Understanding and Improving Belief Propagation

Een wetenschappelijke proeve op het gebied van de
Natuurwetenschappen, Wiskunde en Informatica

Proefschrift

ter verkrijging van de graad van doctor
aan de Radboud Universiteit Nijmegen,
op gezag van de rector magnificus prof. mr. S.C.J.J. Kortmann,
volgens besluit van het College van Decanen
in het openbaar te verdedigen op woensdag 7 mei 2008
om 13.30 uur precies

door

Joris Marten Mooij

geboren op 11 maart 1980
te Nijmegen

Promotor:

Prof. dr. H.J. Kappen

Manuscriptcommissie:

Prof. dr. N.P. Landsman
Prof. dr. M. Opper (University of Southampton)
Prof. dr. Z. Ghahramani (University of Cambridge)
Prof. dr. T.S. Jaakkola (Massachusetts Institute of Technology)
Dr. T. Heskes

The research reported here was sponsored by the Interactive Collaborative Information Systems (ICIS) project (supported by the Dutch Ministry of Economic Affairs, grant BSIK03024) and by the Dutch Technology Foundation (STW).

ISBN 978-90-9022787-0
Gedrukt door PrintPartners Ipskamp, Enschede

Contents

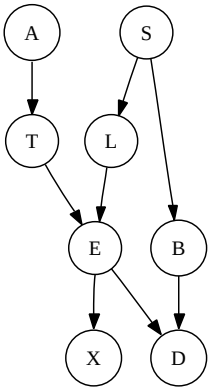
Chapter 1

Introduction

This chapter gives a short introduction to graphical models, explains the Belief Propagation algorithm that is central to this thesis and motivates the research reported in later chapters. The first section uses intuitively appealing examples to illustrate the most important concepts and should be readable even for those who have no background in science. Hopefully, it succeeds in giving a relatively clear answer to the question "Can you explain what your research is about?" that often causes the author some difficulties at birthday parties. The second section does assume a background in science. It gives more precise definitions of the concepts introduced earlier and it may be skipped by the less or differently specialized reader. The final section gives a short introduction to the research questions that are studied in this thesis.

1.1 A gentle introduction to graphical models

Central to the research reported in this thesis are the concepts of probability theory and graph theory, which are both branches of mathematics that occur widely in many different applications. Quite recently, these two branches of mathematics have been combined in the field of *graphical models*. In this section I will explain by means of two "canonical" examples the concept of graphical models. Graphical models can be roughly divided into two types, called *Bayesian networks* and *Markov random fields*. The concept of a Bayesian network will be introduced in the first subsection using an example from the context of medical diagnosis. In the second subsection, we will discuss the basic trade-off in the calculation of (approximations to) probabilities, namely that of computation time and accuracy. In the third subsection, the concept of a Markov random field will be explained using an example from the field of image processing. The fourth subsection is a short summary and



Random variable	Meaning
A	Recent trip to Asia
T	Patient has tuberculosis
S	Patient is a smoker
L	Patient has lung cancer
B	Patient has bronchitis
E	Patient has T and/or L
X	Chest X-Ray is positive
D	Patient has dyspnoea

Figure 1.1: The ASIA network, an example of a Bayesian network.

briefly describes the research questions addressed in this thesis.

1.1.1 The ASIA network: an example of a Bayesian network

To explain the concept of a Bayesian network, I will make use of the highly simplified and stylized hypothetical example of a doctor who tries to find the most probable diagnosis that explains the symptoms of a patient. This example, called the ASIA network, is borrowed from Lauritzen and Spiegelhalter [1988].

The ASIA network is a simple example of a Bayesian network. It describes the probabilistic relationships between different random variables, which in this particular example correspond to possible diseases, possible symptoms, risk factors and test results. The ASIA network illustrates the mathematical modeling of reasoning in the presence of uncertainty as it occurs in medical diagnosis.

A graphical representation of the ASIA network is given in figure 1.1. The *nodes* of the graph (visualized as circles) represent random variables. The *edges* of the graph connecting the nodes (visualized as arrows between the circles) represent probabilistic dependencies between the random variables. Qualitatively, the model represents the following (highly simplified) medical knowledge. A recent trip to Asia (A) increases the chance of contracting tuberculosis (T). Smoking (S) is a risk factor for both lung cancer (L) and bronchitis (B). The presence of either (E) tuberculosis or lung cancer can be detected by an X-ray (X), but the X-ray cannot distinguish between them. Dyspnoea (D), or shortness of breath, may be caused by either (E) tuberculosis or lung cancer, but also by bronchitis (B). In this particular Bayesian network, all these random variables can have two possible values: either "yes" or "no", which we will abbreviate as "y" and "n", respectively.[1]

This model can be used to answer several questions in the following hypothetical situation. Imagine a patient who complains about dyspnoea and has recently visited

[1] In general, the possible number of values of random variables is unlimited.

Asia. The doctor would like to know the probabilities that each of the diseases (lung cancer, tuberculosis and bronchitis) is present. Suppose that tuberculosis can be ruled out by another test, how would that change the belief in lung cancer? Further, would knowing smoking history or getting an X-ray be most informative about the probability of lung cancer? Finally, which information was the most important for forming the diagnosis?

In order to proceed, it will be convenient to introduce some notation from probability theory. The probability that some statement F is true is denoted by $\mathbb{P}(F)$. Probabilities are numbers between 0 and 1, where $\mathbb{P}(F) = 0$ means that F is false with absolute certainty, and $\mathbb{P}(F) = 1$ means that F is true with absolute certainty, and if $\mathbb{P}(F)$ is anything in between, it means that it is not certain whether F is true or false. If $\mathbb{P}(F)$ is close to 0, it is unlikely that F is true, wherease if $\mathbb{P}(F)$ is close to 1 it is likely that F is true. For our purposes, the statement F can be any instantiation of (one or more of) the random variables that we are considering. For example, the statement can be "the patient has bronchitis", which is an instantiation of the random variable B, that can be abbreviated as "$B = \mathrm{y}$". Another possible statement is "the patient does not have bronchitis", which we can write as "$B = \mathrm{n}$". Thus $\mathbb{P}(B = \mathrm{y})$ is the probability that the patient has bronchitis and $\mathbb{P}(B = \mathrm{n})$ is the probability that the patient does not have bronchitis. Both probabilities have to sum to one: $\mathbb{P}(B = \mathrm{y}) + \mathbb{P}(B = \mathrm{n}) = 1$, because the patient either has or does not have bronchitis. The statement can also be a more complicated combination of random variables, e.g., $\mathbb{P}(S = \mathrm{y}, L = \mathrm{n})$ is the probability that the patient smokes but does not have lung cancer.

In addition we need another notion and notation from probability theory, namely that of *conditional* probabilities. If we are given more information about the patient, probabilities may change. For example, the probability that the patient has lung cancer increases if we learn that the patient smokes. For statements F and G, the conditional probability of F, given that G is true, is denoted as $\mathbb{P}(F \,|\, G)$. As before, the statements F and G are instantiations of (some of the) random variables that we are considering. For example, the conditional probability that the patient has lung cancer given that the patient smokes is denoted as $\mathbb{P}(L = \mathrm{y} \,|\, S = \mathrm{y})$. The value of this conditional probability is higher than $\mathbb{P}(L = \mathrm{y})$, which is the probability that the patient has lung cancer if we have no further information about whether the patient smokes or not. Another example of a conditional probability is $\mathbb{P}(D = \mathrm{y} \,|\, B = \mathrm{y}, E = \mathrm{n})$; this is the probability that a patient has dyspnoea, given that the patient has bronchitis but has neither tuberculosis nor lung cancer.

The numerical values for the probabilities can be provided by medical studies. For example, according to the results of Villeneuve and Mao [1994], the lifetime probability of developing lung cancer, given that one is a smoker, is about 14%, whereas it is only about 1.4% if one has never smoked.[2] The complete *conditional*

[2]In reality, the probability of developing lung cancer is different for males and females and depends on many other variables, such as age and the smoking history of the patient. We will come back to this point later.

probability table for L given S (i.e., whether the patient develops lung cancer, given the smoking status of the patient), is then:

$\mathbb{P}(L \mid S)$	$S = \text{y}$	$S = \text{n}$
$L = \text{y}$	14%	1.4%
$L = \text{n}$	86%	98.6%

Note that each column sums to 100%, which expresses that with absolute certainty the patient either develops lung cancer or not. This conditional probability table for $\mathbb{P}(L \mid S)$ corresponds with the edge from S to L in figure 1.1.

Another conditional probability table that we can easily specify (even without consulting medical studies) is $\mathbb{P}(E \mid T, L)$:

$\mathbb{P}(E \mid T, L)$	$T = \text{y}$	$T = \text{n}$	$T = \text{y}$	$T = \text{n}$
	$L = \text{y}$	$L = \text{y}$	$L = \text{n}$	$L = \text{n}$
$E = \text{y}$	100%	100%	100%	0%
$E = \text{n}$	0%	0%	0%	100%

This simply reflects the definition of "T and/or L" in terms of T and L according to elementary logics. The conditional probability table for $\mathbb{P}(E \mid T, L)$ corresponds with the edges from T and L to E in figure 1.1.

Another probability table (not a conditional one) that is relevant here is $\mathbb{P}(S)$, the probability that the patient smokes. In 2006, the percentage of smokers in The Netherlands was 29.6%.[3] Therefore a realistic probability table for $\mathbb{P}(S)$ is:

S	$\mathbb{P}(S)$
y	29.6%
n	70.4%

This probability table corresponds with the node S in figure 1.1.

In order to give a complete quantitative specification of the graphical model shown in figure 1.1, one would have to specify each of the following probability tables: $\mathbb{P}(A)$, $\mathbb{P}(T \mid A)$, $\mathbb{P}(L \mid S)$, $\mathbb{P}(B \mid S)$, $\mathbb{P}(D \mid B, E)$, $\mathbb{P}(E \mid T, L)$, $\mathbb{P}(X \mid E)$ and $\mathbb{P}(S)$. Note how this corresponds with the graph: for each random variable, we need the probability distribution of that variable *conditional* on its parents. By the *parents* of a variable we mean those random variables that point directly towards it. For example, the parents of D are E and B, whereas S has no parents. This means that we have to specify the conditional probability table for $\mathbb{P}(D \mid E, S)$ and the probability table of $\mathbb{P}(S)$. We will not explicitly give all these (conditional) probability tables here but will assume that they are known and that the graphical model illustrated in figure 1.1 is thus completely specified. Then, the complete

[3] According to the CBS (Centraal Bureau voor Statistiek).

joint probability distribution of all the random variables can be obtained simply by multiplying all the (conditional) probability tables:

$$\begin{aligned}\mathbb{P}(A,T,L,S,B,X,E,D)&\\ = \mathbb{P}(A) \times \mathbb{P}(T \mid A) \times \mathbb{P}(L \mid S) \times \mathbb{P}(B \mid S)&\\ \times \mathbb{P}(E \mid T,L) \times \mathbb{P}(D \mid B,E) \times \mathbb{P}(X \mid E) \times \mathbb{P}(S).&\end{aligned} \tag{1.1}$$

This formula should be read as follows. Given an instantiation of all 8 random variables, e.g., $A = n, T = n, L = n, S = y, B = n, X = n, E = n, D = n$, we can calculate the probability of that instantiation by multiplying the corresponding values in the smaller probability tables:

$$\begin{aligned}\mathbb{P}(A = n, T = n, L = n, S = y, B = n, X = n, E = n, D = n)&\\ = \mathbb{P}(A = n) \times \mathbb{P}(T = n \mid A = n) \times \mathbb{P}(L = n \mid S = y)&\\ \times \mathbb{P}(B = n \mid S = y) \times \mathbb{P}(E = n \mid T = n, L = n)&\\ \times \mathbb{P}(D = n \mid B = n, E = n) \times \mathbb{P}(X = n \mid E = n) \times \mathbb{P}(S = y)&\end{aligned}$$

which will give us some number (0.150837 if you use the original model by Lauritzen and Spiegelhalter [1988]). Because the model consists of 8 binary (i.e., yes/no valued) random variables, the complete probability table for the joint probability distribution $\mathbb{P}(A,T,L,S,B,X,E,D)$ would contain $2^8 = 256$ entries: one value for each possible assignment of all the random variables. Part of this table is given in table 1.1.[4]

The completely specified model can be used to answer many different questions that a doctor might be interested in. Let us return to the hypothetical example of the patient who complains about dyspnoea and has recently visited Asia. The doctor would like to know the probability that each of the diseases (T, L and B) is present. Thus, e.g., the doctor would like to know the value of:

$$\mathbb{P}(T = \mathrm{y} \mid D = \mathrm{y}, A = \mathrm{y})$$

according to the model. An elementary fact of probability theory tells us that we can calculate this quantity by dividing two other probabilities:

$$\mathbb{P}(T = \mathrm{y} \mid D = \mathrm{y}, A = \mathrm{y}) = \frac{\mathbb{P}(T = \mathrm{y}, D = \mathrm{y}, A = \mathrm{y})}{\mathbb{P}(D = \mathrm{y}, A = \mathrm{y})}. \tag{1.2}$$

Another elementary result of probability says that, if we would like to calculate a probability of some instantiation of a particular subset of random variables, we

[4]Note that by representing the joint probability distribution as a Bayesian network we actually need less than 256 numbers to specify the complete probabilistic model; indeed, we need just $2 + 4 + 4 + 4 + 8 + 8 + 4 + 2 = 36$ numbers to specify all the probability tables $\mathbb{P}(A)$, $\mathbb{P}(T \mid A)$, $\mathbb{P}(L \mid S)$, $\mathbb{P}(B \mid S)$, $\mathbb{P}(E \mid T,L)$, $\mathbb{P}(D \mid B,E)$, $\mathbb{P}(X \mid E)$, $\mathbb{P}(S)$. In fact, we can use even less numbers because the columns of the tables sum to one. This *efficiency* of representation is one of the advantages of using a Bayesian network to specify a probability distribution over the alternative of "simply" writing down the complete joint probability table. By using equation (1.1), we can calculate any probability that we need.

