x)$ is defined as

{eq:hudef}
$$f(x) = \frac{1}{\beta} \mathrm{atanh}\left[\tanh(\beta)\tanh(\beta x)\right] \ . \tag{14.8}$$

The mapping $u \mapsto f(u + B)$ is differentiable with derivative bounded by $\tanh\beta < 1$. Therefore the fixed point equation $u = f(u + B)$ has a unique

[44] Notice that our definition differs by a factor $1/2\beta$ from the standard log-likelihood definitions in Statistics. This factor is introduced to make contact with statistical physics definitions.

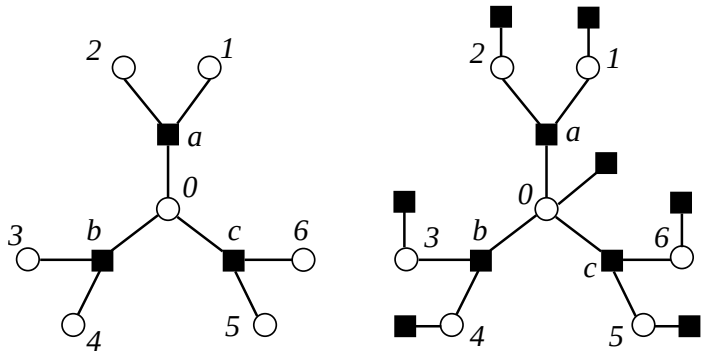


FIG. 14.2. Left: A simple parity check code with 7 variables and 3 checks. Right: the factor graph corresponding to the problem of finding the sent codeword, given a received message.

{fig:treecode1}

solution u_*, and $u_{\rightarrow i} \to u_*$ as $i \to \infty$. Consider a very long chain, and a node in the bulk $j \in [\varepsilon N, (1-\varepsilon)N]$. Then, as $N \to \infty$, both $u_{\rightarrow j}$ and $u_{j\leftarrow}$ converge to u^*, so that $\langle \sigma_j \rangle \to \tanh[\beta(2u^* + B)]$. This is the bulk magnetization. If on the other hand we consider a spin on the boundary we get a smaller magnetization $\langle \sigma_1 \rangle = \langle \sigma_N \rangle \to \tanh[\beta(u^* + B)]$.

Exercise 14.1 Use the recursion (14.7) to show that, when N and j go to infinity, $\langle \sigma_j \rangle = M + O(\lambda^j, \lambda^{N-j})$ where $M = \tanh(2u_* + B)$ and $\lambda = f'(u_* + B)$. Compare this with the treatment of the one-dimensional Ising model in Section 2.5.

The above method can be generalized to the computation of joint distributions of two or more variables. Consider for instance the joint distribution $p(\sigma_j, \sigma_k)$, for $k > j$. Since we already know how to compute the marginal $p(\sigma_j)$, it is sufficient to consider the conditional distribution $p(\sigma_k|\sigma_j)$. For each of the two values of σ_j, the conditional distribution of $\sigma_{j+1}, \cdots, \sigma_N$ takes a form analogous to Eq. (14.1) but with σ_j fixed. Therefore, the marginal $p(\sigma_k|\sigma_j)$ can be computed through the same algorithm as before. The only difference is in the initial condition that becomes $\widehat{\nu}_{\rightarrow j}(+1) = 1$, $\widehat{\nu}_{\rightarrow j}(-1) = 0$ (if we condition on $\sigma_j = +1$) and $\widehat{\nu}_{\rightarrow j}(+1) = 0$, $\widehat{\nu}_{\rightarrow j}(-1) = 1$ (if we condition on $\sigma_j = -1$).

Exercise 14.2 Compute the correlation function $\langle \sigma_j \sigma_k \rangle$, when $j, k \in [N\varepsilon, N(1-\varepsilon)]$ and $N \to \infty$. Check that, when $B = 0$, $\langle \sigma_j \sigma_k \rangle = (\tanh \beta)^{|j-k|}$. Find a simpler derivation of this last result.

14.1.2 *Example 2: a tree-parity-check code*

Our second example deals with a decoding problem. Consider the simple linear code whose factor graph is reproduced in Fig. 14.1.2, left frame. It has block-length $N = 7$ and codewords satisfy the 3 parity check equations:

$$x_0 \oplus x_1 \oplus x_2 = \mathsf{0}\,, \tag{14.9}$$
$$x_0 \oplus x_3 \oplus x_4 = \mathsf{0}\,, \tag{14.10}$$
$$x_0 \oplus x_5 \oplus x_6 = \mathsf{0}\,. \tag{14.11}$$

One of the codewords is sent through a BSC(p). Assume that the received message is $\underline{y} = (1,0,0,0,0,1,0)$. The conditional distribution for $\underline{x}$ to be the transmitted codeword, given the received $\underline{y}$ takes the usual form:

$$p(\underline{x}|\underline{y}) \cong \mathbb{I}(x_0 \oplus x_1 \oplus x_2 = \mathsf{0})\mathbb{I}(x_0 \oplus x_3 \oplus x_4 = \mathsf{0})\mathbb{I}(x_0 \oplus x_5 \oplus x_6 = \mathsf{0}) \prod_{i=0}^{6} Q(y_i|x_i)\ ,$$

where $Q(\mathsf{0}|\mathsf{0}) = Q(\mathsf{1}|\mathsf{1}) = 1-p$ and $Q(\mathsf{1}|\mathsf{0}) = Q(\mathsf{0}|\mathsf{1}) = p$. The corresponding factor graph is drawn in Fig. 14.1.2, right frame.

In order to implement symbol MAP decoding, cf. Chapter 6, we need to compute the conditional distribution of each bit. The computation is straightforward but it is illuminating to recast it as a message passing procedure, similar to the one in the Ising chain example. Consider for instance bit x_0. We start from the boundary. In the absence of the check a, the marginal of x_1 would be $\nu_{1\to a} = (1-p, p)$ (we use here the convention of writing distributions $\nu(x)$ over a binary variable as two dimensional vectors $(\nu(\mathsf{0}), \nu(\mathsf{1}))$). This is interpreted as a message sent from variable 1 to check a.

Variable 2 sends an analogous message $\nu_{2\to a}$ to a (in the present example, this happens to be equal to $\nu_{1\to a}$). Knowing these two messages, we can compute the contribution to the marginal probability distribution of variable x_0 coming from the part of the factor graph containing the whole branch connected to x_0 through the check a:

{eq:BPCodeExample}
$$\widehat{\nu}_{a\to 0}(x_0) \cong \sum_{x_1,x_2} \mathbb{I}(x_0 \oplus x_1 \oplus x_2 = \mathsf{0})\ \nu_{1\to a}(x_1)\nu_{2\to a}(x_2)\ . \tag{14.12}$$

Clearly, $\widehat{\nu}_{a\to 0}(x_0)$ is the marginal distribution of x_0 in the modified factor graph that does not include either factor node b or c, and in which the received symbol y_0 has been erased. This is analogous to the messages $\widehat{\nu}_{\to j}(\sigma_j)$ used in the Ising chain example. The main difference is that the underlying factor graph is no longer a line, but a tree. As a consequence, the recursion (14.12) is no longer linear in the incoming messages. Using the rule (14.12), and analogous ones for $\widehat{\nu}_{b\to 0}(x_0)$, $\widehat{\nu}_{c\to 0}(x_0)$, we obtain:

$$\widehat{\nu}_{a\to 0} = (p^2 + (1-p)^2,\ 2p(1-p))\,,$$
$$\widehat{\nu}_{b\to 0} = (p^2 + (1-p)^2,\ 2p(1-p))\,,$$
$$\widehat{\nu}_{c\to 0} = (2p(1-p),\ p^2 + (1-p)^2)\,.$$

The marginal probability distribution of the variable x_0 is finally obtained by taking into account the contributions of each subtree, together with the channel output for bit x_0:

$$p(x_0) \cong Q(y_0|x_0)\,\widehat{\nu}_{a\to 0}(x_0)\widehat{\nu}_{b\to 0}(x_0)\widehat{\nu}_{c\to 0}(x_0)$$
$$\cong \big(2p^2(1-p)[p^2+(1-p)^2]^2,\ \ 4p^2(1-p)^3[p^2+(1-p)^2]\big)$$

In particular, the MAP symbol decoding of the symbol x_0 is always $x_0 = 0$ in this case, for any $p < 1/2$.

An important fact emerges from this simple calculation. Instead of performing a summation over $2^7 = 128$ configurations, we were able to compute the marginal at x_0 doing 6 summations (one for every factor node a, b, c and for every value of x_0), each one over 2 summands, cf. Eq. (14.12). Such complexity reduction was achieved by merely rearranging the order of sums and multiplications in the marginal computation.

Exercise 14.3 Show that the message $\nu_{0\to a}(x_0)$ is equal to $(1/2, 1/2)$, and deduce that $p(x_1) \cong ((1-p)^2, p^2)$.

14.2 Belief Propagation on tree graphs

{se:BPtrees}

We shall define belief propagation and analyze it in the simplest possible setting: tree graphical models. In this case it solves several computational problems in an efficient and distributed fashion.

14.2.1 *Three problems*

Let us consider a graphical model such that the associated factor graph is a tree (we shall call it a **tree-graphical model**). We use the same notations as in Section 9.1.1. The model describes N random variables $(x_1, \dots, x_N) \equiv \underline{x}$ taking values in a finite alphabet $\mathcal{X}$, whose joint probability distribution has the form

$$p(\underline{x}) = \frac{1}{Z}\prod_{a=1}^{M}\psi_a(\underline{x}_{\partial a})\,. \tag{14.13}$$

where $\underline{x}_{\partial a} \equiv \{x_i \,|\, i \in \partial a\}$. The set $\partial a \subseteq [N]$, of size $|\partial a|$, contains all variables involved in constraint a. We always use indices $i, j, k, \dots$ for the variables and $a, b, c, \dots$ for the function nodes. The set of indices ∂i involves all function nodes a connected to i.

When the factor graph has no loop the following are among the basic problems that can be solved efficiently with a message-passing procedure:

1. Compute the marginal distributions of one variable, $p(x_i)$, or the joint distribution of a small number of variables.
2. Sample from $p(\underline{x})$, i.e. draw independent random configurations $\underline{x}$ with distribution $p(\underline{x})$.
3. Compute the partition function Z, or equivalently, in statistical physics language, the free-entropy.

These three tasks can be accomplished using belief propagation, which is the obvious generalization of the procedure exemplified in the previous section.

14.2.2 *The BP equations*

Belief propagation is an iterative 'message passing' algorithm. The basic variables on which it acts are messages associated with directed edges on the factor graph. For each edge (i, a) (where i is a variable node and a a function node) there are, at the t-th iteration, two messages $\nu_{i\to a}^{(t)}$ and $\widehat{\nu}_{a\to i}^{(t)}$. Messages take values in the space of probability distributions over the single variable space $\mathcal{X}$. Therefore $\nu_{i\to a}^{(t)} = \{\nu_{i\to a}^{(t)}(x_i) : x_i \in \mathcal{X}\}$, with $\nu_{i\to a}^{(t)}(x_i) \geq 0$ and $\sum_{x_i} \nu_{i\to a}^{(t)}(x_i) = 1$.

In tree-graphical models, the messages converge when $t \to \infty$ to fixed point values (see Theorem 14.1). These coincide with single variable marginals in modified graphical models, as we saw in the two examples of the previous section. More precisely $\nu_{i\to a}^{(\infty)}(x_i)$ is the marginal distribution of variable x_i in a modified graphical model which does not include the factor a (i.e. the product in Eq. (14.13) does not include a). Analogously $\widehat{\nu}_{a\to i}^{(\infty)}(x_i)$ is the distribution of x_i in a graphical model where all factors in ∂i, except a, have been erased.

Messages are updated through local computations at the nodes of the factor graph. By *local* we mean that a given node updates the outgoing messages on the basis of incoming ones at the previous iterations. This is a characteristic feature of message passing algorithms, while different algorithms in this family differ in the precise form of the update equations. The **belief propagation (BP)**, or **sum-product** update rules, are:

$$\nu_{j\to a}^{(t+1)}(x_j) \cong \prod_{b\in\partial j\setminus a} \widehat{\nu}_{b\to j}^{(t)}(x_j)\,, \tag{14.14}$$

$$\widehat{\nu}_{a\to j}^{(t)}(x_j) \cong \sum_{\underline{x}_{\partial a\setminus j}} \psi_a(\underline{x}_{\partial a}) \prod_{k\in\partial a\setminus j} \nu_{k\to a}^{(t)}(x_k)\,, \tag{14.15}$$

where $\setminus$ denotes set subtraction. It is understood that, when $\partial j\setminus a$ is an empty set, $\nu_{j\to a}(x_j)$ is the uniform distribution. Similarly, if $\partial a\setminus j$ is empty, then $\widehat{\nu}_{a\to j}(x_j) = \psi_a(x_j)$. A pictorial illustration of these rules is provided in Fig. 14.2.2. A BP fixed point of these equations is a set of t-independent messages $\nu_{i\to a}^{(t)} = \nu_{i\to a}$, $\widehat{\nu}_{a\to i}^{(t)} = \widehat{\nu}_{a\to i}$ which satisfy Eqs. (14.14), (14.15). From these one obtains $2|\mathcal{E}|$ equations (one equation for each oriented edge of the factor graph) relating $2|\mathcal{E}|$ messages. We will often refer to these fixed point conditions as to the **BP equations**.

After t iterations, one can estimate the marginal distribution $p(x_i)$ of variable i using the set of *all* incoming messages. The BP estimate is:

$$\nu_i^{(t)}(x_i) \cong \prod_{a\in\partial i} \widehat{\nu}_{a\to i}^{(t-1)}(x_i)\,. \tag{14.16}$$

In writing the update rules, we are assuming that the update is done in parallel at all the variable nodes, then in parallel at all function nodes and so on. Clearly, in this case, the iteration number must be incremented either at variable nodes or at factor nodes, but not necessarily at both. This is what happens in Eqs. (14.14),

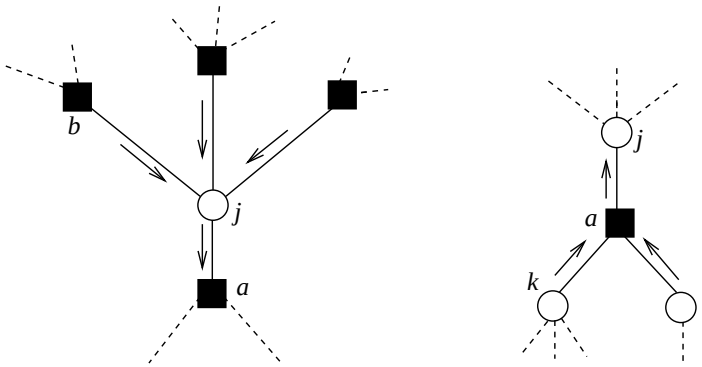


FIG. 14.3. Left: portion of the factor graph involved in the computation of $\nu^{(t+1)}_{j\to a}(x_j)$. This message is a function of the 'incoming messages' $\widehat{\nu}^{(t)}_{b\to j}(x_j)$, with $b \neq a$. Right: portion of the factor graph involved in the computation of $\widehat{\nu}^{(t)}_{a\to j}(x_j)$. This message is a function of the 'incoming messages' $\nu^{(t)}_{k\to a}(x_k)$, with $k \neq j$. {fig:BPiter_gen}

(14.15). Other update schedules are possible and sometimes useful. For the sake of simplicity we shall however stick to the parallel one introduced above.

In order to fully define the algorithm, we need to specify an initial condition. It is a widespread habit to set initial messages to the uniform distribution over $\mathcal{X}$ (i.e. $\nu^{(0)}_{i\to a}(x_i) = 1/|\mathcal{X}|$). On the other hand, it can be useful to explore several distinct (random) initial conditions. This can be done defining some probability measure P over the space $\mathfrak{M}(\mathcal{X})$ of distributions over $\mathcal{X}$ (i.e. the $|\mathcal{X}|$-dimensional simplex) and taking $\nu^{(0)}_{i\to a}(\,\cdot\,)$ iid random variables with distribution P.

Among all message passing algorithms, BP is uniquely characterized by the property of computing exact marginals on tree-graphical models. {thm:BPtrees}

Theorem 14.1. (BP is exact on trees) *Consider a tree-graphical model with diameter t_* (which means that t_* is the maximum distance between any two variable nodes). Then*

1. *Irrespective of the initial condition, the BP update (14.14), (14.15) converges after at most t_* iterations. In other words, for any edge (ia), and any $t > t_*$ $\nu^{(t)}_{i\to a} = \nu^{*}_{i\to a}$, $\widehat{\nu}^{(t)}_{a\to i} = \widehat{\nu}^{*}_{a\to i}$.*
2. *The fixed point messages provide the exact marginals: for any variable node i, and ant $t > t_*$, $\nu^{(t)}_{i}(x_i) = p(x_i)$.*

Proof: As exemplified in the previous Section, on tree factor graphs BP is just a clever way to organize the sum over configurations to compute marginals. In this sense the theorem is obvious.

Let us sketch a formal proof, leaving a few details to the reader. Given a directed edge $u \to v$, we define $\mathbb{T}(u \to v)$ as the sub-tree rooted on this edge. This is the subtree containing all nodes w which can be connected to v by a

non-reverting path whose last step is $u \to v$. Let $t_*(u \to v)$ be the *depth* of $\mathbb{T}(u \to v)$ (the maximal distance from a leaf to u). Consider the graphical model obtained by retaining only nodes in the subtree $\mathbb{T}(u \to v)$ (if v is a factor node, then it must be removed, and if it is a variable node, it must be retained). We will show that, for any edge (u, v), and any number of iterations $t > t_*(u \to v)$, the message $\nu^{(t)}_{u\to v}$ (or $\widehat{\nu}^{(t)}_{u\to v}$ if u is a factor node) coincides with the marginal distribution of the root variable with respect to this sub-tree graphical model. In other words, for tree graphs the interpretation of BP messages in terms of modified marginals is correct.

This claim is proved by induction on the tree depth $t_*(u \to v)$. The base step of the induction is trivial. Assume $u = i$ to be a variable node, and $v = a$ a factor node. Then $\mathbb{T}(i \to a)$ is the graph formed by the unique node i. By definition, for any $t \geq 1$, $\nu^{(t)}_{i\to a}(x_i) = 1/|\mathcal{X}|$ is the uniform distribution, which coincides with the marginal of the trivial graphical model associated to $\mathbb{T}(i \to a)$.

The induction step is easy as well. Assuming the claim to be true for $t_*(u \to v) \leq \tau$, one has to show that it holds when $t_*(u \to v) = \tau+1$. To this end assume again $u = i$ to be a variable node and $v = a$ a factor node, take any $t > \tau+1$ and compute $\nu^{(t+1)}_{i\to a}(x_i)$ using Eqs. (14.14), (14.15) in terms of messages $\nu^{(t)}_{j\to b}(x_j)$ in the subtrees for $b \in \partial i \setminus a$ and $j \in \partial b \setminus i$. By the induction hypothesis, and since the depth of the sub-tree $T(j \to b)$ is at most τ, $\nu^{(t)}_{j\to b}(x_j)$ is the root marginal in such a subtree. It turns out that, combining the marginals at roots of subtrees $\mathbb{T}(j \to b)$ using Eqs. (14.14), (14.15), one obtains the marginal at the root of $\mathbb{T}(i \to a)$. This proves the claim. □

14.2.3 *Correlations and energy*

The use of BP is not limited to computing one variable marginals. Suppose we want to compute the joint probability distribution $p(x_i, x_j)$ of two variables x_i and x_j. Since BP already enables to compute $p(x_i)$, this task is equivalent to computing the conditional distribution $p(x_j \mid x_i)$. Given a model that factorizes as in Eq. (14.13), the conditional distribution of $\underline{x} = (x_1, \dots, x_N)$ given $x_i = x$ takes the form

$$p(\underline{x}|x_i = x) \cong \prod_{a=1}^{M} \psi_a(\underline{x}_{\partial a})\, \mathbb{I}(x_i = x)\,. \tag{14.17}$$

In other words, it is sufficient to add to the original graph a new function node of degree 1 connected to variable node i, which fixes $x_i = x$. One can then run BP on the modified factor graph and obtain estimates $\nu^{(t)}_j(x_j|x_i = x)$ for the conditional marginal of x_j.

This strategy is easily generalized to the joint distribution of any number m of variables. The complexity grows however exponentially in the number of variables involved, since we have to condition over $|\mathcal{X}|^{m-1}$ possible assignments.

Happily, for tree-graphical models, the marginal distribution of any number of variables admits an explicit expression in terms of messages. Let F_R be a

subset of function nodes, V_R be the subset of variable nodes adjacent to F_R, R the induced subgraph, and $\underline{x}_R$ the corresponding variables. Without loss of generality, we shall assume R to be connected. Further, denote by ∂R the subset of function nodes that are not in F_R, but are adjacent to a variable node in V_R.

Then, for $a \in \partial R$ there exists a unique $i \in \partial a \cap V_R$, that we denote by $i(a)$. It then follows immediately from Theorem 14.1, and from the characterization of messages proved there that the joint distribution of variables in R is

$$p(\underline{x}_R) = \frac{1}{Z_R} \prod_{a \in F_R} \psi_a(\underline{x}_{\partial a}) \prod_{a \in \partial R} \widehat{\nu}^*_{a \to i(a)}(x_i(a)) \,, \tag{14.18}$$

where $\widehat{\nu}^*_{a \to i}(\,\cdot\,)$ are the fixed point BP messages.

Exercise 14.4 Let us use the above result to write the joint distribution of variables along a path in a tree factor graph. Consider two variable nodes i, j, and let $R = (V_R, F_R, E_R)$ be the subgraph induced by nodes on the path between i, and j. For any function node $a \in R$, denote by $i(a)$, $j(a)$ the variable nodes in R that are adjacent to a. Show that the joint distribution of the variables along this path, $\underline{x}_R = \{x_l : l \in V_R\}$, takes the form.

$$p(\underline{x}_R) = \frac{1}{Z_R} \prod_{a \in F_R} \widehat{\psi}_a(x_{i(a)}, x_{j(a)}) \prod_{l \in V_R} \widehat{\psi}_l(x_l) \,. \tag{14.19}$$

In other words $p(\underline{x}_R)$ factorizes according to the subgraph R. Write expressions for the compatibility functions $\widehat{\psi}_a(\,\cdot\,,\,\cdot\,)$, $\widehat{\psi}_l(\,\cdot\,)$ in terms of the original compatibility functions and of the messages to nodes in W.

A particularly useful case is the computation of the internal energy. In physics problems, the compatibility functions in Eq. (14.13) take the form $\psi_a(\underline{x}_{\partial a}) = e^{-\beta E_a(\underline{x}_{\partial a})}$, where β is the inverse temperature and $E_a(\underline{x}_{\partial a})$ is the energy function characterizing constraint a. Of course, any graphical model can be written in this form (allowing for $E_a(\underline{x}_{\partial a}) = +\infty$ in the case of hard constraints), adopting for instance the convention $\beta = 1$, that we will use hereafter. The internal energy U is the expectation value of the total energy:

$$U = -\sum_{\underline{x}} p(\underline{x}) \sum_{a=1}^{M} \log \psi_a(\underline{x}_{\partial a}) \,. \tag{14.20}$$

This can be computed in terms of BP messages using Eq. (14.18) with $F_R = \{a\}$. If we further use Eq. (14.14) to express products of check-to-variable messages in terms of variable-to-check ones, we get

$$U = -\sum_{a=1}^{M} \frac{1}{Z_a} \sum_{\underline{x}_{\partial a}} \psi_a(\underline{x}_{\partial a}) \log \psi_a(\underline{x}_{\partial a}) \prod_{i \in \partial a} \nu^*_{i \to a}(x_j) \;, \tag{14.21}$$

{eq:BP_energy}

where $Z_a \equiv \sum_{\underline{x}_{\partial a}} \psi_a(\underline{x}_{\partial a}) \prod_{i\in\partial a} \nu^*_{i\to a}(x_j)$. Notice that in this expression the internal energy is a sum of 'local' terms, one for each compatibility function.

On loopy graph Eqs. (14.18) and (14.21) are no longer valid, and indeed BP does not necessarily converge to fixed point messages $\{\nu^*_{i\to a}, \widehat{\nu}^*_{a\to i}\}$. However one can replace fixed point messages with BP messages after any number t of iterations and take these as *definitions* of the BP estimates for the corresponding quantities. From Eq. (14.18) one obtains an estimate of the joint distribution of a subset of variables, call it $\nu^{(t)}(\underline{x}_R)$, and from (14.21) an estimate of the internal energy.

{sec:TreeVariational}

14.2.4 *Entropy*

Remember that the entropy of a distribution p over $\mathcal{X}^V$ is defined as $H[p] = -\sum_{\underline{x}} p(\underline{x}) \log p(\underline{x})$. In a tree graphical model the entropy, like the internal energy, has a simple expression in terms of local quantities. This follows from an important decomposition property. Let us denote by $p_a(\underline{x}_{\partial a})$ the marginal probability distribution of all the variables involved in the compatibility function a, and by $p_i(x_i)$ the marginal probability distribution of variable x_i.

Theorem 14.2 *In a tree graphical model, the joint probability distribution $p(\underline{x})$ of all the variables can be written in terms of the marginals $p_a(\underline{x}_{\partial a})$ and $p_i(x_i)$ as:*

{eq:entrop_tree_thm}

$$p(\underline{x}) = \prod_{a\in F} p_a(\underline{x}_{\partial a}) \prod_{i\in V} p_i(x_i)^{1-|\partial i|} \; . \tag{14.22}$$

Proof: A simple proof is by induction on the number M of factors. Relation (14.22) holds for $M = 1$ (since the degrees ∂i are all equal to one). Let us assume that it is valid for any factor graph with up to M factors, and consider a specific factor graph G with $M+1$ factors. Since G is a tree, it contains at least one factor node such that all its adjacent variable nodes have degree 1, except at most one of them. Call such a factor node a, and let i be the only neighbor with degree larger than one (the case in which no such neighbor exists is treated analogously). Further, let $\underline{x}_\sim$ be the vector of variables in ∂a that are not in $\partial a \setminus i$. Then, the Markov property together with Bayes rule yields

$$\mathbb{P}(\underline{x}) = \mathbb{P}(\underline{x}_\sim)\mathbb{P}(\underline{x}|\underline{x}_\sim) = \mathbb{P}(\underline{x}_\sim)\mathbb{P}(\underline{x}_{\partial a\setminus i}|x_i) = \mathbb{P}(\underline{x}_\sim) p_a(\underline{x}_{\partial a}) p_i(x_i)^{-1} \; . \tag{14.23}$$

It is easy to check that $\mathbb{P}(\underline{x}_\sim)$ factorizes according to the subgraph obtained by removing the factor node a as well as the variable nodes in $\partial a\setminus i$. In fact it can be written as the product of the compatibility function in $G\setminus a$ times an additional factor due to the sum over $\underline{x}_{\partial a\setminus i}$. Since the degree of i in the reduced graph is smaller by one, and using the induction hypothesis, we get

$$\mathbb{P}(\underline{x}_\sim) = \prod_{b\in F\setminus a} p_b(\underline{x}_{\partial b}) \prod_{j\in V\setminus i} p_j(x_j)^{1-|\partial j|} \, p_i(x_i)^{2-|\partial i|} \; . \tag{14.24}$$

The proof is completed by putting together Eqs. (14.23) and (14.24). □

As an immediate consequence of (14.22), the entropy of a tree graphical model can be expressed as sums of local terms:

{eq:BP_entropy}
$$H[p] = -\sum_{a\in F} p_a(\underline{x}_{\partial a}) \log p_a(\underline{x}_{\partial a}) - \sum_{i\in V}(1-|\partial i|)\, p_i(x_i)\log p_i(x_i)\ . \qquad (14.25)$$

It is also easy to express the free-entropy $\Phi = \log Z$ in terms of local quantities. Recalling that $\Phi = H[p] - U[p]$ ($U[p]$ is here the internal energy given by Eq. (14.21)) we get $\Phi = \mathbb{F}[p]$, where

$$\mathbb{F}[p] = -\sum_{a\in F} p_a(\underline{x}_{\partial a}) \log\left\{\frac{p_a(\underline{x}_{\partial a})}{\psi_a(\underline{x}_{\partial a})}\right\} - \sum_{i\in V}(1-|\partial i|)p_i(x_i)\log p_i(x_i)\ . \qquad (14.26)$$

Expressing local marginals in terms of messages, via Eq. (14.18), we can in turn write the free-entropy as a function of the fixed point messages. We shall introduce the function $\mathbb{F}_*(\underline{\nu})$, that computes the free-entropy in terms of $2|E|$ messages $\underline{\nu} = \{\nu_{i\to a}(\,\cdot\,), \widehat{\nu}_{a\to i}(\,\cdot\,)\}$:

{eq:BP_free_entropy}
$$\mathbb{F}_*(\underline{\nu}) = \sum_{a\in F}\mathbb{F}_a(\underline{\nu}) + \sum_{i\in V}\mathbb{F}_i(\underline{\nu}) - \sum_{(ia)\in E}\mathbb{F}_{ia}(\underline{\nu}) \qquad (14.27)$$

where:

{eq:BP_free_entropy2}
$$\mathbb{F}_a(\underline{\nu}) = \log\left[\sum_{\underline{x}_{\partial a}}\psi_a(\underline{x}_{\partial a})\prod_{i\in\partial a}\nu_{i\to a}(x_i)\right],\quad \mathbb{F}_i(\underline{\nu}) = \log\left[\sum_{x_i}\prod_{b\in\partial i}\widehat{\nu}_{b\to i}(x_i)\right],$$
$$\mathbb{F}_{ai}(\underline{\nu}) = \log\left[\sum_{x_i}\nu_{i\to a}(x_i)\widehat{\nu}_{a\to i}(x_i)\right]. \qquad (14.28)$$

It is not hard to show that, evaluating this functional on the BP fixed point $\underline{\nu}^*$, one gets $\mathbb{F}_*(\underline{\nu}^*) = \mathbb{F}[p] = \Phi$ thus recovering the correct free-entropy. The function $\mathbb{F}_*(\underline{\nu})$ defined in (14.27) is known as the **Bethe free-entropy** (when multiplied by a factor $-1/\beta$, it is called the **Bethe free energy**). The above observations are important enough to be highlighted in a Theorem.

Theorem 14.3. (Bethe free-entropy is exact on trees) *Consider a tree graphical model. Let $\{p_a, p_i\}$ denote its local marginals, and $\underline{\nu}^* = \{\nu^*_{i\to a}, \widehat{\nu}^*_{a\to i}\}$ be the fixed point BP messages. Then $\Phi = \log Z = \mathbb{F}[p] = \mathbb{F}_*(\underline{\nu}^*)$.*

Notice that in the above statement we have used the correct local marginals in $\mathbb{F}[\,\cdot\,]$ and the fixed point messages in $\mathbb{F}_*(\,\cdot\,)$. In Section 14.4 we will reconsider the Bethe free-entropy for more general graphical models, and regard it as functions over the space of all 'possible' marginals/messages.

Exercise 14.5 Consider the satisfiability instance in Fig. 14.4, left. Show by exhaustive enumeration that it has only two satisfying assignments, $\underline{x} = (0,1,1,1,0)$ and $(0,1,1,1,1)$. Re-derive this result using BP. Namely, compute the entropy of the uniform measure over satisfying assignments, and check that its value is indeed $\log 2$. The BP fixed point is shown in Fig. 14.4, right.

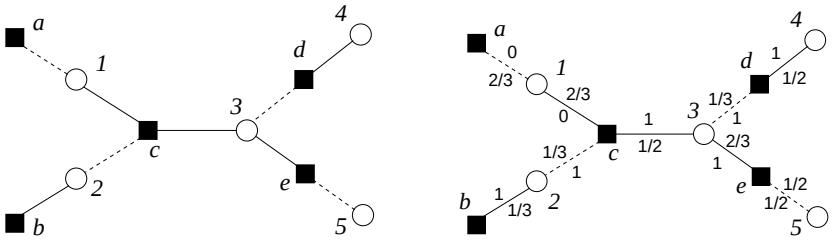


fig:BPSATsimple

FIG. 14.4. Left: the factor graph of a small satisfiability instance with 5 variables and 5 clauses. A dashed line means that the variable appears negated in the adjacent clause. Right: the set of fixed point BP messages for the uniform measure over solutions of this instance. All messages are normalized, and we show their weight on the value "True". For any edge (a, i) (a being the clause and i the variable), the weight corresponding to the message $\widehat{\nu}_{a\to i}$ is shown above the edge, and the weight corresponding to $\nu_{i\to a}$ below the edge.

{ex:BPPHI_mod}

Exercise 14.6 In many systems some of the function nodes have degree 1 and amount to a local redefinition of the reference measure over $\mathcal{X}$. It is then convenient to single out these factors. Let us write $p(\underline{x}) \cong \prod_{a\in F}\psi_a(\underline{x}_{\partial a})\prod_{i\in V}\psi_i(x_i)$, where the second product runs over degree-1 function nodes (indexed by the adjacent variable node), while the factors ψ_a have degree at least 2. In the computation of $\mathbb{F}_*$, the introduction of ψ_i adds N extra factor nodes and subtracts N extra 'edge' terms corresponding to the edge between the variable node i and the function node corresponding to ψ_i. Show that these two effects cancel, and that the net effect is to replace the variable node contribution in Eq. (14.27) with

$$\mathbb{F}_i(\underline{\nu}) = \log\left[\sum_{x_i}\psi_i(x_i)\prod_{a\in\partial i}\widehat{\nu}_{a\to i}(x_i)\right]. \qquad (14.29)$$

The problem of sampling from the distribution $p(\underline{x})$ over the large-dimensional space $\mathcal{X}^N$ reduces to the one of computing one-variable marginals of $p(\underline{x})$, conditional on a subset of the other variables. In other words, if we have a black box that computes $p(x_i|\underline{x}_U)$ for any subset $U \subseteq V$, it can be used to sample a random configuration $\underline{x}$. The standard procedure for doing this is called **sequential importance sampling**. Let us describe it in the case of tree-graphical models, using BP as this 'black box':

BP-Guided Sampling (Graphical model (G, ψ))

initialize BP messages;
initialize $U = \emptyset$;
for $t = 1, \dots, N$:
    run BP until convergence;
    choose $i \in V \setminus U$;
    compute the BP marginal $\nu_i(x_i)$;
    choose x_i^* distributed according to ν_i;
    fix $x_i = x_i^*$ and set $U \leftarrow U \cup \{i\}$;
    add a factor $\mathbb{I}(x_i = x_i^*)$ to the graphical model;
end
return $\underline{x}^*$.

14.2.5 *Pairwise models* {sec:Pairwise}

Pairwise graphical models, i.e. graphical models such that all factor nodes have degree 2, form an important class. Such a model can be conveniently represented as an ordinary graph $G = (V, E)$ over variable nodes. An edge joins two variable each time they are the arguments of the same compatibility function. The corresponding probability distribution reads

$$p(\underline{x}) = \frac{1}{Z} \prod_{(ij)\in E} \psi_{ij}(x_i, x_j)\,. \tag{14.30}$$

Function nodes can be identified with edges $(ij) \in E$.

In this case belief propagation may be described as operating directly on G. Further, one of the two types of messages can be easily eliminated: we shall work uniquely with variable-to-function messages, that we will denote as $\nu_{i\to j}^{(t)}(x_i)$, a shortcut for $\nu_{i\to(ij)}^{(t)}(x_i)$. The BP updates then read

$$\nu_{i\to j}^{(t+1)}(x_i) \cong \prod_{l\in\partial i\setminus j} \sum_{x_l} \psi_{il}(x_i, x_l)\, \nu_{l\to i}^{(t)}(x_l)\,. \tag{14.31}$$

Simplified expressions can be derived in this case for the joint distribution of several variables, cf. Eq. (14.18), as well as for the free-entropy, cf. Eq. (14.27). We leave this as an exercise for the reader. ⋆

14.3 Optimization: Max-Product and Min-Sum {se:MaxProd}

Message passing algorithms are not limited to computing marginals. Imagine that you are given a probability distribution $p(\,\cdot\,)$ as in Eq. (14.13), and you are asked to find a configuration[45] $\underline{x}$ which maximizes the probability $p(\underline{x})$. This task is important for many applications, ranging from MAP estimation (e.g. in image reconstruction) to word MAP decoding.

It is not hard to devise a message passing algorithm adapted to this task, which correctly solves the problem on trees.

[45] Such a configuration, $\underline{x}_*$, such that $p(\underline{x}) \le p(\underline{x}_*)$ for any $\underline{x}$, is called a **mode** of $p(\,\cdot\,)$.

14.3.1 *Max-marginals*

The role of marginal probabilities is here played by the so-called **max-marginals**

$$M_i(x_i^*) = \max_{\underline{x}} \{p(\underline{x}) : x_i = x_i^*\} . \tag{14.32}$$

In the same way as sampling and computing partition functions can be reduced to computing marginals, optimization can be reduced to computing max-marginals. In other words, given a black box that computes max-marginals, optimization can be performed efficiently.

Consider first the simpler case in which the max-marginals are non-degenerate, i.e. for each $i \in V$, there exist x_i^* such that $M_i(x_i^*) > M_i(x_i)$ (strictly) for any $x_i \neq x_i^*$. Then the unique maximizing configuration is given by $\underline{x}^* = (x_1^*, \dots, x_N^*)$.

In the general case, the following 'decimation' procedure, which is closely related to the BP-guided sampling algorithm in Section 14.2.4, returns one of the maximizing configurations. Choose an ordering of the variables, say $(1, \dots, N)$. Compute $M_1(x_1)$ and let x_1^* be one of the values maximizing[46] it: $x_1^* = \arg\max M_1(x_1)$. Fix x_1 to take this value, i.e. modify the graphical model by introducing the factor $\mathbb{I}(x_1 = x_1^*)$ (this corresponds to considering the conditional distribution $p(\underline{x}|x_1 = x_1^*)$). Compute $M_2(x_2)$ for the new model, fix x_2 to $x_2^* = \arg\max M_2(x_2)$ and iterate this procedure fixing sequentially all the x_i's.

14.3.2 *Message passing*

It is clear from the above that max-marginals only need to be computed up to a multiplicative normalization. We shall therefore stick to our convention of denoting by $\cong$ equality between max-marginals up to an overall normalization. Adapting the message passing update rules to the computation of max-marginals is not hard: it is sufficient to replace sums with maximizations. This yields the following **Max-Product** update rules:

$$\nu_{i\to a}^{(t+1)}(x_i) \cong \prod_{b\in\partial i\setminus a} \widehat{\nu}_{b\to i}^{(t)}(x_i) , \tag{14.33}$$

$$\widehat{\nu}_{a\to i}^{(t)}(x_i) \cong \max_{\underline{x}_{\partial a\setminus i}} \left\{ \psi_a(\underline{x}_{\partial a}) \prod_{j\in\partial a\setminus i} \nu_{j\to a}^{(t)}(x_j) \right\} . \tag{14.34}$$

The fixed-point conditions for this recursion are also called **Max-Product equations**. As in BP, it is understood that, when $\partial j \setminus a$ is an empty set, $\nu_{j\to a}(x_j) \cong 1$ is the uniform distribution. Similarly, if $\partial a \setminus j$ is empty, then $\widehat{\nu}_{a\to j}(x_j) \cong \psi_a(x_j)$. After any number of iterations, an estimate of the max-marginals is obtained as follows

[46] Here and below, $\arg\max F(x)$ denotes the set of values of x that maximize $F(x)$, and when we write $x^* = \arg\max F(x)$, what we really mean is $x^* \in \arg\max F(x)$.

$$\nu_i^{(t)}(x_i) \cong \prod_{a\in\partial i} \widehat{\nu}_{a\to i}^{(t-1)}(x_i)\,. \tag{14.35}$$

As in the case of BP, the main motivation for the above updates comes from the analysis of graphical models on trees.

{th:maxprod}

Theorem 14.4. (Max-Product is exact on trees) *Consider a tree graphical model with diameter t_*. Then*

1. *Irrespective of the initialization, the Max-Product updates (14.33), (14.34) converge after at most t_* iterations. In other words, for any edge (i,a), and any $t > t_*$ $\nu_{i\to a}^{(t)} = \nu_{i\to a}^*$, $\widehat{\nu}_{a\to i}^{(t)} = \widehat{\nu}_{a\to i}^*$.*
2. *The max-marginals are estimated correctly, i.e. for any variable node i, and ant $t > t_*$, $\nu_i^{(t)}(x_i) = M_i(x_i)$.*

The proof follows closely the one of Theorem 14.1, and is left as an exercise for the reader.

{ex:distribut}

Exercise 14.7 The crucial property used both in both Theorems 14.1 and 14.4 is the distributive property of sum and max with respect to the product. Consider for instance a function of the form $f(x_1,x_2,x_3) = \psi_1(x_1,x_2)\psi_2(x_1,x_3)$. Then one can decompose the sum and max as

$$\sum_{x_1,x_2,x_3} f(x_1,x_2,x_3) = \sum_{x_1}\left[\left(\sum_{x_2}\psi_1(x_1,x_2)\right)\left(\sum_{x_3}\psi_2(x_1,x_3)\right)\right], \tag{14.36}$$

$$\max_{x_1,x_2,x_3} f(x_1,x_2,x_3) = \max_{x_1}\left[\left(\max_{x_2}\psi_1(x_1,x_2)\right)\left(\max_{x_3}\psi_2(x_1,x_3)\right)\right] \tag{14.37}$$

Formulate a general 'marginalization' problem (with the ordinary sum and product substituted by general operations with a distributive property) and describe a message passing algorithm that solves it on trees.

The Max-Product messages $\nu_{i\to a}^{(t)}(\,\cdot\,)$, $\widehat{\nu}_{a\to i}^{(t)}(\,\cdot\,)$ admit an interpretation which is analogous to the one of Sum-Product messages. For instance $\nu_{i\to a}^{(t)}(\,\cdot\,)$ is an estimate of the max-marginal of variable x_i with respect to the modified graphical model in which factor node a is removed from the graph. Along with the proof of Theorem 14.4, it is easy to show that, on a tree-graphical model, fixed point messages do indeed coincide with max-marginals of such modified graphical models.

The problem of finding the mode of a distribution that factorizes as in Eq. (14.13) has an alternative formulation, namely minimizing a cost (energy) function that can be written as the sum of local terms:

$$E(\underline{x}) = \sum_{a\in F} E_a(\underline{x}_{\partial a})\,. \tag{14.38}$$

The problems are mapped onto each other by writing $\psi_a(\underline{x}_{\partial a}) = e^{-\beta E_a(\underline{x}_{\partial a})}$ (with β some positive constant). A set of message passing rules that is better adapted to the last formulation is obtained by taking the logarithm of Eqs. (14.33), (14.34). The corresponding algorithm is called **Min-Sum**:

$$J_{i\to a}^{(t+1)}(x_i) = \sum_{b\in\partial i\setminus a} \widehat{J}_{b\to i}^{(t)}(x_i) + C_{i\to a}^{(t)} \,, \tag{14.39}$$

$$\widehat{J}_{a\to i}^{(t)}(x_i) = \min_{\underline{x}_{\partial a\setminus i}} \left[E_a(\underline{x}_{\partial a}) + \sum_{j\in\partial a\setminus i} J_{j\to a}^{(t)}(x_j) \right] + \widehat{C}_{a\to i}^{(t)} \,. \tag{14.40}$$

The corresponding fixed-point equations are also known in statistical physics as the **energetic cavity equations**. Notice that, since the Max-Product marginals are relevant up to a multiplicative constant, the Min-Sum messages are defined up to an overall additive constant. In the following we will choose the constant $C_{i\to a}^{(t)}$ (respectively $\widehat{C}_{a\to i}^{(t)}$) such that $\min_{x_i} J_{i\to a}^{(t+1)}(x_i) = 0$ (respectively $\min_{x_i} \widehat{J}_{a\to i}^{(t)}(x_i) = 0$). The analogous of the max-marginal estimate in Eq. (14.35) is provided by the following log-max-marginal

$$J_i^{(t)}(x_i) = \sum_{a\in\partial i} \widehat{J}_{a\to i}^{(t-1)}(x_i) + C_i^{(t)} \,. \tag{14.41}$$

In the case of tree graphical models, the minimum energy $U_* = \min_{\underline{x}} E(\underline{x})$ can be immediately written in terms of the fixed point messages $\{J_{i\to a}^*, \widehat{J}_{i\to a}^*\}$. We get indeed

$$U_* = \sum_a E_a(\underline{x}_{\partial a}^*) \,, \tag{14.42}$$

$$\underline{x}_{\partial a}^* = \arg\min_{\underline{x}_{\partial a}} \left\{ E_a(\underline{x}_{\partial a}) + \sum_{i\in\partial a} \widehat{J}_{i\to a}^*(x_i) \right\} . \tag{14.43}$$

In the case of non-tree graphs, this can be taken as a prescription to obtain a Max-Product estimate $U_*^{(t)}$ of the minimum energy. One has just to replace the fixed point messages in Eq. (14.43) with the ones obtained after t iterations. Finally, a minimizing configuration $\underline{x}^*$ can be obtained through the decimation procedure described in the previous Section.

{ex:minsum_ene_bis}

Exercise 14.8 Show that U_* is also given by $U_* = \sum_{a\in F}\epsilon_a + \sum_{i\in V}\epsilon_i - \sum_{(ia)\in E}\epsilon_{ia}$, where:

$$\epsilon_a = \min_{\underline{x}_{\partial a}}\left[E_a(\underline{x}_{\partial a}) + \sum_{j\in\partial a} J^*_{j\to a}(x_j)\right], \qquad \epsilon_i = \min_{x_i}\left[\sum_{a\in\partial i}\widehat{J}^*_{a\to i}(x_i)\right],$$
$$\epsilon_{ia} = \min_{x_i}\left[J^*_{i\to a}(x_i) + \widehat{J}^*_{a\to i}(x_i)\right]. \qquad (14.44)$$

Hints: (*i*) Define $x^*_i(a) = \arg\min\left[\widehat{J}^*_{a\to i}(x_i) + J^*_{i\to a}(x_i)\right]$, and show that the minima in Eqs. (14.44) are achieved for $x_i = x^*_i(a)$ (for ϵ_i and ϵ_{ai}), and for $\underline{x}^*_{\partial a} = \{x^*_i(a)\}_{i\in\partial a}$ (for ϵ_a); (*ii*) Show that $\sum_{(ia)}\widehat{J}^*_{a\to i}(x^*_i(a)) = \sum_i \epsilon_i$.

14.3.3 *Warning propagation*

{se:warning_prop}

A frequently encountered case is that of constraint satisfaction problems, where the energy function just counts the number of violated constraints:

$$E_a(\underline{x}_{\partial a}) = \begin{cases} 0 & \text{if constraint } a \text{ is satisfied,} \\ 1 & \text{otherwise.} \end{cases} \qquad (14.45)$$

The messages' structure can be simplified considerably in this case. More precisely, if the messages are initialized in such a way that $\widehat{J}^{(0)}_{a\to i} \in \{0,1\}$, this condition is preserved by the Min-Sum updates (14.40), (14.39) at any subsequent time. Let us prove this statement by induction. Suppose it holds up to time $t-1$. From Eq. (14.40) it follows that $J^{(t)}_{i\to a}(x_i)$ is a non-negative integer. Consider now Eq. (14.39). Since both $J^{(t)}_{j\to a}(x_j)$ and $E_a(\underline{x}_{\partial a})$ are integers, $\widehat{J}^{(t)}_{a\to i}(x_i)$, the minimum of the right hand side is a non-negative integer as well. Further, since for each $j \in \partial a \setminus i$ there exists x^*_j such that $J^{(t)}_{j\to a}(x^*_j) = 0$, the minimum in Eq. (14.39) is at most 1, which proves our claim.

This argument also shows that the outcome of the minimization in Eq. (14.39) only depends on which entries of the messages $J^{(t)}_{j\to a}(\,\cdot\,)$ are vanishing. If there exists an assignment x^*_j, such that $J^{(t)}_{j\to a}(x^*_j) = 0$ for each $j \in \partial a \setminus i$, and $E_a(x_i, \underline{x}^*_{\partial a\setminus i}) = 0$, then the value of the minimum is 0. Otherwise it is 1.

In other words, instead of keeping track of the messages $J_{i\to a}(\,\cdot\,)$, one can use their 'projections'

$$\mathtt{J}_{i\to a}(x_i) = \min\{1,\, J_{i\to a}(x_i)\}\ . \qquad (14.46)$$

{eq_gge_def}

Proposition 14.5 *Consider an optimization problem with cost function of the form (14.38) with $E_a(\underline{x}_{\partial a}) \in \{0,1\}$, and assume the Min-Sum algorithm to be initialized with $\widehat{J}_{a\to i}(x_i) \in \{0,1\}$ for all edges (i,a). Then, after any number of iterations, the function node-to-variable node messages coincide with the ones computed with the following update rules*

$$\mathtt{J}_{i\to a}^{(t+1)}(x_i) = \min\left\{1, \sum_{b\in\partial i\setminus a} \widehat{J}_{b\to i}^{(t)}(x_i) + C_{i\to a}^{(t)}\right\}, \tag{14.47}$$

$$\widehat{J}_{a\to i}^{(t)}(x_i) = \min_{\underline{x}_{\partial a\setminus i}}\left\{E_a(\underline{x}_{\partial a}) + \sum_{j\in\partial a\setminus i} \mathtt{J}_{j\to a}^{(t)}(x_j)\right\} + \widehat{C}_{a\to i}^{(t)}, \tag{14.48}$$

where $C_{i\to a}^{(t)}$, $\widehat{C}_{a\to i}^{(t)}$ *are normalization constants determined by* $\min_{x_i} \widehat{J}_{a\to i}(x_i) = 0$ *and* $\min_{x_i} \mathtt{J}_{i\to a}(x_i) = 0$.

Finally, the ground state energy takes the same form as (14.44), with $\mathtt{J}_{i\to a}(\,\cdot\,)$ *replacing* $J_{i\to a}(\,\cdot\,)$.

We shall call **warning propagation** the simplified Min-Sum algorithm with update equations (14.48), (14.47).

The name is due to the remark that the messages $\mathtt{J}_{i\to a}(\,\cdot\,)$ can be interpreted as the following warnings:

> $\mathtt{J}_{i\to a}(x_i) = 1 \;\to$ *"according to the set of constraints* $b \in \partial i \setminus a$*, the* i*-th variable should not take value* x_i*."*
>
> $\mathtt{J}_{i\to a}(x_i) = 0 \to$ *"according to the set of constraints* $b \in \partial i \setminus a$*, the* i*-th variable can take value* x_i*."*

Warning propagation provides a procedure for finding all direct implications of some partial assignment of the variables in a constraint satisfaction problem. For instance, in satisfiability it finds all implications found by unit clause propagation, cf. Section 10.2.

{se:BPloops}

14.4 Loopy BP

We have seen how message passing algorithms can be used to efficiently treat tree-graphical models. In particular they allow to exactly sample, compute marginals, partition functions, modes of distributions that factorize according to tree factor graphs. It would be very important for a number of applications to accomplish the same tasks when the underlying factor graph is no longer a tree.

It is tempting to use the BP equations in this more general context, hoping to get approximate results for large graphical models. We shall be dealing mostly with NP-hard problems, and there is no general guarantee of performance. Indeed, an important unsolved challenge is to identify classes of graphical models where the following questions could be answered:

1. Is there any set of messages $\{\nu_{i\to a}^*, \widehat{\nu}_{a\to i}^*\}$ that reproduces the local marginals of $p(\,\cdot\,)$ through Eq. (14.18), within some prescribed accuracy?
2. Do such messages correspond to an (approximate) fixed point of the BP update rules (14.14), (14.15)?
3. Do the BP update rules have at least one (approximate) fixed point? Is it unique?

4. Does such a fixed point have non-empty 'basin of attraction' with respect to Eqs. (14.14), (14.15)? Does this basin of attraction include all possible (or 'meaningful') initializations?

We shall not treat these questions in depth, as a general theory is lacking. We shall rather describe the (rather sophisticated) picture that has emerged, building on a mixture of physical intuition and methods, empirical observations, and rigorous proofs.

{ex_2DIsing_Bethe}

Exercise 14.9 Consider a ferromagnetic Ising model on the two dimensional grid with periodic boundary conditions (i.e. 'wrapped' on a torus), defined in Section 9.1.2, cf. Fig. 9.7. Ising spins σ_i, $i \in V$ are associated to the vertices of the grid, and interact along the edges:

$$p(\underline{\sigma}) = \frac{1}{Z}\, e^{\beta \sum_{(ij)\in E} \sigma_i \sigma_j}\,. \tag{14.49}$$

(*a*) Describe the associated factor graph.
(*b*) Write the BP equations.
(*c*) Look for a solution that is invariant under translation $\nu_{i\to a}(\sigma_i) = \nu(\sigma_i)$, $\widehat{\nu}_{a\to i}(\sigma_i) = \widehat{\nu}(\sigma_i)$: write the equations satisfied by $\nu(\,\cdot\,)$, $\widehat{\nu}(\,\cdot\,)$.
(*d*) Parameterize $\nu(\sigma)$ in terms of the log-likelihood $h = \frac{1}{2}\log\frac{\nu(+1)}{\nu(-1)}$ and show that h satisfies the equation $\tanh(\beta h) = \tanh(\beta)\tanh(3\beta h)$.
(*e*) Study this equation and show that, for $3\tanh\beta > 1$, it has three distinct solutions corresponding to three BP fixed points.
(*f*) Consider iterating the BP updates starting from a translation invariant initial condition. Does the iteration converge to a fixed point? Which one?
(*g*) Discuss the appearance of three BP fixed points in relation with the structure of the distribution $p(\underline{\sigma})$, and the paramagnetic-ferromagnetic transition. What is the approximate value of the critical temperature obtained from BP? Compare with the exact value $\beta_{\rm c} = \frac{1}{2}\log(1+\sqrt{2})$.
(*h*) What results does one obtain for an Ising model on a d-dimensional (instead of two-dimensional) grid?

14.4.1 *Bethe free-entropy and variational methods*

As we saw in Section 14.2.4, the free-entropy of a tree graphical model has a simple expression in terms of local marginals, cf. Eq. (14.26). We can use it in graphs with loops with the hope that it provides a good estimate of the actual free-entropy. In spirit this approach is similar to the 'mean field' free-entropy introduced in Chapter 2, although it differs from it in several respects.

In order to define precisely the Bethe free-entropy, we must first describe a space of 'possible' local marginals. A minimalistic approach is to restrict ourselves to the so-called 'locally consistent marginals'. A set of **locally consis-**

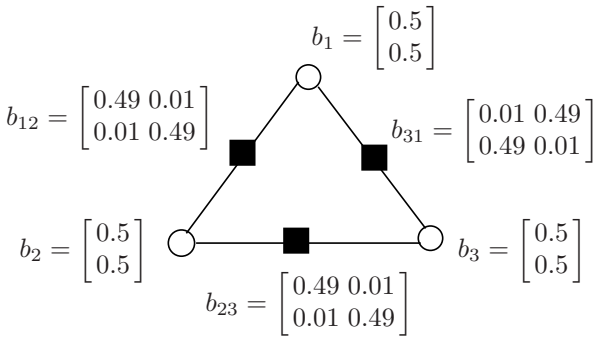


FIG. 14.5. A set of locally consistent marginals that cannot arise as the marginals of any global distribution.

{fig:FactorTriangle}

tent marginals is a collection of distributions $b_i(\,\cdot\,)$ over $\mathcal{X}$, for each $i \in V$, and $b_a(\,\cdot\,)$ over $\mathcal{X}^{\partial a}$ for each $a \in F$. Being distributions they must be non-negative, $b_i(x_i) \geq 0$ $b_a(\underline{x}_{\partial a}) \geq 0$, and they must satisfy the normalization condition

$$\sum_{x_i} b_i(x_i) = 1 \quad \forall i \in V\,, \qquad \sum_{\underline{x}_{\partial a}} b_a(\underline{x}_{\partial a}) = 1 \quad \forall a \in F\,. \tag{14.50}$$

To be 'locally consistent', they must satisfy the marginalization condition:

$$\sum_{\underline{x}_{\partial a \setminus i}} b_a(\underline{x}_{\partial a}) = b_i(x_i) \qquad \forall a \in F\,,\ \ \forall i \in \partial a\,. \tag{14.51}$$

Given a factor graph G, we shall denote the set locally consistent marginals as $\mathsf{LOC}(G)$, and the Bethe free-entropy will be defined as a real valued function on this space.

It is important to stress that, although the marginals of any probability distribution $p(\underline{x})$ over $\underline{x} = (x_1, \dots, x_N)$ must be locally consistent, the converse is not true: one can find sets of locally consistent marginals that do not correspond to any distribution. To stress this point, locally consistent marginals are sometimes called "**beliefs**".

Exercise 14.10 Consider the graphical model in Fig. 14.4.1, on binary variables (x_1, x_2, x_3), $x_i \in \{0, 1\}$. A set beliefs is written in the same figure in the vector/matrix form:

$$b_i = \begin{bmatrix} b_i(0) \\ b_i(1) \end{bmatrix} \quad ; \quad b_{ij} = \begin{bmatrix} b_{ij}(00) & b_{ij}(01) \\ b_{ij}(10) & b_{ij}(11) \end{bmatrix}\,. \tag{14.52}$$

Check that this set of beliefs is locally consistent but cannot be the marginals of any distribution $p(x_1, x_2, x_3)$.

Given a set of locally consistent marginals $\underline{b} = \{b_a, b_i\}$, we associate to it a **Bethe free-entropy** exactly as in Eq. (14.26)

$$\mathbb{F}[\underline{b}] = -\sum_{a\in F} b_a(\underline{x}_{\partial a}) \log\left\{\frac{b_a(\underline{x}_{\partial a})}{\psi_a(\underline{x}_{\partial a})}\right\} - \sum_{i\in V}(1-|\partial i|)\, b_i(x_i)\log b_i(x_i)\,. \quad (14.53)$$

The analogy with naive mean field suggests that stationary points (and in particular maxima) of the Bethe free-entropy should play an important role. This is partially confirmed by the following result.

{propo:BPvsFreeEnergy}

Proposition 14.6 *Assume $\psi_a(\underline{x}_{\partial a}) > 0$ for each a and $\underline{x}_{\partial a}$. Then the stationary points of the Bethe free-entropy $\mathbb{F}[\underline{b}]$ are in one-to-one correspondence with the fixed points of BP.*

As it will appear from the proof, this correspondence between BP fixed points and stationary points of $\mathbb{F}[\underline{b}]$ is completely explicit.

Proof: We want to check stationarity with respect to variations of $\underline{b}$ within the set $\mathsf{LOC}(G)$, that is defined by the constraints (14.50), (14.51), as well as $b_a(\underline{x}_{\partial a}) \geq 0$, $b_i(x_i) \geq 0$. We thus introduce a set of Lagrange multipliers $\underline{\lambda} = \{\lambda_i,\, i\in V;\ \lambda_{ai}(x_i),\, (a,i)\in E,\, x_i\in\mathcal{X}\}$, where λ_i corresponds to the normalization of $b_i(\,\cdot\,)$ and $\lambda_{ai}(x_i)$ to the marginal of b_a coinciding with b_i. We then define the Lagrangian

$$\mathcal{L}(\underline{b},\underline{\lambda}) = \mathbb{F}[\underline{b}] - \sum_{a\in F}\lambda_i\left[\sum_{x_i} b_i(x_i) - 1\right] - \sum_{(ia),x_i}\lambda_{ai}(x_i)\left[\sum_{\underline{x}_{\partial a\setminus i}} b_a(\underline{x}_{\partial a}) - b_i(x_i)\right]. \quad (14.54)$$

Notice that we did not introduce a Lagrange multiplier for the normalization of $b_a(\underline{x}_{\partial a})$ as this follows from the two constraints already enforced. The stationarity conditions with respect to b_i and b_a imply:

$$b_i(x_i) \cong e^{-\frac{1}{|\partial i|-1}\sum_{a\in\partial i}\lambda_{ai}(x_i)}\,, \quad b_a(\underline{x}_{\partial a}) \cong \psi_a(\underline{x}_{\partial a})\, e^{-\sum_{i\in\partial a}\lambda_{ai}(x_i)}\,. \quad (14.55)$$

The Lagrange multipliers must be chosen in such a way that Eq. (14.51) is fulfilled. Any such set of Lagrange multipliers yields a stationary point of $\mathbb{F}[\underline{b}]$. Once the $\lambda_{ai}(x_j)$ are found, the computation of the normalization constants in these expressions fixes λ_i. Conversely, any stationary point corresponds to a set of Lagrange multipliers satisfying the stated condition.

It remains to show that sets of Lagrange multipliers such that $\sum_{\underline{x}_{\partial a\setminus i}} b_a(\underline{x}_{\partial a}) = b_i(x_i)$ are in one-to-one correspondence with BP fixed points. In order to see this, define the messages

$$\nu_{i\to a}(x_i) \cong e^{-\lambda_{ai}(x_i)}\,, \quad \widehat{\nu}_{a\to i}(x_i) \cong \sum_{\underline{x}_{\partial a\setminus i}} \psi_a(\underline{x}_{\partial a})\, e^{-\sum_{j\in\partial a\setminus i}\lambda_{aj}(x_j)}\,. \quad (14.56)$$

It is clear from the definition that such messages satisfy

$$\widehat{\nu}_{a\to i}(x_i) \cong \sum_{\underline{x}_{\partial a\setminus i}} \psi_a(\underline{x}_{\partial a}) \prod_{j\in\partial a\setminus i} \nu_{i\to a}(x_i)\ . \tag{14.57}$$

Further, using the second of Eqs. (14.55) together with (14.56) we get $\sum_{\underline{x}_{\partial a\setminus i}} b_a(\underline{x}_{\partial a}) \cong \nu_{i\to a}(x_i)\widehat{\nu}_{a\to i}(x_i)$. On the other hand, from the first of Eqs. (14.55) together with (14.56), we get $b_i(x_i) \cong \prod_b \nu_{i\to b}(x_i)^{\frac{1}{|\partial i|-1}}$. The marginalization condition thus implies

$$\prod_{b\in\partial i} \nu_{i\to b}(x_i)^{\frac{1}{|\partial i|-1}} \cong \nu_{i\to a}(x_i)\widehat{\nu}_{a\to i}(x_i)\,. \tag{14.58}$$

Taking the product of these equalities for $a \in \partial i\backslash b$, and eliminating $\prod_{a\in\partial i\backslash b} \nu_{i\to a}(x_i)$ from the resulting equation (which is possible if $\psi_a(\underline{x}_{\partial a}) > 0$), we get

$$\nu_{i\to b}(x_i) \cong \prod_{a\in\partial i\backslash b} \widehat{\nu}_{a\to i}(x_i)\,. \tag{14.59}$$

At this point we recognize in Eqs. (14.57), (14.59) the fixed point condition for BP, cf. Eqs. (14.14), (14.14). Conversely, given any solution of Eqs. (14.57), (14.59) one can define a set of Lagrange multipliers using the first of Eqs. (14.56). It follows that from the fixed point condition that the second Eq. (14.56) is fulfilled as well, and that the marginalization condition holds. □

An important consequence of this proposition is the existence of BP fixed points.

{cor:bp_fp}

Corollary 14.7 *Assume $\psi_a(\underline{x}_a) > 0$ for each a and $\underline{x}_{\partial a}$. Then BP has at least one fixed point.*

Proof: Since $\mathbb{F}[\underline{b}]$ is bounded and continuous in $\mathsf{LOC}(G)$ (which is closed), it takes its maximum at some point $\underline{b}^* \in \mathsf{LOC}(G)$. Using the condition $\psi_a(\underline{x}_a) > 0$ it is easy to see that such a maximum is reached in the relative interior of $\mathsf{LOC}(G)$, i.e. that $b^*_a(\underline{x}_{\partial a}) > 0$, $b^*_i(x_i) > 0$ strictly. As a consequence $\underline{b}^*$ must be a stationary point and therefore, by Proposition 14.6, there is a BP fixed point associated with it. □

The 'variational principle' provided by Proposition 14.6 is particularly suggestive as it is analogous to naive mean field bounds. For practical applications it is sometimes more convenient to use the free-entropy functional $\mathbb{F}_*(\underline{\nu})$ (14.27). This can be regarded as a function from the space of messages to reals: $\mathbb{F} : \mathfrak{M}(\mathcal{X})^{|\vec{E}|} \to \mathbb{R}$ (remember that $\mathfrak{M}(\mathcal{X})$ denotes the set of measures over $\mathcal{X}$, and $\vec{E}$ is the set of directed edges in the factor graph)[47]. It enjoys the following principle.

{propo:BPvsFreeEnergy2}

Proposition 14.8 *The stationary points of the Bethe free-entropy $\mathbb{F}_*(\underline{\nu})$ are fixed points of belief propagation. Conversely, any fixed point $\underline{\nu}$ of belief propagation such that $\mathbb{F}_*(\underline{\nu})$ is finite, is also a stationary point of $\mathbb{F}_*(\underline{\nu})$.*

[47] On a tree $\mathbb{F}_*(\underline{\nu})$ is (up to a change of variables) the Lagrangian dual of $\mathbb{F}(\underline{b})$

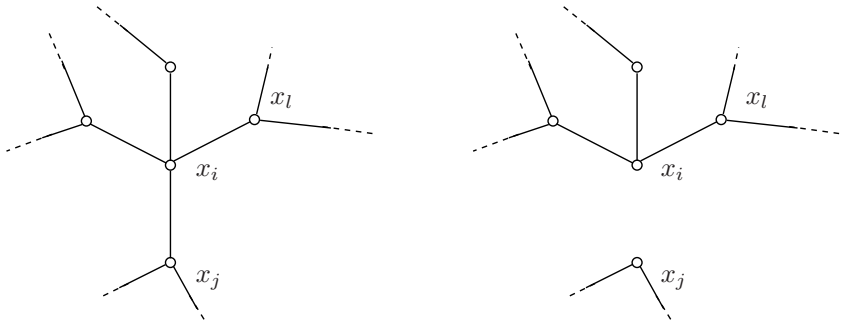


FIG. 14.6. Neighborhood of node i in a pairwise graphical model. Right: the modified graphical model used to define message $\nu_{i\to j}(x_i)$. {fig:Cavity1}

The proof is simple calculus and is left to the reader.

It turns out that for tree graphs and for unicyclic graphs, $\mathbb{F}[\underline{b}]$ is convex, and the above results then prove the existence and unicity of BP fixed points. But for general graphs $\mathbb{F}[\underline{b}]$ is non-convex and may have multiple stationary points.