Table 1.1: (Part of the) probability table for the full joint probability distribution of all 8 random variables in the ASIA network. Only part of the table is shown here; the full table has $2^8 = 256$ entries.

A	T	L	S	B	X	E	D	$\mathbb{P}(A, T, L, S, B, X, E, D)$
n	n	n	n	n	n	n	n	0.290362
n	n	n	n	n	n	n	y	0.032262
n	n	n	n	n	n	y	n	0.000000
n	n	n	n	n	n	y	y	0.000000
n	n	n	n	n	y	n	n	0.015282
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	n	n	y	n	n	n	n	0.150837
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
y	y	y	y	y	y	y	y	0.000013

have to sum the joint probability of all the random variables over *all* the possible instantiations of the *other* random variables. As an example, to calculate the numerator of the fraction in equation (1.2), we would like to calculate the probability $\mathbb{P}(T = \mathrm{y}, D = \mathrm{y}, A = \mathrm{y})$ of the instantiation $T = \mathrm{y}, D = \mathrm{y}, A = \mathrm{y}$ of the three random variables T, D, A. Thus we have to sum over all possible instantiations of the other random variables S, L, E, B, X. In mathematical notation:

$$\begin{aligned}
&\mathbb{P}(T = \mathrm{y}, D = \mathrm{y}, A = \mathrm{y}) \\
&\quad = \sum_{S,L,E,B,X} \mathbb{P}(A = \mathrm{y}, T = \mathrm{y}, L, S, B, X, E, D = \mathrm{y}) \\
&\quad = \sum_{S,L,E,B,X} \mathbb{P}(A = \mathrm{y})\ \mathbb{P}(T = \mathrm{y} \,|\, A = \mathrm{y})\ \mathbb{P}(L \,|\, S)\ \mathbb{P}(B \,|\, S) \times \\
&\qquad\qquad \mathbb{P}(E \,|\, T = \mathrm{y}, L)\ \mathbb{P}(D = \mathrm{y} \,|\, B, E)\ \mathbb{P}(X \,|\, E)\ \mathbb{P}(S),
\end{aligned}$$

where we used equation (1.1) to write out the joint probability in terms of the smaller probability tables. Because each random variable can have two possible values, there are $2 \times 2 \times 2 \times 2 \times 2 = 2^5 = 32$ possible instantiations of the random variables S, L, E, B, X that contribute to the sum. A similar formula can be derived for the denominator $\mathbb{P}(D = \mathrm{y}, A = \mathrm{y})$ of the fraction in equation (1.2); there we are interested in the probability of an instantiation of the random variables D, A and therefore we have to sum over all possible instantiations of the six other random variables T, S, L, E, B, X, which gives a sum of $2^6 = 64$ terms. In this way, we can calculate that $\mathbb{P}(T = \mathrm{y} \,|\, D = \mathrm{y}, A = \mathrm{y}) \approx 8.7751\%$ if one uses the same model specification as in [Lauritzen and Spiegelhalter, 1988].

Obviously, it would take a human quite some time to do these calculations.

However, computers can calculate very fast nowadays and we could instruct a computer in such a way that it performs precisely these calculations. Thereby it could assist the doctor by calculating (according to the model) the probabilities that each of the diseases is present, given that the patient has dyspnoea and has recently visited Asia. In a similar way, the elementary rules of probability theory can be applied in a rather straightforward manner to answer more difficult questions, like "If tuberculosis were ruled out by another test, how would that change the belief in lung cancer?", "Would knowing smoking history or getting an X-ray contribute most information about cancer, given that smoking may 'explain away' the dyspnoea since bronchitis is considered a possibility?" and "When all information is in, can we identify which was the most influential in forming our judgment?"

For readers that have not had any previous exposure to probability theory, the reasoning above may be difficult to understand. However, the important point is the following: any probability distribution that the doctor may be interested in (concerning the random variables in the model) in order to obtain a diagnosis for the patient, can be calculated using elementary probability theory and the precise specification of the Bayesian network. For a human this would be a lengthy calculation, but a computer can do these calculations very fast (at least for this particular Bayesian network).

As a final note, let us return to the probability that one develops lung cancer given that one smokes. One might object that this probability depends in reality on many other factors, such as the gender, the age, the amount of smoking and the number of years that the patient has been smoking. However, we can in principle easily improve the realism of the model to take these dependences into account, e.g., by adding nodes for gender (G), age (Y), smoking history (H), adding edges from these new nodes to L (lung cancer) and replacing the conditional probability table $\mathbb{P}(L \mid S)$ by a more complicated table $\mathbb{P}(L \mid S, G, Y, H)$ where the probability of developing lung cancer depends on more variables than in our simple model. This illustrates the *modularity* inherent in this way of modeling: if new medical studies result in more accurate knowledge about the chances of getting lung cancer from smoking, one only needs to modify the model locally (i.e., only change the model in the neighborhood of the nodes S and L). The rules of probability theory will ensure that answers to questions like "What disease most likely causes the positive X-ray?" depend on the *complete* model; improving the realism of the part of the model involving lung cancer and smoking will also automatically give a more accurate answer to those questions.

1.1.2 The trade-off between computation time and accuracy

For this small and simplified model, the calculations involved could even be done by hand. However, for larger and more realistic models, which may involve tens of thousands of variables which interact in highly complicated ways, the computational complexity to calculate the answer to questions like "What is the most likely disease

Table 1.2: The number of possible instantiations (i.e., joint assignments of all variables) as a function of the number N of binary variables.

N	2^N, the number of possible instantiations
1	2
2	4
3	8
4	16
5	32
10	1024
20	1048576
50	1125899906842624
100	1267650600228229401496703205376
200	1606938044258990275541962092341162602522202993782792835301376

that causes these symptoms?" explodes. Although it is easy in principle to write down a formula that gives the answer to that question (this would be a rather long formula, but similar to the ones we have seen before), to actually calculate the result would involve adding enormous amounts of numbers. Indeed, in order to calculate a probability involving a few random variables, we have to add many probabilities, namely one for each possible instantiation of *all the other random variables* that we are not interested in. The number of such instantiations quickly increases with the number of variables, as shown in table 1.2. Even if we include only 200 diseases and symptoms in our model, in order to calculate the probability of one disease given a few symptoms would require adding an enormous amount of terms. Although a modern desktop computer can do many additions per second (about one billion, i.e., 1000000000), we conclude that for a realistic model involving thousands of variables, the patient will have died before the computer can calculate the probability of a single disease, even if we use the fastest supercomputer on earth.[5] Thus although for small models we can actually calculate the probabilities of interest according to the model, it is completely impractical for large realistic models.

It turns out that for the specific case of medical diagnosis, using certain assumptions on the probability distributions and several clever tricks (which are outside the scope of this introduction) one can significantly decrease the computation time needed to calculate the probabilities. Such a model, called PROMEDAS, which contains thousands of diseases, symptoms and tests, is currently being developed in Nijmegen. It can calculate probabilities of interest within a few seconds on an

[5] In fact, it is likely that the earth and maybe even the universe have ceased to exist before a computer (using current technology) will have calculated the probability of interest if one uses this "brute force" method of calculating probabilities. On the other hand, computers get faster each year: processing speed roughly doubles every 24 months. Extrapolating this variant of "Moore's law" into the far future (which is not very realistic), it would take about three centuries before the calculation could be done within one day.


Figure 1.2: Left: reference image. Right: input image. The reference image defines the background and the input image consists of some interesting foreground imposed on the background. The task is to decide which part of the input image belongs to the foreground.

ordinary computer.[6]

However, there are many other applications (for which these simplifying assumptions cannot be made and the clever tricks cannot be applied) where the exact calculation of probabilities is impossible to perform within a reasonable amount of time. In these cases, one can try to calculate *approximate* probabilities using advanced approximation methods which have been specially developed for this purpose. If the approximate result can be calculated within a reasonable amount of time and its accuracy is enough for the application considered (e.g., knowing the probabilities that some disease causes the observed symptoms to ten decimal places is usually not necessary, one or two decimal places may be more than enough), then this forms a viable alternative to the exact calculation. This illustrates the basic trade-off in the field known as *approximate inference*: computation time versus accuracy.

1.1.3 Image processing: an example of a Markov random field

To introduce the concept of a Markov random field, another type of graphical models, I will use an example from the field of image processing. The task that we will consider is that of separating foreground from background. Suppose that we have two images, one of which is the *reference image* that defines the background, and one where there is some foreground in front of the background, which we call the *input image* (see figure 1.2 for an example). By comparing the input image with the reference image, we try to infer which part of the input image is foreground and which part belongs to the background. We can then extract only the foreground part of the input image, filtering out the background. This may have applications in surveillance (surveillance cameras only need to store the interesting parts of the

[6] A demonstration version of Promedas is available at `http://www.promedas.nl/`

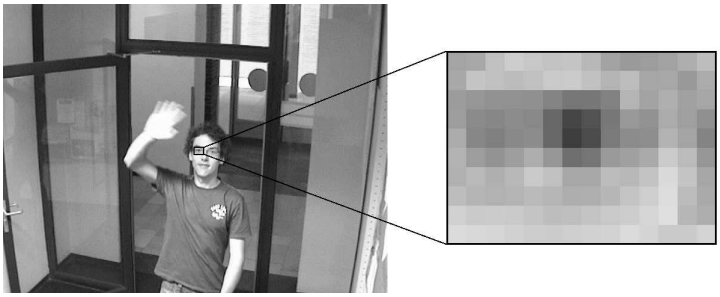


Figure 1.3: An image consists of many *pixels*, small squares with a uniform intensity. The images used here consist of $640 \times 480 = 307200$ pixels.

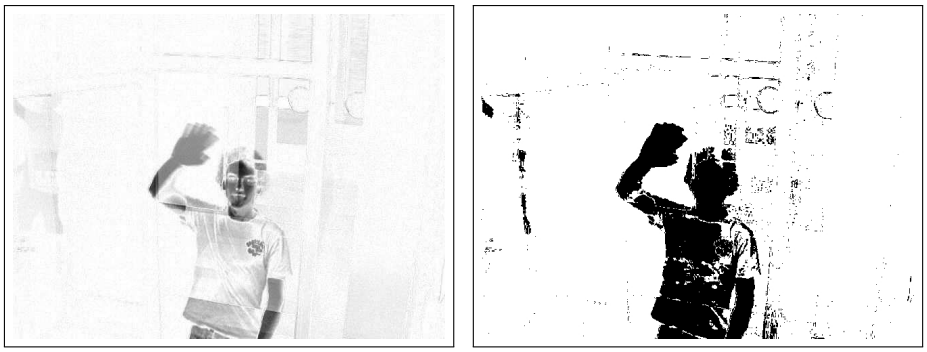


Figure 1.4: Left: difference between input and reference image. Right: simple estimate of foreground, obtained by thresholding the difference image on the left.

video frames, i.e., the foreground, and thus save storage space) and video conferencing (if we only transmit the foreground, this will save bandwidth and thus costs), but I have chosen this example mainly for educational purposes.

As illustrated in figure 1.3, an image is digitally represented as a grid of many *pixels*: small squares that have a uniform intensity. The intensity of a pixel is a number between 0 and 255, where 0 is black and 255 is white, and anything in between is a shade of gray, where a larger intensity value corresponds with a lighter shade of gray. The images used in this example are 640 pixels in width and 480 pixels in height and were made by a surveillance camera. We will denote the intensity of a pixel at some location i of the reference image by R_i and the intensity of a pixel at the same location i of the input image by I_i.