14.4.2 *Correlations*

What is the origin of the error made when using BP in an arbitrary graph with loops, and under what conditions can it be small? In order to understand this point, let us consider for notational simplicity a pairwise graphical model, cf. Eq. (14.2.5). The generalization to other models is straightforward. Taking seriously the probabilistic interpretation of messages, we want to compute the marginal distribution $\nu_{i\to j}(x_i)$ of x_i in the modified graphical model that does not include the factor $\psi_{ij}(x_i, x_j)$ (see Fig. 14.6). Call $p_{\partial i\setminus j}(\underline{x}_{\partial i\setminus j})$ the joint distribution of all variables in $\partial i \setminus j$ in the model where all the factors $\psi_{il}(x_i, x_l)$, $l \in \partial i$, have been removed. Then:

$$\nu_{i\to j}(x_i) \cong \sum_{\underline{x}_{\partial i\setminus j}} \prod_{l\in\partial i\setminus j} \psi_{il}(x_i, x_l) p_{\partial i\setminus j}(\underline{x}_{\partial i\setminus j})\,. \tag{14.60}$$

Comparing this expression to the BP equations, cf. Eq. (14.31), we deduce that the messages $\{\nu_{i\to j}\}$ solve these equations if

$$p_{\partial i\setminus j}(\underline{x}_{\partial i\setminus j}) = \prod_{l\in\partial i\setminus j} \nu_{l\to i}(x_l)\,. \tag{14.61}$$

We can think that this happens when two conditions are fulfilled:

1. Under $p_{\partial i\setminus j}(\,\cdot\,)$, the variables $\{x_l : l \in \partial i\setminus j\}$ are independent: $p_{\partial i\setminus j}(\underline{x}_{\partial i\setminus j}) = \prod_{l\in\partial i\setminus j} p_{\partial i\setminus j}(x_l)$.
2. The marginal of each of these variables under $p_{\partial i\setminus j}(\,\cdot\,)$ is equal to the corresponding message $\nu_{l\to i}(x_l)$. In other words the two graphical models obtained by removing all the compatibility functions that involve x_i

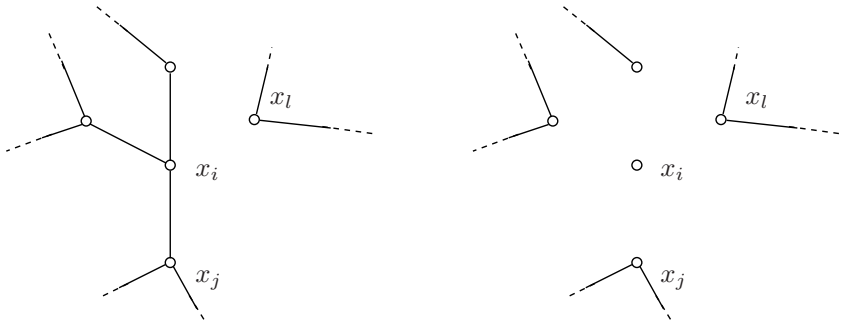


FIG. 14.7. Left: Modified graphical model used to define $\nu_{l\to i}(x_l)$. Right: Modified graphical model corresponding to the cavity distribution of the neighbors of i, $p_{\partial i\backslash j}(\underline{x}_{\partial i\backslash j})$.

{fig:Cavity2}

(namely, the model $p_{\partial i\backslash j}(\,\cdot\,)$) and by removing only $\psi_{il}(x_i, x_l)$ must have the same marginal for variable x_l, cf. Fig. 14.7.

These two conditions are obviously fulfilled for tree graphical models. They are also approximately fulfilled if correlations among variables $\{x_l : l \in \partial i\}$ are 'small' under $p_{\partial i\backslash j}(\,\cdot\,)$. As we have seen, in many cases of practical interest (LDPC codes, random K-SAT, etc.) the factor graph is locally tree-like. In other words, when removing node i, the variables $\{x_l : l \in \partial i\}$ are with high probability far apart from each other. This suggests that, in such models, the conditions 1, 2 above may indeed hold in the large size limit, provided far apart variables are weakly correlated. A simple illustration of this phenomenon is provided in the exercises below. The following Chapters will investigate further this property and discuss how to cope with cases in which it does not hold.

Exercise 14.11 Consider the anti-ferromagnetic Ising model on a ring, with variables $(\sigma_1, \dots, \sigma_N) \equiv \underline{\sigma}$, $\sigma_i \in \{+1, -1\}$ and distribution

$$p(\underline{\sigma}) = \frac{1}{Z}\, e^{-\beta \sum_{i=1}^{N} \sigma_i \sigma_{i+1}} \tag{14.62}$$

where $\sigma_{N+1} \equiv \sigma_i$. This is a pairwise graphical model whose graph G is the ring over N vertices.

1. Write the BP update rules for this model (see Section 14.2.5).
2. Express the update rules in terms of log-likelihoods $h_{i\rightarrow}^{(t)} \equiv \frac{1}{2} \log \frac{\nu_{i\rightarrow i+1}^{(t)}(+1)}{\nu_{i\rightarrow i+1}^{(t)}(-1)}$, and $h_{\leftarrow i}^{(t)} \equiv \frac{1}{2} \log \frac{\nu_{i\rightarrow i-1}^{(t)}(+1)}{\nu_{i\rightarrow i-1}^{(t)}(-1)}$.
3. Show that, for any $\beta \in [0, \infty)$, and any initialization, the BP updates converge to the unique fixed point $h_{\leftarrow i} = h_{i\rightarrow} = 0$ for all i.
4. Assume $\beta = +\infty$ and N even. Show that any set of log-likelihoods of the form $h_{i\rightarrow} = (-1)^i a$, $h_{\leftarrow i} = (-1)^i b$, with $a, b \in [-1, 1]$, is a fixed point.
5. Consider now $\beta = \infty$ and N odd, and show that the only fixed point is $h_{\leftarrow i} = h_{i\rightarrow} = 0$. Find an initialization of the messages such that BP does not converge to this fixed point.

Exercise 14.12 Consider the ferromagnetic Ising model on a ring with magnetic field. This is defined through the distribution

$$p(\underline{\sigma}) = \frac{1}{Z}\, e^{\beta \sum_{i=1}^{N} \sigma_i \sigma_{i+1} + B \sum_{i=1}^{N} \sigma_i} \tag{14.63}$$

where $\sigma_{N+1} \equiv \sigma_i$. Notice that with respect to the previous exercise we changed a sign in the exponent.

1,2. As in the previous exercise.
3. Show that, for any $\beta \in [0, \infty)$, and any initialization, the BP updates converge to the unique fixed point $h_{\leftarrow i} = h_{i\rightarrow} = h_*(\beta, B)$ for all i.
4. Let $\langle \sigma_i \rangle$ be the expectation of spin σ_i with respect to the measure $p(\,\cdot\,)$, and $\langle \sigma_i \rangle_{\rm BP}$ the corresponding BP estimate. Show that $|\langle \sigma_i \rangle - \langle \sigma_i \rangle_{\rm BP}| = O(\lambda^N)$ for some $\lambda \in (0, 1)$.

14.5 General message passing algorithms

Both the sum-product and max-product (or min-sum) algorithms are instances of a more general class of **message passing** algorithms. All the algorithms in this family share some common features that we now highlight.

Given a factor graph, a message-passing algorithm is defined by the following ingredients:

1. An alphabet of messages M. This can be either continuous or discrete. The algorithm operates on messages $\nu_{i\to a}^{(t)}, \widehat{\nu}_{a\to i}^{(t)} \in \mathsf{M}$ associated with the directed edges in the factor graph.
2. Update functions $\Psi_{i\to a} : \mathsf{M}^{|\partial i\setminus a|} \to \mathsf{M}$ and $\Phi_{a\to i} : \mathsf{M}^{|\partial a\setminus i|} \to \mathsf{M}$ that describe how to update messages.
3. An initialization, i.e. a mapping from the directed edges in the factor graph to M (it can be a random mapping). We shall denote by $\nu_{i\to a}^{(0)}$, $\widehat{\nu}_{a\to i}^{(0)}$ the image of such a mapping.
4. A decision rule, i.e. a local function from messages to a space of 'decisions' among which we are interested to make a choice. Since we will be mostly interested in computing marginals (or max-marginals), we shall assume the decision rule to be given by a family of functions $\widehat{\Psi}_i : \mathsf{M}^{|\partial i|} \to \mathfrak{M}(\mathcal{X})$.

Notice the characterizing feature of message passing algorithms: messages outgoing from a node are functions of messages incoming on the same node through the other edges.

Given these ingredients, a message passing algorithm with parallel updating is defined as follows. Assign the values of initial messages $\nu_{i\to a}^{(0)}$, $\widehat{\nu}_{a\to i}^{(0)}$ according to the initialization rule. Then, for any $t \ge 0$, update messages through local operations at variable/check nodes as follows:

$$\nu_{i\to a}^{(t+1)} = \Psi_{i\to a}(\{\widehat{\nu}_{b\to i}^{(t)} : b \in \partial i \setminus a\}), \tag{14.64}$$

$$\widehat{\nu}_{a\to i}^{(t)} = \Phi_{a\to i}(\{\nu_{j\to a} : j \in \partial a \setminus i\}). \tag{14.65}$$

Finally, after a pre-established number of iterations t, take the decision using the rules $\widehat{\Psi}_i$, namely return

$$\nu_i^{(t)}(x_i) = \widehat{\Psi}_i(\{\widehat{\nu}_{b\to i}^{(t-1)} : b \in \partial i\})(x_i). \tag{14.66}$$

Many variants are possible concerning the update schedule. For instance in sequential updating one can pick up a directed edge uniformly at random and compute the corresponding message. Another possibility is to generate a random permutation of the edges and update the messages according to this permutation. We shall not discuss these 'details', but the reader should be aware that they can be important in practice: some update schemes may converge better than others.

Exercise 14.13 Recast the sum-product and min-sum algorithms in the general message passing framework. In particular, specify the messages alphabet, the update and decision rules.

{sec:ProbAnaBP}

14.6 Probabilistic analysis

In the following Chapters we shall repeatedly be concerned with the analysis of message passing algorithms on random graphical models. In this context messages become random variables, and their distribution can be characterized in the large system limit, as we will now see.

{sec:AssumptionsDE}

14.6.1 *Assumptions*

Before proceeding, it is necessary to formulate a few technical assumptions under which the approach works. The basic idea is that, in a 'random graphical model', distinct nodes should be essentially independent. Specifically, we shall consider below a setting which already includes many cases of interest; it is easy to extend our analysis to even more general situations.

A **random graphical model** is a (random) probability distribution on $\underline{x} = (x_1, \dots, x_N)$ of the form[48]

$$p(\underline{x}) \cong \prod_{a \in F} \psi_a(\underline{x}_{\partial a}) \prod_{i \in V} \psi_i(x_i) \ , \tag{14.67}$$

where the factor graph $G = (V, F, E)$, and the various factors ψ_a, ψ_i, are independent random variables. More precisely, we assume that the factor graph is distributed according to one of the ensembles $\mathbb{G}_N(K, \alpha)$ or $\mathbb{D}_N(\Lambda, P)$ (see Chapter 9).

The random factors are assumed to be distributed as follows. For any given degree k, we are given a list of possible factors $\psi^{(k)}(x_1, \dots, x_k; \widehat{J})$, indexed by a 'label' $\widehat{J} \in \mathsf{J}$, and a distribution $P^{(k)}$ over the set of possible labels J. For each function node $a \in F$ of degrees $|\partial a| = k$, a label $\widehat{J}_a$ is drawn with distribution $P^{(k)}$, and the function $\psi_a(\,\cdot\,)$ is taken equal to $\psi^{(k)}(\,\cdot\,; \widehat{J}_a)$. Analogously, the factors ψ_i are drawn from a list of possible $\{\psi(\,\cdot\,; J)\}$, indexed by the label J which is drawn from a distribution P. The random graphical model is fully characterized by the graph ensemble, the set of distributions $P^{(k)}$, P, and lists of factors $\{\psi^{(k)}(\,\cdot\,; \widehat{J})\}$, $\{\psi(\,\cdot\,; J)\}$.

We need to make some assumptions on the message update rules. Specifically, we assume that the variable-to-function node update rules $\Psi_{i \to a}$ depend on $i \to a$ only through $|\partial i|$ and J_i, and the function-to-variable node update rules $\Phi_{a \to i}$ depend on $a \to i$ only through $|\partial a|$ and $\widehat{J}_a$. With a slight abuse of notation, we shall denote the update functions as:

$$\Psi_{i \to a}(\{\widehat{\nu}_{b \to i} : b \in \partial i \setminus a\}) = \Psi_l(\widehat{\nu}_1, \dots, \widehat{\nu}_l; J_i) \, , \tag{14.68}$$

$$\Phi_{a \to i}(\{\nu_{j \to a} : j \in \partial a \setminus i\}) = \Phi_k(\nu_1, \dots, \nu_k; \widehat{J}_a) \, , \tag{14.69}$$

where we let $l \equiv |\partial i| - 1$, $k \equiv |\partial a| - 1$, $\{\widehat{\nu}_1, \dots, \widehat{\nu}_l\} \equiv \{\widehat{\nu}_{b \to i} : b \in \partial i \setminus a\}$ and $\{\nu_1, \dots, \nu_k\} \equiv \{\nu_{j \to a} : j \in \partial a \setminus i\}$. A similar notation will be used for the decision rule $\widehat{\Psi}$.

[48] Notice that the factors ψ_i, $i \in V$ could have been included as degree 1 function nodes as we do in (14.13); including them explicitly yields a description of density evolution which is more symmetric between variables and factors, and applies more directly to decoding

Exercise 14.14 Let $G = (V, E)$ be a uniformly random graph with $M = N\alpha$ edges over N vertices, and let λ_i, $i \in V$ be iid random variables uniform in $[0, \bar{\lambda}]$. Recall that an independent set for G is a subset of the vertices $S \subseteq V$ such that if $i, j \in S$, then (ij) is not an edge. Consider the following weighted measure over independent sets

$$p(S) = \frac{1}{Z}\, \mathbb{I}(S \text{ is an independent set}) \prod_{i \in S} \lambda_i \,. \tag{14.70}$$

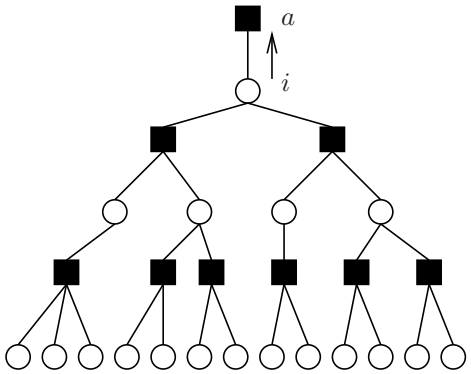
1. Write the distribution $p(S)$ as a graphical model with binary variables and define the corresponding factor graph.
2. Describe the BP algorithm to compute its marginals.
3. Show that this model is a random graphical model.

{sec:DE_eqs}

14.6.2 *Density evolution equations*

Consider a random graphical model, with factor graph $G = (V, F, E)$ and let (i, a) be a uniformly random edge in G. Let $\nu^{(t)}_{i\to a}$ be the message sent by the BP algorithm in iteration t along edge (i, a). We assume that the initial messages $\nu^{(0)}_{i\to a}$, $\widehat{\nu}^{(0)}_{a\to i}$ are iid random variables, with distribution independent of N. A considerable amount of information is contained in the distribution of $\nu^{(t)}_{i\to a}$ and $\widehat{\nu}^{(t)}_{a\to i}$ with respect to the model realization. We are interested in characterizing these distributions in the large system limit $N \to \infty$. Our analysis will assume that both the message alphabet M and the node labels alphabet J are subsets of $\mathbb{R}^d$ for some fixed d, and that the update functions $\Psi_{i\to a}$, $\Phi_{a\to i}$ are continuous with respect to the usual topology of $\mathbb{R}^d$.

It is convenient to introduce the **directed neighborhood** of radius t of the directed edge $i \to a$: $\mathsf{B}_{i\to a,t}(G)$. This is defined as the subgraph of G that includes all the variable nodes which can be reached from i through a non-reversing path of length at most t, whose first step *is not* the edge (i, a). It includes as well all the function nodes connected only to the above specified variable nodes- see Fig. 14.8. Let us consider, to be definite, the case where G is a random factor graph from the $\mathbb{D}_N(\Lambda, P)$ ensemble. Then $\mathsf{B}_{i\to a,t}(F)$ converges in distribution to a tree with a well defined distribution as $N \to \infty$. This is completely analogous to what happens for *undirected* neighborhoods treated in Section 9.5. The corresponding tree ensemble $\mathbb{T}^{\rightarrow}_t(\Lambda, P)$ is defined as follows. For $t = 0$, the tree is formed by a single variable node. To construct a tree of radius $t \geq 1$, first construct a tree of radius $t - 1$. Then, for each of the variable nodes i at distance $t - 1$ from the root, draw an independent integer l_i with distribution λ_l and connect i to $l_i - 1$ new function nodes. Finally, for each of the newly added function nodes a, draw an independent random integer K_a with distribution ρ_k and connect a to $K_a - 1$ new variable nodes. For illustrative reasons, we shall occasionally add a 'root edge' as $i \to a$ in Fig. 14.8.



{fig:FactorTree}

FIG. 14.8. A radius 2 directed neighborhood $\mathsf{B}_{i\to a,2}(F)$.

Exercise 14.15 Consider a random graph from the regular $\mathbb{D}_N(\Lambda, P)$ ensemble with $\Lambda_2 = 1$, $P_3 = 1$ (each variable node has degree 2 and each function node degree 3). The three possible radius-1 directed neighborhoods appearing in such factor graph are depicted in Fig. 14.9.

(*a*) Show that the probability that a given edge (i, a) has neighborhoods as in (B) or (C) is $O(1/N)$.

(*b*) Deduce that $\mathsf{B}_{i\to a,1}(F) \stackrel{\mathrm{d}}{\to} \mathsf{T}_1$ where T_1 is distributed according to the tree model $\mathbb{T}^{\to}_1(2, 3)$ (i.e. it is the tree on Fig. 14.9, (A)).

(*c*) Discuss the case of a radius-t neighborhood.

For our purposes it is necessary to include in the description of the neighborhood $\mathsf{B}_{i\to a,t}(F)$, the value of the labels $J_i, \widehat{J}_b$ for function nodes b in this neighborhood. It is understood that the tree model $\mathbb{T}^{\to}_t(\Lambda, P)$ includes labels as well: these have to be drawn as iid random variables independent of the tree and with the same distribution as in the original graphical model.

Now consider the message $\nu^{(t)}_{i\to a}$. This is a function of the factor graph G, of the labels $\{J_j\}$, $\{\widehat{J}_b\}$ and of the initial condition $\{\nu^{(0)}_{j\to b}\}$. However, a moment of thought shows that its dependence on G and on the labels occurs only through the radius-$(t + 1)$ directed neighborhood $\mathsf{B}_{i\to a,t+1}(F)$. Its dependence on the initial condition is only through the messages $\nu^{(0)}_{j\to b}$ for $j, b \in \mathsf{B}_{i\to a,t}(F)$.

In view of the above discussion, let us pretend for a moment that the neighborhood of (i, a) *is* a random tree T_{t+1} with distribution $\mathbb{T}^{\to}_{t+1}(\Lambda, P)$. We define $\nu^{(t)}$ to be the message passed through the root edge of such a random neighborhood after t message passing iterations. Since $\mathsf{B}_{i\to a,t+1}(F)$ converges in distribution to the tree T_{t+1}, we find that[49] $\nu^{(t)}_{i\to a} \stackrel{\mathrm{d}}{\to} \nu^{(t)}$ as $N \to \infty$.

[49]The mathematically suspicious reader may wonder about the topology we are assuming on the message space space. In fact no assumption is necessary if the distribution of labels $J_i, \widehat{J}_a$

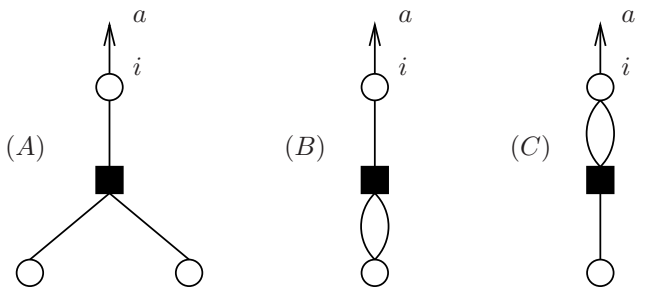


FIG. 14.9. The three possible radius–1 directed neighborhoods in a random factor graph from the regular $\mathbb{D}_N(2,3)$ graph ensemble. {fig:FactorTreeRadius1}

We have shown that, as $N \to \infty$, the distribution of $\nu^{(t)}_{i\to a}$ converges to the one of a well defined (N-independent) random variable $\nu^{(t)}$. The next step consists in finding a recursive characterization of $\nu^{(t)}$. Consider a random tree from the $\mathbb{T}^{\rightarrow}_r(\Lambda, P)$ ensemble and let $j \to b$ be an edge directed towards the root, at distance d from it. The directed subtree rooted at $j \to b$ is distributed according to $\mathbb{T}^{\rightarrow}_{r-d}(\Lambda, P)$. Therefore the message passed through it after $r-d-1$ (or more) iterations is distributed as $\nu^{(r-d-1)}$. The degree of the root variable node i (including the root edge) has distribution λ_l. Each check node connected to i has a number of other neighbors (distinct from i) which is a random variable distributed according to ρ_k. These facts imply the following distributional equations for $\nu^{(t)}$ and $\widehat{\nu}^{(t)}$:

$$\nu^{(t+1)} \stackrel{\mathrm{d}}{=} \Psi_l(\widehat{\nu}^{(t)}_1, \dots, \widehat{\nu}^{(t)}_l; J)\,, \qquad \widehat{\nu}^{(t)} \stackrel{\mathrm{d}}{=} \Phi_k(\nu^{(t)}_1, \dots, \nu^{(t)}_k; \widehat{J})\,. \quad (14.71)$$

Here $\widehat{\nu}^{(t)}_b$, $b \in \{1, \dots, l-1\}$ are independent copies of $\widehat{\nu}^{(t)}$, $\nu^{(t)}_j$, $j \in \{1, \dots, k-1\}$ are independent copies of $\nu^{(t)}$, l and k are independent random integers distributed, respectively, according to λ_l and ρ_k, $\widehat{J}$ is distributed as $P^{(k)}_{\widehat{J}}$ and J is distributed as P_J. It is understood that the recursion is initiated with $\nu^{(0)} \stackrel{\mathrm{d}}{=} \nu^{(0)}_{i\to a}$, $\widehat{\nu}^{(0)} \stackrel{\mathrm{d}}{=} \widehat{\nu}^{(0)}_{a\to i}$.

In information theory, the equations (14.71), or sometimes the sequence of random variables $\{\nu^{(t)}, \widehat{\nu}^{(t)}\}$, is referred to as **density evolution**. In probabilistic combinatorics, they are also called **recursive distributional equations**. We have proved the following characterization of the messages distribution:

Proposition 14.9 *Consider a random graphical model satisfying assumptions 1-4 in Section* 14.6.1. *Let $t \geq 0$ and (ia) be a uniformly random edge in the factor graph. Then, as $N \to \infty$, the message $\nu^{(t)}_{i\to a}$ ($\widehat{\nu}^{(t)}_{i\to a}$) converges in distribution to the random variable $\nu^{(t)}$ (respectively $\widehat{\nu}^{(t)}$) defined through the density evolution equations (14.71).*

is independent of N. If it is N dependent but converges, then the topology must be such that the messages updates are continuous with respect to it.

We shall discuss several applications of density evolution in the following Chapters. Here we just mention that density evolution allows to compute the asymptotic distribution of message passing decisions at a uniformly random site i. Recall that the general message passing decision after t iterations is taken using the rule (14.66), with $\widehat{\Psi}_i(\{\widehat{\nu}_b\}) = \widehat{\Psi}_l(\widehat{\nu}_1, \dots, \widehat{\nu}_l; J_i)$ (where $l \equiv |\partial i|$). Arguing as in the previous paragraphs it is easy to show that, in the large N limit, $\mu_i^{(t)} \stackrel{\mathrm{d}}{\to} \mu^{(t)}$, where the random variable $\mu^{(t)}$ is distributed according to:

$$\mu^{(t)} \stackrel{\mathrm{d}}{=} \widehat{\Psi}_l(\widehat{\nu}_1^{(t-1)}, \dots, \widehat{\nu}_l^{(t-1)}; J)\,. \tag{14.72}$$

As above $\widehat{\nu}_1^{(t-1)}, \dots, \widehat{\nu}_l^{(t-1)}$ are iid copies of $\widehat{\nu}^{(t-1)}$, J is an independent copy of the variable node label J_i, and l is a random integer distributed according to Λ_l.

14.6.3 *The replica symmetric cavity method*

{se:RS_cavity}

The replica symmetric (RS) cavity method of statistical mechanics adopts a point of view which is very close to the previous one, but less algorithmic. Instead of considering the BP update rules as an iterative message passing rule, it focuses on the fixed point BP equations themselves.

The idea is to compute the partition function recursively, by adding one variable node at a time. Equivalently one may think of taking one variable node out of the system and computing the change in the partition function. The name of the method comes exactly from this image: one digs a 'cavity' in the system.

As an example, take the original factor graph, delete the factor node a and all the edges incident on it. If the graph is a tree, this procedure separates it into $|\partial a|$ disconnected trees. Consider now the tree-graphical model described by the connected component containing the variable $j \in \partial a$. Denote the corresponding partition function, when the variable j is fixed to the value x_j, by $Z_{j\to a}(x_j)$. These partial partition functions can be computed iteratively as:

$$Z_{j\to a}(x_j) = \prod_{b\in\partial j\setminus a} \left[\sum_{\underline{x}_{\partial b\setminus j}} \psi_b(\underline{x}_{\partial b}) \prod_{k\in\partial b\setminus j} Z_{k\to b}(x_k) \right]\,. \tag{14.73}$$

{eq:Z_rs_recursion}

The equations obtained by letting $j \to b$ be a generic directed edge in G, are called **cavity equations**, or **Bethe equations**.

The cavity equations are mathematically identical to the BP equations, with two important conceptual differences: (*i*) One is naturally led to think that the equations (14.73) must have a fixed point, and to give special importance to it; (*ii*) The partial partition functions are unnormalized messages, and, as we will see in Chapter **??**, their normalization provides some useful information. The relation between BP messages and partial partition functions is

$$\nu_{j\to a}(x_j) = \frac{Z_{j\to a}(x_j)}{\sum_y Z_{j\to a}(y)}\,. \tag{14.74}$$

{eq:CavityNorm}

Within the cavity approach, the **replica symmetry assumption** consists in pretending that, for random graphical models as introduced above, and in the large N limit:

1. There exists a solution (or quasi-solution[50]) to these equations.
2. This solution provides good approximations of the marginals of the graphical model.
3. The messages in this solution are distributed according to a density evolution fixed point.

The last statement amounts to assuming that the normalized variable-to-factor messages $\nu_{i\to a}$, cf. Eq. (14.74), converge in distribution to a random variable ν, that solves the distributional equations:

$$\nu \stackrel{\mathrm{d}}{=} \Psi(\widehat{\nu}_1,\dots,\widehat{\nu}_{k-1};J)\,, \qquad \widehat{\nu} \stackrel{\mathrm{d}}{=} \Phi(\nu_1,\dots,\nu_{l-1};\widehat{J})\,. \tag{14.75}$$

Here we use the same notations as in Eq. (14.71): $\widehat{\nu}_b$, $b\in\{1,\dots,l-1\}$ are independent copies of $\widehat{\nu}^{(t)}$; $\nu_j^{(t)}$, $j\in\{1,\dots,k-1\}$ are independent copies of $\nu^{(t)}$; l and k are independent random integers distributed, respectively, according to λ_l and ρ_k; J, $\widehat{J}$ are distributed as the variable and function nodes labels J_i, $\widehat{J}_a$.

Using the distributions of ν and $\widehat{\nu}$, the expected Bethe free-entropy per variable $\mathbb{F}/N$ can be computed by taking the expectation of Eq. (14.27). The result is:

{eq:RS_free_entropy}

$$\mathbb{F}/N = \mathbb{F}_V + a\mathbb{F}_F - b\mathbb{F}_E \tag{14.76}$$

where a is the average number of function nodes per variable, and b is the average number of edges per variable: In the $\mathbb{D}_N(\Lambda,P)$ ensemble one has $a=\Lambda'(1)/P'(1)$ and $b=\Lambda'(1)$; Within the $\mathbb{G}_N(K,\alpha)$ ensemble, $a=\alpha$ and $b=K\alpha$. The contributions of variable nodes $\mathbb{F}_V$, function nodes $\mathbb{F}_F$, and edges $\mathbb{F}_E$ are:

$$\begin{aligned}
\mathbb{F}_V &= \mathbb{E}_{l,J,\{\widehat{\nu}\}}\log\left[\sum_x \psi(x;J)\,\widehat{\nu}_1(x)\cdots\widehat{\nu}_l(x)\right]\,,\\
\mathbb{F}_F &= \mathbb{E}_{k,\widehat{J},\{\nu\}}\log\left[\sum_{x_1,\dots,x_k} \psi^{(k)}(x_1,\dots,x_k;\widehat{J})\,\nu_1(x_1)\cdots\nu_k(x_k)\right]\,,\\
\mathbb{F}_E &= \mathbb{E}_{\nu,\widehat{\nu}}\log\left[\sum_x \nu(x)\widehat{\nu}(x)\right]\,.
\end{aligned} \tag{14.77}$$

In these expressions, $\mathbb{E}$ denotes expectation with respect to the random variables in subscript. For instance, if G is distributed according to the $\mathbb{D}_N(\Lambda,P)$ ensemble, $\mathbb{E}_{l,J,\{\widehat{\nu}\}}$ implies that l is drawn from distribution Λ, J is drawn from P, and $\widehat{\nu}_1,\dots\widehat{\nu}_l$ are l independent copies of the random variable $\widehat{\nu}$.

[50] A quasi-solution is a set of messages $\nu_{j\to a}$ such that the average difference between the left and right hand sides of the BP equations goes to zero in the large N limit

Instead of estimating the partition function, the cavity method can be used to compute the ground state energy. One then uses Min-Sum like messages instead of those in (14.73). The method is then called the 'energetic cavity method', we leave to the reader the task of writing the corresponding average ground state energy per variable.

14.6.4 *Numerical methods*

Generically, the RS cavity equations (14.75), as well as density evolution (14.71), cannot be solved in close form, and one uses numerical methods to estimate the distribution of the random variables ν, $\widehat{\nu}$. Here we limit ourselves to describing a stochastic approach that has the advantage of being extremely versatile and simple to implement. It has been used in coding theory under the name of 'sampled density evolution' or 'Monte Carlo', and is known in statistical physics as **population dynamics**, a name which we shall adopt in the following.

The idea is to approximate the distribution of ν (or $\widehat{\nu}$) through a sample of (ideally) N iid copies of ν (respectively $\widehat{\nu}$). As N gets large, the empirical distribution of such a sample should converge to the actual distribution of m. We shall call such a sample $\{\nu_i\} \equiv \{\nu_1, \dots, \nu_N\}$ (or $\{\widehat{\nu}_i\} \equiv \{\widehat{\nu}_1, \dots, \widehat{\nu}_N\}$) a **population**.

The algorithm is described by the pseudo-code below. As inputs it requires the population size N, the maximum number of iterations T and a specification of the ensemble of (random) graphical models. The latter consists in a description of the (edge perspective) degree distributions λ, ρ , of the variable node labels P, and of the factor node labels $P^{(k)}$

POPULATION DYNAMICS (Model ensemble, Size N, Iterations T)

Initialize $\{\nu_i^{(0)}\}$;
for $t = 1, \dots, T$:
    for $i = 1, \dots, N$:
        Draw an integer k with distribution ρ;
        Draw $i(1), \dots, i(k-1)$ uniformly in $\{1, \dots, N\}$;
        Draw $\widehat{J}$ with distribution $P^{(k)}$;
        Set $\widehat{\nu}_i^{(t)} = \Phi_k(\nu_{i(1)}^{(t-1)}, \dots, \nu_{i(k-1)}^{(t-1)}; \widehat{J})$;
    end;
    for $i = 1, \dots, N$:
        Draw an integer l with distribution λ;
        Draw $i(1), \dots, i(l-1)$ uniformly in $\{1, \dots, N\}$;
        Draw J with distribution P;
        Set $\widehat{\nu}_i^{(t)} = \Psi_l(\nu_{i(1)}^{(t)}, \dots, \nu_{i(l-1)}^{(t)}; J)$;
    end;
end;
return $\{\nu_i^{(T)}\}$ and $\{\widehat{\nu}_i^{(T)}\}$.

In step 1 the initialization is done by drawing $\nu_1^{(0)}, \dots, \nu_N^{(0)}$ independently with the same distribution P that was used for the initialization of BP.

It is not hard to show that, for any fixed T, the empirical distribution of $\{\nu_i^{(T)}\}$ (respectively $\{\widehat{\nu}_i^{(T)}\}$) converges, as $N \to \infty$ to the distribution of the density evolution random variable $\nu^{(t)}$ ($\widehat{\nu}^{(t)}$). The limit $T \to \infty$ is trickier. Let us first assume that density evolution has a unique fixed point, and $\nu^{(t)}$, $\widehat{\nu}^{(t)}$ converges to such a fixed point. Then we expect the empirical distribution of $\{\nu_i^{(T)}\}$ to converge to such a fixed point if the $N \to \infty$ limit is taken after $T \to \infty$ as well. Finally, when density evolution has more than a fixed point, the situation is even more subtle. The population $\{\nu_i^{(T)}\}$ evolve according to a finite (although large-) dimensional Markov chain, and one could expect it to converge to its unique fixed point. This seems to spoil the application of population dynamics in these cases (that are probably the most interesting ones). Luckily, the convergence of population dynamics to its unique fixed point appears to happen on a time scale that increases very rapidly with N. For large N and on moderate time scales T, it converges instead to one of several 'quasi-fixed points' that correspond to the density evolution fixed points.

In practice, one can monitor the effective convergence of the algorithm by computing, after any number of iterations t, averages of the form

$$\langle \varphi \rangle_t \equiv \frac{1}{N} \sum_{i=1}^{N} \varphi(\nu_i^{(t)}) \,, \tag{14.78}$$

for a smooth function $\varphi : \mathfrak{M}(\mathcal{X}) \to \mathbb{R}$. If these averages are well settled (up to statistical fluctuations of order $1/\sqrt{N}$), this is interpreted as a signal that the iteration has converged.

The populations produced by the above algorithm can be used to to estimate expectation with respect to the density evolution random variables ν, $\widehat{\nu}$. For instance, the expression in Eq. (14.78) is an estimate for $\mathbb{E}\{\varphi(\nu)\}$. When $\varphi = \varphi(\nu_!, \dots, \nu_l)$ is a function of l iid copies of ν, the above formula is modified as

$$\langle \varphi \rangle_t \equiv \frac{1}{R} \sum_{n=1}^{R} \varphi(\nu_{i(1)}^{(t)}, \dots, \nu_{i(l)}^{(t)}) \,. \tag{14.79}$$

Here R is a large number (typically of the same order as N), and $i(1), \dots, i(l)$ are iid indices in $\{1, \dots, N\}$. Of course such estimates will be reasonable only if $l \ll N$.

A particularly important example is the computation of the the free entropy (14.76). Each of the terms $\mathbb{F}_V$, $\mathbb{F}_F$ and $\mathbb{F}_E$ can be estimated as indicated. The precision of these estimates can be improved by repeating the computation for several iterations and averaging the result.

Notes

Belief propagation equations have been rediscovered several times. They were developed by Pearl (Pearl, 1988) as exact algorithm for probabilistic inference

in acyclic Bayesian networks. In the early 60's, Gallager had introduced them as an iterative procedure for decoding low density parity check codes (Gallager, 1963). Gallager described several message passing procedures and, among them, the sum-product algorithm. Always for decoding, the basic idea of this algorithm was rediscovered in several works in the 90's, and, in particular, in (Berrou and Glavieux, 1996).

In the physics context, the history is even longer. In 1935, Bethe used a free energy functional written in terms of pseudo-marginals to approximate the partition function of the ferromagnetic Ising model (Bethe, 1935). Bethe equations were of the simple form discussed Exercise 14.9, because of the homogeneity (translation invariance) of the underlying model. Their generalization to inhomogeneous systems, which has a natural algorithmic interpretation, waited until the application of Bethe's method to spin glasses (Thouless, Anderson and Palmer, 1977; Klein, Schowalter and Shukla, 1979; Katsura, Inawashiro and Fujiki, 1979; Morita, 1979; Nakanishi, 1981).

The review paper (Kschischang, Frey and Loeliger, 2001) gives a general overview of belief propagation in the factor graphs framework. The role of the distributive property, mentioned in Exercise 14.7, is emphasized in (Aji and McEliece, 2000). On tree graphs, belief propagation can be regarded as an instance of the junction-tree algorithm (Lauritzen, 1996). This approach consists in building a tree associated to the graphical model under study, by grouping some of its variables. Belief propagation is then applied to this tree.

Although implicit in these earlier works, the equivalence between BP, Bethe approximation, and sum-product algorithm was only recognized in the 90's. The turbo-decoding and sum-product algorithm were shown to be instances of BP in (McEliece, MacKay and Cheng, 1998). A variational derivation of the turbo decoding algorithm was proposed in (Montanari and Sourlas, 2000). The equivalence between BP and Bethe approximation was first put forward in (Kabashima and Saad, 1998) and, in a more general setting, in (Yedidia, Freeman and Weiss, 2001; Yedidia, Freeman and Weiss, 2005).

The last paper proved, in particular, the variational formulation in Proposition 14.8. This suggests to look for fixed points of BP by seeking directly stationary points of the Bethe free-entropy, without iterating the BP equations. An efficient such procedure, based on the observation that the Bethe free-entropy can be written as the difference between a convex and a concave function, was proposed in (Yuille, 2002). An alternative approach consists in constructing convex surrogates of the Bethe free energy (Wainwright, Jaakkola and Willsky, 2005*b*; Wainwright, Jaakkola and Willsky, 2005*a*), which allow to define provably convergent message passing procedures.

Bethe approximation can also be regarded as a first step in a hierarchy of variational methods describing exactly larger and larger clusters of variables. This point of view was first developed by Kikuchi (Kikuchi, 1951), leading to the so called 'cluster variational method' in physics. The algorithmic version of this approach is referred to as 'generalized BP,' and is described in detail in (Yedidia,

Freeman and Weiss, 2005).

The analysis of iterative message passing algorithms on random graphical models dates back to Gallager work (Gallager, 1963). These ideas were developed into a systematic method, also thanks to efficient numerical techniques, in (Richardson and Urbanke, 2001*b*) who coined the name 'density evolution.' The point of view taken in this book is however closer to the one of 'local weak convergence' (Aldous and Steele, 2003).

In physics, the replica symmetric cavity method for sparse random graphical models, was first discussed in (Mézard and Parisi, 1987). The use of population dynamics first appeared in (Abou-Chacra, Anderson and Thouless, 1973), and was further developed for spin glasses in (Mézard and Parisi, 2001), but this paper mainly deals with RSB effects which will be the object of Chapter **??**.

15

DECODING WITH BELIEF PROPAGATION

As we have seen in Section 6.1, symbol MAP decoding of error correcting codes can be regarded as a statistical inference problem. If $p(\underline{x}|\underline{y})$ denotes the conditional distribution of the channel input $\underline{x}$, given the output $\underline{y}$, one aims at computing its single bit marginals $p(x_i|\underline{y})$. It is a very natural idea to accomplish this task using belief propagation (BP).

However, it is not hard to realize that an error correcting code cannot achieve good performances unless the associated factor graph has loops. As a consequence, belief propagation has to be regarded only as an approximate inference algorithm in this context. A major concern of the theory is to establish conditions for its optimality, and, more generally, the relation between message passing and optimal (exact symbol MAP) decoding.

In this Chapter we discuss belief propagation decoding of the LDPC ensembles introduced in Chapter 11. The message passing approach can be generalized to several other applications within information and communications theory: other code ensembles, source coding, channels with memory, etc.... Here we shall keep to the 'canonical' example of channel coding as most of the theory has been developed in this context.

BP decoding is defined in Section 15.1. One of the main tools in the analysis is the 'density evolution' method that we discuss in Section 15.2. This allows to determine the threshold for reliable communication under BP decoding, and to optimize accordingly the code ensemble. The whole process is considerably simpler for the erasure channel, which is treated in Section 15.3. Finally, Section 15.4 explains the relation between optimal (MAP) decoding and BP decoding in the large block-length limit: the two approaches can be considered in the same unified framework of the Bethe free energy.

15.1 BP decoding: the algorithm

{sec:DefinitionBPDecoding}

In this chapter, we shall consider communication over a **binary input output symmetric memoryless channel (BMS)**. This is a channel in which the transmitted codeword is binary, $\underline{x} \in \{0,1\}^N$, and the output $\underline{y}$ is a sequence of N letters y_i from an alphabet[51] $\mathcal{Y} \subset \mathbb{R}$. The probability of receiving letter y when bit x is sent, $Q(y|x)$, enjoys the symmetry property $Q(y|0) = Q(-y|1)$.

Let us suppose that a LDPC error correcting code is used in this communication. The conditional probability for the channel input being $\underline{x} \in \{0,1\}^N$ given the output $\underline{y}$ is

[51] The case of a general output alphabet $\mathcal{Y}$ reduces in fact to this one.

$$p(\underline{x}|\underline{y}) = \frac{1}{Z(\underline{y})} \prod_{i=1}^{N} Q(y_i|x_i) \prod_{a=1}^{M} \mathbb{I}(x_{i_1^a} \oplus \cdots \oplus x_{i^a_{k(a)}} = 0)\,, \tag{15.1}$$

The factor graph associated with this distribution is the same as for the code membership function, cf. Fig. 9.6 in Chapter 9. An edge joins a variable node i to a check node a whenever the variable x_i appears in the a-th parity check equation.

Messages $\nu_{i\to a}(x_i)$, $\widehat{\nu}_{a\to i}(x_i)$, are exchanged along the edges. We shall assume a parallel updating of BP messages, as introduced in Section 14.2:

$$\nu^{(t+1)}_{i\to a}(x_i) \cong Q(y_i|x_i) \prod_{b\in\partial i\setminus a} \widehat{\nu}^{(t)}_{b\to i}(x_i)\,, \tag{15.2}$$

$$\widehat{\nu}^{(t)}_{a\to i}(x_i) \cong \sum_{\{x_j\}} \mathbb{I}(x_i \oplus x_{j_1} \oplus \cdots \oplus x_{j_{k-1}} = 0) \prod_{j\in\partial a\setminus i} \nu^{(t)}_{j\to a}(x_j)\,, \tag{15.3}$$

where we used the notation $\partial a \equiv \{i, j_1, \dots, j_{k-1}\}$, and the symbol $\cong$ denotes as usual 'equality up to a normalization constant'. We expect that the asymptotic performances (for instance, the asymptotic bit error rate) of such BP decoding should be not sensitive to the precise update schedule. On the other hand, this schedule can have an important influence on the speed of convergence, and on performances at moderate N. Here we shall not address these issues.

The BP estimate for the marginal distribution at node i at time t, also called 'belief' or '**soft decision**', is

$$\mu^{(t)}_i(x_i) \;\cong\; Q(y_i|x_i) \prod_{b\in\partial i} \widehat{\nu}^{(t-1)}_{b\to i}(x_i)\,. \tag{15.4}$$

Based on this estimate, the optimal BP decision for bit i at time t (sometimes called **'hard decision'**) is

$$\widehat{x}^{(t)}_i = \arg\max_{x_i} \mu^{(t)}_i(x_i)\,. \tag{15.5}$$

In order to completely specify the algorithm, one should address two more issues: (1) How are the messages initialized, and (2) After how many iterations t, does one make the hard decision (15.5).

In practice, one usually initializes the messages to $\nu^{(0)}_{i\to a}(\mathsf{0}) = \nu^{(0)}_{i\to a}(\mathsf{1}) = 1/2$. One alternative choice, that is sometimes useful for theoretical reasons, is to take the messages $\nu^{(0)}_{i\to a}(\,\cdot\,)$ as independent random variables, for instance by choosing $\nu^{(0)}_{i\to a}(\mathsf{0})$ uniformly on $[0,1]$.

As for the number of iterations, one would like to have a stopping criterion. In practice, a convenient criterion is to check whether $\underline{\widehat{x}}^{(t)}$ is a codeword, and to stop if this is the case. If this condition is not fulfilled, the algorithm is stopped after a fixed number of iterations $t_{\max}$. On the other hand, for analysis purposes,

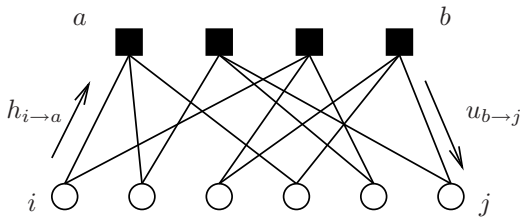


FIG. 15.1. Factor graph of a (2,3) regular LDPC code, and notation for the belief propagation messages.

{fig:FactorMess}

we shall rather fix $t_{\max}$ and assume that belief propagation is run always for $t_{\max}$ iterations, regardless whether a valid codeword is reached at an earlier stage.

Since the messages are distributions over binary valued variables, we describe them as in (**??**) by the log-likelihoods:

$$h_{i\to a} = \frac{1}{2}\,\log\frac{\nu_{i\to a}(0)}{\nu_{i\to a}(1)}\,, \qquad u_{a\to i} = \frac{1}{2}\,\log\frac{\widehat{\nu}_{a\to i}(0)}{\widehat{\nu}_{a\to i}(1)}\,. \tag{15.6}$$

We further introduce the a-priori log-likelihood for bit i, given the received message y_i:

$$B_i = \frac{1}{2}\,\log\frac{Q(y_i|0)}{Q(y_i|1)}\,. \tag{15.7}$$

For instance in a BSC channel with flip probability p, one has $B_i = \frac{1}{2}\log\frac{1-p}{p}$ on variable nodes which have received $y_i = 0$, and $B_i = -\frac{1}{2}\log\frac{1-p}{p}$ on those with $y_i = 1$. The BP update equations (15.2), (15.3) read in this notation (see Fig. 15.1):

$$h^{(t+1)}_{i\to a} = B_i + \sum_{b\in\partial i\setminus a} u^{(t)}_{b\to i}\,, \qquad u^{(t)}_{a\to i} = \mathrm{atanh}\Big\{\prod_{j\in\partial a\setminus i}\tanh h^{(t)}_{j\to a}\Big\}\,. \tag{15.8}$$

The hard-decision decoding rule depends on the over-all BP log-likelihood

$$h^{(t+1)}_{i} = B_i + \sum_{b\in\partial i\setminus a} u^{(t)}_{b\to i}\,, \tag{15.9}$$

and is given by (using for definiteness a fair coin outcome in case of a tie):

$$\widehat{x}^{(t)}_i(\underline{y}) = \begin{cases} 0 & \text{if } h^{(t)}_i > 0,\\ 1 & \text{if } h^{(t)}_i < 0,\\ 0 \text{ or } 1 & \text{with probability } 1/2 \text{ if } h^{(t)}_i = 0. \end{cases} \tag{15.10}$$

15.2 Analysis: density evolution

{sec:DensityEvolutionDecoding

In this section we consider BP decoding of random codes from the $\mathrm{LDPC}_N(\Lambda, P)$ ensemble in the large block-length limit. The code ensemble is specified by the

degree distributions of variable nodes $\Lambda = \{\Lambda_l\}$ and of check nodes, $P = \{P_k\}$. We assume for simplicity that messages are initialized to $u^{(0)}_{a\to i} = 0$.

Because of the symmetry of the channel, under the above hypotheses, the bit (or block) error probability is independent of the transmitted codeword. The explicit derivation of this fact is outlined in Exercise 15.1 below. This is also true for any other meaningful performance measures. We shall use this freedom to assume that the all-zero codeword has been transmitted. We shall first write the density evolution recursion as a special case of the one written in Section **??**. It turns out that this recursion can be analyzed in quite some detail, and in particular one can show that the decoding performance improves as t increases. The analysis hinges on two important properties of BP decoding and density evolution, related to the notions of 'symmetry' and 'physical degradation'.

{ex:cw_indep}

Exercise 15.1 Independence of the transmitted codeword. Assume the codeword $\underline{x}$ has been transmitted and let $B_i(\underline{x})$, $u^{(t)}_{a\to i}(\underline{x})$, $h^{(t)}_{i\to a}(\underline{x})$ be the corresponding channel log-likelihoods and messages. These are regarded as random variables (because of the randomness in the channel realization). Let furthermore $\sigma_i = \sigma_i(\underline{x}) = +1$ if $x_i = 0$, and $= -1$ otherwise.

(*a*) Prove that the distribution of $\sigma_i B_i$ is independent of $\underline{x}$.

(*b*) Use the equations (15.8) to prove by induction over t that the (joint) distribution of $\{\sigma_i h^{(t)}_{i\to a}, \sigma_i u^{(t)}_{a\to i}\}$ is independent of $\underline{x}$.

(*c*) Use Eq. (15.9) to show that the distribution of $\{\sigma_i H^{(t)}_i\}$ is independent of $\underline{x}$ for any $t \geq 0$. Finally, prove that the distribution of the 'error vector' $\underline{z}^{(t)} \equiv \underline{x} \oplus \underline{\widehat{x}}^{(t)}(\underline{y})$ is independent of $\underline{x}$ as well. Write the bit and block error rate in terms of the distribution of $\underline{z}^{(t)}$.

15.2.1 *Density evolution equations*

Let us consider the distribution of messages after a fixed number t of iterations. As we saw in Section **??**, in the large N limit, the directed neighborhood of any given edge is with high probability a tree. This implies the following recursive distributional characterization for $h^{(t)}$ and $u^{(t)}$:

$$h^{(t+1)} \stackrel{\mathrm{d}}{=} B + \sum_{b=1}^{l-1} u^{(t)}_b\,, \qquad u^{(t)} \stackrel{\mathrm{d}}{=} \operatorname{atanh}\Big\{ \prod_{j=1}^{k-1} \tanh h^{(t)}_j \Big\}\,. \tag{15.11}$$

Here $u^{(t)}_b$, $b \in \{1,\dots,l-1\}$ are independent copies of $u^{(t)}$, $h^{(t)}_j$, $j \in \{1,\dots,k-1\}$ are independent copies of $h^{(t)}$, l and k are independent random integers distributed, respectively, according to λ_l and ρ_k. Finally, $B = \frac{1}{2}\log\frac{Q(y|0)}{Q(y|1)}$ where y is independently distributed according to $Q(y|0)$. The recursion is initiated with $u^{(0)} = 0$.

Let us finally consider the BP log-likelihood at site i. The same arguments as above imply $h^{(t)}_i \stackrel{\mathrm{d}}{\to} h^{(t)}_*$, where the distribution of $h^{(t)}_*$ is defined by

$$h_*^{(t+1)} \stackrel{\mathrm{d}}{=} B + \sum_{b=1}^{l} u_b^{(t)} \,, \tag{15.12}$$

with l a random integer distributed according to Λ_l. In particular, if we let $\mathrm{P}_{\mathrm{b}}^{(N,t)}$ be the expected (over a $\mathrm{LDPC}_N(\Lambda, P)$ ensemble) bit error rate for the decoding rule (15.10), then:

$$\lim_{N\to\infty} \mathrm{P}_{\mathrm{b}}^{(N,t)} = \mathbb{P}\big\{h_*^{(t)} < 0\big\} + \frac{1}{2}\mathbb{P}\big\{h_*^{(t)} = 0\big\} \,. \tag{15.13}$$

The suspicious reader will notice that this statement is non-trivial, because $f(x) = \mathbb{I}(x < 0) + \frac{1}{2}\mathbb{I}(x = 0)$ is not a continuous function. We shall prove it below using the symmetry property of the distribution of $h_i^{(t)}$,which allows to write the bit error rate as the expectation of a continuous function (cf. Exercise 15.2).

15.2.2 *Basic properties:* 1. *Symmetry*

{sec:Symmetry}

A real random variable Z (or, equivalently, its distribution) is said to be **symmetric** if

$$\mathbb{E}\left\{f(-Z)\right\} = \mathbb{E}\left\{e^{-2Z} f(Z)\right\} \,. \tag{15.14}$$

for any function f such that one of the expectations exists. If Z has a density $p(z)$, then the above condition is equivalent to $p(-z) = e^{-2z}p(z)$.

Symmetric variables appear quite naturally in the description of BMS channels:

{propo:channel_sym}

Proposition 15.1 *Consider a BMS channel with transition probability $Q(y|x)$. Let Y be the channel output conditional to input* 0 *(this is a random variable with distribution $Q(y|0)$), and let $B \equiv \frac{1}{2}\log\frac{Q(Y|0)}{Q(Y|1)}$. Then B is a symmetric random variable.*

Conversely, if Z is a symmetric random variable, there exists a BMS channel whose log-likelihood ratio, conditioned on the input being 0 *is distributed as Z.*

Proof: To avoid technicalities, we prove this claim when the output alphabet $\mathcal{Y}$ is a *discrete* subset of $\mathbb{R}$. Then, using channel symmetry in the form $Q(y|0) = Q(-y|1)$, we get

$$\begin{aligned}\mathbb{E}\left\{f(-B)\right\} &= \sum_y Q(y|0)\; f\left(\frac{1}{2}\log\frac{Q(y|1)}{Q(y|0)}\right) = \sum_y Q(y|1)\; f\left(\frac{1}{2}\log\frac{Q(y|0)}{Q(y|1)}\right) = \\ &= \sum_y Q(y|0)\,\frac{Q(y|1)}{Q(y|0)}\; f\left(\frac{1}{2}\log\frac{Q(y|0)}{Q(y|1)}\right) = \mathbb{E}\left\{e^{-2B} f(B)\right\} \,. \end{aligned} \tag{15.15}$$

We now prove the converse. Let Z be a symmetric random variable. We build a channel with output alphabet $\mathbb{R}$ as follows: Under input 0, the output is distributed as Z, and under input 1, it is distributed as $-Z$. In terms of densities

$$Q(z|\mathtt{0}) = p(z)\,, \qquad Q(z|\mathtt{1}) = p(-z)\,. \tag{15.16}$$

This is a BMS channel with the desired property. Of course this construction is not unique. □

Example 15.2 Consider the binary erasure channel BEC(ϵ). If the channel input is 0, then Y can take two values, either 0 (with probability $1-\epsilon$) or $*$ (probability ϵ). The distribution of B, $\mathbb{P}_B = (1-\epsilon)\,\delta_\infty + \epsilon\,\delta_0$, is symmetric. In particular, this is true for the two extreme cases: $\epsilon = 0$ (a noiseless channel) and $\epsilon = 1$ (a completely noisy channel: $\mathbb{P}_B = \delta_0$).

Example 15.3 Consider a binary symmetric channel BSC(p). The log-likelihood B can take two values, either $b_0 = \frac{1}{2}\log\frac{1-p}{p}$ (input 0 and output 0) or $-b_0$ (input 0 and output 1). Its distribution, $\mathbb{P}_B = (1-p)\,\delta_{b_0} + p\,\delta_{-b_0}$ is symmetric.

Example 15.4 Finally consider the binary white noise additive Gaussian channel BAWGN(σ^2). If the channel input is 0, the output Y has probability density

$$q(y) = \frac{1}{\sqrt{2\pi\sigma^2}}\,\exp\left\{-\frac{(y-1)^2}{2\sigma^2}\right\}\,, \tag{15.17}$$

i.e. it is a Gaussian of mean 1 and variance σ^2. The output density upon input 1 is determined by the channel symmetry (i.e. a Gaussian of mean -1 and variance σ^2). The log-likelihood under output y is easily checked to be $b = y/\sigma^2$. Therefore B also has a symmetric Gaussian density, namely:

$$p(b) = \sqrt{\frac{\sigma^2}{2\pi}}\,\exp\left\{-\frac{\sigma^2}{2}\left(b-\frac{1}{\sigma^2}\right)^2\right\}\,. \tag{15.18}$$

The variables appearing in density evolution are symmetric as well. The argument is based on the symmetry of the channel log-likelihood, and the fact that symmetry is preserved by the operations in BP evolution: If Z_1 and Z_2 are two independent symmetric random variables (not necessarily identically distributed), ⋆ it is straightforward to show that $Z = Z_1 + Z_2$, and $Z' = \mathrm{atanh}[\tanh Z_1 \tanh Z_2]$ are both symmetric.

Consider now communication of the all-zero codeword over a BMS channel using a LDPC code, but let us first assume that the factor graph associated with the code is a tree. We apply BP decoding with a symmetric random initial condition like e.g. $u^{(0)}_{a\to i} = 0$. The messages passed during the decoding procedure can be regarded as random variables, because of the random received symbols y_i (which yield random log-likelihoods B_i). Furthermore, messages incoming at

a given node are independent since they are functions of B_i's (and of initial conditions) on disjoint subtrees. From the above remarks, and looking at the BP equations (15.8) it follows that the messages $u^{(t)}_{a\to i}$, and $h^{(t)}_{i\to a}$, as well as the overall log-likelihoods $h^{(t)}_i$ are symmetric random variables at all $t \geq 0$. Therefore:

{propo:SymmetryBP}

Proposition 15.5 *Consider BP decoding of an LDPC code under the above assumptions. If* $\mathsf{B}_{i\to a,t+1}(F)$ *is a tree, then* $h^{(t)}_{i\to a}$ *is a symmetric random variable. Analogously, if* $\mathsf{B}_{i,t+1}(F)$ *is a tree, then* $H^{(t)}_i$ *is a symmetric random variable.*

{propo:SymmetryDE}

Proposition 15.6 *The density evolution random variables* $\{h^{(t)}, u^{(t)}, H^{(t)}_*\}$ *are symmetric.*

{ex:SymmetryBER}

Exercise 15.2 Using Proposition 15.5, and the fact that, for any finite t $\mathsf{B}_{i\to a,t+1}(F)$ is a tree with high probability as $N \to \infty$, show that

$$\lim_{N\to\infty} \mathrm{P}^{(N,t)}_{\mathrm{b}} = \lim_{N\to\infty} \mathbb{E}\left\{\frac{1}{N}\sum_{i=1}^{N} f(h^{(t)}_i)\right\}, \qquad (15.19)$$

where $f(x) = 1/2$ for $x \leq 0$ and $f(x) = e^{-2x}/2$ otherwise.

The symmetry property is a generalization of the Nishimori condition that we encountered in spin glasses. As can be recognized from Eq. (12.7) this condition is satisfied if and only if for each coupling constant J, βJ is a symmetric random variable. While in spin glasses symmetry occurs only at very special values of the temperature, it is a natural property in the decoding problem. Further it does not hold uniquely for the BP log-likelihood, but also for the actual (MAP) log-likelihood of a bit, as shown in the exercise below.

{ex:MAPSymmetric}

Exercise 15.3 Consider the actual (MAP) log-likelihood for bit i (as opposed to its BP approximation). This is defined as

$$h_i = \frac{1}{2} \log \frac{\mathbb{P}\{x_i = \mathtt{0} | \underline{y}\}}{\mathbb{P}\{x_i = \mathtt{1} | \underline{y}\}} \,. \tag{15.20}$$

If we condition on the all-zero codeword being transmitted, then the random variable h_i is symmetric. This can be shown as follows.

(*a*) Show that $h_i = \frac{1}{2} \log \frac{Q(y_i|0)}{Q(y_i|1)} + g_i$ where g_i depends on $y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_N$, but not on y_i. Suppose that a codeword $\underline{z} \neq \underline{\mathtt{0}}$ has been transmitted, and let $h_i(\underline{z})$ be the corresponding log-likelihood for bit x_i. Show that $h_i(\underline{z}) \stackrel{\mathrm{d}}{=} h_i$ if $z_i = \mathtt{0}$, and $h_i(\underline{z}) \stackrel{\mathrm{d}}{=} -h_i$ if $z_i = \mathtt{1}$.

(*b*) Consider the following process. A bit z_i is chosen uniformly at random. Then a codeword $\underline{z}$ is chosen uniformly at random conditioned on the value of z_i, and transmitted through a BMS channel, yielding an output $\underline{y}$. Finally, the log-likelihood $h_i(\underline{z})$ is computed. Hiding the intermediate steps in a black box, this can be seen as a communication channel: $z_i \to h_i(\underline{z})$. Show this is a BMS channel.

(*c*) Show that h_i is a symmetric random variable.

15.2.3 *Basic properties:* 2. *Physical degradation*

It turns out that BP decoding gets better when the number of iterations t increases (although it does not necessarily converge to the correct values). This is an extremely useful result, which does not hold when BP is applied to a general inference problems. A precise formulation of this statement is provided by the notion of physical degradation. This notion is first defined in terms of BMS channels, and then extended to symmetric random variables. This allows to apply it to the random variables encountered in BP decoding and density evolution.

Let us start with the case of BMS channels. Consider two such channels, denoted as BMS(1) and BMS(2), denote by $\{Q_1(y|x)\}$, $\{Q_2(y|x)\}$ their transition matrices and by $\mathcal{Y}_1$, $\mathcal{Y}_2$ the corresponding output alphabets. We say that BMS(2) is **physically degraded** with respect to BMS(1) if there exists a third channel C with input alphabet $\mathcal{Y}_1$ and output $\mathcal{Y}_2$ such that BMS(2) can be regarded as the concatenation of BMS(1) and C. By this we mean that passing a bit through BMS(1) and then feeding the output to C is statistically equivalent to passing the bit through BMS(2). If the transition matrix of C is $\{R(y_2|y_1)\}$, this can be written in formulae as

$$Q_2(y_2|x) = \sum_{y_1 \in \mathcal{Y}_1} R(y_2|y_1)\, Q_1(y_1|x) \,, \tag{15.21}$$

where, to simplify the notation, we assumed $\mathcal{Y}_1$ to be discrete. A pictorial representation of this relationship is provided by Fig. 15.2. A formal way of expressing

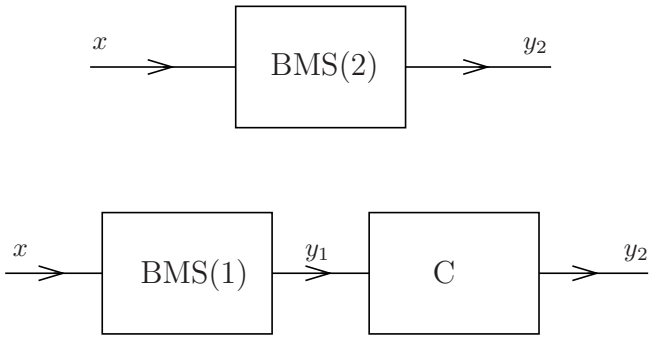


FIG. 15.2. The channel BMS(2) (top) is said to be physically degraded with respect to BMS(1) if it is equivalent to the concatenation of BMS(1) with a second channel C.

{fig:PhysDegr}

the same idea is that there exists a Markov chain $X \to Y_1 \to Y_2$.

Whenever BMS(2) is physically degraded with respect to BMS(1) we shall write BMS(1) $\preceq$ BMS(2) (which is read as: BMS(1) is 'less noisy than' BMS(2)). Physical degradation is a partial ordering: If BMS(1) $\preceq$ BMS(2) and BMS(2) $\preceq$ BMS(3), then BMS(1) $\preceq$ BMS(3). Furthermore, if BMS(1) $\preceq$ BMS(2) and BMS(2) $\preceq$ BMS(1), then BMS(1) = BMS(2). However, given two binary memoryless symmetric channels, they are not necessarily ordered by physical degradation (i.e. it can be that neither BMS(1) $\preceq$ BMS(2) nor BMS(2) $\preceq$ BMS(1)). ⋆

Here are a few examples of channel pairs ordered by physical degradation.

Example 15.7 Let $\epsilon_1, \epsilon_2 \in [0,1]$ with $\epsilon_1 \leq \epsilon_2$. Then the corresponding erasure channels are ordered by physical degradation, namely BEC(ϵ_1) $\preceq$ BEC(ϵ_2).

Consider in fact a channel C that has input and output alphabet $\mathcal{Y} = \{0, 1, *\}$ (the symbol $*$ representing an erasure). On inputs 0, 1, it transmits the input unchanged with probability $1-x$ and erases it with probability x. On input $*$ it outputs an erasure. If we concatenate this channel at the output of BEC(ϵ_1), we obtain a channel BEC(ϵ), with $\epsilon = 1-(1-x)(1-\epsilon)$ (the probability that a bit *is not* erased is the product of the probability that it is not erased by each of the component channels). The claim is thus proved by taking $x = (\epsilon_2 - \epsilon_1)/(1-\epsilon_1)$ (without loss of generality we can assume $\epsilon_1 < 1$).

Exercise 15.4 If $p_1, p_2 \in [0, 1/2]$ with $p_1 \leq p_2$, then BSC(p_1) $\preceq$ BSC(p_2). This can be proved by showing that BSC(p_2) is equivalent to the concatenation of BSC(p_1) with a second binary symmetric channel BSC(x). What value of the crossover probability x should one take?

Exercise 15.5 If $\sigma_1^2, \sigma_2^2 \in [0, \infty)$ with $\sigma_1^2 \leq \sigma_2^2$, show that BAWGN($\sigma_1^2$) $\preceq$ BAWGN(σ_2^2).

If BMS(1) $\preceq$ BMS(2), most measures of the channel 'reliability' are ordered accordingly. Let us discuss here two important such measures: (1) conditional entropy and (2) bit error rate.

(1): Let Y_1 and Y_2 be the outputs of passing a uniformly random bit, respectively, through channels BMS(1) and BMS(2). Then $H(X|Y_1) \leq H(X|Y_2)$ (the uncertainty on the transmitted is larger for the 'noisier' channel). This follows immediately from the fact that $X \to Y_1 \to Y_2$ is a Markov chain by applying the data processing inequality, cf. Sec. **??**.

(2) Assume the outputs of channels BMS(1), BMS(2) are y_1 and y_2. The MAP decision rule for x knowing y_a is $\widehat{x}_a(y_a) = \arg\max_x \mathbb{P}\{X = x|Y_a = y_a\}$, with $a = 1, 2$. The corresponding bit error rate is $\mathrm{P}_{\mathrm{b}}^{(a)} = \mathbb{P}\{\widehat{x}_a(y_a) \neq x\}$. Let us show that $\mathrm{P}_{\mathrm{b}}^{(1)} \leq \mathrm{P}_{\mathrm{b}}^{(2)}$. As BMS(1) $\preceq$ BMS(2), there is a channel C be the channel such that BMS(1) concatenated with C is equivalent to BMS(2). Then $\mathrm{P}_{\mathrm{b}}^{(2)}$ can be regarded as the bit error rate for a non-MAP decision rule given y_1. The rule is: transmit y_1 through C, denote by y_2 the output, and then compute $\widehat{x}_2(y_2)$. This non-MAP decision rule cannot be better than the MAP rule applied directly to y_1.

Since symmetric random variables can be associated with BMS channels (see Proposition 15.1), the notion of physical degradation of channels can be extended to symmetric random variables. Let Z_1, Z_2 be two symmetric random variables and BMS(1), BMS(2) the associated BMS channels, constructed as in the proof of proposition 15.1. We say that Z_2 is physically degraded with respect to Z_1 (and we write $Z_1 \preceq Z_2$) if BMS(2) is physically degraded with respect to BMS(1). It can be proved that this definition is in fact independent of the choice of BMS(1), BMS(2) within the family of BMS channels associated to Z_1, Z_2.

The interesting result is that BP decoding behaves in the intuitively most natural way with respect to physical degradation. As above, we fix a particular LDPC code and look at BP message as random variables due to the randomness in the received vector $\underline{y}$.