A crude way to separate foreground from background is to consider the differences between the images. More precisely, for each location i, we can calculate the absolute value of the difference in intensity for both pixels corresponding to that location, i.e., $d_i := |I_i - R_i|$.[7] This can be done using some of the more advanced image processing applications, e.g., GIMP or Adobe® PhotoShop®. Figure 1.4

[7] For a real number x, its *absolute value* $|x|$ is x if $x \geq 0$ and $-x$ if $x < 0$.

shows the difference image, where the absolute difference d_i between the intensity values I_i and R_i of the input and reference image is represented by a shade of gray (black corresponding to the maximum possible difference of 255 and white corresponding to the minimum possible difference of 0). We can now choose some threshold value c, and decide that all pixels i for which the absolute difference d_i is larger than c belong to the foreground, and all pixels for which the absolute difference d_i is smaller than c belong to the background. The result is shown in figure 1.4. This is a fast method, but the quality of the result is not satisfying: instead of identifying the whole person as foreground, it only identifies parts of the person as foreground, omitting many little and a few larger regions that (according to the human eye) clearly belong to the foreground. On the other hand, many little parts of the background, where the intensities of the reference and input image differ slightly because of changed lightning conditions, get incorrectly classified as foreground. In order to improve the classification, we would like to somehow impose the criterion that we are only interested in large contiguous foreground objects: we would like to catch a burglar, not a fly.

The key idea is to also take into account neighboring pixels. Every pixel (except those on the border of the image) has four neighboring pixels: one to the left, one to the right, one above and one below. We are going to construct a probability model such that if the absolute difference d_i is large, the probability that the pixel at location i belongs to the foreground should be high. Furthermore, if the majority of the neighboring pixels of the pixel at location i belong to the foreground with high probability, then the probability that the pixel itself belongs to the foreground should also be high. Vice versa, if the neighboring pixels belong to the background with high probability, then the probability that the pixel itself belongs to the background should increase.

For each location i, we introduce a random variable x_i that can have two possible values: either $x_i = -1$, which means "the pixel at location i belongs to the background", or $x_i = +1$, which means "the pixel at location i belongs to the foreground". We are going to construct a probability distribution that takes into account all the $640 \times 480 = 307200$ random variables x_i. We choose this probability distribution in such a way that $\mathbb{P}(x_i = 1)$ is large if $d_i > c$ (in words, the probability that pixel i belongs to the foreground is large if the difference between input and reference image at that location is large) and $\mathbb{P}(x_i = 1)$ is small if $d_i < c$. Note that $\mathbb{P}(x_i = 1) + \mathbb{P}(x_i = -1) = 1$ because the pixel at location i either belongs to the foreground or to the background. Thus, if $\mathbb{P}(x_i = 1)$ is large then $\mathbb{P}(x_i = -1)$ is small and vice versa. Furthermore, the probability $\mathbb{P}(x_i = 1)$ depends on the probabilities $\mathbb{P}(x_j = 1)$ for other pixels j. Indeed, if j is a neighbor of i, then we should have that $\mathbb{P}(x_i = 1 \,|\, x_j = 1)$ is larger than $\mathbb{P}(x_i = 1)$, i.e., the probability that pixel i belongs to the foreground should increase when we learn that its neighbor j belongs to the foreground.

The actual construction of this probability distribution is a bit technical, but in principle we only translate our qualitative description above into a more quantita-

tive and precise mathematical formulation. The complete probability distribution $\mathbb{P}(\{x_i\})$ is a function of all the 307200 random variables x_i (we write $\{x_i\}$ when we refer to the whole collection of random variables x_i for all pixels i).[8] The full probability distribution will be a product of two types of factors: a "local evidence" factor $\psi_i(x_i)$ for each pixel i and a "compatibility" factor $\psi_{ij}(x_i, x_j)$ for each pair $\{i, j\}$ of neighboring pixels i and j. We take the full probability distribution to be the product of all these factors:

$$\mathbb{P}(\{x_i\}) = \frac{1}{Z}\left(\prod_i \psi_i(x_i)\right)\left(\prod_{\{i,j\}} \psi_{ij}(x_i, x_j)\right). \tag{1.3}$$

This probability distribution is an example of a *Markov random field*. We have used a convenient mathematical abbreviation for what would otherwise be an enormous formula: $\prod_i \psi_i(x_i)$ means that we have to take the product of the functions $\psi_i(x_i)$ for each possible pixel location i (in this case, that would be a product of 307200 factors). Similarly, $\prod_{\{i,j\}} \psi_{ij}(x_i, x_j)$ means that we have to take the product of the functions $\psi_{ij}(x_i, x_j)$ for each possible pair of neighboring pixels i and j (which would be a product of 613280 factors). As an example, if we would have images consisting of 3 pixels in one row (with labels i, j, k), writing out equation (1.3) would give:

$$\mathbb{P}(x_i, x_j, x_k) = \frac{1}{Z}\Big(\psi_i(x_i)\psi_j(x_j)\psi_k(x_k)\Big)\Big(\psi_{ij}(x_i, x_j)\psi_{jk}(x_j, x_k)\Big). \tag{1.4}$$

Obviously, I will not write out equation (1.3) for the larger images considered here, because that would just be a waste of paper; instead, please use your imagination. Graphical representations of the Markov random field defined in equation (1.3) are given in figure 1.5 for two cases, the first being the example of 3 pixels on a row, the second a slightly larger image of 5×5 pixels. The nodes in the graphs (represented as circles) correspond to random variables and the edges (represented as lines connecting the circles) correspond to the compatibility functions.

The constant Z is the *normalization constant*, which ensures that the function $\mathbb{P}(\{x_i\})$ is actually a *probability* distribution: if we add the values of $\mathbb{P}(\{x_i\})$ for all possible configurations of $\{x_i\}$, i.e., for all possible background/foreground assignments (which is an enormous number of terms, namely 2^{307200}; imagine how large this number is by comparing the values in table 1.2), we should obtain 1. For the

[8] Compare this with the ASIA network, where the complete probability distribution was a function of only 8 random variables.

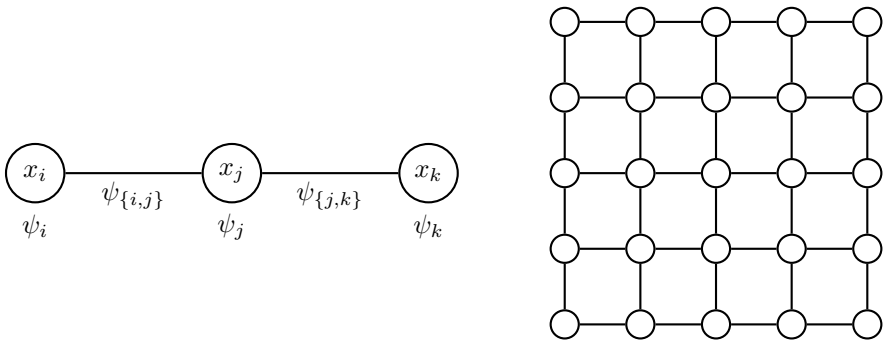


Figure 1.5: Examples of Markov random fields. Left: corresponding with three pixels on a row, labeled i, j and k. Right: corresponding with an image of 5×5 pixels.

simple example of images consisting of only 3 pixels on a row, this means that

$$\begin{aligned}
&\sum_{x_i=\pm 1} \sum_{x_j=\pm 1} \sum_{x_k=\pm 1} \mathbb{P}(x_i, x_j, x_k) = \\
&\quad = \mathbb{P}(x_i = 1, x_j = 1, x_k = 1) + \mathbb{P}(x_i = 1, x_j = 1, x_k = -1) \\
&\qquad + \mathbb{P}(x_i = 1, x_j = -1, x_k = 1) + \mathbb{P}(x_i = 1, x_j = -1, x_k = -1) \\
&\qquad + \mathbb{P}(x_i = -1, x_j = 1, x_k = 1) + \mathbb{P}(x_i = -1, x_j = 1, x_k = -1) \\
&\qquad + \mathbb{P}(x_i = -1, x_j = -1, x_k = 1) + \mathbb{P}(x_i = -1, x_j = -1, x_k = -1) \\
&\quad = 1.
\end{aligned}$$

Using equation (1.4) for each of the eight terms in this sum, we obtain an equation for Z, which can be solved for the value of the normalization constant Z. A similar, but much longer equation, can be used in principle (but not in practice) to calculate the value of Z for the 640×480 images that we are interested in.

We still have to specify which functions $\psi_i(x_i)$ and $\psi_{ij}(x_i, x_j)$ we will use. For the "local evidence" functions, we use

$$\psi_i(x_i) = \exp\left(x_i \, \theta \tanh \frac{d_i - c}{w}\right)$$

This function is shown in figure 1.6 for three different choices of the parameters θ and w. The parameter θ determines the height of the curves, whereas the parameter w determines how steep the curves are. Note that $\psi_i(x_i = 1)$ is large if d_i is larger than the threshold c (i.e., if the difference between input and reference image at location i is larger than the threshold) and small if d_i is smaller than the threshold c; for $\psi_i(x_i = -1)$, the opposite is true. This means that the contribution of the local evidence function to the joint probability is as discussed before: a larger difference between input and reference image at location i increases the probability that $x_i = 1$ (and at the same time, decreases the probability that $x_i = -1$).

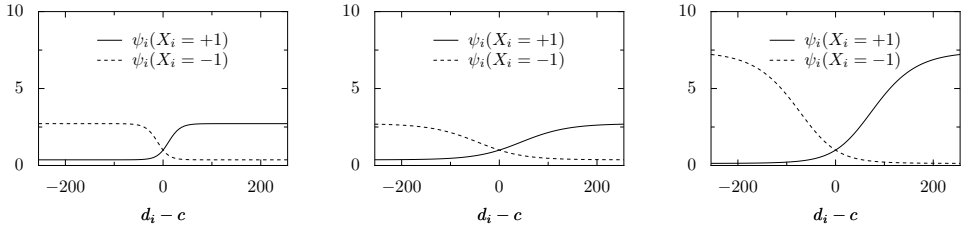


Figure 1.6: Local evidence functions $\psi_i(x_i)$, as a function of $d_i - c$, the difference of d_i with the threshold c. Left: $\theta = 1$, $w = 25$; center: $\theta = 1$, $w = 100$; right: $\theta = 2$, $w = 100$. The parameter θ determines the height of the curves, the parameter w the steepness.

Let i and j be the locations of two neighboring pixels. For the compatibility factor $\psi_{ij}(x_i, x_j)$, we use

$$\psi_{ij}(x_i, x_j) = \exp\Big(Jx_ix_j\Big),$$

where $J > 0$ is a parameter that describes how "compatible" pixels i and j are. The compatibility factor is large if $x_i = x_j$ and small if $x_i \neq x_j$. The larger J, the larger the difference between those two values. In other words, the parameter J determines how much the neighboring locations i and j influence each other regarding their values of x_i and x_j. If J is large, then it will be highly unlikely that x_i differs from x_j, which will result in a larger probability for configurations having large contiguous pieces of foreground and background.

For the readers with less mathematical background, the discussion above may be difficult to understand. The important point is that we have constructed a probability distribution that describes the probability that each pixel is either foreground or background (based on the input and reference image and a few parameters that can be adjusted to obtain optimal results), which satisfies our two desiderata: (i) the larger the difference between input and reference image at some location, the larger the probability that the input image at that location is actually part of the foreground; (ii) neighboring pixels influence each other, i.e., if the neighborhood of some pixel is with high probability part of the foreground, then the probability that the pixel itself is part of the foreground should increase.

In theory, we could now, for each pixel i, calculate the probability $\mathbb{P}(x_i = 1)$ that the pixel is part of the foreground, according to the probability distribution in (1.3), by multiplying the 307200 local evidence factors and the 613280 compatibility factors together, and summing over all the configurations of the random variables that we are *not* interested in (i.e., all configurations of the x_j for $j \neq i$). However, this sum consists of $2^{307200-1}$ terms and it is a completely hopeless task to calculate the value of this sum. We have again encountered the computational complexity explosion that occurs in probability distributions that depend on a large number of random variables.[9] Thus, while we have solved the problem in theory by con-

[9] Actually, by repeatedly applying the *distributive law* (which says that $a(b+c) = ab+ac$), one

Figure 1.7: Top left: result of applying the approximation method Belief Propagation to the probability distribution in equation (1.3), using $J = 3$, $\theta = 3.5$, $w = 40$ and $c = 20$. Top right: the corresponding filtered input image, where the background has been removed. Bottom left: simple threshold image for comparison. Bottom right: result of a different approximation method (Mean Field), which is not as good as the Belief Propagation result.

structing a probability distribution that can be used to calculate for each pixel the probability that it is either foreground or background, it is not possible to do these calculations exactly within a reasonable amount of time (using hardware currently available).