{propo:PhysDegr}

Proposition 15.8 *Consider communication over a BMS channel using an LDPC code under the all-zero codeword assumption, and BP decoding with standard initial condition $X = 0$. If $\mathsf{B}_{i,r}(F)$ is a tree, then $h_i^{(0)} \succeq h_i^{(1)} \succeq \cdots \succeq h_i^{(t-1)} \succeq h_i^{(t)}$ for any $t \leq r-1$. Analogously, if $\mathsf{B}_{i\to a,r}(F)$ is a tree, then $h_{i\to a}^{(0)} \succeq h_{i\to a}^{(1)} \succeq \cdots \succeq h_{i\to a}^{(t-1)} \succeq h_{i\to a}^{(t)}$ for any $t \leq r-1$.*

We shall not prove this proposition in full generality here, but rather prove its most useful consequence for our purpose, namely the fact that the bit error rate is monotonously decreasing with t.

Proof: Under the all-zero codeword assumption, the bit error rate is $\mathbb{P}\{\widehat{x}_i^{(t)} = \mathtt{1}\} = \mathbb{P}\{h_i^{(t)} < 0\}$ (for the sake of simplicity we neglect here the case $h_i^{(t)} = 0$). Assume $\mathsf{B}_{i,r}(F)$ to be a tree and fix $t \leq r-1$. Then we want to show that $\mathbb{P}\{h_i^{(t)} < 0\} \leq \mathbb{P}\{h_i^{(t-1)} < 0\}$. The BP log-likelihood after T iterations on the original graph, $h_i^{(t)}$, is equal to the actual (MAP) log-likelihood for the reduced

model defined on the tree $\mathsf{B}_{i,t+1}(F)$. More precisely, let us call $\mathfrak{C}_{i,t}$ the LDPC code associated to the factor graph $\mathsf{B}_{i,t+1}(F)$, and imagine the following process. A uniformly random codeword in $\mathfrak{C}_{i,t}$ is transmitted through the BMS channel yielding output $\underline{y}_t$. Define the log-likelihood ratio for bit x_i

$$\widehat{h}_i^{(t)} = \frac{1}{2}\log\left\{\frac{\mathbb{P}(x_i = \mathtt{0}|\underline{y}_t)}{\mathbb{P}(x_i = \mathtt{1}|\underline{y}_t)}\right\}, \tag{15.22}$$

and denote the map estimate for x_i and $\widehat{x}_i$. It is not hard to show that $h_i^{(t)}$ is distributed as $\widehat{h}_i^{(t)}$ under the condition $x_i = 0$. In particular, $\mathbb{P}\{\widehat{x}_i = \mathtt{1}|x_i = \mathtt{0}\} = \mathbb{P}\{h_i^{(t)} < 0\}$.

In the above example, instead of MAP decoding one can imagine to scratch all the received symbols at distance t from i, and then performing MAP decoding on the reduced information. Call $\widehat{x}'_i$ the resulting estimate. The vector of non-erased symbols is $\underline{y}_{t-1}$. The corresponding log-likelihood is clearly the BP log-likelihood after $t-1$ iterations. Therefore $\mathbb{P}\{\widehat{x}'_i = \mathtt{1}|x_i = \mathtt{0}\} = \mathbb{P}\{h_i^{(t-1)} < 0\}$. By optimality of the MAP decision rule $\mathbb{P}\{\widehat{x}_i \neq x_i\} \leq \mathbb{P}\{\widehat{x}'_i \neq x_i\}$, which proves our claim. □

In the case of random LDPC codes $\mathsf{B}_{i,r}(F)$ is a tree with high probability for any fixed r, in the large block length limit. Therefore Proposition 15.8 has an immediate consequence in the asymptotic setting.

{propo:PhysDegrDE}

Proposition 15.9 *The density evolution random variables are ordered by physical degradation. Namely, $h^{(0)} \succeq h^{(1)} \succeq \cdots \succeq h^{(t-1)} \succeq h^{(t)} \succeq \cdots$. Analogously $h_*^{(0)} \succeq h_*^{(1)} \succeq \cdots \succeq h_*^{(t-1)} \succeq h_*^{(t)} \succeq \cdots$. As a consequence, the asymptotic bit error rate after a fixed number t of iterations $\mathrm{P}_{\mathrm{b}}^{(t)} \equiv \lim_{N\to\infty} \mathrm{P}_{\mathrm{b}}^{(N,t)}$ is monotonically decreasing with t.*

Exercise 15.6 An alternative measure of the reliability of $h_i^{(t)}$ is provided by the conditional entropy. Assuming that a uniformly random codeword is transmitted, this is given by $H_i(t) = H(X_i|h_i^{(t)})$.

(*a*) Prove that, if $\mathsf{B}_{i,r}(F)$ is a tree, then $H_i(t)$ is monotonically decreasing with t for $t \leq r-1$.

(*b*) Assume that, under the all-zero codeword assumption $h_i^{(t)}$ has density $\mathsf{p}_t(.)$. Show that $H_i(t) = \int \log(1+e^{-2z})\, \mathrm{d}\mathsf{p}_t(z)$. (Hint: remember that $\mathsf{p}_t(.)$ is a symmetric distribution).

15.2.4 *Numerical implementation and threshold*

Density evolution is a useful tool because it can be simulated efficiently. One can estimate numerically the distributions of the density evolution variables $\{h^{(t)}, u^{(t)}\}$, as well as $\{h_*^{(t)}\}$. As we have seen this gives access to the properties of BP decoding in the large block-length limit, such as the bit error rate $\mathrm{P}_{\mathrm{b}}^{(t)}$ after t iterations.

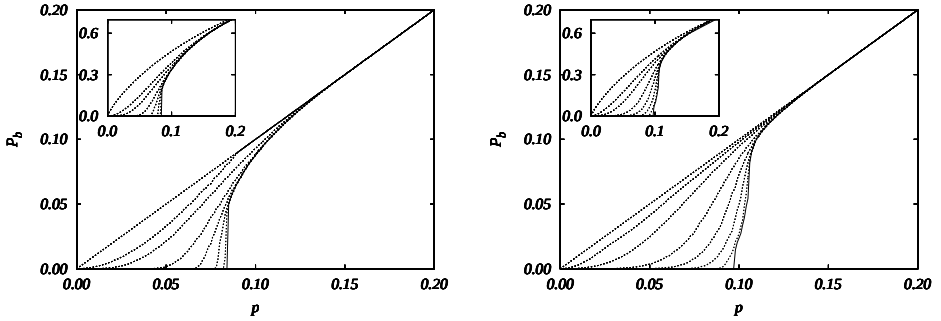


FIG. 15.3. Predicted performances of two LDPC ensembles on a BSC channel. The curves have been obtained through a numerical solution of density evolution, using population dynamics algorithm with population size $5 \cdot 10^5$. On the left, the $(3, 6)$ regular ensemble. On the right, an optimized irregular ensemble with the same design rate $R_{\rm des} = 1/2$. Its degree distribution pair is $\Lambda(x) = 0.4871\, x^2 + 0.3128\, x^3 + 0.0421\, x^4 + 0.1580\, x^{10}$, $P(x) = 0.6797\, x^7 + 0.3203\, x^8$. Dotted curves give the bit error rate obtained after $t = 1$, 2, 3, 6, 11, 21, 51 iterations (from top to bottom), and bold continuous lines to the limit $t \to \infty$. In the inset we plot the expected conditional entropy $\mathbb{E}H(X_i|\underline{Y})$. {fig:DE}

A possible approach[52] consists in representing the distributions by samples of some fixed size. This leads to the population dynamics algorithm discussed in Section 14.6.2. In Fig. 15.3 we report the results of population dynamics computations for two different LDPC ensembles used on a BSC channel with crossover probability p. We consider two performance measures: the bit error rate $\mathrm{P}_{\rm b}^{(t)}$ and the conditional entropy $H^{(t)}$. As follows from proposition 15.9, they are monotonically decreasing functions of the number of iterations. One can also show that they are monotonically increasing functions of p. As $\mathrm{P}_{\rm b}^{(t)}$ is non-negative and decreasing in t, it has a finite limit $\mathrm{P}_{\rm b}^{\rm BP} \equiv \lim_{t\to\infty} \mathrm{P}_{\rm b}^{(t)}$, which is itself non-decreasing in p (the limit curve $\mathrm{P}_{\rm b}^{\rm BP}$ is estimated in Fig. 15.3 by choosing t large enough so that $\mathrm{P}_{\rm b}^{(t)}$ is independent of t within the numerical accuracy). One defines the **BP threshold** as

$$p_{\rm d} \equiv \sup\left\{ p \in [0, 1/2] \,:\, \mathrm{P}_{\rm b}^{\rm BP}(p) = 0 \right\} . \tag{15.23}$$

Analogous definitions can be provided for other channel families such as the erasure BEC(ϵ) or Gaussian BAWGN(σ^2) channels. In general, the definition

[52] An alternative approach is as follows. Both maps (15.11) can be regarded as convolutions of probability densities for an appropriate choice of the message variables. The first one is immediate in terms of log-likelihoods. For the second map, one can use variables $r^{(t)} = (\mathrm{sign}\, h^{(t)}, \log|\tanh h^{(t)}|)$, $s^{(t)} = (\mathrm{sign}\, u^{(t)}, \log|\tanh y^{(t)}|)$. By using fast Fourier transform to implement convolutions, this can result in a significant speedup of the calculation.

{TableBPThresholds}

l	k	$R_{\rm des}$	$p_{\rm d}$	Shannon limit
3	4	1/4	0.1669(2)	0.2145018
3	5	2/5	0.1138(2)	0.1461024
3	6	1/2	0.0840(2)	0.1100279
4	6	1/3	0.1169(2)	0.1739524

Table 15.1 *Belief propagation thresholds for the BSC channel, for a few regular* LDPC *ensembles. The third column is the design rate* $1 - l/k$.

(15.23) can be extended to any family of BMS channels BMS(p) indexed by a real parameter p which orders the channels in terms of physical degradation.

Numerical simulation of density evolution allows to determine the BP threshold $p_{\rm d}$ with good accuracy. In Table 15.2.4 we report the results of a few such results. Let us stress that the threshold $p_{\rm d}$ has an important practical meaning. For any $p < p_{\rm d}$ one can achieve arbitrarily small bit error rate with high probability by just picking one random code from the ensemble $\mathrm{LDPC}_N(\Lambda, P)$ with large N and decoding it using BP with a large enough (but independent of N) number of iterations. For $p > p_{\rm d}$ the bit error rate is asymptotically lower bounded by $\mathrm{P}_{\rm b}^{\rm BP}(p) > 0$ for any fixed number of iterations (in practice it turns out that doing more iterations, say n^a, does not help). The value of $p_{\rm d}$ is therefore a primary measure of the performance of a code.

The design of good LDPC codes thus involves a choice of the degree distribution pair (Λ, P) with the largest BP threshold $p_{\rm d}$, given a certain design rate $R_{\rm des} = 1 - P'(1)/\Lambda'(1)$. For general BMS channels, this can be done numerically. One computes the threshold noise level for a given degree distribution pair using density evolution, and maximizes it by a local search procedure. As we shall see in Section 15.3, the optimization can be carried out analytically for the BEC. Figure 15.3 shows the example of an optimized irregular ensemble with rate 1/2, including variable nodes of degrees $2, 3, 4$ and 10 and check nodes of degree 7 and 8. Its threshold is $p_{\rm d} \approx 0.097$ (while Shannon's limit is 0.110).

Note that this ensemble has a finite fraction of variable nodes of degree 2. We can use the analysis in Chapter 11 to compute its weight enumerator function. It turns out that the parameter of A in Eq. (11.23) is positive. This optimized ensemble has a large number of codewords with small weight. It is surprising, and not very intuitive, that a code where there exist many codewords close to the one which is sent has nevertheless a large BP threshold $p_{\rm d}$. It turns out that this phenomenon is pretty general: the code ensembles that approach Shannon capacity turn out to have bad distance properties, without any gap at short distance in the weight enumerator function. The low-weight codewords are not harmless. They degrade the code performances at moderate block-length N, below the threshold $p_{\rm d}$. Further they prevent the block error probability from vanishing as N goes to infinity (in each codeword a fraction $1/N$ of the bits is decoded incorrectly). This phenomenon is referred to as the **error floor**.

Exercise 15.7 while the BP threshold (15.23) was defined in terms of the bit error rate, any other 'reasonable' measure of error on the decoding of a single bit would give the same result. This can be shown as follows. Let Z be a symmetric random variable and $\mathrm{P_b} \equiv \mathbb{P}\{Z<0\} + \frac{1}{2}\mathbb{P}\{Z=0\}$. Show that, for any $\Delta > 0$, $\mathbb{P}\{Z < \Delta\} \leq (2 + e^{2\Delta})\mathrm{P_b}$.

Consider then a sequence of symmetric random variables $\{Z^{(t)}\}$, such that the sequence of $\mathrm{P_b}^{(t)} \to 0$ defined as before goes to 0. Show that the distribution of $Z^{(t)}$ becomes a Dirac delta at plus infinity as $t \to \infty$.

15.2.5 *Local stability*

Beside numerical computation, it is useful to derive simple analytical bounds on the BP threshold. One of the most interesting is provided by a local stability analysis. It applies to any BMS channel, and the result depends on the specific channel only through its Bhattacharya parameter $\mathfrak{B} \equiv \sum_y \sqrt{Q(y|0)Q(y|1)}$. This parameter $\mathfrak{B} \leq 1$, that we already encountered in Chap.11, is a measure of the channel noise level. It preserves the ordering by physical degradation (i.e. the Bhattacharya parameters of two channels BMS(1) $\preceq$ BMS(2) satisfy $\mathfrak{B}(1) \leq \mathfrak{B}(2)$), as can be checked by explicit computation. ⋆

The local stability condition depends on the LDPC code through the fraction of vertices with degree 2, $\Lambda_2 = \lambda'(0)$, and the value of $\rho'(1) = \frac{\sum_k P_k k(k-1)}{\sum_k P_k k}$. It is expressed as:

{thm:LocalStability}

Theorem 15.10 *Consider communication over a binary memoryless symmetric channel with Bhattacharyia parameter $\mathfrak{B}$, using random elements from the ensemble* $\mathrm{LDPC}_N(\Lambda, P)$ *and belief propagation decoding with an arbitrary symmetric initial condition X (by this we mean a couple $(X(0), X(1))$). Let $\mathrm{P_b}^{(t,N)}$ be the bit error rate after t iterations, and $\mathrm{P_b}^{(t)} = \lim_{N\to\infty} \mathrm{P_b}^{(t,N)}$.*

1. *If $\lambda'(0)\rho'(1)\mathfrak{B} < 1$, then there exists $\xi > 0$ such that, if $\mathrm{P_b}^{(t)} < \xi$ for some ξ, then $\mathrm{P_b}^{(t)} \to 0$ as $t \to \infty$.*
2. *If $\lambda'(0)\rho'(1)\mathfrak{B} > 1$, then there exists $\xi > 0$ such that $\mathrm{P_b}^{(t)} > \xi$ for any t.*

Corollary 15.11 *Define the* ***local stability threshold*** p_{loc} *as*

$$p_{\mathrm{loc}} = \inf\left\{\, p \;\mid\; \lambda'(0)\rho'(1)\mathfrak{B}(p) > 1 \right\}. \tag{15.24}$$

The BP threshold p_{d} for decoding a communication over an ordered channel family $\mathrm{BMS}(p)$ *using random codes from the* $\mathrm{LDPC}_N(\Lambda, P)$ *ensemble satisfies:*

$$p_{\mathrm{d}} \leq p_{\mathrm{loc}}\,.$$

We shall not give the full proof of the theorem, but will explain the stability argument that underlies it. If the minimum variable node degree is 2 or larger, the density evolution recursions (15.11) have as a fixed point $h, u \stackrel{\mathrm{d}}{=} Z_\infty$, where Z_∞ is

the random variable that takes value $+\infty$ with probability 1. The BP threshold $p_{\rm d}$ is the largest value of the channel parameter such that $\{h^{(t)}, u^{(t)}\}$ converge to this fixed point as $t \to \infty$. It is then quite natural to ask what happens if the density evolution recursion is initiated with some random initial condition that is 'close enough' to Z_∞. To this end, we consider the initial condition

$$X = \begin{cases} 0 & \text{with probability } \epsilon, \\ +\infty & \text{with probability } 1-\epsilon. \end{cases} \tag{15.25}$$

This is nothing but the log-likelihood distribution for a bit revealed through a binary erasure channel, with erasure probability ϵ.

Let us now apply the density evolution recursions (15.11) with initial condition $u^{(0)} \stackrel{\rm d}{=} X$. At the first step we have $h^{(1)} \stackrel{\rm d}{=} B + \sum_{b=1}^{l-1} X_b$, where $\{X_b\}$ are iid copies of X. Therefore $h^{(1)} = +\infty$ unless $X_1 = \cdots = X_{l-1} = 0$, in which case $h^{(1)} \stackrel{\rm d}{=} B$. We have therefore

$$\text{With probability } \lambda_l \ : h^{(1)} = \begin{cases} B & \text{with prob. } \epsilon^{l-1}, \\ +\infty & \text{with prob. } 1-\epsilon^{l-1}. \end{cases} \tag{15.26}$$

where B is distributed as the channel log-likelihood. Since we are interested in the behavior 'close' to the fixed point Z_∞, we linearize in ϵ, thus getting

$$h^{(1)} = \begin{cases} B & \text{with prob. } \lambda_2\epsilon + O(\epsilon^2), \\ +\infty & \text{with prob. } 1-\lambda_2\epsilon + O(\epsilon^2), \\ \cdots & \text{with prob. } O(\epsilon^2). \end{cases} \tag{15.27}$$

The last line is absent here, but it will become necessary at next iterations. It signals that $h^{(1)}$ could take some other value with a negligible probability.

Next consider the first iteration at check node side: $u^{(1)} = \operatorname{atanh}\{\prod_{j=1}^{k-1} \tanh h_j^{(1)}\}$. At first order in ϵ, we have to consider only two cases. Either $h_1^{(1)} = \cdots = h_{k-1}^{(1)} = +\infty$ (this happens with probability $1-(k-1)\lambda_2\epsilon + O(\epsilon^2)$), or one of the log-likelihoods is distributed like B (with probability $(k-1)\lambda_2\epsilon + O(\epsilon^2)$). Averaging over the distribution of k, we get

$$u^{(1)} = \begin{cases} B & \text{with prob. } \lambda_2\rho'(1)\epsilon + O(\epsilon^2), \\ +\infty & \text{with prob. } 1-\lambda_2\rho'(1)\epsilon + O(\epsilon^2), \\ \cdots & \text{with prob. } O(\epsilon^2). \end{cases} \tag{15.28}$$

Repeating the argument t times (and recalling that $\lambda_2 = \lambda'(0)$), we get

$$h^{(t)} = \begin{cases} B_1 + \cdots + B_t & \text{with prob. } (\lambda'(0)\rho'(1))^t\epsilon + O(\epsilon^2), \\ +\infty & \text{with prob. } 1-(\lambda'(0)\rho'(1))^t\epsilon + O(\epsilon^2), \\ \cdots & \text{with prob. } O(\epsilon^2). \end{cases} \tag{15.29}$$

The bit error rate vanishes if and only $\mathrm{P(t;\epsilon)} = \mathbb{P}\{\mathrm{h^{(t)}} \le 0\}$ goes to 0 as $t \to \infty$. The above calculation shows that

$$\mathrm{P(t;\epsilon) = \epsilon(\lambda'(0)\rho'(1))^t \mathbb{P}\{B_1 + \cdots + B_t \le 0\} + O(\epsilon^2)}\,. \tag{15.30}$$

The probability of $B_1 + \cdots + B_t \le 0$ is computed, to leading exponential order, using the large deviations estimates of Section 4.2. In particular

$$\mathbb{P}\{B_1 + \cdots + B_t \le 0\} \doteq \left\{ \inf_{z \ge 0} \mathbb{E}\,[e^{-zB}] \right\}^t\,. \tag{15.31}$$

⋆ We leave to the reader the exerciseof showing that, since B is a symmetric random variable, $\mathbb{E}\, e^{-zB}$ is minimized for $z = 1$, thus yielding

$$\mathbb{P}\{B_1 + \cdots + B_t \le 0\} \doteq \mathfrak{B}^t\,. \tag{15.32}$$

As a consequence, the order ϵ coefficient In Eq. (15.30) behaves, to leading exponential order, as $(\lambda'(0)\rho'(1)\mathfrak{B})^t$. Depending whether $\lambda'(0)\rho'(1)\mathfrak{B} < 1$ or $\lambda'(0)\rho'(1)\mathfrak{B} > 1$ density evolution converges or not to the error-free fixed point if initiated sufficiently close to it. The full proof relies on these ideas, but it requires to control the terms of higher order in ϵ, and other initial conditions as well.

{sec:ErasureCodes}

15.3 BP decoding of the erasure channel

In this Section we focus on the channel BEC(ϵ). The analysis can be greatly simplified in this case: the BP decoding algorithm has a simple interpretation, and the density evolution equations can be studied analytically. This allows to construct capacity achieving ensembles.

15.3.1 *BP, peeling and stopping sets*

We consider BP decoding, with initial condition $u^{(0)}_{a\to i} = 0$. As can be seen from Eq. (15.7), the channel log likelihood B_i can take three values: $+\infty$ (if a `0` has been received at position i), $-\infty$ (if a `1` has been received at position i), 0 (if an erasure occurred at position i).

It follows from the update equations (15.8) that the messages exchanged at any subsequent time take values in $\{-\infty, 0, +\infty\}$ as well. Consider first the equation at check nodes. If one of the incoming messages $h^{(t)}_{j\to a}$ is 0, then $u^{(t)}_{a\to i} = 0$ as well. If on the other hand $h^{(t)}_{j\to a} = \pm\infty$ for all incoming messages, then $u^{(t)}_{a\to i} = \pm\infty$ (the sign being the product of the incoming signs). Next consider the update equation at variable nodes. If $u^{(t)}_{b\to i} = 0$ for all the incoming messages, and $B_i = 0$ as well, then of course $h^{(t+1)}_{i\to a} = 0$. If on the other hand some of the incoming messages, or the received value B_i take value $\pm\infty$, then $h^{(t+1)}_{i\to a}$ takes the same value. Notice that there can never be contradicting messages (i.e. both $+\infty$ and $-\infty$) incoming at a variable node.

Exercise 15.8 Show that, if contradicting messages were sent to the same variable node, this would imply that the transmitted message was not a codeword.

The meaning of the three possible messages $\pm\infty$ and 0, and of the update equations is very clear in this case. Each time the message $h^{(t)}_{i\to a}$, or $u^{(t)}_{a\to i}$ is $+\infty$ (respectively, $-\infty$), this means that the bit x_i is `0` (respectively `1`) in all codewords that coincide with the channel output on the non-erased positions: the value of x_i is perfectly known. Vice-versa, if, $h^{(t)}_{i\to a} = 0$ (or $u^{(t)}_{a\to i} = 0$) the bit x_i is currently considered equally likely to be `0` or `1`.

The algorithm is very simple: each message changes value at most one time, either from 0 to $+\infty$, or from 0 to $-\infty$.

Exercise 15.9 To show this, consider the first time, t_1 at which a message $h^{(t)}_{i\to a}$ changes from $+\infty$ to 0. Find out what has happened at time $t_1 - 1$.

Therefore a fixed point is reached after a number of updates smaller or equal to the number of edges $N\Lambda'(1)$. There is also a clear stopping criterion: if in one update round no progress is made (i.e. if $h^{(t)}_{i\to a} = h^{(t+1)}_{i\to a}$ for all directed edges $i \to a$) then no progress will be made at successive rounds.

An alternative decoding formulation of BP decoding is the so-called **peeling algorithm**. The idea is to view decoding as a linear algebra problem. The code is defined through a linear system over $\mathbb{Z}_2$, of the form $\mathbb{H}\underline{x} = \underline{0}$. The output of an erasure channel fixes a fraction of the bits in the vector $\underline{x}$ (the non-erased ones). One is left with a linear system $\mathcal{L}$ over the remaining erased bits (not necessarily an homogeneous one). Decoding amounts to using this new linear system to determine the bits erased by the channel. If an equation in $\mathcal{L}$ contains a single variable x_i with non vanishing coefficient, it can be used to determine x_i, and replace it everywhere. One can then repeat this operation recursively until either all the variables have been fixed (in which case decoding is successful), or the residual linear systems includes only equations over two or more variables (in which case the decoder gets stuck).

Exercise 15.10 An explicit characterization of the fixed points of the peeling algorithm can be given in terms of **stopping sets** (or **2-cores**). A stopping set is a subset S of variable nodes in the factor graph such that each check has a number of neighbors in S which is either zero, or at least 2.

(*a*) Let S be the subset of undetermined bits when the peeling algorithm stops. Show that S is a stopping set.

(*b*) Show that the union of two stopping sets is a stopping set. Deduce that, given a subset of variable nodes U, there exists a unique 'largest' stopping set contained in U that contains any other stopping set in U.

(*c*) Let U be the set of erased bits. Show that S is the largest stopping set contained in U.

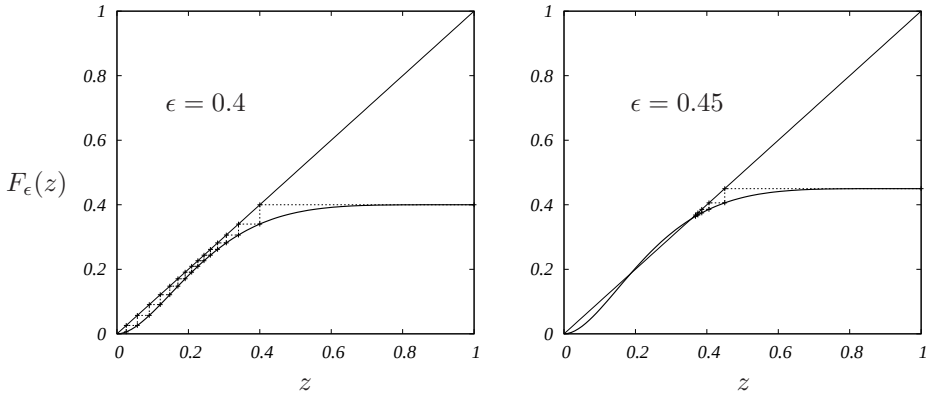


FIG. 15.4. Density evolution for the $(3, 6)$ LDPC ensemble over the erasure channel BEC(ϵ), for two values of ϵ below and above the BP threshold $\epsilon_{\rm d} = 0.4294$.

{fig:DEBEC}

Exercise 15.11 Let us prove that the peeling algorithm is indeed equivalent to BP decoding. As in the previous exercise, we denote by S the largest stopping set contained in the erased set U.

(*a*) Prove that, for any edge (i, a) with $i \in S$, $u^{(t)}_{a\to i} = h^{(t)}_{a\to i} = 0$ at all times.

(*b*) Vice-versa, let S' be the set of bits that are undetermined by BP after a fixed point is reached. Show that S' is a stopping set.

(*c*) Deduce that $S' = S$ (use the maximality property of S).

15.3.2 *Density evolution*

Consider BP decoding of an LDPC$_N(\Lambda, P)$ code after communication through a binary erasure channel. Under the assumption that the all-zero codeword has been transmitted, messages will take values in $\{0, +\infty\}$, and their distribution can be parameterized by a single real number. We let z_t denote the probability that $h^{(t)} = 0$, and by $\hat{z}_t$ the probability that $u^{(t)} = 0$. The density evolution recursions (15.11) translate into the following recursion on $\{z_t, \hat{z}_t\}$:

$$z_{t+1} = \epsilon\lambda(\hat{z}_t)\,, \qquad \hat{z}_t = 1 - \rho(1 - z_t)\,. \tag{15.33}$$

We can also eliminate $\hat{z}_t$ from this recursion to get $z_{t+1} = F_\epsilon(z_t)$, where we defined $F_\epsilon(z) \equiv \epsilon\lambda(1 - \rho(1 - z))$. The bit error rate after t iterations in the large block-length limit is $\mathrm{P}^{(t)}_{\rm b} = \epsilon\Lambda(\hat{z}_t)$.

In Fig. 15.4 we show as an illustration the recursion $z_{t+1} = F_\epsilon(z_t)$ for the $(3, 6)$ regular ensemble. The edge perspective degree distributions are $\lambda(z) = z^2$ and $\rho(z) = z^5$, so that $F_\epsilon(z) = \epsilon[1 - (1 - z)^2]^5$. Notice that $F_\epsilon(z)$ is a monotonously increasing function with $F_\epsilon(0) = 0$ (if the minimum variable node degree is at least 2), and $F_\epsilon(1) = \epsilon < 1$. As a consequence the sequence $\{z_t\}$ is decreasing

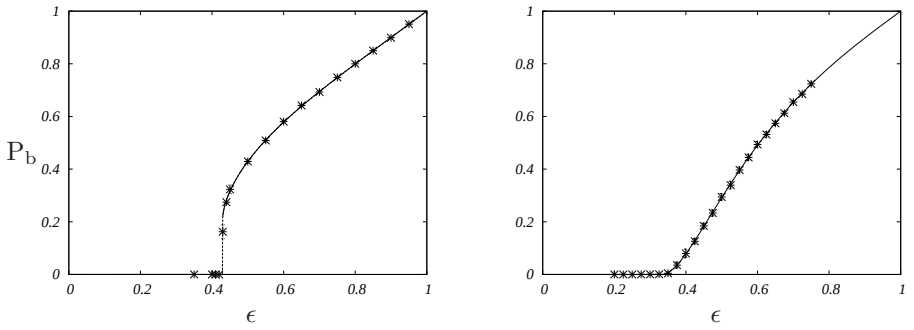


FIG. 15.5. The bit error rate under belief propagation decoding for the $(3,6)$ (left) and $(2,4)$ (right) ensembles. The prediction of density evolution (bold lines) is compared to numerical simulations (averaged over 10 code/channel realizations with block-length $N = 10^4$). For the $(3,6)$ ensemble $\epsilon_{\rm BP} \approx 0.4294 < \epsilon_{\rm loc} = \infty$, the transition is discontinuous. For the $(2,4)$ ensemble $\epsilon_{\rm BP} = \epsilon_{\rm loc} = 1/4$, the transition is continuous. {fig:36vs24bec}

and converges at large t to the largest fixed point of F_ϵ. In particular $z_t \to 0$ (and consequently $\mathrm{P}_{\rm b}^{\rm BP} = 0$) if and only if $F_\epsilon(z) < z$ for all $z \in (0,1]$. This yields the following explicit characterization of the BP threshold:

$$\epsilon_{\rm d} = \inf\left\{ \frac{z}{\lambda(1-\rho(1-z))} \;:\quad z \in (0,1] \right\} . \tag{15.34}$$

It is instructive to compare this characterization with the local stability threshold that in this case reads $\epsilon_{\rm loc} = 1/\lambda'(0)\rho'(1)$. It is obvious that $\epsilon_{\rm d} \le \epsilon_{\rm loc}$, since $\epsilon_{\rm loc} = \lim_{z\to 0} z/\lambda(1-\rho(1-z))$.

Two cases are possible, as illustrated in Fig. 15.5: either $\epsilon_{\rm d} = \epsilon_{\rm loc}$ or $\epsilon_{\rm d} < \epsilon_{\rm loc}$. Each one corresponds to a different behavior of the bit error rate. If $\epsilon_{\rm d} = \epsilon_{\rm loc}$, then, generically[53], $\mathrm{P}_{\rm b}^{\rm BP}(\epsilon)$ is a continuous function of ϵ at $\epsilon_{\rm d}$ with $\mathrm{P}_{\rm b}^{\rm BP}(\epsilon_{\rm d}+\delta) = C\delta + O(\delta^2)$ just above threshold. If on the other hand $\epsilon_{\rm d} < \epsilon_{\rm loc}$, then $\mathrm{P}_{\rm b}^{\rm BP}(\epsilon)$ is discontinuous at $\epsilon_{\rm d}$ with $\mathrm{P}_{\rm b}^{\rm BP}(\epsilon_{\rm d}+\delta) = \mathrm{P}_{\rm b}^{\rm BP,*} + C\delta^{1/2} + O(\delta)$ just above threshold.

Exercise 15.12 Consider communication over the binary erasure channel using random elements from the regular (k,l) ensemble, in the limit $k,l \to \infty$, with a fixed rate $R = 1 - l/k$. Prove that the BP threshold $\epsilon_{\rm d}$ tends to 0 in this limit.

15.3.3 *Ensemble optimization*

The explicit characterization (15.34) of the BP threshold for the binary erasure channel opens the way to the optimization of the code ensemble.

[53] Other behaviors are possible but they are not 'robust' with respect to a perturbation of the degree sequences.

A possible setup is the following. Fix an erasure probability $\epsilon \in (0,1)$: this is the estimated noise level on the channel that we are going to use. For a given degree sequence pair (λ, ρ), let $\epsilon_{\rm d}(\lambda,\rho)$ denote the corresponding BP threshold, and $R(\lambda,\rho) = 1 - \frac{\sum_k \rho_k/k}{\sum_l \lambda_l/l}$ be the design rate. Our objective is to maximize the rate, while keeping $\epsilon_{\rm d}(\lambda,\rho) \le \epsilon$. Let us assume that the check node degree distribution ρ is given. Finding the optimal variable node degree distribution can then be recast as a (infinite dimensional) linear programming problem:

$$\begin{cases} \text{maximize} & \sum_l \lambda_l/l\,, \\ \text{subject to} & \sum_l \lambda_l = 1 \\ & \lambda_l \ge 0 \quad \forall\, l\,, \\ & \epsilon\lambda(1-\rho(1-z)) \le z \quad \forall\, z \in (0,1]\,. \end{cases} \tag{15.35}$$

Notice that the constraint $\epsilon\lambda(1-\rho(1-z)) \le z$ is conflicting with the requirement of maximizing $\sum_l \lambda_l/l$, since both are increasing functions in each of the variables λ_l. As usual with linear programming, one can show that the objective function is maximized when the constraints saturate i.e. $\epsilon\lambda(1-\rho(1-z)) = z$ for all $z \in (0,1]$. This 'saturation condition' allows to derive λ, for a given ρ.