However, we can calculate *approximations* to the probabilities $\mathbb{P}(x_i = 1)$ using approximation techniques that have been developed for this purpose. These approximation techniques will not yield the exact probabilities according to the probability distribution we constructed, but can give reasonable approximations within a reasonable amount of time. We have applied one such approximation technique, called *Belief Propagation*, and shown the result in figure 1.7. Although it is an approximation, it is clearly a much better approximation to the truth than the one we obtained by the fast local thresholding technique discussed earlier. Note that our

can greatly reduce the computational complexity in this case, reducing it to a sum of "only" 2^{960} terms. Although this is an enormous reduction, it is still impossible to calculate that sum within a reasonable amount of time with current technology.

probability distribution has correctly filled in the missing spots in the body, apart from one hole in the hair. It has also correctly removed the noise in the background, apart from two remaining regions (which are actually shades and reflections caused by the human body). In the incorrectly classified hair region, the reference image and the input image turn out to be almost indistinguishable. A human that does the foreground classification task will probably decide that this region also belongs to the foreground, based on his knowledge about how haircuts are supposed to look. However, our probability distribution does not know anything about haircuts; it has made the decision purely by looking at the difference of the intensities of the input and reference images in that region, and thus we can understand why it makes an error in that region.

1.1.4 Summary

Graphical models are used and studied in various applied statistical and computational fields, e.g., machine learning and artificial intelligence, computational biology, statistical signal/image processing, communication and information theory, and statistical physics. We have seen two examples (one from medical diagnosis, the other from image processing) in which graphical models can be used to model real world problems. A fundamental technical problem that we encountered is the explosion of the computational complexity when the number of random variables increases. We have seen that in some cases, the exact calculation of probabilities of interest is still possible, whereas in other cases, it is completely hopeless. In the latter cases, one can instead use approximation methods to calculate approximations to the probabilities of interest. If the approximations are accurate enough and can be obtained within a reasonable amount of time, this is a viable alternative to the exact calculation.

Over the last century, researchers have developed many different approximation methods, each with their own characteristics of accuracy and computation time. One of the most elementary yet successful approximation methods (not in the least place because it is a very fast method), is Belief Propagation. This is the approximation method that we have applied to solve our image processing problem in the last example. Belief Propagation is the object of further study in the rest of this thesis. Although it has been applied in many different situations, sometimes with spectacular success, the theoretical understanding of its accuracy and the computation time it needs was not very well developed when I started working on this research topic four years ago. It was not fully understood in what circumstances Belief Propagation would actually yield an approximation, how much computation time would be needed to calculate the approximation, and how accurate the approximation would be. The results of my research, reported in the next chapters of this thesis, can be very briefly summarized as contributing better answers to these questions, a deeper understanding of the Belief Propagation method, as well as a way to improve the accuracy of the Belief Propagation approximation.

1.2 A less gentle introduction to Belief Propagation

We continue this introduction by giving more formal definitions of the concepts introduced in the previous section. This requires a stronger mathematical background of the reader.

A *(probabilistic) graphical model* is a convenient representation in terms of a graph of the dependency relations between random variables. The qualitative information provided by the graph, together with quantitative information about these dependencies, forms a modular specification of the joint distribution of the random variables. From a slightly different point of view, the graph represents a factorization of the joint distribution in terms of factors that depend only on local subsets of random variables. The structure of this factorization can be exploited to improve the efficiency of calculations of expectation values and marginals of the joint distribution or as a basis for calculating approximations of quantities of interest.

The class of graphical models can be subdivided into *directed* and *undirected* graphical models. Directed graphical models are also known as *Bayesian networks*, *belief networks*, *causal networks* or *influence diagrams*. We have seen an example of a Bayesian network in subsection 1.1.1, the ASIA network. The subclass of undirected graphical models can be subdivided again into *Markov random fields* (also called *Markov networks*) and *factor graphs*. We have seen an example of a Markov random field (the probability distribution corresponding to the foreground classification task) in subsection 1.1.3.

We will repeatedly use the following notational conventions. Let $N \in \mathbb{N}^*$ and $\mathcal{V} := \{1, 2, \ldots, N\}$. Let $(x_i)_{i\in\mathcal{V}}$ be a family of N discrete random variables, where each variable x_i takes values in a discrete domain $\mathcal{X}_i$. In this thesis, we focus on the case of discrete variables for simplicity; it may be possible to generalize our results towards the case of continuous random variables. We will frequently use the following multi-index notation: let $A = \{i_1, i_2, \ldots, i_m\} \subseteq \mathcal{V}$ with $i_1 < i_2 < \cdots < i_m$; we write $\mathcal{X}_A := \mathcal{X}_{i_1} \times \mathcal{X}_{i_2} \times \cdots \times \mathcal{X}_{i_m}$ and for any family[10] $(Y_i)_{i\in B}$ with $A \subseteq B \subseteq \mathcal{V}$, we write $Y_A := (Y_{i_1}, Y_{i_2}, \ldots, Y_{i_m})$. For example, $x_{\{5,3\}} = (x_3, x_5) \in \mathcal{X}_{\{5,3\}} = \mathcal{X}_3 \times \mathcal{X}_5$.

1.2.1 Bayesian networks

A *directed graph* $\mathcal{G} = (\mathcal{V}, \mathcal{D})$ is a pair of vertices (nodes) $\mathcal{V}$ and directed edges $\mathcal{D} \subseteq \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$. A *directed path* in $\mathcal{G}$ is a sequence of nodes $(i_t)_{t=1}^T$ such that $(i_t, i_{t+1}) \in \mathcal{D}$ for each $t = 1, \ldots, T-1$; if $i_1 = i_T$ then the directed path is called a *directed cycle*. A *directed acyclic graph* $\mathcal{G} = (\mathcal{V}, \mathcal{D})$ is a directed graph with no directed cycles, i.e., there is no (nonempty) directed path in $\mathcal{G}$ that starts and

[10] Note the difference between a *family* and a *set*: a family $(Y_i)_{i\in B}$ is a mapping from some set B to another set which contains $\{Y_i : i \in B\}$. We use families as the generalization to arbitrary index sets of (ordered) n-tuples, and sets if the ordering or the number of occurrences of each element is unimportant.

ends at the same vertex. For $i \in \mathcal{V}$, we define the set $\mathrm{par}(i)$ of *parent nodes* of i to be the set of nodes that point directly towards i, i.e., $\mathrm{par}(i) := \{j \in \mathcal{V} : (j,i) \in \mathcal{D}\}$.

A *Bayesian network* consists of a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{D})$ and a family $(P_i)_{i\in\mathcal{V}}$ of (conditional) probability distributions, one for each vertex in $\mathcal{V}$. Each vertex $i \in \mathcal{V}$ represents a random variable x_i which takes values in a finite set $\mathcal{X}_i$. If $\mathrm{par}(i) = \emptyset$, then P_i is a probability distribution for x_i; otherwise, P_i is a conditional probability distribution for x_i given $x_{\mathrm{par}(i)}$. The joint probability distribution represented by the Bayesian network is the product of all the probability distributions $(P_i)_{i\in\mathcal{V}}$:

$$\mathbb{P}(x_{\mathcal{V}}) = \prod_{i\in\mathcal{V}} P_i(x_i \,|\, x_{\mathrm{par}(i)}). \tag{1.5}$$

where $P_i(x_i \,|\, x_{\mathrm{par}(i)}) = P_i(x_i)$ if $\mathrm{par}(i) = \emptyset$.

Causal networks

A Bayesian network describes conditional independencies of a set of random variables, not necessarily their causal relations. However, causal relations can be modeled by the closely related *causal Bayesian network*. The additional semantics of the causal Bayesian network specify that if a random variable x_i is actively caused to be in a state ξ (an operation written as "$\mathrm{do}(x_i = \xi)$"), then the probability distribution changes to the one of the Bayesian network obtained by cutting the edges from $\mathrm{par}(i)$ to i, and setting x_i to the caused value ξ [Pearl, 2000], i.e., by defining $P_i(x_i) = \delta_\xi(x_i)$. Note that this is very different from *observing* that x_i is in some state ξ; the latter is modeled by conditioning on $x_i = \xi$, i.e., by calculating

$$\mathbb{P}(x_{\mathcal{V}} \,|\, x_i = \xi) = \frac{\mathbb{P}(x_{\mathcal{V}}, x_i = \xi)}{\mathbb{P}(x_i = \xi)}.$$

1.2.2 Markov random fields

An *undirected graph* is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of a set of nodes $\mathcal{V}$ and a set of undirected edges $\mathcal{E} \subseteq \{\{i,j\} : i,j \in \mathcal{V}, i \neq j\}$. A *clique* of $\mathcal{G}$ is a subset $C \subseteq \mathcal{V}$ that is fully connected in $\mathcal{G}$, i.e., $\{i,j\} \in \mathcal{E}$ for all $i,j \in C$ with $i \neq j$. A clique is *maximal* if it is not a strict subset of another clique.

A *Markov random field* (or *Markov network*) consists of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a family of *potential functions* (also called *factors* or *clique potentials*) $(\psi_I)_{I\in\mathcal{C}}$, where $\mathcal{C}$ is the set of all maximal cliques of the graph $\mathcal{G}$. Each vertex $i \in \mathcal{V}$ represents a random variable x_i which takes values in the finite set $\mathcal{X}_i$ and each edge $\{i,j\} \in \mathcal{E}$ represents an "interaction" between the random variables x_i and x_j. The potential functions are nonnegative functions $\psi_I : \mathcal{X}_I \to [0,\infty)$ that depend on the random variables in the clique $I \in \mathcal{C}$. The joint probability distribution represented by the Markov random field is given by:

$$\mathbb{P}(x_{\mathcal{V}}) = \frac{1}{Z} \prod_{I\in\mathcal{C}} \psi_I(x_I)$$

where x_I is the state of the random variables in the clique I, and the normalizing constant Z (also called *partition function*) is defined as

$$Z = \sum_{x_{\mathcal{V}} \in \mathcal{X}_{\mathcal{V}}} \prod_{I \in \mathcal{C}} \psi_I(x_I).$$

An example of a Markov random field that is studied in statistical physics is the Ising model.

1.2.3 Factor graphs

Both Bayesian networks and Markov random fields can be represented in a unifying representation, called *factor graph* [Kschischang *et al.*, 2001]. Factor graphs explicitly express the factorization structure of the corresponding probability distribution.

We consider a probability distribution over $x_{\mathcal{V}} \in \mathcal{X}_{\mathcal{V}}$ that can be written as a product of *factors* $(\psi_I)_{I \in \mathcal{F}}$:

$$\mathbb{P}(x_{\mathcal{V}}) = \frac{1}{Z} \prod_{I \in \mathcal{F}} \psi_I(x_{N_I}), \qquad Z = \sum_{x \in \mathcal{X}_{\mathcal{V}}} \prod_{I \in \mathcal{F}} \psi_I(x_{N_I}). \tag{1.6}$$

For each factor index $I \in \mathcal{F}$, there is an associated subset $N_I \subseteq \mathcal{V}$ of variable indices and the factor ψ_I is a nonnegative function $\psi_I : \mathcal{X}_{N_I} \to [0, \infty)$. For a Bayesian network, each factor corresponds to a (conditional) probability table, whereas for a Markov random field, each factor corresponds to a maximal clique of the undirected graph.

We can represent the structure of the probability distribution (1.6) using a *factor graph* $(\mathcal{V}, \mathcal{F}, \mathcal{E})$. This is a bipartite graph, consisting of *variable nodes* $i \in \mathcal{V}$, *factor nodes* $I \in \mathcal{F}$, and an undirected edge $\{i, I\}$ between $i \in \mathcal{V}$ and $I \in \mathcal{F}$ if and only if $i \in N_I$, i.e., if ψ_I depends on x_i. We will represent factor nodes visually as rectangles and variable nodes as circles. Examples of factor graphs, corresponding to the Asia network in figure 1.1 and the Markov random fields in figure 1.5 are shown in figure 1.8.

It is trivial to represent a Bayesian network or a Markov random field as a factor graph, and also trivial to represent a factor graph as a Markov random field, but it is less trivial to represent a factor graph as a Bayesian network. In this thesis, we will regard Bayesian networks and Markov random fields as special cases of factor graphs.

The neighbors of a factor node $I \in \mathcal{F}$ are precisely the variables N_I, and the neighbors N_i of a variable node $i \in \mathcal{V}$ are the factors that depend on that variable, i.e., $N_i := \{I \in \mathcal{F} : i \in N_I\}$. Further, we define for each variable $i \in \mathcal{V}$ the set $\Delta i := \bigcup_{I \in N_i} N_I$ consisting of all variables that appear in some factor in which variable i participates, and the set $\partial i := \Delta i \setminus \{i\}$, the *Markov blanket* of i. For $J \subseteq \mathcal{V} \setminus \Delta i$, x_i is conditionally independent of x_J given the Markov blanket $x_{\partial i}$ of i:

$$\mathbb{P}(x_i \,|\, x_J, x_{\partial i}) = \mathbb{P}(x_i \,|\, x_{\partial i}) \qquad \text{for } J \subseteq \mathcal{V} \setminus \Delta i.$$

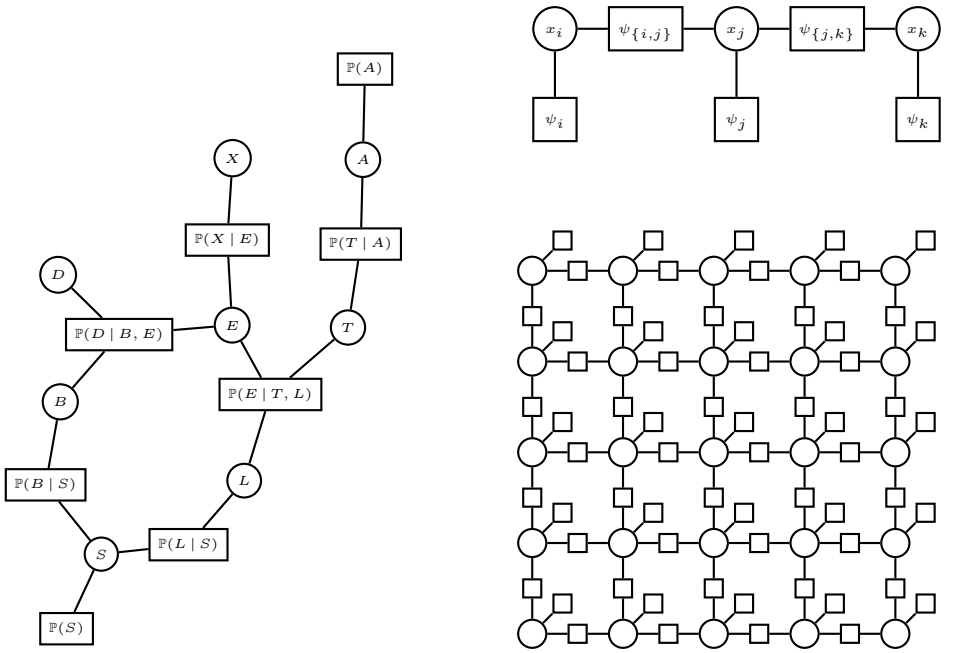


Figure 1.8: Examples of factor graphs, corresponding with the Bayesian network in figure 1.1 and the two Markov random fields in figure 1.5.

We will generally use uppercase letters for indices of factors ($I, J, K, \ldots \in \mathcal{F}$) and lowercase letters for indices of variables ($i, j, k, \ldots \in \mathcal{V}$).

1.2.4 Inference in graphical models

In this thesis, we will define *inference* in a graphical model to be the task of calculating marginal probabilities of subsets of random variables, possibly conditional on observed values of another subset of random variables. This can be done exactly (*exact inference*) or in an approximate way (*approximate inference*). Another inference task that is often considered is to calculate the MAP state, i.e., the joint assignment of a subset of random variables which has the largest probability (possibly conditional on observed values of another subset of random variables). We focus on the first inference problem, i.e., the approximation of marginal probabilities.

In general, the normalizing constant Z in (1.6) (also called *partition function*) is not known and exact computation of Z is infeasible, due to the fact that the number of terms to be summed is exponential in N. Similarly, computing marginal distributions $\mathbb{P}(x_J)$ of $\mathbb{P}(x_\mathcal{V})$ for subsets of variables $J \subseteq \mathcal{V}$ is known to be NP-hard [Cooper, 1990]. Furthermore, approximate inference within given error bounds is NP-hard [Dagum and Luby, 1993; Roth, 1996]. Because of the many applications

in which inference plays a role, the development and understanding of approximate inference methods is thus an important topic of research.

1.2.5 Belief Propagation: an approximate inference method

Belief Propagation (BP) is a popular algorithm for approximate inference that has been reinvented many times in different fields. It is also known as *Loopy Belief Propagation* (where the adjective "loopy" is used to emphasize that it is used on a graphical model with cycles), the *Sum-Product Algorithm* and the *Bethe-Peierls approximation*. In artificial intelligence, it is commonly attributed to Pearl [1988]. In the context of error-correcting (LDPC) codes, it was already proposed by Gallager [1963]. The earliest description of Belief Propagation in the statistical physics literature known to the author is [Nakanishi, 1981] (for the special case of a binary, pairwise Markov random field). A few years ago it became clear [Yedidia *et al.*, 2001] that the BP algorithm is strongly related to the Bethe-Peierls approximation, which was invented originally in statistical mechanics [Bethe, 1935; Peierls, 1936]; this discovery led to a renewed interest in Belief Propagation and related inference methods. We will henceforth use the acronym "BP" since it can be interpreted as being an abbreviation for either "Belief Propagation" or "Bethe-Peierls approximation".

BP calculates approximations to the factor marginals $(\mathbb{P}(x_I))_{I\in\mathcal{F}}$ and the variable marginals $(\mathbb{P}(x_i))_{i\in\mathcal{V}}$ of the probability distribution (1.6) of a factor graph [Kschischang *et al.*, 2001; Yedidia *et al.*, 2005]. The calculation is done by message-passing on the factor graph. Each node passes messages to its neighbors: variable nodes pass messages to factor nodes and factor nodes pass messages to variable nodes. The outgoing messages are functions of the incoming messages at each node. This iterative process is repeated using some schedule that describes the sequence of message updates in time. This process can either converge to some fixed point or go on *ad infinitum*. If BP converges, the approximate marginals (called *beliefs*) can be calculated from the fixed point messages.

For the factor graph formulation of BP (see also figure 1.9), it is convenient to discriminate between two types of messages: messages $\mu_{I\to i} : \mathcal{X}_i \to [0,\infty)$ sent from factors $I \in \mathcal{F}$ to neighboring variables $i \in N_I$ and messages $\mu_{i\to I} : \mathcal{X}_i \to [0,\infty)$ from variables $i \in \mathcal{V}$ to neighboring factors $I \in N_i$. The messages that are sent by a node depend on the incoming messages; the new messages, designated by μ', are given in terms of the incoming messages by the following *BP update equations*:

$$\mu'_{j\to I}(x_j) \propto \prod_{J\in N_j\setminus I} \mu_{J\to j}(x_j) \qquad \forall j \in \mathcal{V},\ \forall I \in N_j, \tag{1.7}$$

$$\mu'_{I\to i}(x_i) \propto \sum_{x_{N_I\setminus i}} \psi_I(x_{N_I}) \prod_{j\in N_I\setminus i} \mu_{j\to I}(x_j) \qquad \forall I \in \mathcal{F},\ \forall i \in N_I. \tag{1.8}$$

Usually, one normalizes the messages such that $\sum_{x_i\in\mathcal{X}_i} \mu(x_i) = 1$. This is only done

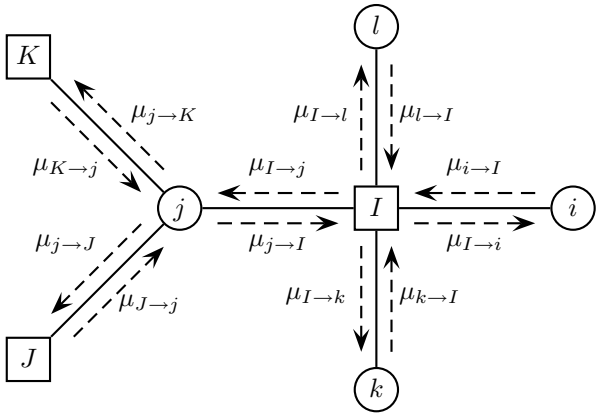


Figure 1.9: Part of the factor graph illustrating the BP update rules (1.7) and (1.8). The factor nodes $I, J, K \in \mathcal{F}$ are drawn as rectangles, the variable nodes $i, j, k, l \in \mathcal{V}$ as circles. Note that $N_j \setminus I = \{J, K\}$ and $N_I \setminus i = \{j, k, l\}$.

for numerical stability reasons; in fact, the final results are invariant under rescaling each message with an arbitrary positive scale factor.

Alternatively, we can use the following update equations, which are formulated in terms of only the messages sent from factors to variables:

$$\mu'_{I\to i}(x_i) \propto \sum_{x_{N_I\setminus i}} \psi_I(x_{N_I}) \prod_{j\in N_I\setminus i} \prod_{J\in N_j\setminus I} \mu_{J\to j}(x_j) \qquad \forall I \in \mathcal{F},\, \forall i \in N_I. \tag{1.9}$$

This equation is obtained by substituting (1.7) into (1.8).

The initial messages at $t = 0$ are often taken to be uniform, i.e.,

$$\mu^{(0)}_{I\to i}(x_i) \propto 1 \qquad \forall I \in \mathcal{F},\ \forall i \in N_I.$$

There is some freedom of choice in the ordering of the message updates. For $I \in \mathcal{F}$ and $i \in N_I$, we also write $I \to i$ for the directed edge (I, i). Let $\mathcal{D} := \{(I, i) : I \in \mathcal{F}, i \in N_I\}$. With an *update schedule* we mean a particular ordering of the BP update equations in time. Formally, an update schedule is a pair (e, τ), where e is a (possibly random) sequence of edges $(e_t)_{t\in\mathbb{N}}$ with $e_t \in \mathcal{D}$ for all $t \in \mathbb{N}$ and τ is a sequence of functions $(\tau_t)_{t\in\mathbb{N}}$ where each τ_t is a function $\mathcal{D} \to \{0, 1, 2, \dots, t\}$. The BP algorithm according to the update schedule (e, τ) is specified by the following update of the messages $\mu^{(t)}$ at time t to the new messages $\mu^{(t+1)}$ at time t:

$$\mu^{(t+1)}_e = \begin{cases} \mu'_e\big((\mu_d^{(\tau_t(d))})_{d\in\mathcal{D}}\big) & \text{if } e = e_t \\ \mu^{(t)}_e & \text{if } e \neq e_t \end{cases}$$

i.e., only the message on edge e_t is updated using the update rule (1.9), using as input the messages at previous times $\tau_t(d)$. Some update schedules that are often used are:

- *Parallel* updates: calculate all new messages as a function of the current messages and then simultaneously set all messages to their new values.
- *Sequential updates in fixed order*: determine some fixed linear ordering of $\mathcal{D}$ and update in that order, using the most recent messages available for each update.
- *Sequential updates in random order*: before doing a batch of $\#(\mathcal{D})$ message updates, construct a new random linear ordering of $\mathcal{D}$ and update the next batch in that order, using the most recent messages available for each individual update.
- *Random updates*: at each timestep t, draw a random element from $\mathcal{D}$ and update the corresponding message, using the most recent messages available.
- *Maximum residual updating* [Elidan *et al.*, 2006]: calculate all residuals, i.e., the differences between the updated and current messages, and update only the message with the largest residual (according to some measure). Then, only the residuals that depend on the updated message need to be recalculated and the process repeats.

If the messages converge to some fixed point $\mu^{(\infty)}$, the approximate marginals, often called *beliefs*, are calculated as follows:

$$\mathbb{P}(x_i) \approx b_i(x_i) \propto \prod_{I \in N_i} \mu_{I \to i}^{(\infty)}(x_i) \qquad \forall i \in \mathcal{V},$$
$$\mathbb{P}(x_I) \approx b_I(x_I) \propto \psi_I(x_{N_I}) \prod_{i \in I} \mu_{i \to I}^{(\infty)}(x_i) \qquad \forall I \in \mathcal{F}.$$

The beliefs are normalized such that

$$\sum_{x_i \in \mathcal{X}_i} b_i(x_i) = 1 \quad \forall i \in \mathcal{V}, \qquad \sum_{x_I \in \mathcal{X}_I} b_I(x_I) = 1 \quad \forall I \in \mathcal{F}.$$

If the factor graph is acyclic, i.e., it contains no loops, then BP with any reasonable update schedule will converge towards a unique fixed point within a finite number of iterations and the beliefs can be shown to be exact. However, if the factor graph contains cycles, then the BP marginals are only approximations to the exact marginals; in some cases, these approximations can be surprisingly accurate, in other cases, they can be highly inaccurate.