We shall do this in the simple case where the check nodes have uniform degree k, i.e. $\rho(z) = z^{k-1}$. The saturation condition implies $\lambda(z) = \frac{1}{\epsilon}[1-(1-z)^{\frac{1}{k-1}}]$. By Taylor expanding this expression we get, for $l \ge 2$

$$\lambda_l = \frac{(-1)^l}{\epsilon}\, \frac{\Gamma\left(\frac{1}{k-1}+1\right)}{\Gamma(l)\,\Gamma\left(\frac{1}{k-1}-l+2\right)}\,. \tag{15.36}$$

In particular $\lambda_2 = \frac{1}{(k-1)\epsilon}$, $\lambda_3 = \frac{(k-2)}{2(k-1)^2\epsilon}$, and $\lambda_l \simeq \lambda_\infty l^{-k/(k-1)}$ as $l \to \infty$. Unhappily this degree sequence does not satisfy the normalization condition in (15.35). In fact $\sum_l \lambda_l = \lambda(1) = 1/\epsilon$. This problem can however be overcome by truncating the series and letting $k \to \infty$, as shown in the exercise below. The final result is that a sequence of LDPC ensembles can be found that allows for reliable communication under BP decoding, at a rate that asymptotically achieved the channel capacity $C(\epsilon) = 1-\epsilon$. This is stated more formally below.

Theorem 15.12 *Let $\epsilon \in (0,1)$. Then there exists a sequence of degree distribution pairs $\{(\lambda^{(k)},\rho^{(k)})\}_k$, with $\rho^{(k)}(x) = x^{k-1}$ such that $\epsilon_{\rm d}(\lambda^{(k)},\rho^{(k)}) > \epsilon$ and $R(\lambda^{(k)},\rho^{(k)}) \to 1-\epsilon$.*

The precise construction of the sequence $(\lambda^{(k)},\rho^{(k)})$ is outlined in the next exercise. In Fig. 15.6 we show the BP error probability curves for this sequence of ensembles.

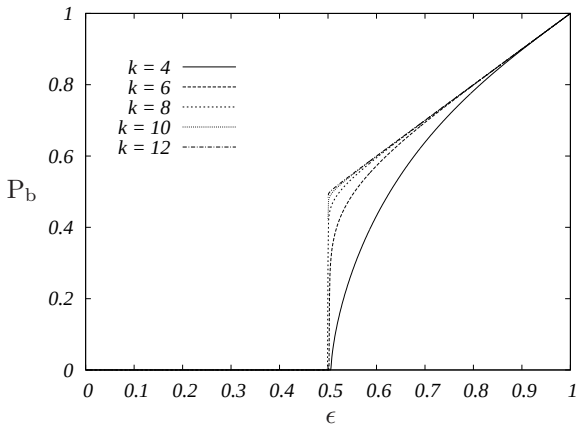


FIG. 15.6. Belief propagation bit error rate for $\mathrm{LDPC}_N(\Lambda, P)$ ensembles from the capacity achieving sequence $(\lambda^{(k)}, \rho^{(k)})$ defined in the main text. The sequence is constructed in such a way to achieve capacity at the noise level $\epsilon = 0.5$ (the corresponding capacity is $C(\epsilon) = 1 - \epsilon = 0.5$). The 5 ensembles considered here have design rates $R_{\mathrm{des}} = 0.42253$, 0.48097, 0.49594, 0.49894, 0.49976 (respectively for $k = 4$, 6, 8, 10, 12).

{fig:CapacityAchieving}

Exercise 15.13 Let $\rho^{(k)}(z) = z^{k-1}$, $\hat{\lambda}^{(k)}(z) = \frac{1}{\epsilon}[1 - (1-z)^{1/(k-1)}]$, and $z_L = \sum_{l=2}^{L} \hat{\lambda}_l^{(k)}$. Define $L(k, \epsilon)$ as the smallest value of L such that $z_L \geq 1$. Finally, set $\lambda_l^{(k)} = \hat{\lambda}_l^{(k)} / z_{L(k,\epsilon)}$ if $l \leq L(k, \epsilon)$ and $\lambda_l^{(k)} = 0$ otherwise.

(a) Show that $\epsilon\lambda^{(k)}(1-\rho^{(k)}(1-z)) < z$ for all $z \in (0, 1]$, and, as a consequence $\epsilon_{\mathrm{d}}(\lambda^{(k)}, \rho^{(k)}) > \epsilon$. [Hint: Use the fact that the coefficients λ_l in Eq. (15.36) are non-negative and hence $\lambda^{(k)}(x) \leq \hat{\lambda}^{(k)}(z)/z_{L(k,\epsilon)}$.]

(b) Show that, for any sequence $l(k)$, $\hat{\lambda}_{l(k)}^{(k)} \to 0$ as $k \to \infty$. Deduce that $L(k, \epsilon) \to \infty$ and $z_{L(k,\epsilon)} \to 1$ as $k \to \infty$.

(b) Prove that $\lim_{k\to\infty} R(\lambda^{(k)}, \rho^{(k)}) = \lim_{k\to\infty} 1 - \epsilon z_{L(k,\epsilon)} = 1 - \epsilon$.

15.4 Bethe free energy and optimal decoding

{sec:OptimalVSBP}

So far we have studied the performance of $\mathrm{LDPC}_N(\Lambda, P)$ ensembles under BP message passing decoding, in the large block-length limit. Remarkably, sharp asymptotic predictions can be obtained for optimal decoding as well, and they involve the same mathematical objects, namely messages distributions. We shall focus here on symbol MAP decoding for a channel family $\{\mathrm{BMS}(p)\}$ ordered by physical degradation. Analogously to Chapter 11, we can define a threshold p_{MAP} depending on the LDPC ensemble, such that MAP decoding allows to communicate reliably at all noise levels below p_{MAP}. We shall now compute p_{MAP} using the Bethe free energy.

Let us consider the entropy density $\mathfrak{h}_N = (1/N)\,\mathbb{E}H_N(\underline{X}|\underline{Y})$, averaged over the code ensemble. Intuitively speaking, this quantity allows to estimate the typical number of inputs with non-negligible probability for a given channel output. If $\mathfrak{h}_N$ is bounded away from 0 as $N \to \infty$, the typical channel output is likely to correspond to an exponential number of inputs. If on the other hand $\mathfrak{h}_N \to 0$, the correct input has to be searched among a sub-exponential number of candidates. A precise relation with the error probability is provided by Fano's inequality:

Proposition 15.13 *Let $\mathrm{P}_{\mathrm{b}}^N$ the symbol error probability for communication using a code of block-length N. Then*

$$\mathcal{H}(\mathrm{P}_{\mathrm{b}}^N) \geq \frac{H_N(\underline{X}|\underline{Y})}{N}\,.$$

In particular, if the entropy density $H_N(\underline{X}|\underline{Y})/N$ is bounded away from 0, so is $\mathrm{P}_{\mathrm{b}}^N$.

Although this gives only a bound, it suggests to identify the MAP threshold as the largest noise level such that $\mathfrak{h}_N \to 0$ as $N \to \infty$:

$$p_{\mathrm{MAP}} \equiv \sup\Big\{p\,:\,\lim_{N\to\infty}\mathfrak{h}_N = 0\Big\}\,. \tag{15.37}$$

The conditional entropy $H_N(\underline{X}|\underline{Y})$ is directly related to the free entropy of the model defined in (15.1). More precisely we have

$$H_N(\underline{X}|\underline{Y}) = \mathbb{E}_{\underline{y}} \log_2 Z(\underline{y}) - N\sum_y Q(y|0)\log_2 Q(y|0)\,, \tag{15.38}$$

where $\mathbb{E}_{\underline{y}}$ denotes expectation with respect to the output vector $\underline{y}$. In order to derive this expression, we first use the entropy chain rule to write (dropping the subscript N)

$$H(\underline{X}|\underline{Y}) = H(\underline{Y}|\underline{X}) + H(\underline{X}) - H(\underline{Y})\,. \tag{15.39}$$

Since the input message is uniform over the code $H(\underline{X}) = NR$. Further, since the channel is memoryless and symmetric $H(\underline{Y}|\underline{X}) = \sum_i H(Y_i|X_i) = NH(Y_i|X_i = 0) = -N\sum_y Q(y|0)\log_2 Q(y|0)$. Finally, rewriting the distribution (15.1) as

$$p(\underline{x}|\underline{y}) = \frac{|\mathfrak{C}|}{Z(\underline{y})}p(\underline{y},\underline{x})\,, \tag{15.40}$$

we can identify (by Bayes theorem) $Z(\underline{y}) = |\mathfrak{C}|\,p(\underline{y})$. The expression (15.38) follows by putting together these contributions.

The free-entropy $\mathbb{E}_y \log_2 Z(y)$ is the non-trivial term in Eq. (15.38). For LDPC codes, in the large N limit, it is natural to compute it using the Bethe approximation of Section 14.2.4. Suppose $\underline{u} = \{u_{a\to i}\}$, $\underline{h} = \{h_{i\to a}\}$ is a set of messages which solves the BP equations

$$h_{i\to a} = B_i + \sum_{b\in\partial i\setminus a} u_{b\to i}\,, \qquad u_{a\to i} = \operatorname{atanh}\left\{\prod_{j\in\partial a\setminus i}\tanh h_{j\to a}\right\}. \tag{15.41}$$

Then the corresponding Bethe free-entropy follows from Eq. (14.28):

$$\Phi(\underline{u},\underline{h}) = -\sum_{(ia)\in E}\log_2\left[\sum_{x_i} P_{u_{a\to i}}(x_i)P_{h_{i\to a}}(x_i)\right] \tag{15.42}$$
$$+\sum_{i=1}^{N}\log_2\left[\sum_{x_i} Q(y_i|x_i)\prod_{a\in\partial i} P_{u_{a\to i}}(x_i)\right] + \sum_{a=1}^{M}\log_2\left[\sum_{\underline{x}_a}\mathbb{I}_a(\underline{x})\prod_{i\in\partial a} P_{h_{i\to a}}(x_i)\right].$$

where we denote by $P_u(x)$ the distribution of a bit x whose log likelihood ratio is u, given by: $P_u(0) = 1/(1+e^{-2u})$, $P_u(1) = e^{-2u}/(1+e^{-2u})$.

We are interested in the expectation of this quantity with respect to the code and channel realization, in the $N\to\infty$ limit. We assume that messages are asymptotically identically distributed $u_{a\to i} \stackrel{\mathrm{d}}{=} u$, $h_{i\to a} \stackrel{\mathrm{d}}{=} u$, and that messages incoming in the same node along distinct edges are asymptotically independent. Under these hypotheses we get:

$$\lim_{N\to\infty}\frac{1}{N}\mathbb{E}_y\,\Phi(\widehat{\mathtt{m}},\underline{h}) = \phi - \sum_y Q(y|0)\log_2 Q(y|0)\,, \tag{15.43}$$

where

$$\phi = -\Lambda'(1)\,\mathbb{E}_{u,h}\log_2\left[\sum_x P_u(x)P_h(x)\right] + \mathbb{E}_l\mathbb{E}_y\mathbb{E}_{\{u_l\}}\log_2\left[\sum_x \frac{Q(y|x)}{Q(y,0)}\prod_{i=1}^{l} P_{u_l}(x)\right] -$$
$$+\frac{\Lambda'(1)}{P'(1)}\,\mathbb{E}_k\mathbb{E}_{\{h_i\}}\log_2\left[\sum_{x_1\dots x_k}\mathbb{I}_a(\underline{x})\prod_{i=1}^{k} P_{h_i}(x_i)\right]. \tag{15.44}$$

Here k and l are distributed according to P_k and Λ_l respectively, and $u_1, u_2, \dots$ (respectively $h_1, h_2, \dots$) are i.i.d.'s and distributed as u (respectively as h).

If the Bethe free energy is correct, ϕ should give the conditional entropy $\mathfrak{h}_N$. It turns out that this guess can be turned into a rigorous inequality:

Theorem 15.14 *If u, h are symmetric random variables satisfying the distributional identity $u \stackrel{\mathrm{d}}{=} \operatorname{atanh}\left\{\prod_{i=1}^{k-1}\tanh h_i\right\}$, then*

$$\lim_{N\to\infty}\mathfrak{h}_N \geq \phi_{u,h}\,. \tag{15.45}$$

{TableMAPThresholds}

l	k	$R_{\rm des}$	$p_{\rm d}$	$p_{\rm MAP}$	Shannon limit
3	4	1/4	0.1669(2)	0.2101(1)	0.2145018
3	5	2/5	0.1138(2)	0.1384(1)	0.1461024
3	6	1/2	0.0840(2)	0.1010(2)	0.1100279
4	6	1/3	0.1169(2)	0.1726(1)	0.1739524

Table 15.2 *MAP thresholds for the BSC channel are compared to the BP decoding thresholds, for a few regular* LDPC *ensembles*

It is natural to conjecture that the correct limit is obtained by optimizing the above lower bound, i.e.

$$\lim_{N\to\infty} \mathfrak{h}_N = \sup_{u,h} \phi_{u,h}\,, \tag{15.46}$$

where, once again the sup is taken over the couples of symmetric random variables u, h satisfying $u \stackrel{\rm d}{=} \operatorname{atanh}\left\{\prod_{i=1}^{k-1} \tanh h_i\right\}$ and $h \stackrel{\rm d}{=} B + \sum_{a=1}^{l-1} u_a$.

This conjecture has indeed been proved in the case of communication over the binary erasure channel for a large class of LDPC ensembles (including, for instance, regular ones).

The above expression is interesting because it establishes a bridge between BP and MAP decoding. For instance, it is immediate to show that it implies $p_{\rm BP} \le p_{\rm MAP}$:

Exercise 15.14 (*a*) Recall that $u, h = +\infty$ constitute a density evolution fixed point for any noise level. Show that $\phi_{h,u} = 0$ on such a fixed point.
(*b*) Use ordering by physical degradation to show that, if any other fixed point exists, then density evolution converges to it.
(*c*) Deduce that $p_{\rm BP} \le p_{\rm MAP}$.

Evaluating the expression (15.46) implies an a priori infinite dimensional optimization problem. In practice good approximations can be obtained through the following procedure:

1. Initialize h, u to a couple of symmetric random variables $h^{(0)}$, $u^{(0)}$.
2. Implement numerically the density evolution recursion (15.11) and iterate it until an approximate fixed point is attained.
3. Evaluate the functional $\phi_{u,h}$ on such a fixed point, after enforcing $u \stackrel{\rm d}{=} \operatorname{atanh}\left\{\prod_{i=1}^{k-1} \tanh h_i\right\}$ exactly.

The above procedure can be repeated for several different initializations $u^{(0)}$, $h^{(0)}$. The largest of the corresponding values of $\phi_{u,v}$ is then picked as an estimate for $\lim_{N\to\infty} \mathfrak{h}_N$.

While this procedure is not guaranteed to exhaust all the possible density evolution fixed points, it allows to compute a sequence of lower bounds to the

conditional entropy density. Further, in analogy with exactly solvable cases (such as the binary erasure channel) one expects a small finite number of density evolution fixed points. In particular, for regular ensembles and $p > p_{\rm BP}$, a unique (stable) fixed point is expected to exist apart from the no-error one $u, h = +\infty$. In Table 15.4 we present the corresponding MAP thresholds for a few regular ensembles.

Notes

Belief propagation was first applied to the decoding problem by Robert Gallager in his Ph. D. thesis (Gallager, 1963), and denoted there as the 'sum-product' algorithm. Several low-complexity alternative message-passing approaches were introduced in the same work, along with the basic ideas of their analysis.

The analysis of iterative decoding of irregular ensembles over the erasure channel was pioneered by Luby and co-workers in (Luby, Mitzenmacher, Shokrollahi, Spielman and Stemann, 1997; Luby, Mitzenmacher, Shokrollahi and Spielman, 1998; Luby, Mitzenmacher, Shokrollahi and Spielman, 2001*a*; Luby, Mitzenmacher, Shokrollahi and Spielman, 2001*b*). These papers also presented the first examples of capacity achieving sequences.

Density evolution for general binary memoryless symmetric channels was introduced in (Richardson and Urbanke, 2001*b*). The whole subject is surveyed in the review (Richardson and Urbanke, 2001*a*) as well as in the upcoming book (Richardson and Urbanke, 2006). One important property we left out is 'concentration:' the error probability under message passing decoding is, for most of the codes, close to its ensemble average, that is predicted by density evolution.

The design of capacity approaching LDPC ensembles for general BMS channels is discussed in (Chung, G. David Forney, Richardson and Urbanke, 2001; Richardson, Shokrollahi and Urbanke, 2001).

Since message passing allows for efficient decoding, one may wonder whether encoding (whose complexity is, a priori, $O(N^2)$) might become the bottleneck. Efficient encoding schemes are discussed in (Richardson and Urbanke, 2001*c*).

Tight bounds for the threshold under MAP decoding were first proved in (Montanari, 2005), and subsequently generalized in (Macris, 2005). An alternative proof technique uses the so-called area theorem and the related 'Maxwell construction' (Méasson, Montanari, Richardson and Urbanke, 2005*b*). Tightness of these bounds for the erasure channel was proved in (Méasson, Montanari and Urbanke, 2005*a*).

The analysis we describe in this Chapter is valid in the large block-length limit $N \to \infty$. In practical applications a large block-length translates into a corresponding communication delay. This has motivated a number of works that aims at estimating and optimizing LDPC codes at moderate block-lengths. Some pointers to this large literature can be found in (Di, Proietti, Richardson, Telatar and Urbanke, 2002; Amraoui, Montanari, Richardson and Urbanke, 2004; Amraoui, Montanari and Urbanke, 2007; Wang, Kulkarni and Poor, 2006; Kötter and Vontobel, 2003; Stepanov, Chernyak, Chertkov and Vasic, 2005).

16

THE ASSIGNMENT PROBLEM

Consider N 'agents' and N 'jobs', and suppose you are given the $N \times N$ matrix $\{E_{ij}\}$, where E_{ij} is the cost for having job j executed by agent i. Finding an assignment of agents to jobs that minimizes the cost is one of the most classical combinatorial optimization problems.

The minimum cost (also referred to as 'maximum weight') assignment problem is important both because of its many applications and because it can be solved in polynomial time. This motivated a number theoretical developments, both from the algorithms and the probability viewpoints.

Here we will study it as an application domain for message passing techniques. The assignment problem is in fact a success story of this approach. Given a generic instance of the assignment problem, the associated factor graph is not locally tree like. Nevertheless, the Min-Sum algorithm can be proved to converge to an optimal solution in polynomial time. Belief propagation (Sum-Product algorithm) can also be used for computing weighted sums over assignments, although much weaker guarantees exist in this case. A significant amount of work has been devoted to the study of random instances, mostly in the case where the costs E_{ij} are iid random variables. Typical properties (such as the cost of the optimal assignment) can be computed heuristically within the replica symmetric cavity method. It turns out that these calculations can be made fully rigorous.

In spite of this success of the replica symmetric cavity method, one must be warned that apparently harmless modifications of the problem can spoil it. One instance is the generalization of minimal cost assignment to multi-indices (say matching agents with jobs and houses). Even random instances of this problem are not described by the replica symmetric scenario. The more sophisticated replica symmetry breaking ideas, described in the next chapters, are required.

After defining the problem in Sec. 16.1, in Sec. 16.2 we compute the asymptotic optimal cost for random instances using the cavity method. In order to do this we write the Min-Sum equations. In Sec. 16.3 we prove convergence of the Min-Sum iteration to the optimal assignment. Section 16.4 contains a combinatorial proof that confirm the cavity result and provides sharper estimates. In sect. 16.5 we discuss a generalization of the assignment problem to a multi-assignment case.

{se:assign_def}

16.1 Assignment and random assignment ensembles

An instance of the assignment problem is determined by a cost matrix $\{E_{ij}\}$, indexed by $i \in A$ (the 'agents' set) and $j \in B$ (the 'jobs' set), with $|A| =$

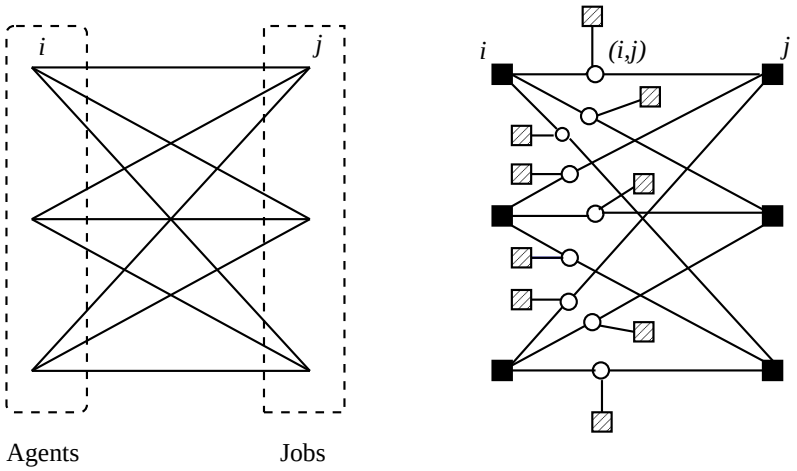


FIG. 16.1. Left: graphical representation of a small assignment problem with 3 agents and 3 jobs. Each edge carries a cost (not shown), the problem is to find a perfect matching, i.e. a set of 3 edges which are vertex disjoint, of minimal cost. Right: The factor graph corresponding to the representation (16.2) of this problem. Dashed squares are the function nodes associate with edge weights.

{fig:assignment_def}

$|B| = N$. We shall often identify A and B with the set $\{1, \dots, N\}$ and use indifferently the terms cost or energy in the following. An assignment is a one-to-one mapping of agents to jobs, that is a permutation π of $\{1, \dots, N\}$. The cost of an assignment π is $E(\pi) = \sum_{i=1}^{N} E_{i\pi(i)}$. The optimization problem consists in finding a permutation that minimizes $E(\pi)$.

We shall often use a graphical description of the problem as a weighted complete bipartite graph over vertices sets A and B. Each of the N^2 edges (i, j) carries a weight E_{ij}. The problem is to find a perfect matching in this graph (a subset M of edges such that every vertex is adjacent to exactly one edge in M), of minimal weight (see Fig. 16.1).

In the following we shall be interested in two types of questions. The first is to understand whether a minimum cost assignment for a given instance can be found efficiently through a message passing strategy. The second will be to analyze the typical properties of ensembles of random instances where the N^2 costs E_{ij} are iid random variables drawn from a distribution with density $\rho(E)$. One particularly convenient choice is that of exponentially distributed variables with probability density function $\rho(E) = e^{-E}\,\mathbb{I}(E \geq 0)$. Although the cavity method allows to tackle more general distribution, assuming exponential costs greatly simplifies the combinatorial approach.

16.2 Message passing and its probabilistic analysis

{se:cavity_assign}

16.2.1 *Statistical physics formulation and counting*

Following the general statistical physics approach, it is of interest to relax the optimization problem by introducing a finite inverse temperature β. The corresponding computational problem associates a weight to each possible matching, as follows.

Consider the complete bipartite graph over vertices sets A (agents), B (jobs). To any edge (i,j), $i \in A$, $j \in B$, we associate a variable which is an 'occupation number' $n_{ij} \in \{0,1\}$ encoding membership of edge (ij) in the matching: $n_{ij} = 1$ means that job j is done by agent i. We impose that the subset of edges (i,j) with $n_{ij} = 1$ be a matching of the complete bipartite graph:

{eq:matching_constraints}

$$\sum_{j\in B} n_{ij} \le 1 \quad \forall i \in A\,, \qquad \sum_{i\in A} n_{ij} \le 1 \quad \forall j \in B\,. \tag{16.1}$$

Let us denote by $\underline{n} = \{n_{ij} : i \in A\, j \in B\}$ the matrix of occupation numbers, and define the probability distribution

$$p(\underline{n}) = \frac{1}{Z} \prod_{i\in A} \mathbb{I}\left(\sum_{j\in B} n_{ij} \le 1\right) \prod_{j\in B} \mathbb{I}\left(\sum_{i\in A} n_{ij} \le 1\right) \prod_{(ij)} e^{-\beta n_{ij}(E_{ij}-2\gamma)} \tag{16.2}$$

The support of $p(\underline{n})$ corresponds to matchings, thanks to the 'hard constraints' enforcing conditions (16.1). The factor $\exp\left(2\beta\gamma \sum_{(ij)} n_{ij}\right)$ can be interpreted as a 'soft constraint': as $\gamma \to \infty$, the distribution concentrates on perfect matchings (the factor 2 is introduced here for future convenience). On the other hand, in the limit $\beta \to \infty$, the distribution (16.2) concentrates on the minimal cost assignments. The optimization problem is thus recovered in the double limit $\gamma \to \infty$ followed by $\beta \to \infty$.

There is a large degree of arbitrariness in the choice of which constraint should be 'softened' and how. The present one makes the whole problem most similar to the general class of graphical models that we study in this book. The factor graph obtained from (16.2) has the following structure (see Fig.16.1). It contains N^2 variable nodes, each associated with an edge (i,j) in the complete bipartite graph over vertices sets A, B. It also includes N^2 function nodes of degree one, one for each variable node, and $2N$ function nodes of degree N, associated with the vertices in A and B. The variable node (i,j), $i \in A$, $j \in B$ is connected to the two function nodes corresponding to i and j, as well as to the one corresponding to edge (i,j). The first two enforce the hard constraints (16.1); the third one corresponds to the weight $\exp\left[-\beta(E_{ij} - 2\gamma)n_{ij}\right]$.

In the case of random instances, we will be particularly interested in the thermodynamic limit $N \to \infty$. In order for this limit to be non-trivial the distribution (16.2) must be dominated neither by energy nor by entropy. Consider the case of iid costs $E_{ij} \ge 0$ with exponential density $\rho(E) = e^{-E}$. One can then

argue that low energy assignments have, with high probability, energy of order $O(1)$ as $N \to \infty$. The hand-waving reason is that for a given agent $i \in A$, and any fixed k, the k lowest costs among the ones of jobs that can be assigned to him (namely among $\{E_{ij} : j \in B\}$) are of order $O(1/N)$. The exercise below sketches a more formal proof. Since the entropy[54] is linear in N, we need to re-scale the costs for the two contributions to be of the same order.

To summarize, throughout our cavity analysis, we shall assume the edge cost to be drawn according to the 'rescaled pdf' $\hat{\rho}(E) = \frac{1}{N}\exp(-E/N)$. This choice ensures that the occupied edges in the best assignment have finite cost in the large N limit.

Exercise 16.1 Assume the energies E_{ij} to be iid exponential variables of mean η. Consider the 'greedy mapping' obtained by mapping each vertex $i \in A$ to that $j = \pi_1(i) \in B$ that minimizes E_{ij}, and call $E_1 = \sum_i E_{i,\pi_1(i)}$ the corresponding energy.

(*a*) Show that $\mathbb{E}\, E_1 = \eta$.

(*b*) Of course π_1 is not necessary injective and therefore not a valid matching. Let C be the number of collisions (i.e. the number of vertices $j \in B$ such that there exist several i with $\pi_1(i) = j$). Show that $\mathbb{E}\, C = N(1 - 2/e) + O(1)$, and that C is tightly concentrated around its expectation.

(*c*) Consider the following 'fix'. Construct π_1 in the greedy fashion described above, and let $\pi_2(i) = \pi_1(i)$ whenever i is the unique vertex mapped to $\pi_1(i)$. For each collision vertex $j \in B$, and each $i \in A$ such that $\pi_1(i) = j$, let j' be the vertex in B such that $E_{ij'}$ takes the smallest value among the vertices still un-matched. What is the expectation of the resulting energy $E_2 = \sum_i E_{i,\pi_2(i)}$? What is the number of residual collisions?

(*d*) How can this construction be continued?

16.2.2 *The belief propagation equations*

The BP equations for this problem are a particular instantiation of the general ones in (14.14,14.15). We will generically denote by i (respectively, j) a vertex in the set A (respectively, in B), in the complete bipartite graph, cf. Fig. 16.1

To be definite, let us write explicitly the equation for updating messages flowing from right to left (from vertices $j \in B$ to $i \in A$) in the graph of Fig. 16.1:

$$\nu_{ij\to i}(n_{ij}) \cong \widehat{\nu}_{j\to ij}(n_{ij})\, e^{-\beta n_{ij}(E_{ij}-2\gamma)}\,, \tag{16.3}$$

$$\widehat{\nu}_{j\to ij}(n_{ij}) \cong \sum_{\{n_{kj}\}} \mathbb{I}\Big[n_{ij} + \sum_{k\in A\setminus i} n_{kj} \le 1\Big] \prod_{k\in A\setminus i} \nu_{kj\to j}(n_{kj})\,. \tag{16.4}$$

The equations for messages moving from A to B, $\nu_{ij\to j}$ and $\widehat{\nu}_{i\to ij}$, are obtained by inverting the role of the two sets.

[54] The total number of assignments is $N!$ which would imply an entropy of order $N \log N$. However, if we limit the choices of $\pi(i)$ to those $j \in B$ such that the cost E_{ij} is comparable with the optimal one, the entropy becomes $O(N)$.

Since the variables n_{ij} take values in $\{0,1\}$, messages can be parameterized by a single real number, as usual. In the present case it is convenient to introduce rescaled log-likelihood ratios as follows:

{eq:BP_assignmentLLRDef}
$$x^{\rm L}_{j\to i} \equiv \gamma + \frac{1}{\beta}\log\left\{\frac{\widehat{\nu}_{j\to ij}(1)}{\widehat{\nu}_{j\to ij}(0)}\right\}\,, \quad x^{\rm R}_{i\to j} \equiv \gamma + \frac{1}{\beta}\log\left\{\frac{\widehat{\nu}_{i\to ij}(1)}{\widehat{\nu}_{i\to ij}(0)}\right\}\,, \qquad (16.5)$$

Variable-to-function node messages do not enter this definition, but they are easily expressed in terms of the quantities $x^{\rm L}_{i\to j}$, $x^{\rm R}_{i\to j}$ using Eq. (16.3). The BP equations (16.3), (16.4) can be written as:

{eq:BP_assignmentLLR}
$$\begin{aligned} x^{\rm L}_{j\to i} &= -\frac{1}{\beta}\,\log\Big\{e^{-\beta\gamma} + \sum_{k\in A\setminus i} e^{-\beta E_{kj}+\beta x^{\rm R}_{k\to j}}\Big\}\,,\\ x^{\rm R}_{i\to j} &= -\frac{1}{\beta}\,\log\Big\{e^{-\beta\gamma} + \sum_{k\in B\setminus j} e^{-\beta E_{ik}+\beta x^{\rm L}_{k\to i}}\Big\}\,. \end{aligned} \qquad (16.6)$$

Notice that the factor graph representation in Fig. 16.1, right frame, was helpful in writing down these equations. However, any reference to the factor graph disappeared in the latter, simplified form. This can be regarded as a a message passing procedure operating on the original complete bipartite graph, cf. Fig. 16.1, left frame.

{ex:marginals_assign}

Exercise 16.2 [Marginals] Consider the expectation value of n_{ij} with respect to the measure (16.2). Show that its BP estimate is $t_{ij}/(1+t_{ij})$, where $t_{ij} \equiv e^{\beta(x^{\rm L}_{j\to i}+x^{\rm R}_{i\to j}-E_{ij})}$

The Bethe free-entropy $\mathbb{F}(\underline{\nu})$ can be computed using the general formulae (14.27), (14.28). Writing it in terms of the log-likelihood ratio messages $\{x^{\rm R}_{i\to j}, x^{\rm L}_{j\to i}\}$ is straightforward but tedious. The resulting BP estimate for the free-entropy $\log Z$ is:

{free_entropy_assignment}
$$\begin{aligned} \mathbb{F}(\underline{x}) = 2N\beta\gamma - \sum_{i\in A, j\in B}\log\left[1+e^{-\beta(E_{ij}-x^{R}_{i\to j}-x^{L}_{j\to i})}\right] + \\ + \sum_{i\in A}\log\left[e^{-\beta\gamma}+\sum_j e^{-\beta(E_{ij}-x^{L}_{j\to i})}\right] + \sum_{j\in B}\log\left[e^{-\beta\gamma}+\sum_i e^{-\beta(E_{ij}-x^{R}_{i\to j})}\right]\,. \end{aligned} \qquad (16.7)$$

The exercise below provides a few guidelines for this computation.

Exercise 16.3 Consider the Bethe free-entropy (14.27) for the model (16.2).

(*a*) Show that it contains three types of function node terms, one type of variable node term, and three types of mixed (edge) terms.

(*b*) Show that the function node term associated with the weight $e^{-\beta n_{ij}(E_{ij}-2\gamma)}$ exactly cancels with the mixed term involving this same factor node and the variable node (i,j).

(*c*) Write explicitly each of the remaining terms, express it in terms of the messages messages $\{x^{\rm R}_{i\to j}, x^{\rm L}_{j\to i}\}$, and derive the result (16.7).
[Hint: The calculation can be simplified by recalling that the expression (14.27) does not change value if each message is independently rescaled]

16.2.3 *Zero temperature: The Min-Sum algorithm*

{sec:MinSumAssignment}

The BP equations (16.6) simplify in the double limit $\gamma\to\infty$ followed by $\beta\to\infty$ which is relevant for the minimum cost assignment problem. Assuming that $\{x^{\rm R}_{i\to j}, x^{\rm L}_{j\to i}\}$ remain finite in this limit, we get:

$$x^{\rm L}_{j\to i} = \min_{k\in A\setminus i}\left(E_{kj}-x^{\rm R}_{k\to j}\right)\,, \quad x^{\rm R}_{i\to j} = \min_{k\in B\setminus j}\left(E_{ik}-x^{\rm L}_{k\to i}\right)\,. \tag{16.8}$$

{eq:BP_assignment_T0}

Alternatively, the same equations can be obtained directly as the Min-Sum update rules. This derivation is outlined in the exercise below.