A fixed point $\mu^{(\infty)}$ always exists if all factors are strictly positive [Yedidia *et al.*, 2005]. However, the existence of a fixed point does not necessarily imply convergence towards the fixed point. Indeed, fixed points may be unstable, and there may be multiple fixed points (corresponding to different final beliefs).

If BP does not converge, one can try to *damp* the message updates as a possible remedy. The new message is then a convex combination of the old and the updated message, either according to:

$$\mu_d^{(t+1)} = \epsilon \mu_d^{(t)} + (1 - \epsilon) \mu'_d \qquad d \in \mathcal{D},$$

or in the logarithmic domain:

$$\log \mu_d^{(t+1)} = \epsilon \log \mu_d^{(t)} + (1-\epsilon) \log \mu'_d \qquad d \in \mathcal{D},$$

for some $\epsilon \in [0, 1)$.

Other methods that can be applied if BP does not converge (or converges too slowly) are *double-loop methods* [Yuille, 2002; Heskes *et al.*, 2003; Heskes, 2006]. Yedidia *et al.* [2001] showed that fixed points of BP correspond to stationary points of the Bethe free energy. The double-loop methods exploit this correspondence by directly minimizing the Bethe free energy. The corresponding non-convex constrained minimization problem can be solved by performing a sequence of convex constrained minimizations of upper bounds on the Bethe free energy. In this way, the method is guaranteed to converge to a minimum of the Bethe free energy, which corresponds with a BP fixed point.

1.2.6 Related approximate inference algorithms

BP can be regarded as the most elementary one in a family of related algorithms, consisting of

- the Max-Product algorithm [Weiss and Freeman, 2001], a zero temperature version of BP;
- Generalized Belief Propagation (GBP) [Yedidia *et al.*, 2005] and the Cluster Variation Method (CVM) [Pelizzola, 2005], where variables are clustered in "regions" or "clusters" in order to increase accuracy;
- Double-loop algorithms [Yuille, 2002; Heskes *et al.*, 2003; Heskes, 2006], where the inner loop is equivalent to GBP;
- Expectation Propagation (EP) [Minka, 2001; Welling *et al.*, 2005] and the Expectation Consistent (EC) approximation [Opper and Winter, 2005], which can be regarded as generalizations of BP [Heskes *et al.*, 2005];
- Survey Propagation (SP) [Braunstein and Zecchina, 2004; Braunstein *et al.*, 2005], which turned out to be equivalent to a special case of the BP algorithm;
- Fractional Belief Propagation [Wiegerinck and Heskes, 2003].

A good theoretical understanding of BP may therefore be beneficial to understanding these other algorithms as well. In this thesis, we focus on BP because of its simplicity and its successes in solving nontrivial problems.

1.2.7 Applications of Belief Propagation

We have given two examples of situations where approximate inference can be used to solve real world problems. In recent years, mainly due to increased computational

power, the number of applications of approximate inference methods has seen an enormous growth. To convey some sense of the diversity of these applications, we provide a few references to a small subset of these applications (found by searching on the internet for scientific articles reporting the application of Belief Propagation).

Many applications can be found in vision and image processing. Indeed, BP has been applied to stereo vision [Felzenszwalb and Huttenlocher, 2004, 2006; Sun *et al.*, 2003; Tappen and Freeman, 2003], super-resolution [Freeman *et al.*, 2000, 2002; Gupta *et al.*, 2005], shape matching [Coughlan and Ferreira, 2002; Coughlan and Shen, 2004], image reconstruction [Felzenszwalb and Huttenlocher, 2006; Tanaka, 2002], inference of human upper body motion [Gao and Shi, 2004], panaroma generation [Brunton and Shu, 2006], surface reconstruction [Petrovic *et al.*, 2001], skin detection [Zheng *et al.*, 2004], hand tracking [Sudderth *et al.*, 2004], inferring facial components [Sudderth *et al.*, 2003], and unwrapping phase images [Frey *et al.*, 2002]. Very successful applications can also be found in error correcting codes, e.g., Turbo Codes [McEliece *et al.*, 1998] and Low Density Parity Check codes [Gallager, 1963; Frey and MacKay, 1997]. In combinatorial optimization, in particular satisfiability and graph coloring, an algorithm called Survey Propagation recently redefined the state-of-the-art [Mézard and Zecchina, 2002; Braunstein *et al.*, 2005]. Later it was discovered that it was actually equivalent to a special case of BP [Braunstein and Zecchina, 2004]. BP has been applied for diagnosis, for example in medical diagnosis [Murphy *et al.*, 1999; Wemmenhove *et al.*, 2007]. In computer science, it was suggested as a natural algorithm in sensor networks [Ihler *et al.*, 2005c; Crick and Pfeffer, 2003], for data cleaning [Chu *et al.*, 2005] and for content distribution in peer-to-peer networks [Bickson *et al.*, 2006]. In biology, it has been used to predict protein folding [Kamisetty *et al.*, 2006]. Finally, conform the latest fashions, BP has even been used for solving Sudokus [Dangauthier, 2006].

1.3 Outline of this thesis

In this section, we briefly motivate the research questions addressed in this thesis and summarize the results obtained in later chapters.

In practice, there are at least three important issues when applying BP to concrete problems: (i) it is usually not known *a priori* whether BP will converge and how many iterations are needed; (ii) if BP converges, it is not known how large the error of the resulting approximate marginals is; (iii) if the error is too large for the application, can the error be reduced in some way?

The issues of convergence and accuracy may actually be interrelated: the "folklore" is that convergence difficulties of BP often indicate low quality of the corresponding Bethe approximation. This would imply that the pragmatic solution for the convergence problem (forcing BP to converge by means of damping, the use of other update schemes or applying double-loop methods) would yield low quality results. Furthermore, if we could quantify the relation between error and convergence

rate, this might yield a practical way of estimating the error from the observed rate of convergence.

For the case of a graphical model with a single loop, these questions have been solved by Weiss [2000]. However, it has turned out to be difficult to generalize that work to factor graphs with more than one loop. Significant progress has been made in recent years regarding the question under what conditions BP converges [Tatikonda and Jordan, 2002; Tatikonda, 2003; Ihler *et al.*, 2005b,a], on the uniqueness of fixed points [Heskes, 2004], and on the accuracy of the marginals [Tatikonda, 2003; Taga and Mase, 2006a], but the theoretical understanding was (and is) still incomplete. Further, even though many methods have been proposed in recent years to reduce the error of BP marginals, these methods are all in some sense "local" (although more global than BP). We felt that it should be possible to take into account longer loops in the factor graph (which may be important when the interactions along those loops are strong), instead of only taking into account short loops (as usually done with GBP).

These questions have been the motivation for the research reported in the next chapters. We finish this introductory chapter with a short summary of all the following chapters.

Convergence of BP

In chapter 2, we study the question of convergence and uniqueness of the fixed point for parallel, undamped BP. We derive novel conditions that guarantee convergence of BP to a unique fixed point, irrespective of the initial messages. The conditions are applicable to arbitrary factor graphs with discrete variables and factors that contain zeros. For the special case of binary variables with pairwise interactions, we derive stronger results that take into account single-variable factors and the type of pairwise interactions (attractive, mixed or repulsive). We show empirically that these bounds improve upon existing bounds.

Phase transitions and BP

While we focussed on undamped parallel BP in chapter 2, in the next chapter, we investigate the influence of damping and the use of alternative update schemes. We focus on the special case of binary variables with pairwise interactions and zero local fields in the interest of simplicity. Whereas in the previous chapter we studied the global ("uniform") convergence properties of BP, in chapter 3 we analyze the *local* stability of the "high-temperature" fixed point of BP.[11] Further, we investigate the relationship between the properties of this fixed point and the properties of the corresponding stationary point of the Bethe free energy.

[11] If the interactions are weak enough, BP has a unique fixed point. In statistical physics, weak interactions correspond to high temperatures. Therefore, we call this particular fixed point the *high-temperature* BP fixed point.

We distinguish three cases for the interactions: ferromagnetic (attractive), antiferromagnetic (repulsive) and spin-glass (mixed). We prove that the convergence conditions for undamped, parallel BP derived in chapter 2 are sharp in the ferromagnetic case. Also, the use of damping would only slow down convergence to the high-temperature fixed point. In contrast, in the antiferromagnetic case, the use of damping or sequential updates significantly improves the range of instances on which BP converges. In the spin-glass case, we observe that damping only slightly improves convergence of BP.

Further, we show how one can estimate analytically the temperature (interaction strength) at which the high-temperature BP fixed point becomes unstable for random graphs with arbitrary degree distributions and random interactions, extending the worst-case results with some average-case results. The results we obtain are in agreement with the results of the replica method from statistical physics. This provides a link between statistical physics and the properties of the BP algorithm. In particular, it leads to the conclusion that the behavior of BP is closely related to the phase transitions in the underlying graphical model.

Reducing the BP error

In the fourth chapter, we show how the accuracy of BP can be improved by taking into account the influence of loops in the graphical model. Extending a method proposed by Montanari and Rizzo [2005], we propose a novel way of generalizing the BP update equations by dropping the basic BP assumption of independence of incoming messages. We call this method the Loop Correction (LC) method.

The basic idea behind the Loop Correction method is the following. A *cavity distribution* of some variable in a graphical model is the probability distribution on its Markov blanket for a modified graphical model, in which all factors involving that variable have been removed, thereby breaking all the loops involving that variable. The Loop Correction method consists of two steps: first, the cavity distributions of all variables are estimated (using some approximate inference method), and second, these initial estimates are improved by a message-passing algorithm, which reduces the errors in the estimated cavity distributions.

If the initial cavity approximations are taken to be uniform (or completely factorized) distributions, the Loop Correction algorithm reduces to the BP algorithm. In that sense, it can be considered to be a generalization of BP. On the other hand, if the initial cavity approximations contain the effective interactions between variables in the cavity, application of the Loop Correction method usually gives significantly better results than the original (uncorrected) approximate inference algorithm used to estimate the cavity approximations. Indeed, we often observe that the loop-corrected error is approximately the square of the error of the uncorrected approximate inference method.

We report the results of an extensive experimental comparison of various approximate inference methods on a variety of graphical models, including real world

networks. We conclude that the LC method obtains the most accurate results in general, at the cost of significantly increased computation time compared to BP.

Bounds on marginal probabilities

In the final chapter, we further develop some of the ideas that arose out of the convergence analysis in chapter 2 and the cavity interpretation in chapter 4. From chapter 2, we take the idea of studying how the distance between two different message vectors (for the same factor graph) evolves during BP iterations. From chapter 4, we take the cavity interpretation that relates the exact marginals to the BP marginals. The key insight exploited in chapter 5 is that by combining and extending these ideas, it is possible to derive rigorous bounds on the exact single-variable marginals. By construction, the same bounds also apply to the BP beliefs. We also derive a related method that propagates bounds over a "self-avoiding-walk tree", inspired by recent results of Ihler [2007]. We show empirically that our bounds often outperform existing bounds in terms of accuracy and/or computation time. We apply the bounds on factor graphs arising in a medical diagnosis application and show that the bounds can yield nontrivial results.

Chapter 2

Sufficient conditions for convergence of BP

We derive novel conditions that guarantee convergence of Belief Propagation (BP) to a unique fixed point, irrespective of the initial messages, for parallel (synchronous) updates. The computational complexity of the conditions is polynomial in the number of variables. In contrast with previously existing conditions, our results are directly applicable to arbitrary factor graphs (with discrete variables) and are shown to be valid also in the case of factors containing zeros, under some additional conditions. We compare our bounds with existing ones, numerically and, if possible, analytically. For binary variables with pairwise interactions, we derive sufficient conditions that take into account local evidence (i.e., single-variable factors) and the type of pairwise interactions (attractive or repulsive). It is shown empirically that this bound outperforms existing bounds.

2.1 Introduction

Belief Propagation [Pearl, 1988; Kschischang *et al.*, 2001], also known as "Loopy Belief Propagation" and as the "Sum-Product Algorithm", which we will henceforth abbreviate as BP, is a popular algorithm for approximate inference in graphical models. Applications can be found in diverse areas such as error correcting codes (iterative channel decoding algorithms for Turbo Codes and Low Density Parity Check Codes [McEliece *et al.*, 1998]), combinatorial optimization (satisfiability problems such as 3-SAT and graph coloring [Braunstein and Zecchina, 2004]) and computer vision (stereo matching [Sun *et al.*, 2003] and image restoration [Tanaka, 2002]). BP

can be regarded as the most elementary one in a family of related algorithms, consisting of double-loop algorithms [Heskes *et al.*, 2003], GBP [Yedidia *et al.*, 2005], EP [Minka, 2001], EC [Opper and Winter, 2005], the Max-Product Algorithm [Weiss and Freeman, 2001], the Survey Propagation Algorithm [Braunstein and Zecchina, 2004; Braunstein *et al.*, 2005] and Fractional BP [Wiegerinck and Heskes, 2003]. A good understanding of BP may therefore be beneficial to understanding these other algorithms as well.