Exercise 16.4 Consider the Min-Sum equations (14.39), (14.40), applied to the graphical model (16.2).

(*a*) Show that the message arriving on variable node (ij) from the adjacent degree-1 factor node is equal to $n_{ij}(E_{ij}-2\gamma)$.

(*b*) Write the update equations for the other messages and eliminate the variable-to-function node messages $J_{ij\to i}(n_{ij})$, $J_{ij\to j}(n_{ij})$, in favor of the function-to-variable ones. Show that the resulting equations for function-to-variable messages read (cf. Fig. 16.1):

$$\begin{aligned}
\widehat{J}_{i\to ij}(1) &= \sum_{k\in B\setminus j}\widehat{J}_{k\to ik}(0)\,,\\
\widehat{J}_{i\to ij}(0) &= \sum_{k\in B\setminus j}\widehat{J}_{k\to ik}(0) + \min\{0; T_{ij}\}\\
T_{ij} &= \min_{l\in B\setminus j}\left\{\widehat{J}_{l\to il}(1) - \widehat{J}_{l\to il}(0) + E_{il} - 2\gamma\right\}.
\end{aligned}$$

(*c*) Define $x^{\rm R}_{i\to j} = \widehat{J}_{i\to ij}(0) - \widehat{J}_{i\to ij}(1) + \gamma$, and analogously $x^{\rm L}_{i\to j} = \widehat{J}_{j\to ij}(0) - \widehat{J}_{j\to ij}(1) + \gamma$. Write the above Min-Sum equations in terms of $\{x^{\rm R}_{i\to j}, x^{\rm L}_{j\to i}\}$.

(*d*) Show that, in the large γ limit, the update equations for the x-messages coincide with Eq. (16.8).

The Bethe estimate for the ground state energy (the cost of the optimal assignment) can be obtained by taking the $\gamma, \beta \to \infty$ limit of the free energy $-\mathbb{F}(\underline{x})/\beta$, whereby $\mathbb{F}(\underline{x})$ is the Bethe approximation for the log-partition function $\log Z$, cf. Eq. (16.7). Alternatively, we can use the fact that Min-Sum estimates the max-marginals of the graphical model (16.2). More precisely, for each pair (i,j), $i \in A$, $j \in B$, we define

$$J_{ij}(n_{ij}) \equiv n_{ij}(E_{ij} - 2\gamma) + \widehat{J}_{i\to ij}(n_{ij}) + \widehat{J}_{j\to ij}(n_{ij})\,, \tag{16.9}$$

$$n^*_{ij} \equiv \arg\min_{n\in\{0,1\}} J_{ij}(n)\,. \tag{16.10}$$

The interpretation of these quantities is that $e^{-J_{ij}(n)}$ is the message passing estimate for the max-marginal n_{ij} with respect to the distribution (16.2). Let us neglect the case of multiple optimal assignment (in particular, the probability of such an event vanishes for the random ensembles we shall consider). Under the assumption that message passing estimates are accurate, n_{ij} necessarily take the value n^*_{ij} in the optimal assignment, see Section 14.3. The resulting ground state energy estimate is $E_{\rm gs} = \sum_{ij} n^*_{ij} E_{ij}$.

In the limit $\gamma \to \infty$, Eq. (16.10) reduces to a simple **'inclusion principle'**: an edge ij is present in the optimal assignment (i.e. $n^*_{ij} = 1$) if and only if $E_{ij} \le x^R_{i\to j} + x^L_{j\to i}$. We invite the reader to compare this fact to the result of Exercise 16.2.

{sec:AssignmentDE}

16.2.4 *Distributional fixed point and* $\zeta(2)$

Let us now consider random instances of the assignment problem. For the sake of simplicity we assume that the edge costs E_{ij} are iid exponential random variables with mean N. We want to use the general density evolution technique of Section 14.6.2, to analyze Min-Sum message passing, cf. Eqs. (16.8).

The skeptical reader might notice that the assignment problem does not fit the general framework for density evolution, since the associated graph (the complete bipartite graph) is not locally tree like. Density evolution can nevertheless justified, through the following limiting procedure. Remove from the factor graph all the variables (ij), $i \in A$, $j \in B$ such that $E_{ij} > E_{\rm max}$, and the edges attached to them. Remembering that typical edge costs are of order $\Theta(N)$, it is easy to
⋆ check that the resulting graph is a sparse factor graph and therefore density evolution applies. On the other hand, one can prove that the error made in introducing a finite cutoff $E_{\rm max}$ is bounded uniformly in N by a quantity that vanishes as $E_{\rm max} \to \infty$, which justifies the use of density evolution. In the following we shall take the shortcut of writing density evolution equations for finite N without any cut-off and formally take the $N \to \infty$ limit on them.

Since the Min-Sum equations (16.8) involve minima, it is convenient to introduce the distribution function $\mathsf{A}_{N,t}(x) = \mathbb{P}\{x^{(t)}_{i\to j} \ge x\}$, where t indicates the iteration number, and $x^{(t)}$ refer to right moving messages (going from A to B) when t is even, and to left moving messages when t is odd. Then, density

evolution reads $\mathsf{A}_{N,t+1}(x) = [1 - \mathbb{E}\,\mathsf{A}_{N,t}(E-x)]^{N-1}$, where $\mathbb{E}$ denotes expectation with respect to E (that is an exponential random variable of mean N). Within the cavity method, one seeks fixed points of this recursion. These are the distributions that solve

$$\mathsf{A}_N(x) = [1 - \mathbb{E}\,\mathsf{A}_N(E-x)]^{N-1}\,. \tag{16.11}$$ {eq:FixPointFiniteN}

We want now to take the $N \to \infty$ limit. Assuming the fixed point $\mathsf{A}_N(x)$ has a (weak) limit $\mathsf{A}(x)$ we have

$$\mathbb{E}\,\mathsf{A}_N(E-x) = \frac{1}{N}\int_{-x}^{\infty} \mathsf{A}_N(y)\, e^{-(x+y)/N}\mathrm{d}y = \frac{1}{N}\int_{-x}^{\infty} \mathsf{A}(y)\,\mathrm{d}y + o(1/N) \tag{16.12}$$

It follows from Eq. (16.11) that the limit message distribution must satisfy the equation

$$\mathsf{A}(x) = \exp\left\{-\int_{-x}^{\infty} \mathsf{A}(y)\,\mathrm{d}y\right\}\,. \tag{16.13}$$ {eq:FixPointDEMatching}

This equation has the unique solution $\mathsf{A}(x) = 1/(1+e^x)$ corresponding to the density $\mathsf{a}(x) = \mathsf{A}'(x) = 1/[4\cosh^2(x)]$. It can be shown that density evolution does indeed converge to this fixed point.

Within the hypothesis of replica symmetry, cf. Sec. 14.6.3, we can use the above fixed point distribution to compute the asymptotic ground state energy (minimum cost). The most direct method is to use the inclusion principle: an edge (ij) is present in the optimal assignment if and only if $E_{ij} < x^{\mathrm{R}}_{i\to j} + x^{\mathrm{L}}_{j\to i}$. Therefore the conditional probability for (ij) to be in the optimal assignment, given its energy $E_{ij} = E$ is given by:

$$q(E) = \int \mathbb{I}(x_1 + x_2 \geq E)\;\; \mathsf{a}(x_1)\mathsf{a}(x_2)\,\mathrm{d}x_1\mathrm{d}x_2 = \frac{1+(E-1)e^E}{(e^E-1)^2} \tag{16.14}$$ {eq:rs_assignment2}

The expected cost E_* of the optimal assignment is equal to the number of edges, N^2, times the expectation of the edge cost, times the probability that the edge is in the optimal assignment. Asymptotically we have $E_* = N^2\mathbb{E}\{Eq(E)\}$:

$$\begin{aligned} E_* &= N^2 \int_0^{\infty} E\; N^{-1} e^{-E/N}\, q(E)\,\mathrm{d}E\; + o(N) \\ &= N\int_0^{\infty} E \frac{1+(E-1)e^E}{(e^E-1)^2}\mathrm{d}E + o(N) = N\zeta(2) + o(N)\,, \end{aligned}$$

where

$$\zeta(2) \equiv \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \approx 1.64493406684823\,. \tag{16.15}$$

Recall that this result holds when the edge weights are exponential random variables of mean N. If we reconsider the case of exponential random variables of mean 1, we get $E_* = \zeta(2) + o(1)$.

The reader can verify that the above derivation does not depend on the full distribution of the edges costs, but only on its behavior near $E = 0$. More precisely, for any edge costs distribution with a density $\rho(E)$ such that $\rho(0) = 1$, the cost of the optimal assignment converges to $\zeta(2)$.

Exercise 16.5 Suppose that the pdf of the costs $\rho(E)$ has support $\mathbb{R}_+$, and that $\rho(E) \simeq E^r$, for some $r > 0$, when $E \downarrow 0$.

(*a*) Show that, in order to have an optimal weight of order N, the edge costs must be rescaled by letting $E_{ij} = N^{r/r+1}\widetilde{E}_{ij}$ where $\widetilde{E}_{ij}$ have density ρ (i.e. typical costs must be of order $N^{r/r+1}$).

(*b*) Show that, within the replica symmetric cavity method, the asymptotic ($N \to \infty$) message distribution satisfies the following distributional equation

{eq:Gdef_rqcq}

$$\mathsf{A}(x) = \exp\left\{-\int_{-x}^{\infty}(x+y)^r\,\mathsf{A}(y)\,\mathrm{d}y\right\} \qquad (16.16)$$

(*c*) Assume that the solution $\mathsf{A}(x)$ to Eq. (16.16) is unique and that replica symmetry holds. Show that the expected ground state energy (in the problem with rescaled edge costs) is $E_* = N\epsilon_r + o(N)$, where $\epsilon_r \equiv \int \mathsf{A}(x)\,\log\frac{1}{\mathsf{A}(x)}\,\mathrm{d}x$. As a consequence the optimal cost in the initial problem is $N^{r/(r+1)}e_r(1+o(1))$.

(*d*) Equation (16.16) can be solved numerically with the population dynamics algorithm of Section 14.6.3. Write the corresponding program and show that the costs of the optimal matching for $r = 1, 2$ are: $e_1 \approx 0.8086$, $e_2 \approx 0.6382$.

ratureStabilityMatching}

16.2.5 *Non-zero temperature and stability analysis*

The reader may wonder whether the heuristic discussion of the previous sections can be justified. While a rigorous justification would lead us too far, we want to discuss, still at a heuristic level, the consistency of the approach. In particular we want to argue that BP provides good approximations to the marginals of the distribution (16.2), and that density evolution can be used to analyze its behavior on random instances.

Intuitively, two conditions should be verified for the approach to be valid: (*i*) The underlying factor graph should be locally tree-like; (*ii*) The correlation between two variables n_{ij}, n_{kl} should decay rapidly with the distance between edges (ij) and (kl) on such a graph.

At first sight it looks that condition (*i*) is far from holding, since our factor graph is constructed from a complete bipartite graph. As mentioned in the previous Section, the locally tree like structure emerges if one notices that only edges with cost of order 1 are relevant (as above we are assuming that edge costs have been rescaled or, equivalently, drawn with probability density function $\hat{\rho}(E) = N^{-1}\exp(-E/N)$). In order to further investigate this point, we modify the model (16.2) by pruning from the original graph all edges with cost

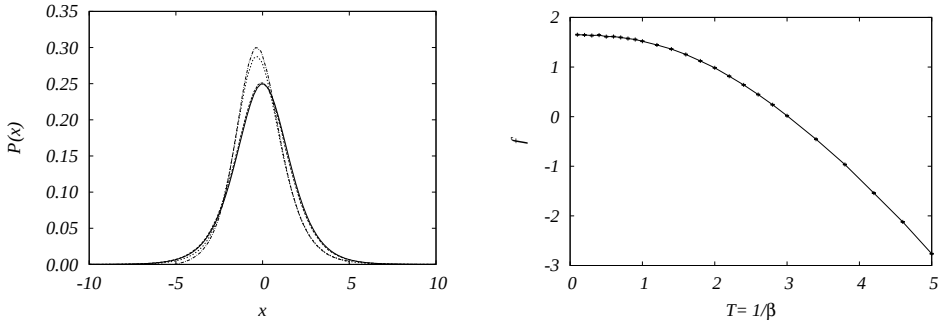


FIG. 16.2. Left frame: Estimate of the probability distribution of the messages $x_{i\to j}$ obtained by population dynamics. Here we consider the modified ensemble in which costly edges (with $E_{ij} > 2\gamma$) have been removed. The three curves, from top to bottom, correspond to: $(\beta = 1, \gamma = 5)$, $(\beta = 1, \gamma = 60)$,and $(\beta = 10, \gamma = 60)$. The last curve is indistinguishable from the analytical result for $(\beta = \infty, \gamma = \infty)$: $\mathsf{a}(x) = 1/[4\cosh^2(x/2)]$, also shown. The curves with larger γ are indistinguishable from the curve $\gamma = 60$ on this scale. The algorithm uses a population of size 10^5, and the whole population is updated 100 times. Right: Free energy versus temperature $T = 1/\beta$, computed using Eq. (16.18). The messages distribution was obtained as above with $\gamma = 40$. {fig:assign_pdex}

larger than 2γ. In the large β limit this modification will become irrelevant since the Boltzmann weight (16.2) ensures that these 'costly' edges of the original problem are not occupied. In the modified problem, the degree of any vertex in the graph converges (as $N \to \infty$) to a Poisson random variable with mean 2γ. The costs of 'surviving' edges converge to iid uniform random variables in the interval $[0, 2\gamma]$.

For fixed β and γ, the asymptotic message distribution can be computed from the RS cavity method. The corresponding fixed point equation reads

$$x \stackrel{\mathrm{d}}{=} -\frac{1}{\beta}\log\left[e^{-\beta\gamma} + \sum_{r=1}^{k} e^{-\beta(E_r - x_r)}\right] , \qquad (16.17) \quad \{\texttt{eq:rsd\_assign}\}$$

where k is a Poisson random variable with mean 2γ, E_r are iid uniformly distributed on $[0, 2\gamma]$, and x_r are iid with the same distribution as x. The fixed point distribution can be estimated easily using the population dynamics algorithm of Sec. 14.6.3. Results are shown in Fig. 16.2. For large β, γ the density estimated by this algorithm converges rapidly to the analytical result for $\beta = \gamma = \infty$, namely $\mathsf{a}(x) = 1/[4\cosh^2(x/2)]$.

The messages distribution can be used to compute the expected Bethe free-entropy. Assuming that the messages entering in Eq. (16.7) are independent, we get $\mathbb{E}\,\mathbb{F}(\underline{x}) = -N\beta f(\beta, \gamma) + o(N)$ where

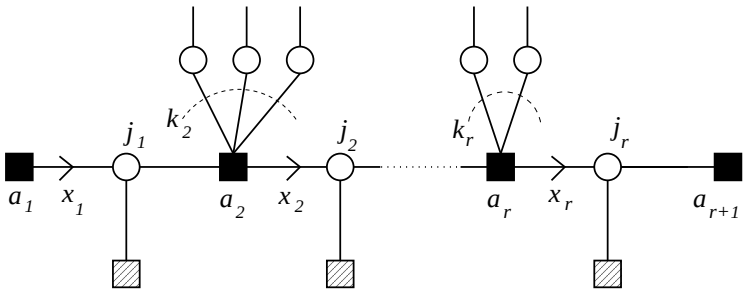


FIG. 16.3. Part of the factor graph used to compute the correlation between x_r and x_1.

{fig:assignment_stab}

$$f(\beta,\gamma) = -2\gamma - \frac{2}{\beta}\mathbb{E}\log\left[e^{-\beta\gamma} + \sum_{j=1}^{k} e^{-\beta(E_j - x_j)}\right] + \frac{2\gamma}{\beta}\mathbb{E}\log\left[1 + e^{-\beta(E_1 - x_1 - x_2)}\right] \tag{16.18}$$

{eq:rs_free_energy_assignment

Having a locally tree-like structure is only a necessary condition for BP to provide good approximations of the marginals. An additional condition is that correlations of distinct variables n_{ij}, n_{kl} decay rapidly enough with the distance between nodes (ij), (kl) in the factor graph. Let us discuss here one particular measure of these correlations, namely the spin glass susceptibility defined in Sec. 12.3.2. In the present case it can be written as

$$\chi_{\rm SG} \equiv \frac{1}{N}\sum_{e,f}\left(\langle n_e n_f\rangle - \langle n_e\rangle\langle n_f\rangle\right)^2 , \tag{16.19}$$

where the sum runs over all pairs of variable nodes $e = (i,j)$, $f = (k,l)$ in the factor graph (equivalently, over all pairs of edges in the original bipartite graph with vertex sets A, B).

If correlations decay fast enough $\chi_{\rm SG}$ should remain bounded as $N \to \infty$. The intuitive explanation goes as follows: From the fluctuation dissipation relation of Sec. 2.3, $\langle n_e n_f\rangle - \langle n_e\rangle\langle n_f\rangle$ is proportional to the change in $\langle n_f\rangle$ when the cost of edge e is perturbed. The sign of such a change will depend upon f, and therefore the resulting change in the expected matching size $\sum_f \langle n_f\rangle$ (namely $\sum_f(\langle n_e n_f\rangle - \langle n_e\rangle\langle n_f\rangle)$) can be either positive or negative. Assuming that this sum obeys a central limit theorem, its typical size is given by the square root of $\sum_f(\langle n_e n_f\rangle - \langle n_e\rangle\langle n_f\rangle)^2$. Averaging over the perturbed edge, we see that $\chi_{\rm SG}$ measures the decay of correlations.

We shall thus estimate $\chi_{\rm SG}$ using the same RS cavity assumption that we used in our computation of the expectations $\langle n_e\rangle$. If the resulting $\chi_{\rm SG}$ is infinite, such an assumption is falsified. In the opposite case, although nothing definite can be said, the assumption is said 'consistent', and the RS-solution is called 'locally stable' (since it is stable to small perturbations).

In order for the susceptibility to be finite, only couples of variable nodes (e, f) whose distance r in the factor graph is bounded should give a significant contribution to the susceptibility. We can then compute

$$\chi_{\rm SG}^{(r)} \equiv \frac{1}{N} \sum_{e,f:\, d(e,f)=r} \left(\langle n_e n_f\rangle - \langle n_e\rangle\langle n_f\rangle\right)^2 \tag{16.20}$$

for fixed r in the $N \to \infty$ limit, and then sum the result over r. For any given r and large N, there is (with high probability) a unique path of length r joining e to f, all the others being of length $\Theta(\log N)$. Denote by $(j_1, j_2, \dots, j_r)$ variable nodes, and by $(a_2, \dots, a_r)$ the function nodes on this path (with $e = j_1$, $f = j_r$), see Fig. 16.2.5.

Consider a fixed point of BP and denote by x_n the (log-likelihood) message passed from a_n to j_n. The BP fixed point equations (16.6) allow to compute x_r as a function of the message x_1 arriving on j_1, and of all the messages incoming on the path $\{a_2, \dots, a_r\}$ from edges outside this path, call them $\{y_{n,p}\}$:

$$\begin{aligned} x_2 &= -\frac{1}{\beta}\log\left\{e^{-\beta\gamma} + e^{-\beta(E_1 - x_1)} + \sum_{p=1}^{k_2} e^{-\beta(E_{2,p} - y_{2,p})}\right\}, \\ &\cdots\cdots\cdots \\ &\cdots\cdots\cdots \\ x_r &= -\frac{1}{\beta}\log\left\{e^{-\beta\gamma} + e^{-\beta(E_r - x_r)} + \sum_{p=1}^{k_r} e^{-\beta(E_{r,p} - y_{r,p})}\right\}. \end{aligned} \tag{16.21}$$

In a random instance, the k_n are iid Poisson random variables with mean 2γ, the E_n and $E_{n,p}$ variables are iid uniform on $[0, 2\gamma]$, and the $y_{n,p}$ are iid random variables with the same distribution as the solution of Eq. (16.17). We shall denote below by $\mathbb{E}_{\rm out}$ the expectation with respect to all of these variables outside the path. Keeping them fixed, a small change δx_1 of the message x_1 leads to a change $\delta x_r = \frac{\partial x_r}{\partial x_1}\delta x_1 = \frac{\partial x_2}{\partial x_1}\frac{\partial x_3}{\partial x_2}\cdots\frac{\partial x_r}{\partial x_{r-1}}\delta x_1$ of x_r. We leave it as an exercise to the reader to show that the correlation function ⋆

$$\langle n_e n_f\rangle - \langle n_e\rangle\langle n_f\rangle = C\frac{\partial x_r}{\partial x_1} = C\prod_{n=2}^{r}\frac{\partial x_n}{\partial x_{n-1}} \tag{16.22}$$

where the proportionality constant C is r-independent. Recalling that the expected number of variable nodes f such that $d(e, f) = r$ grows as $(2\gamma)^r$, and using Eq. (16.20), we have $\mathbb{E}\,\chi_{\rm SG}^{(r)} = C'\, e^{\lambda_r r}$, where

$$\lambda_r(\beta,\gamma) = \log(2\gamma) + \frac{1}{r}\log\left\{\mathbb{E}_{\rm out}\prod_{n=2}^{r}\left(\frac{\partial x_n}{\partial x_{n-1}}\right)^2\right\}. \tag{16.23}$$ {eq:assign_stab}

Therefore, a sufficient condition for the expectation of $\chi_{\rm SG}$ to be finite is to have $\lambda_r(\beta, \gamma)$ negative and bounded away from 0 for large enough r (when this happens, $\mathbb{E}\,\chi_{\rm SG}^{(r)}$ decays exponentially with r).

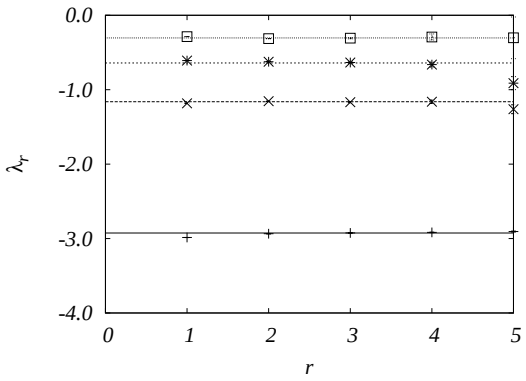


FIG. 16.4. Stability parameter λ_r, defined in Eq. (16.23), plotted versus r, for inverse temperatures $\beta = 10, 5, 2, 1$ (from bottom to top). Lines are guides to the eye. A negative asymptotic value of λ_r at large r shows that the spin glass susceptibility is finite. Data obtained from a population dynamics simulation with a population of 10^6, for $\gamma = 20$.

{fig:assign_stab_dat}

The exponent $\lambda_r(\beta, \gamma)$ can be computed numerically through population dynamics: the population allows to sample iid messages $y_{n,p}$ from the fixed point message density, and the costs E_n, $E_{n,p}$ are sampled uniformly in $[0, 2\gamma]$. The expectation (16.23) can be estimated through a numerical average over large enough populations. Notice that the quantity we are taking expectation of depends exponentially on r. As a consequence, its expectation becomes more difficult to compute as r grows.

In Fig. 16.2.5 we present some estimates of λ_r obtained through this approach. Since λ_r depends very weakly on r, we expect that λ_∞ can be safely estimated from these data. The data are compatible with the following scenario: $\lambda_\infty(\beta, \gamma)$ is negative at all finite β temperatures and vanishes as $1/\beta$ as $\beta \to \infty$. This indicates that $\chi_{\rm SG}$ is finite, so that the replica symmetry assumption is consistent.

16.3 A polynomial message passing algorithm

{se:BP_assign}

Remarkably, the Min-Sum message passing algorithm introduced in Section 16.2.3, can be proved to return the minimum cost assignment on any instance for which the minimum is unique. Let us state again the Min-Sum update equations of Eq. (16.8), writing the iteration number explicitly:

$$x^{\rm L}_{j\to i}(t+1) = \min_{k\in A\setminus i} \left(E_{kj} - x^{\rm R}_{k\to j}(t)\right) \ , \quad x^{\rm R}_{i\to j}(t) = \min_{k\in B\setminus j} \left(E_{ik} - x^{\rm L}_{k\to i}(t)\right) \ . \tag{16.24}$$

{eq:BPiter_assign}

Here, as before, A and B (with $|A| = |B| = N$) are the two vertices sets to be matched, and we keep denoting by i (respectively j) a generic vertex in A (resp. in B).

The algorithm runs as follows:

MIN-SUM ASSIGNMENT (Cost matrix E, Iterations t_*)
1: Set $x^{\rm L}_{j\to i}(0) = x^{\rm R}_{i\to j}(0) = 0$ for any $i \in A$, $j \in B$
2: For all $t \in \{0, 1, \dots, t_*\}$:
3: Compute the messages at time $t+1$ using Eq. (16.24)
4: Set $\pi(i) = \arg\min_{j\in B}\left(E_{ij} - x^{\rm L}_{j\to i}(t_*)\right)$ for each $i \in A$;
5: Output the permutation π ;

This algorithm finds the correct result if the optimal assignment is unique after a large enough number of iterations, as stated in the theorem below. {th:assign_BP_conv}

Theorem 16.1 *Let $W \equiv \max_{ij}|E_{ij}|$ and ϵ be the gap between the cost E^* of the optimal assignment, π^*, and the next best cost: $\epsilon \equiv \min_{\pi(\neq\pi^*)}(E(\pi) - E^*)$, where $E(\pi) \equiv \sum_{i=1}^N E_{i\pi(i)}$. Then, for any $t_* \geq 2NW/\epsilon$, the Min-Sum algorithm above returns the optimal assignment π^*.*

The proof is given in the Sec. 16.3.2, and is based on the notion of computation tree explained in the present context in Sec. 16.3.1.

For practical application of the algorithm to cases where one does not know the gap in advance, it is important to have a stopping criterion for the the algorithm. This can be obtained by noticing that, after convergence, the messages become 'periodic-up-to-a-drift' functions of t. More precisely there exists a period τ and a drift $C > 0$ such that for any $t > 2NW/\epsilon$, and any $i \in A$, $x^{\rm R}_{i\to j}(t+\tau) = x^{\rm R}_{i\to j}(t) + C$ if $j = \arg\min_{k\in B}(E_{ik} + x^{\rm L}_{k\to i}(t))$, and $x^{\rm R}_{i\to j}(t+\tau) = x^{\rm R}_{i\to j}(t) - C$ otherwise. If this happens, we shall write $\underline{x}^{\rm R}(t+\tau) = \underline{x}^{\rm R}(t) + \underline{C}$.

It turns out that: (*i*) If for some time t_0, period τ and constant $C > 0$, one has $\underline{x}^{\rm R}(t_0+\tau) = \underline{x}^{\rm R}(t_0) + \underline{C}$, then $\underline{x}^{\rm R}(t+\tau) = \underline{x}^{\rm R}(t) + \underline{C}$ for any $t \geq t_0$; (*ii*) Under the same condition, the permutation returned by the Min-Sum algorithm is independent of t_* for any $t_* \geq t_0$. We leave the proof of these statement as a (research level) exercise for the reader. It is immediate to see that they imply a ⋆ clear stopping criterion: After any number of iterations t, check whether there exists $t_0 < t$ and $C > 0$, such that $\underline{x}^{\rm R}(t) = \underline{x}^{\rm R}(t_0) + \underline{C}$. If this is the case halt the message passing updates and return the resulting permutation as in point 4 of the above pseudocode.

16.3.1 *The computation tree* {se:minsum_computree}

As we saw in Fig. 16.1, an instance of the assignment problem is characterized by the complete weighted bipartite graph $\mathcal{G}_N$ over vertices sets A, B, with $|A| = |B| = N$. The analysis of the Min Sum algorithm described above uses in a crucial way the notion of **computation tree**.

Given a vertex $i_0 \in A$ (the case $i_0 \in B$ is completely symmetric) the corresponding computation tree of depth t, $\mathbb{T}^t_{i_0}$ is a weighted rooted tree of depth t and degree N, that is constructed recursively as follows. First introduce the root $\hat{i}_0$ that is in correspondence with $i_0 \in A$. For any $j \in B$, add a corresponding

vertex $\hat{j}$ in $\mathbb{T}^t_{i_0}$ and connect it to $\hat{i}_0$. The weight of such an edge is taken to be $E_{\hat{i}_0,\hat{j}} \equiv E_{i_0,j}$. At any subsequent generation, if $\hat{i} \in \mathbb{T}^t_{i_0}$ corresponds to $i \in A$, and its direct ancestor is $\hat{j}$ that corresponds to $j \in B$, add $N-1$ direct descendants of $\hat{i}$ in $\mathbb{T}^t_{i_0}$. Each one of such descendants $\hat{k}$, corresponds to a distinct vertex $k \in B \setminus j$, and the corresponding weight is $E_{\hat{k}\hat{j}} = E_{kj}$.

A more compact description of the computation tree $\mathbb{T}^t_{i_0}$ consists in saying that it is the tree of non-reversing walks[55] rooted at i_0.

Imagine iterating the Min-Sum equations (16.24) on the computation tree $\mathbb{T}^t_{i_0}$ (starting from initial condition $x_{\hat{i}\to\hat{j}}(0) = 0$). Since $\mathbb{T}^t_{i_0}$ has the same local structure as $\mathcal{G}_N$, for any $s \le t$ the messages incoming to the root $\hat{i}_0$ coincide with the ones along the corresponding edges in the original graph $\mathcal{G}_N$: $x_{\hat{j}\to\hat{i}_0}(s) = x_{j\to i_0}(s)$.

In the proof of Theorem 16.1, we will use the basic fact that the Min-Sum algorithm correctly finds the ground state on trees (see theorem 14.4). More precisely, let us define an **internal matching** of a tree to be a subset of the edges such that each non-leaf vertex has one adjacent edge in the subset. In view of the above remark, we have the following property.

{lemma:assign_algo1}

Lemma 16.2 *Define, for any $i \in A$, $\pi^t(i) = \operatorname{argmin}_{j\in B}\left(E_{i,j} - x^L_{j\to i}(t)\right)$. Let $\hat{i}$ denote the root in the computation tree $\mathbb{T}^t_i$, and $\hat{j}$ the direct descendant of $\hat{i}$ that corresponds to $\pi^t(i)$.*

Then the edge $(\hat{i},\hat{j})$ belongs to the internal matching with lowest cost in $\mathbb{T}^t_i$ (assuming this is unique).

Although it follows from general principles, it is a useful exercise to re-derive this result explicitly.

Exercise 16.6 Let r be an internal (non-leaf) vertex in the computation $\mathbb{T}^t_i$, distinct from the root. Denote by S_r the set of its direct descendants (hence $|S_r| = N-1$), T_r the tree of all its descendants. We define a 'cavity internal matching' in T_r as a set of edges where all vertices which are distinct from the root r and are not leaves. Denote by A_r the cost of the optimal cavity internal matching when vertex r is not matched, and B_r its cost when vertex r is matched. Show that:

$$A_r = \sum_{q\in S_r} B_q \;\; ; \;\; B_r = \min_{q\in S_r}\left[A_q + E_{rq} + \sum_{q'\in S_r\setminus\{q\}} B_{q'}\right] \tag{16.25}$$

Show that $x_r = B_r - A_r$ satisfies the same equations as (16.24), and prove Lemma 16.2.

[55] A 'non-reversing walk' on a graph $\mathcal{G}$ is a sequence of vertices $\omega = (i_0, i_1, \dots, i_n)$, such that (i_s, i_{s+1}) is an edge for any $s \in \{0,\dots,n-1\}$, and $i_{s-1} \neq i_{s+1}$ for $s \in \{1,\dots,n-1\}$.

{se:minsum_assign_proof}

16.3.2 *Proof of convergence of the Min-Sum algorithm*

We can now prove Theorem 16.1. It will be convenient to represent assignments as matchings, i.e. subsets of the edges such that each vertex is incident to exactly one edge in the subset. In particular we denote the optimal matching on G as M^*. If π^* is the optimal assignment then $\mathsf{M}^* \equiv \{(i, \pi_*(i)) : i \in A\}$. We denote by π the mapping returned by the Min-Sum algorithm. It is not necessarily injective, therefore the subset of edges $\mathsf{M} = \{(i, \pi(i)) : i \in A\}$ is not necessarily a matching.

The proof is by contradiction. Assume that $\pi \neq \pi^*$. Then there exists at least one vertex in A, call it i_0, such that $\pi(i_0) \neq \pi^*(i_0)$. Consider the depth-t computation tree of i_0, $\mathbb{T}^t_{i_0}$, call $\hat{i}_0$ its root, and denote by $\widehat{\mathsf{M}}$ the optimal internal matching in this graph. Finally, denote by $\widehat{\mathsf{M}}^*$ the internal matching on $\mathbb{T}^t_{i_0}$ which is obtained by projection of the optimal one, M^*. Let $j = \pi(i_0) \in B$, and $\hat{j} \in \mathbb{T}^t_{i_0}$ be the neighbor of $\hat{i}_0$ whose projection on G is j. By Lemma 16.2 $(\hat{i}_0, \hat{j}) \in \widehat{\mathsf{M}}$. On the other hand, since $\pi(i_0) \neq \pi_*(i_0)$, $(\hat{i}_0, \hat{j}) \notin \widehat{\mathsf{M}}^*$. The idea is to construct a new internal matching $\widehat{\mathsf{M}}'$ on $\mathbb{T}^t_{i_0}$, such that: (i) $(\hat{i}_0, \hat{j}) \notin \widehat{\mathsf{M}}'$; (ii) The cost of $\widehat{\mathsf{M}}'$ is strictly smaller than the one $\widehat{\mathsf{M}}$, thus leading to a contradiction.

Intuitively, the improved matching $\widehat{\mathsf{M}}'$ is constructed by modifying $\widehat{\mathsf{M}}$ in such a way as to 'get closer' to $\widehat{\mathsf{M}}^*$. In order to formalize the idea, consider the symmetric difference of $\widehat{\mathsf{M}}$ and $\widehat{\mathsf{M}}^*$, $\widehat{\mathsf{P}}' = \widehat{\mathsf{M}} \triangle \widehat{\mathsf{M}}^*$, i.e. the set of edges which are either in $\widehat{\mathsf{M}}$ or in $\widehat{\mathsf{M}}^*$ but not in both. The edge $(\hat{i}_0, \hat{j})$ belongs to $\widehat{\mathsf{P}}'$. We can therefore consider the connected component of $\widehat{\mathsf{P}}'$ that contains $(\hat{i}_0, \hat{j})$, call it $\widehat{\mathsf{P}}$. A moment of thought reveals that $\widehat{\mathsf{P}}$ is a path on $\mathbb{T}^t_{i_0}$ with end-points on its leaves (see Fig. 16.3.2). Furthermore, its $2t$ edges alternate between edges in $\widehat{\mathsf{M}}$ and in $\widehat{\mathsf{M}}^*$. We can then define $\widehat{\mathsf{M}}' = \widehat{\mathsf{M}} \triangle \widehat{\mathsf{P}}$ (so that $\widehat{\mathsf{M}}'$ is obtained from $\widehat{\mathsf{M}}$ by deleting the edges in $\widehat{\mathsf{P}} \cap \widehat{\mathsf{M}}$ and adding those in $\widehat{\mathsf{P}} \cap \widehat{\mathsf{M}}^*$). We shall now show that, if t is large enough, the cost of $\widehat{\mathsf{M}}'$ is smaller than that of $\widehat{\mathsf{M}}$, in contradiction with the hypothesis.

Consider the projection of $\widehat{\mathsf{P}}$ onto the original complete bipartite graph G, call it $\mathsf{P} \equiv \varphi(\widehat{\mathsf{P}})$ (see Fig. 16.3.2). This is a non-reversing path of length $2t$ on G. As such, it can be decomposed into m simple cycles[56] $\{\mathsf{C}_1, \dots, \mathsf{C}_m\}$ (eventually with repetitions) and at most one even length path Q, whose lengths add up to $2N$. Furthermore, the length of Q is at most $2N - 2$, and the length of each of the cycles at most $2N$. As a consequence $m > t/N$.

Consider now a particular cycle, say C_s. Its edges alternate between edges belonging to the optimal matching M^* and edges not in it. As we assumed that the second best matching in G has cost at least ϵ above the best one, the total cost of edges in $\mathsf{C}_s \setminus \mathsf{M}_*$ is at least ϵ above the total cost of edges in $\mathsf{C}_s \cap \mathsf{M}_*$.

As for the path Q, it is again alternating between edges belonging to M^* and edges outside of M^*. We can order the edges in Q in such a way that the first

[56] A *simple cycle* is a cycle that does not visit the same vertex twice.

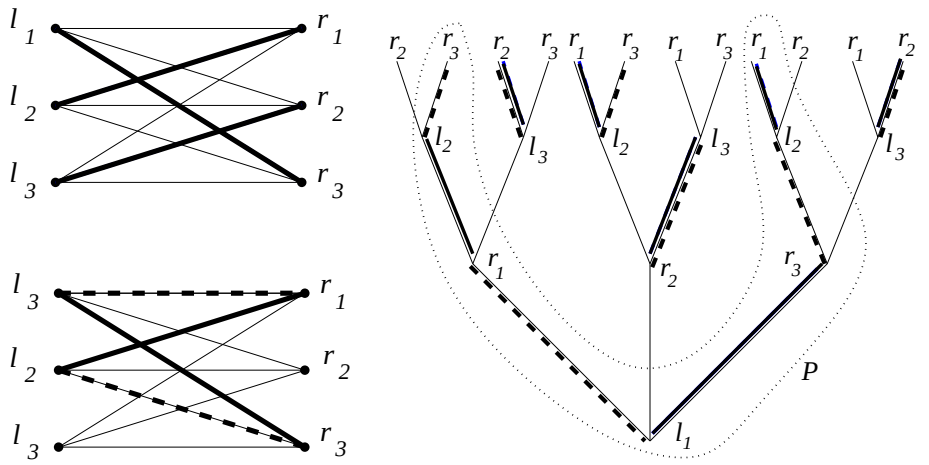


FIG. 16.5. Top left: an instance $\mathcal{G}$ of an assignment problem with $2N = 6$ vertices (costs are not shown). The optimal π^* is composed of the thick edges. Right: the computation tree $\mathbb{T}^2_{l_1}$. The matching π^* is 'lifted' to an internal matching in $\mathbb{T}^2_{l_1}$ composed of the thick edges. Notice that one edge in the original graph has many images in the unwrapped graph. The dashed edges are those of the optimal internal matching in $\mathbb{T}^2_{l_1}$, and the alternating path P is circled (dashed). Bottom left: the projection of P on the original graph; here it consists of a single cycle.

{fig:unwrapped}

one is in M^* and the last one is not. By changing the last step, we can transform it into an alternating cycle, to which the same analysis as above applies. This swapping changes the cost of edges not in Q by at most $2W$. Therefore the cost of the edges in $\mathsf{Q} \setminus \mathsf{M}^*$ is at least the cost of edges in $\mathsf{Q} \cap \mathsf{M}^*$ plus $\epsilon - 2|W|$.

Let $E_{\mathbb{T}}(\widehat{\mathsf{M}})$ denote the cost of matching $\widehat{\mathsf{M}}$ on $\mathbb{T}^t_{i_0}$. By summing the cost differences of the m cycles $\{\mathsf{C}_1, \dots, \mathsf{C}_m\}$ and the path Q, we found that $E_{\mathbb{T}}(\widehat{\mathsf{M}}) \geq E_{\mathbb{T}}(\widehat{\mathsf{M}}') + (m+1)\epsilon - 2W$. Therefore, for $t > 2NW/\epsilon$, $E_{\mathbb{T}}(\widehat{\mathsf{M}}) > E_{\mathbb{T}}(\widehat{\mathsf{M}}')$, in contradiction with our hypothesis.□

16.3.3 *A few remarks*

The alert reader might be puzzled by the following observation. Consider a random instance of the assignment problem with iid edge weights, e.g. exponentially distributed. In Section 16.2.4 we analyzed the Min-Sum algorithm through density evolution and showed that the only fixed point is given by the x-message density $\mathsf{a}(x) = \frac{1}{4}\cosh^2(x/2)$. A little more work shows that, when initiated with $x = 0$ messages density evolution does indeed converge to such a fixed point.

On the other hand, for such a random instance the maximum weight W and the gap between the two best assignments are almost surely finite, so the hypotheses of Thm 16.1 apply. The proof in the last Section implies that the

Min-Sum messages diverge: the messages $x_{i\to\pi^*(i)}$ diverge to $+\infty$, while other ones diverge to $-\infty$ (indeed Min-Sum messages are just the difference between the cost of optimal matching on the computation tree and the cost of the optimal matching that does not include the root).

How can these two behaviors be compatible? The conundrum is that density evolution correctly predicts the messages distribution as long as the number of iterations is kept bounded as $N \to \infty$. On the other hand, the typical scale for messages divergence discussed in the previous Section is NW/ϵ. If the edge weights are exponentials of mean N, the typical gap is $\epsilon = \Theta(1)$, while $W = \Theta(N \log N)$. Therefore the divergence sets in after $t_* = \Theta(N^2 \log N)$ iterations. The two analyses therefore describe completely distinct regimes.

16.4 Combinatorial results

{se:assign_combi}

It turns out that a direct combinatorial analysis allows to prove several non-asymptotic results for ensembles of random assignment problems. Although the techniques are quite specific, the final results are so elegant that they deserve being presented. As an offspring, they also provide rigorous proofs of some of our previous results, like the optimal cost $\zeta(2)$ found in (16.15).

We will consider here the case of edge weights given by iid exponential random variables with **rate** 1. Let us remind that an exponential random variable X with rate α has density $\rho(x) = \alpha\, e^{-\alpha x}$ for $x \geq 0$, and therefore its expectation is $\mathbb{E}[X] = 1/\alpha$. Equivalently, the distribution of X is given by $\mathbb{P}\{X \geq x\} = e^{-\alpha x}$ for $x \geq 0$.

Exponential random variables have several special properties that make them particularly convenient in the present context. The most important is that the minimum of two independent exponential random variables is again exponential. We shall use the following refined version of this statement:

{lemma:exponential_var}

Lemma 16.3 *Let $X_1, \dots, X_n$ be n independent exponential random variables with respective rates $\alpha_1, \dots, \alpha_n$. Then:*

1. *The random variable $X = \min\{X_1, \dots, X_n\}$ is exponential with rate $\alpha \equiv \sum_{i=1}^n \alpha_i$.*
2. *The random variable $I = \arg\min_i X_i$ is independent of X, and has distribution $\mathbb{P}\{I = i\} = \alpha_i/\alpha$.*

Proof: First notice that the minimum of $\{X_1, \dots, X_n\}$ is almost surely achieved by only one of the variables, and therefore the index I in point 2 is well defined. An explicit computation yields, for any $x \geq 0$ and $i \in \{1, \dots, n\}$

$$\begin{aligned}\mathbb{P}\{I = i,\, X \geq x\} &= \int_x^\infty \alpha_i\, e^{-\alpha_i z} \prod_{j(\neq i)} \mathbb{P}\{X_j \geq z\}\, \mathrm{d}z \\ &= \int_x^\infty \alpha_i\, e^{-\alpha z}\, \mathrm{d}z = \frac{\alpha_i}{\alpha}\, e^{-\alpha x}\,. \end{aligned} \tag{16.26}$$

By summing over $i = 1, \dots, n$, we get $\mathbb{P}\{X \geq x\} = e^{-\alpha x}$ which proves point 1.

By taking $x = 0$ in the above expression we get $\mathbb{P}\{I = i\} = \alpha_i/\alpha$. Using these two results, Eq. (16.26) can be rewritten as $\mathbb{P}\{I = i,\, X \geq x\} = \mathbb{P}\{I = i\}\,\mathbb{P}\{X \geq x\}$, which imply that X and I are independent. □

16.4.1 *The Coppersmith-Sorkin and Parisi formulae*

The combinatorial approach is based on a recursion on the size of the problem. It is therefore natural to generalize the assignment problem by allowing for partial matching between two sets of unequal size as follows. Given a set of agents A and a set of jobs B (with $|A| = M$ and $|B| = N$), consider the complete bipartite graph $\mathcal{G}$ over vertex sets A and B. A k-assignment between A and B is defined as a subset of k edges of G that has size k and such that each vertex is adjacent to at most one edge. Given edge costs $\{E_{ij} : i \in A\ j \in B\}$ the optimal k-assignment is the one that minimizes the sum of costs over edges in the matching. The assignment problem considered so far is recovered by setting $k = M = N$. Below we shall assume, without loss of generality, $k \leq M \leq N$:

{thm:CS_conj}

Theorem 16.4. (Coppersmith-Sorkin formula) *Assume the edge costs* $\{E_{ij} : i \in A,\, j \in B\}$ *to be iid exponential random variables of rate* 1*, with* $|A| = M$*,* $|B| = N$*, and let* $C_{k,M,N}$ *denote the expected cost of the optimal* k*-assignment. Then:*

{eq:CSMatching}

$$C_{k,M,N} = \sum_{i,j=0}^{k-1} \mathbb{I}(i + j < k)\, \frac{1}{(M-i)(N-j)}\,. \tag{16.27}$$

This result, that we shall prove in the next sections, yields, as a special case, the expected cost C_N of the complete matching over a bipartite graph with $2N$ vertices:

{coro:Parisi_conj}

Corollary 16.5. (Parisi formula) *Let* $C_N \equiv C_{N,N,N}$ *be the expected cost of the optimal complete matching among vertices sets* A*,* B *with* $|A| = |B| = N$*, assuming iid, exponential, rate* 1*, edge weights. Then*

$$C_N = \sum_{i=1}^{N} \frac{1}{i^2}\,. \tag{16.28}$$

In particular, the expected cost of the optimal assignment when $N \to \infty$ *is* $\zeta(2)$*.*

Proof: By Theorem 16.4 we have $C_N = \sum_{i,j=0}^{N-1} \mathbb{I}(i+j < N)\,(N-i)^{-1}(N-j)^{-1}$. Simplifying equal terms, the difference $C_{N+1} - C_N$ can be written as

$$\sum_{j=0}^{N} \frac{1}{(N+1)(N+1-j)} + \sum_{i=1}^{N} \frac{1}{(N+1-i)(N+1)} - \sum_{r=1}^{N} \frac{1}{(N+1-r)r}\,. \tag{16.29}$$

Applying the identity $\frac{1}{(N+1-r)r} = \frac{1}{(N+1)r} + \frac{1}{(N+1-r)(N+1)}$, this implies $C_{N+1} - C_N = 1/N^2$, which establishes Parisi's formula. □

16.4.2 *From k-assignment to $k+1$-assignment*

The proof of Theorem 16.4 relies on two Lemmas which relate properties of the optimal k-assignment to those of the optimal $(k+1)$-assignment. Let us denote by M_k the optimal k-assignment.

The first Lemma applies to any realization of the edge costs, provided that no two subsets of the edges have equal cost (this happens with probability 1 within our random cost model).

{lemma:nesting}

Lemma 16.6. (Nesting lemma) *Let $k < M \leq N$ and assume that no linear combination of the edge costs $\{E_{ij} : i \in A, j \in B\}$ with coefficients in $\{+1, 0, -1\}$ vanishes. Then every vertex that belongs to M_k also belongs to M_{k+1}.*

The matching M_k consists of k edges which are incident on the vertices $i_1, \ldots, i_k$ in set A, and on $j_1, \ldots, j_k$ in set B. Call A_k the $k \times k$ matrix which is the restriction of E to the lines $i_1, \ldots, i_k$ and the columns $j_1, \ldots, j_k$. The nesting lemma insures that A_{k+1} is obtained from A_k by adding one line (i_{k+1}) and one column (j_{k+1}). Therefore we have a sequence of nested matrices $E^{(1)} \subset E^{(2)} \cdots \subset E^{(M)} = E$ containing the sequence of optimal assignments $\mathsf{M}_1, \mathsf{M}_2, \ldots, \mathsf{M}_M$.

Proof: Color in red all the edges in M_k, in blue all the edges in M_{k+1}, and denote by G_{k+} the bipartite graph induced by edges in $\mathsf{M}_k \cup \mathsf{M}_{k+1}$. Clearly the maximum degree of G_{k+} is *at most* 2, and therefore its connected components are either cycles or paths.

We first notice that no component of G_{k+} can be a cycle. Assume by contradiction that edges $\{u_1, v_1, u_2, v_2, \ldots, u_p, v_p\} \subseteq \mathsf{G}_{k+}$ form such a cycle, with $\{u_1, \ldots, u_p\} \subseteq \mathsf{M}_k$ and $\{v_1, \ldots, v_p\} \subseteq \mathsf{M}_{k+1}$. Since M_k is the optimal k-assignment $E_{u_1} + \cdots + E_{u_p} \leq E_{v_1} + \cdots + E_{v_p}$ (in the opposite case we could decrease its cost by replacing the edges $\{u_1, \ldots, u_p\}$ with $\{v_1, \ldots, v_p\}$, without changing its size). On the other hand, since M_{k+1} is the optimal $(k+1)$-assignment, the same argument implies $E_{u_1} + \cdots + E_{u_p} \geq E_{v_1} + \cdots + E_{v_p}$. These two inequalities imply $E_{u_1} + \cdots + E_{u_p} = E_{v_1} + \cdots + E_{v_p}$, which is impossible by the non-degeneracy hypothesis.

So far we have proved that G_{k+} consists of a collection of disjoint simple paths, made of alternating blue and red edges. Along such paths all vertices have degree 2 except for the two endpoints which have degree 1. Since each path alternates between red and blue edges, the difference in their number is in at most 1 in absolute value. We will show that indeed there can exist only one such path, with one more blue than red edges, thus proving the thesis.

We first notice that G_{k+} cannot contain even paths, with as many red as blue edges. This can be shown using the same argument that we explained above in the case of cycles: either the cost of blue edges along the path is lower than the cost of red ones, which would imply that M_k is not optimal, or vice-versa, the cost of red edges is lower, which would imply that M_{k+1} is not optimal.

We now exclude the existence of a path P of odd length with one more red edge than blue edges. Since the total number of blue edges is larger than the total number of red edges, there should exist at least one path P' with odd length, with

one more blue edge than red edges. We can then consider the double path $\mathsf{P} \cup \mathsf{P}'$, which contains as many red as blue edges and apply to it the same argument as for cycles and even paths.

We thus conclude that the symmetric difference of M_k and M_{k+1} is a path of odd length, with one endpoint $i \in A$ and one $j \in B$. These are the only vertices that are in M_{k+1} but not in M_k. Reciprocally, there is no vertex that is M_k but not in M_{k+1}. □

{lemma:cost_nesting}

Lemma 16.7 *Let $\{u_i : i \in A\}$ and $\{v_j : j \in B\}$ be two collections of positive real numbers and assume that the edges costs $\{E_{ij} : i \in A, j \in B\}$ are independent exponential random variables, the rate of E_{ij} being $u_i v_j$. Denote by $A_k = \{i_1, \dots, i_k\} \subseteq A$, and $B_k = \{j_1, \dots, j_k\} \subseteq B$, the sets of vertices appearing in the optimal k-assignment M_k. Let $I_{k+1} = A_{k+1} \setminus A_k$ and $J_{k+1} = B_{k+1} \setminus B_k$ be the extra vertices which are added in M_{k+1}. Then the conditional distribution of I_{k+1} and J_{k+1} is $\mathbb{P}\{I_{k+1} = i,\, J_{k+1} = j | A_k, B_k\} = Q_{i,j}$, where*

$$Q_{ij} = \frac{u_i v_j}{\left(\sum_{i' \in A \setminus A_k} u_{i'}\right)\left(\sum_{j' \in B \setminus B_k} v_{j'}\right)} \,. \tag{16.30}$$

Proof: Because of the nesting lemma, one of the following must be true: Either the matching M_{k+1} contains edges (I_{k+1}, j_b), and (i_a, J_{k+1}) for some $i_a \in A_k$, $j_b \in B_k$, or it contains the edge (I_{k+1}, J_{k+1}).

Let us fix i_a and j_b and condition on the first event

$$\mathcal{E}_1(i_a, j_b) \equiv \{A_k, B_k,\, (I_{k+1}, j_b), (i_a, J_{k+1}) \in \mathsf{M}_{k+1}\} \,.$$

Then necessarily $E_{I_{k+1}, j_b} = \min\{E_{i j_b} : i \in A \setminus A_k\}$ (because otherwise we could decrease the cost of M_{k+1} by making a different choice for I_{k+1}). Analogously $E_{i_a, J_{k+1}} = \min\{E_{i_A j} : j \in B \setminus B_k\}$. Since the two minima are taken over independent random variables, I_{k+1} and J_{k+1} are independent as well. Further, by Lemma 16.3,

$$\mathbb{P}\{I_{k+1} = i,\, J_{k+1} = j \mid \mathcal{E}_1(i_a, j_b)\} = \frac{u_i v_{j_b}}{\sum_{i' \in A \setminus A_k} u_{i'} v_{j_b}} \, \frac{u_{i_a} v_j}{\sum_{j' \in B \setminus B_k} u_{i_a} v_{j'}} = Q_{ij} \,.$$

If we instead condition on the second event

$$\mathcal{E}_2 \equiv \{A_k, B_k,\, (I_{k+1}, J_{k+1}) \in \mathsf{M}_{k+1}\} \,,$$

then $E_{I_{k+1}, J_{k+1}} = \min\{E_{ij} : i \in A \setminus A_k\, j \in B \setminus B_k\}$ (because otherwise we could decrease the cost of M_{k+1}). By applying again Lemma 16.3 we get

$$\mathbb{P}\{I_{k+1} = i,\, J_{k+1} = j \mid \mathcal{E}_2\} = \frac{u_i v_j}{\sum_{i' \in A \setminus A_k, j' \in B \setminus B_k} u_{i'} v_{j'}} = Q_{ij} \,.$$

Since the resulting probability is Q_{ij} irrespective of the conditioning, it remains the same when we condition on the union of the events $\{\cup_{a,b} \mathcal{E}_1(i_a, j_b)\} \cup \mathcal{E}_2 = \{A_k, B_k\}$. □

16.4.3 *Proof of Theorem 16.4*

In order to prove the Coppersmith-Sorkin (C-S) formula (16.27), we will consider the difference $D_{k,M,N} \equiv C_{k,M,N} - C_{k-1,M,N-1}$, and establish in this section that:

eq:CS_formula_recursion}

$$D_{k,M,N} = \frac{1}{N}\left(\frac{1}{M} + \frac{1}{M-1} + \cdots + \frac{1}{M-k+1}\right) . \tag{16.31}$$

This immediately leads to the C-S formula, by recursion using as a base step the identity $C_{1,M,N-k+1} = \frac{1}{M(N-k+1)}$ (which follows from the fact that it is the minimum of $M(N-k+1)$ iid exponential random variables with rate 1).

Consider a random instance of the problem over vertex sets A and B with $|A| = M$ and $|B| = N$, whose edge costs $\{E_{ij} : i \in A,\, j \in B\}$ are iid exponential random variables with rate 1. Let X be the cost of its optimal k-assignment. Let Y be the cost of the optimal $(k-1)$-assignment for the new problem that is obtained by removing one fixed vertex, say the last one, from B. Our aim is to estimate the expectation value $D_{k,M,N} = \mathbb{E}(X - Y)$,

We shall use an intermediate problem with a cost matrix F of size $(M+1)\times N$ constructed as follows. The first M lines of F are identical to those of E. The matrix element in its last line are N iid exponential random variables of rate λ, independent from E. Denote by W the cost of the edge $(M+1, N)$, and let us call $\mathcal{E}$ the event "the optimal k-assignment in F uses the edge $(M+1, N)$".

We claim that, as $\lambda \to 0$, $\mathbb{P}(\mathcal{E}) = \lambda\mathbb{E}[X - Y] + O(\lambda^2)$. First notice that, if $\mathcal{E}$ is true, then $W + Y < X$, and therefore

$$\mathbb{P}(\mathcal{E}) \le \mathbb{P}\{W + Y < X\} = \mathbb{E}\big[1 - e^{-\lambda(X-Y)}\big] = \lambda\,\mathbb{E}[X - Y] + O(\lambda^2) \tag{16.32}$$

Conversely, if $W < X - Y$, and all the edges from the vertex $M+1$ in A to $B \setminus \{N\}$ have cost at least X, then the optimal k-assignment in F uses the edge $(M+1, N)$. Therefore, using the independence of the edge costs

$$\begin{aligned}
\mathbb{P}(\mathcal{E}) &\ge \mathbb{P}\{W < X - Y;\; E_{M+1,j} \ge X \text{ for } j \le N-1\} = \\
&= \mathbb{E}\Big\{\mathbb{P}\{W < X - Y \mid X, Y\} \prod_{j=1}^{N-1} \mathbb{P}\{E_{M+1,j} \ge X | \, X\}\Big\} \\
&= \mathbb{E}\Big\{\mathbb{P}\{W < X - Y \mid X, Y\}\, e^{-(N-1)\lambda X}\Big\} = \\
&= \mathbb{E}\Big\{\big(1 - e^{-\lambda(X-Y)}\big)\, e^{-(N-1)\lambda X}\Big\} = \lambda\,\mathbb{E}[X - Y] + O(\lambda^2)\,.
\end{aligned} \tag{16.33}$$

We now turn to the evaluation of $\mathbb{P}(\mathcal{E})$, and show that

$$\mathbb{P}(\mathcal{E}) = \frac{1}{N}\Big[1 - \prod_{r=0}^{k-1} \frac{M-r}{M-r+\lambda}\Big] . \tag{16.34}$$

{eq:plemmaCSPf}

Let us denote by α the $M+1$-th vertex in A. By Lemma 16.7, conditional to $\alpha \notin \mathsf{M}_{k-1}$, the probability that $\alpha \in \mathsf{M}_k$ is $\lambda/(M - (k-1) + \lambda)$. By recursion,

this shows that the probability that $\alpha \notin \mathsf{M}_{k-1}$ is $\prod_{r=0}^{k-1} \frac{M-r}{M-r+\lambda}$. Since all the N edges incident on α are statistically equivalent, we get (16.34).

Expanding Eq. (16.34) as $\lambda \to 0$, we get $\mathbb{P}(\mathcal{E}) = \frac{\lambda}{N} \sum_{r=0}^{k-1} \frac{1}{M-r} + O(\lambda^2)$. Since, as shown above, $\mathbb{E}[X - Y] = \lim_{\lambda \to 0} \mathbb{P}(\mathcal{E})/\lambda$, this proves Eq. (16.31), which establishes the C-S formula.□

{se:multi_assign}

16.5 An exercise: multi-index assignment

In Section 16.2.4 we computed the asymptotic minimum cost for random instances of the assignment problem using the cavity method under the replica symmetric (RS) assumption. The result, namely that the cost converges to $\zeta(2)$ for exponential edge weights with mean 1, was confirmed by the combinatorial analysis of Section 16.4. This suggests that the RS assumption is probably correct for this ensemble, an intuition that is further confirmed by the fact that Min-Sum finds the optimal assignment.

Statistical physicists conjecture that there exists a broad class of random combinatorial problems which satisfy the RS assumption. On the other hand, many problems are thought not to satisfy it: the techniques developed for dealing with such problems will be presented in the next chapters. In any case, it is important to have a feeling of the line separating RS from non-RS problems. This is a rather subtle point, here we want to illustrate it by considering a generalization of random assignment: the multi-index random assignment (MIRA) problem. We propose to study the MIRA using the RS cavity method and detect the inconsistency of this approach. Since the present Section is essentially an application of the methods developed above for the assignment, we will skip all technical details. The reader may consider it as a long guided exercise.

One instance of the multi-index assignment problem consists of d sets $A_1, \dots, A_d$, of N vertices, and a cost E_a for every d-uplet $a = (a_1, \dots, a_d) \in A_1 \times \cdots \times A_d$. A 'hyper-edge' a can be occupied ($n_a = 1$) or empty ($n_a = 0$). A matching is a set of hyper-edges which are vertex disjoint (formally: $\sum_{a:\, i \in a} n_a \le 1$ for each r and each $i \in A_r$). The cost of a matching is the sum of the costs of the hyper-edges that it occupies. The problem is to find a perfect matching (i.e. a matching with N occupied hyper-edges) with minimal total cost.

In order to define a random ensemble of multi-index assignment instances, we proceed as for the assignment problem, and assume that the edge costs E_i are iid exponential random variables with mean N^{d-1}. Thus the costs have density

$$\rho(E) = N^{-d+1}\, e^{-E/N^{d-1}}\, \mathbb{I}(E \ge 0)\,. \tag{16.35}$$

The reader is invited to check that under this scaling of the edge costs, the typical optimal cost is extensive, i.e. $\Theta(N)$. The simple random assignment problem considered before corresponds to $d = 2$.

We introduce the probability distribution on matchings that naturally generalizes Eq. (16.2):

$$p(\underline{n}) = \frac{1}{Z} \prod_{a \in \cup_r A_r} \mathbb{I}\Big(\sum_{i:\, a \in i} n_i \le 1 \Big)\, e^{-\beta \sum_i n_i (E_i - 2\gamma)} \,. \tag{16.36}$$

The associated factor graph has N^d variable nodes, each of degree d, corresponding to the original hyper-edges, and dN factor nodes, each of degree N, corresponding to the vertices in $F \equiv A_1 \cup \cdots \cup A_d$. As usual $i, j, \cdots \in V$ denote the variable nodes in the factor graph and $a, b, \cdots \in F$ the function nodes coding for hard constraints.

Using a parameterization analogous to the one for the assignment problem, one finds that the BP equations for this model take the form:

$$\begin{aligned} h_{i\to a} &= \sum_{b \in \partial i \setminus a} x_{b\to i}\,, \\ x_{a\to i} &= -\frac{1}{\beta} \log\Big\{ e^{-\beta\gamma} + \sum_{j \in \partial a \setminus i} e^{-\beta(E_j - h_{j\to a})} \Big\}. \end{aligned} \tag{16.37}$$

{eq:recrs}

In the large β, γ limit they become:

$$h_{i\to a} = \sum_{b \in \partial i \setminus a} x_{b\to i} \quad , \quad x_{a\to i} = \min_{j \in \partial a \setminus i} (E_j - h_{j\to a}). \tag{16.38}$$

Finally, the Bethe free-entropy can be written in terms of x-messages yielding:

$$\begin{aligned} \mathbb{F}[\underline{x}] =& N d \beta\gamma + \sum_{a \in F} \log\Big\{ e^{-\beta\gamma} + \sum_{i \in \partial a} e^{-\beta(E_i - \sum_{b \in \partial i \setminus a} x_{b\to i})} \Big\} \\ & - (d-1) \sum_{i \in V} \log\Big\{ 1 + e^{-\beta(E_i - \sum_{a \in \partial i} x_{j\to a})} \Big\}\,. \end{aligned} \tag{16.39}$$

{eq:BetheMIRA}

Using the RS cavity method, one obtains the following equation for the distribution of x messages in the $N \to \infty$ limit:

$$\mathsf{A}(x) = \exp\left\{ -\int \Big(x + \sum_{j=1}^{d-1} t_j \Big)\, \mathbb{I}\Big(x + \sum_{j=1}^{d-1} t_j \ge 0 \Big) \prod_{j=1}^{d-1} \mathrm{d}\mathsf{A}(t_j) \right\}. \tag{16.40}$$

This reduces to Eq. (16.13) in the case of simple assignment. Under the RS assumption the cost of the optimal assignment is $E_0 = N e_0 + o(N)$, where

$$e_0 = \frac{1}{2} \int \Big(\sum_{j=1}^{d} x_j \Big)^2\, \mathbb{I}\Big(\sum_j x_j > 0 \Big) \prod_{j=1}^{d} \mathrm{d}\mathsf{A}(x_j)\,. \tag{16.41}$$

{eq:energyinclusion}

These equations can be solved numerically to high precision and allow to derive several consequences of the RS assumption. However the resulting predictions (in particular, the cost of the optimal assignment) are *wrong* for $d \ge 3$. There are two observations showing that the RS assumption is inconsistent:

1. Using the Bethe free-entropy expression (16.39) we can compute the asymptotic free energy density as $f(T) = -\mathbb{F}/(N\beta)$, for a finite $\beta = 1/T$. The resulting expression can be estimated numerically via population dynamics, for instance for $d = 3$. It turns out that the entropy density $s(T) = -\mathrm{d}f/\mathrm{d}T$ becomes negative for $T < T_{\rm cr} \approx 2.43$. This is impossible: we are dealing with a statistical mechanics model with a finite state space, thus the entropy must be non-negative.
2. A local stability analysis can be performed analogously to what is done in Section 16.2.5. It turns out that , for $d = 3$, the stability coefficient λ_∞, cf. Eq. (16.23), becomes positive for $T \lesssim 1.6$, indicating an instability of the putative RS solution to small perturbations.

The same findings are generic for $d \geq 3$. A more satisfying set of predictions for such problems can be developed using the RSB cavity method that will be treated in Chap. **??**.

Notes

Rigorous upper bounds on the cost of the optimal random assignment go back to (Walkup, 1979) and (Karp, 1987). The $\zeta(2)$ result for the cost was first obtained in 1985 by (Mézard and Parisi, 1985) using the replica method. The cavity method solution was then found in (Mézard and Parisi, 1986; Krauth and Mézard, 1989), but the presentation in Sec. 16.2 is closer to (Martin, Mézard and Rivoire, 2005). This last paper deals the multi-index assignment and contains answers to the exercise in Sec. 16.5, as well as a proper solution of the problem using the RSB cavity method).

The first rigorous proof of the $\zeta(2)$ result was derived in (Aldous, 2001), using a method which can be regarded as a rigorous version of the cavity method. An essential step in elaborating this proof was the establishment of the existence of the limit, and its description as a minimum cost matching on an infinite tree (Aldous, 1992). An extended review on the 'objective method' on which this convergence result is based can be found in (Aldous and Steele, 2003). A survey of recursive distributional equations like (16.17) occurring in the replica symmetric cavity method is found in (Aldous and Bandyopadhyay, 2005).

On the algorithmic side, the assignment problem is a very well studied problem for many years (Papadimitriou and Steiglitz, 1998), and there exist efficient algorithms based on network flow ideas. The first BP algorithm was found in (Bayati, Shah and Sharma, 2005), it was then simplified in (Bayati, Shah and Sharma, 2006) into the $\mathrm{O}(N^3)$ algorithm presented in Sec. 16.3. This paper also shows that the BP algorithm is basically equivalent to Bertsekas' auction algorithm (Bertsekas, 1988). The periodic-up-to-a-shift stopping criterion is due to (Sportiello, 2004), and the understanding of the existence of diverging time scales for the onset of the drift was found in (Grosso, 2004)

Combinatorial studies of random assignments were initiated by Parisi's conjecture (Parisi, 1998*a*). This was generalized to the Coppersmith-Sorkin conjecture in (Coppersmith and Sorkin, 1999). The same paper also provides a nice