In practice, there are two major obstacles in the application of BP to concrete problems: (i) if BP converges, it is not clear whether the results are a good approximation of the exact marginals; (ii) BP does not always converge, and in these cases gives no approximations at all. These two issues might actually be interrelated: the "folklore" is that failure of BP to converge often indicates low quality of the Bethe approximation on which it is based. This would mean that if one has to "force" BP to converge (e.g., by using damping or double-loop approaches), one may expect the results to be of low quality.

Although BP is an old algorithm that has been reinvented in many fields, a thorough theoretical understanding of the two aforementioned issues and their relation is still lacking. Significant progress has been made in recent years regarding the question under what conditions BP converges [Tatikonda and Jordan, 2002; Tatikonda, 2003; Ihler *et al.*, 2005b][1], on the uniqueness of fixed points [Heskes, 2004], and on the accuracy of the marginals [Tatikonda, 2003], but the theoretical understanding is still incomplete. For the special case of a graphical model consisting of a single loop, it has been shown that convergence rate and accuracy are indeed related [Weiss, 2000].

In this work, we study the question of convergence of BP and derive new sufficient conditions for BP to converge to a unique fixed point. Our results are more general and in certain cases stronger than previously known sufficient conditions.

2.2 Background

To introduce our notation, we give a short treatment of factorizing probability distributions, the corresponding visualizations called factor graphs, and the BP algorithm on factor graphs. For an excellent, more extensive treatment of these topics we refer the reader to [Kschischang *et al.*, 2001].

2.2.1 Factor graphs

Consider N random variables x_i for $i \in \mathcal{V} := \{1, 2, \ldots, N\}$, with x_i taking values in a finite set $\mathcal{X}_i$. We will frequently use the following multi-index notation. Let

[1] After initial submission of this work, we came to the attention of [Ihler *et al.*, 2005a], which contains improved versions of results in [Ihler *et al.*, 2005b], some of which are similar or identical to results presented here (see also section 2.5.2).

$A = \{i_1, i_2, \ldots, i_m\} \subseteq \mathcal{V}$ with $i_1 < i_2 < \ldots i_m$. We write $\mathcal{X}_A := \mathcal{X}_{i_1} \times \mathcal{X}_{i_2} \times \cdots \times \mathcal{X}_{i_m}$ and for any family $(Y_i)_{i \in B}$ with $A \subseteq B \subseteq \mathcal{V}$, we write $Y_A := (Y_{i_1}, Y_{i_2}, \ldots, Y_{i_m})$.

We are interested in the class of probability measures on $\mathcal{X}_\mathcal{V}$ that can be written as a product of *factors* (also called *potentials* or *interactions*):

$$\mathbb{P}(x_1, \ldots, x_N) := \frac{1}{Z} \prod_{I \in \mathcal{F}} \psi_I(x_{N_I}). \tag{2.1}$$

For each factor index $I \in \mathcal{F}$, there is an associated subset $N_I \subseteq \mathcal{V}$ of variable indices and the factor ψ_I is a positive function[2] $\psi_I : \mathcal{X}_{N_I} \to (0, \infty)$. Z is a normalizing constant ensuring that $\sum_{x_\mathcal{V} \in \mathcal{X}_\mathcal{V}} \mathbb{P}(x_\mathcal{V}) = 1$. The class of probability measures described by (2.1) contains Markov random fields as well as Bayesian networks. We will use uppercase letters for indices of factors ($I, J, K, \ldots \in \mathcal{F}$) and lowercase letters for indices of variables ($i, j, k, \ldots \in \mathcal{V}$).

The *factor graph* that corresponds to the probability distribution (2.1) is a bipartite graph with *variable nodes* $i \in \mathcal{V}$, *factor nodes* $I \in \mathcal{F}$ and edges between variable nodes and factor nodes; there is an edge between variable node i and factor node I if and only if the factor ψ_I depends on the variable x_i, i.e., if $i \in N_I$. The neighbors of a factor node $I \in \mathcal{F}$ are precisely the variables N_I, and the neighbors N_i of a variable node $i \in \mathcal{V}$ are the factors that depend on that variable, i.e., $N_i := \{I \in \mathcal{F} : i \in N_I\}$. For each variable node $i \in \mathcal{V}$, we define the set of its neighboring variable nodes by $\partial i := \left(\bigcup_{I \in N_i} N_I\right) \setminus \{i\}$, i.e., ∂i is the set of indices of those variables that interact directly with x_i.

2.2.2 Belief Propagation

Belief Propagation is an algorithm that calculates approximations to the marginals $\big(\mathbb{P}(x_{N_I})\big)_{I \in \mathcal{F}}$ and $\big(\mathbb{P}(x_i)\big)_{i \in \mathcal{V}}$ of the probability measure (2.1). The calculation is done by message-passing on the factor graph: each node passes messages to its neighbors (see also figure 2.1). One usually discriminates between two types of messages: messages $\mu_{I \to i}(x_i)$ from factors to variables and messages $\mu_{i \to I}(x_i)$ from variables to factors (where $i \in \mathcal{V}, I \in N_i$). Both messages are positive functions on $\mathcal{X}_i$, or, equivalently, vectors in $\mathbb{R}^{\mathcal{X}_i}$ (with positive components). The messages that are sent by a node depend on the incoming messages; the new messages, designated by μ', are given in terms of the incoming messages by the following *BP update rules*:[3]

$$\mu'_{j \to I}(x_j) \propto \prod_{J \in N_j \setminus I} \mu_{J \to j}(x_j) \qquad \forall j \in \mathcal{V},\ \forall I \in N_j; \tag{2.2}$$

$$\mu'_{I \to i}(x_i) \propto \sum_{x_{N_I \setminus i}} \psi_I(x_{N_I}) \prod_{j \in N_I \setminus i} \mu_{j \to I}(x_j) \qquad \forall I \in \mathcal{F},\ \forall i \in N_I. \tag{2.3}$$

[2] In subsection 2.4.5 we will loosen this assumption and allow for factors containing zeros.

[3] We abuse notation slightly by writing $X \setminus x$ instead of $X \setminus \{x\}$ for sets X.

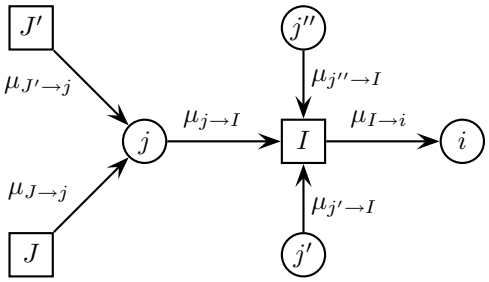


Figure 2.1: Part of the factor graph illustrating the BP update rules (2.2) and (2.3). The factor nodes $I, J, J' \in \mathcal{F}$ are drawn as rectangles, the variable nodes $i, j, j', j'' \in \mathcal{V}$ as circles. Note that $N_j \setminus I = \{J, J'\}$ and $N_I \setminus i = \{j, j', j''\}$. Apart from the messages that have been drawn, each edge also carries a message flowing in the opposite direction.

Usually, one normalizes the messages in the ℓ_1-sense, i.e., such that

$$\sum_{x_i \in \mathcal{X}_i} \mu_{i\to I}(x_i) = 1, \quad \sum_{x_i \in \mathcal{X}_i} \mu_{I\to i}(x_i) = 1 \qquad \forall i \in \mathcal{V},\ \forall I \in N_i.$$

If all messages have converged to some fixed point $\mu^{(\infty)}$, one calculates the approximate marginals, called *beliefs*, as follows:

$$b_i(x_i) \propto \prod_{I \in N_i} \mu^{(\infty)}_{I\to i}(x_i) \approx \mathbb{P}(x_i) \qquad \forall i \in \mathcal{V},$$

$$b_I(x_{N_I}) \propto \psi_I(x_{N_I}) \prod_{i \in N_I} \mu^{(\infty)}_{i\to I}(x_i) \approx \mathbb{P}(x_{N_I}) \qquad \forall I \in \mathcal{F},$$

where the normalization is by definition in ℓ_1-sense. A fixed point always exists if all factors are strictly positive [Yedidia *et al.*, 2005]. However, the existence of a fixed point does not necessarily imply convergence towards the fixed point, and fixed points may be unstable.

Note that the beliefs are invariant under rescaling of the messages

$$\mu^{(\infty)}_{I\to i}(x_i) \mapsto \alpha_{I\to i}\mu^{(\infty)}_{I\to i}(x_i), \qquad \mu^{(\infty)}_{i\to I}(x_i) \mapsto \alpha_{i\to I}\mu^{(\infty)}_{i\to I}(x_i)$$

for arbitrary positive constants α, which shows that the precise way of normalizing the messages in (2.2) and (2.3) is irrelevant. For numerical stability however, some way of normalization (not necessarily in ℓ_1-sense) is desired to ensure that the messages stay in some compact domain.

In the following, we will formulate everything in terms of the messages $\mu_{I\to i}(x_i)$ from factors to variables; the update equations are then obtained by substituting (2.2) in (2.3):

$$\mu'_{I\to i}(x_i) = C_{I\to i} \sum_{x_{N_I \setminus i}} \psi_I(x_{N_I}) \prod_{j \in N_I \setminus i} \prod_{J \in N_j \setminus I} \mu_{J\to j}(x_j). \tag{2.4}$$

with $C_{I\to i}$ such that $\sum_{x_i \in \mathcal{X}_i} \mu'_{I\to i}(x_i) = 1$. We consider here BP with a *parallel* update scheme, which means that all message updates (2.4) are done in parallel.

2.3 Special case: binary variables with pairwise interactions

In this section we investigate the simple case of binary variables (i.e., $\#(\mathcal{X}_i) = 2$ for all $i \in \mathcal{V}$), and in addition we assume that all potentials consist of at most two variables ("pairwise interactions"). Although this is a special case of the more general theory to be presented later on, we start with this simple case because it illustrates most of the underlying ideas without getting involved with the additional technicalities of the general case.

We will assume that all variables are ± 1-valued, i.e., $\mathcal{X}_i = \{-1, +1\}$ for all $i \in \mathcal{V}$. We take the factor index set as $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$ with $\mathcal{F}_1 = \mathcal{V}$ (the "local evidence") and $\mathcal{F}_2 \subseteq \big\{\{i,j\} : i,j \in \mathcal{V}, i \neq j\big\}$ (the "pair potentials"). The probability measure (2.1) can then be written as

$$\mathbb{P}(x_{\mathcal{V}}) = \frac{1}{Z} \exp\left(\sum_{\{i,j\} \in \mathcal{F}_2} J_{ij} x_i x_j + \sum_{i \in \mathcal{F}_1} \theta_i x_i \right) \tag{2.5}$$

for some choice of the parameters J_{ij} ("couplings") and θ_i ("local fields"), with $\psi_i(x_i) = \exp(\theta_i x_i)$ for $i \in \mathcal{F}_1$ and $\psi_{ij}(x_i, x_j) = \exp(J_{ij} x_i x_j)$ for $\{i,j\} \in \mathcal{F}_2$.

Note from (2.4) that the messages sent from single-variable factors $\mathcal{F}_1$ to variables are constant. Thus the question whether messages converge can be decided by studying only the messages sent from pair potentials $\mathcal{F}_2$ to variables. It turns out to be advantageous to use the following "natural" parameterization of the messages (often used in statistical physics):

$$\tanh \nu_{i \to j} := \mu_{\{i,j\} \to j}(x_j = 1) - \mu_{\{i,j\} \to j}(x_j = -1), \tag{2.6}$$

where $\nu_{i \to j} \in \mathbb{R}$ is now interpreted as a message sent from variable i to variable j (instead of a message sent from the factor $\{i,j\}$ to variable j). Note that in the pairwise case, the product over $j \in N_I \setminus i$ in (2.4) becomes trivial. Some elementary algebraic manipulations show that the BP update equations (2.4) become particularly simple in this parameterization: they can be written as

$$\tanh(\nu'_{i \to j}) = \tanh(J_{ij}) \tanh\left(\theta_i + \sum_{t \in \partial i \setminus j} \nu_{t \to i} \right), \tag{2.7}$$

where $\partial i = \{t \in \mathcal{V} : \{i,t\} \in \mathcal{F}_2\}$ are the variables that interact with i via a pair potential.

Defining the set of ordered pairs $\mathcal{D} := \{i \to j : \{i,j\} \in \mathcal{F}_2\}$, we see that the parallel BP update is a mapping $f : \mathbb{R}^{\mathcal{D}} \to \mathbb{R}^{\mathcal{D}}$; (2.7) specifies the component $\big(f(\nu)\big)_{i \to j} := \nu'_{i \to j}$ in terms of the components of ν. Our goal is now to derive sufficient conditions under which the mapping f is a contraction. For this we need some elementary but powerful mathematical theorems.

2.3.1 Normed spaces, contractions and bounds

In this subsection we introduce some (standard) notation and remind the reader of some elementary but important properties of vector norms, matrix norms, contractions and the Mean Value Theorem in arbitrary normed vector spaces, which are the main mathematical ingredients for our basic tool, Lemma 2.3. The reader familiar with these topics can skip this subsection and proceed directly to Lemma 2.3 in section 2.3.2.

Let $(V, \|\cdot\|)$ be a normed finite-dimensional real vector space. Examples of norms that will be important later on are the ℓ_1-norm on $\mathbb{R}^N$, defined by

$$\|x\|_1 := \sum_{i=1}^{N} |x_i|$$

and the ℓ_∞-norm on $\mathbb{R}^N$, defined by

$$\|x\|_\infty := \max_{i\in\{1,\dots,N\}} |x_i| .$$

A norm on a vector space V induces a metric on V by the definition $d(v,w) := \|v-w\|$. The resulting metric space is complete.[4]

Let (X,d) be a metric space. A mapping $f : X \to X$ is called a *contraction with respect to* d if there exists $0 \le K < 1$ such that

$$d(f(x), f(y)) \le K d(x,y) \qquad \text{for all } x, y \in X. \tag{2.8}$$

In case d is induced by a norm $\|\cdot\|$, we will call a contraction with respect to d a $\|\cdot\|$-contraction. If (X,d) is complete, we can apply the following theorem, due to Banach:

Theorem 2.1 (Contracting Mapping Principle) *Let $f : X \to X$ be a contraction of a complete metric space (X,d). Then f has a unique fixed point $x_\infty \in X$ and for any $x \in X$, the sequence $x, f(x), f^2(x), \dots$ obtained by iterating f converges to x_∞. The rate of convergence is at least linear, since $d(f(x), x_\infty) \le K d(x, x_\infty)$ for all $x \in X$.*

Proof. Can be found in many textbooks on analysis. □

Note that linear convergence means that the error decreases exponentially, indeed $d(f^n(x), x_\infty) \le CK^n$ for some C.

Let $(V, \|\cdot\|)$ be a normed space. The norm induces a *matrix norm* (also called *operator norm*) on linear mappings $A : V \to V$, defined as follows:

$$\|A\| := \sup_{\substack{v\in V,\\ \|v\|\le 1}} \|Av\| .$$

[4] Completeness is a topological property which we will not further discuss, but we need this to apply Theorem 2.1.

The ℓ_1-norm on $\mathbb{R}^N$ induces the following matrix norm:

$$\|A\|_1 = \max_{j\in\{1,\dots,N\}} \sum_{i=1}^{N} |A_{ij}| \tag{2.9}$$

where $A_{ij} := (Ae_j)_i$ with e_j the j^{th} canonical basis vector. The ℓ_∞-norm on $\mathbb{R}^N$ induces the following matrix norm:

$$\|A\|_\infty = \max_{i\in\{1,\dots,N\}} \sum_{j=1}^{N} |A_{ij}| \,. \tag{2.10}$$

In the following consequence of the well-known Mean Value Theorem, the matrix norm of the derivative ("Jacobian") $f'(v)$ at $v \in V$ of a differentiable mapping $f : V \to V$ is used to bound the distance of the f-images of two vectors:

Lemma 2.2 *Let $(V, \|\cdot\|)$ be a normed space and $f : V \to V$ a differentiable mapping. Then, for $x, y \in V$:*

$$\|f(y) - f(x)\| \le \|y - x\| \cdot \sup_{z\in[x,y]} \|f'(z)\|$$

where we wrote $[x, y]$ for the segment $\{\lambda x + (1-\lambda)y : \lambda \in [0, 1]\}$ joining x and y.

Proof. See [Dieudonné, 1969, Thm. 8.5.4]. □

2.3.2 The basic tool

Combining Theorem 2.1 and Lemma 2.2 immediately yields our basic tool:

Lemma 2.3 *Let $(V, \|\cdot\|)$ be a normed space, $f : V \to V$ differentiable and suppose that*

$$\sup_{v\in V} \|f'(v)\| < 1.$$

Then f is a $\|\cdot\|$-contraction by Lemma 2.2. Hence, for any $v \in V$, the sequence $v, f(v), f^2(v), \dots$ converges to a unique fixed point $v_\infty \in V$ with a convergence rate that is at least linear by Theorem 2.1. □

2.3.3 Sufficient conditions for BP to be a contraction

We apply Lemma 2.3 to the case at hand: the parallel BP update mapping $f : \mathbb{R}^{\mathcal{D}} \to \mathbb{R}^{\mathcal{D}}$, written out in components in (2.7). Different choices of the vector norm on $\mathbb{R}^{\mathcal{D}}$ will yield different sufficient conditions for whether iterating f will converge to a unique fixed point. We will study two examples: the ℓ_1 norm and the ℓ_∞ norm.

The derivative of f is easily calculated from (2.7) and is given by

$$\Big(f'(\nu)\Big)_{i\to j,k\to l} = \frac{\partial \nu'_{i\to j}}{\partial \nu_{k\to l}} = A_{i\to j,k\to l} B_{i\to j}(\nu) \tag{2.11}$$

where[5]

$$B_{i\to j}(\nu) := \frac{1-\tanh^2\left(\theta_i + \sum_{t\in\partial i\setminus j}\nu_{t\to i}\right)}{1-\tanh^2\left(\nu'_{i\to j}(\nu)\right)}\operatorname{sgn} J_{ij} \tag{2.12}$$

$$A_{i\to j,k\to l} := \tanh|J_{ij}|\,\delta_{il}\mathbf{1}_{\partial i\setminus j}(k). \tag{2.13}$$

Note that we have absorbed all ν-dependence in the factor $B_{i\to j}(\nu)$; the reason for this will become apparent later on. The factor $A_{i\to j,k\to l}$ is nonnegative and independent of ν and captures the structure of the graphical model. Note that $\sup_{\nu\in V}|B_{i\to j}(\nu)| = 1$, implying that

$$\left|\frac{\partial\nu'_{i\to j}}{\partial\nu_{k\to l}}\right| \le A_{i\to j,k\to l} \tag{2.14}$$

everywhere on V.

Example: the ℓ_∞-norm

The ℓ_∞-norm on $\mathbb{R}^{\mathcal{D}}$ yields the following condition:

Corollary 2.4 *For binary variables with pairwise interactions and probability distribution (2.5), if*

$$\max_{i\in\mathcal{V}}\left(\left(\#(\partial i)-1\right)\max_{j\in\partial i}\tanh|J_{ij}|\right) < 1, \tag{2.15}$$

BP is an ℓ_∞-contraction and converges to a unique fixed point, irrespective of the initial messages.

Proof. Using (2.10), (2.13) and (2.14):

$$\begin{aligned}\|f'(\nu)\|_\infty &= \max_{i\to j}\sum_{k\to l}\left|\frac{\partial\nu'_{i\to j}}{\partial\nu_{k\to l}}\right| \le \max_{i\to j}\sum_{k\to l}\tanh|J_{ij}|\,\delta_{il}\mathbf{1}_{\partial i\setminus j}(k)\\ &= \max_{i\in\mathcal{V}}\max_{j\in\partial i}\sum_{k\in\partial i\setminus j}\tanh|J_{ij}| = \max_{i\in\mathcal{V}}\left(\left(\#(\partial i)-1\right)\max_{j\in\partial i}\tanh|J_{ij}|\right)\end{aligned}$$

and now simply apply Lemma 2.3. □

Another example: the ℓ_1-norm

Using the ℓ_1-norm instead, we find:

Corollary 2.5 *For binary variables with pairwise interactions and probability distribution (2.5), if*

$$\max_{i\in\mathcal{V}}\max_{k\in\partial i}\sum_{j\in\partial i\setminus k}\tanh|J_{ij}| < 1, \tag{2.16}$$

BP is an ℓ_1-contraction and converges to a unique fixed point, irrespective of the initial messages.

[5] For a set X, we define the indicator function $\mathbf{1}_X$ of X by $\mathbf{1}_X(x) = 1$ if $x \in X$ and $\mathbf{1}_X(x) = 0$ if $x \notin X$.

Proof. Similar to the proof of Corollary 2.4, now using (2.9) instead of (2.10):

$$\|f'(\nu)\|_1 \leq \max_{k\to l} \sum_{i\to j} \tanh|J_{ij}|\,\delta_{il}\mathbf{1}_{\partial i\setminus j}(k) = \max_{i\in\mathcal{V}} \max_{k\in\partial i} \sum_{j\in\partial i\setminus k} \tanh|J_{ij}|\,.$$

□

It is easy to see that condition (2.16) is implied by (2.15), but not conversely; thus in this case the ℓ_1-norm yields a tighter bound than the ℓ_∞-norm.

2.3.4 Beyond norms: the spectral radius

Instead of pursuing a search for the optimal norm, we will derive a criterion for convergence based on the spectral radius of the matrix (2.13). The key idea is to look at several iterations of BP at once. This will yield a significantly stronger condition for convergence of BP to a unique fixed point.

For a square matrix A, we denote by $\sigma(A)$ its *spectrum*, i.e., the set of eigenvalues of A. By $\rho(A)$ we denote its *spectral radius*, which is defined as $\rho(A) := \sup|\sigma(A)|$, i.e., the largest modulus of eigenvalues of A.[6]

Lemma 2.6 *Let $f : X \to X$ be a mapping, d a metric on X and suppose that f^N is a d-contraction for some $N \in \mathbb{N}$. Then f has a unique fixed point x_∞ and for any $x \in X$, the sequence $x, f(x), f^2(x), \ldots$ obtained by iterating f converges to x_∞.*

Proof. Take any $x \in X$. Consider the N sequences obtained by iterating f^N, starting respectively in x, $f(x)$, …, $f^{N-1}(x)$:

$$\begin{aligned} &x, f^N(x), f^{2N}(x), \ldots \\ &f(x), f^{N+1}(x), f^{2N+1}(x), \ldots \\ &\vdots \\ &f^{N-1}(x), f^{2N-1}(x), f^{3N-1}(x), \ldots \end{aligned}$$

Each sequence converges to x_∞ since f^N is a d-contraction with fixed point x_∞. But then the sequence $x, f(x), f^2(x), \ldots$ must converge to x_∞. □

Theorem 2.7 *Let $f : \mathbb{R}^m \to \mathbb{R}^m$ be differentiable and suppose that $f'(x) = B(x)A$, where A has nonnegative entries and B is diagonal with bounded entries $|B_{ii}(x)| \leq 1$. If $\rho(A) < 1$ then for any $x \in \mathbb{R}^m$, the sequence x, $f(x)$, $f^2(x), \ldots$ obtained by iterating f converges to a fixed point x_∞, which does not depend on x.*

Proof. For a matrix B, we will denote by $|B|$ the matrix with entries $|B|_{ij} = |B_{ij}|$. For two matrices B, C we will write $B \leq C$ if $B_{ij} \leq C_{ij}$ for all entries (i, j). Note that if $|B| \leq |C|$, then $\|B\|_1 \leq \|C\|_1$. Also note that $|BC| \leq |B|\,|C|$. Finally, if $0 \leq A$ and $B \leq C$, then $AB \leq AC$ and $BA \leq CA$.

[6] One should not confuse the spectral *radius* $\rho(A)$ with the spectral *norm* $\|A\|_2 = \sqrt{\rho(A^TA)}$ of A, the matrix norm induced by the ℓ_2-norm.

Using these observations and the chain rule, we have for any $n \in \mathbb{N}^*$ and any $x \in \mathbb{R}^m$:

$$|(f^n)'(x)| = \left| \prod_{i=1}^{n} f'\big(f^{i-1}(x)\big) \right| \leq \prod_{i=1}^{n} \Big(\big|B\big(f^{i-1}(x)\big)\big|\, A \Big) \leq A^n,$$

hence $\|(f^n)'(x)\|_1 \leq \|A^n\|_1$.

By the Gelfand spectral radius theorem,

$$\lim_{n\to\infty} \big(\|A^n\|_1 \big)^{1/n} = \rho(A).$$

Choose $\epsilon > 0$ such that $\rho(A) + \epsilon < 1$. For some N, $\big\|A^N\big\|_1 \leq (\rho(A)+\epsilon)^N < 1$. Hence for all $x \in \mathbb{R}^m$, $\big\|(f^N)'(x)\big\|_1 < 1$. Applying Lemma 2.3, we conclude that f^N is a ℓ_1-contraction. Now apply Lemma 2.6. □

Using (2.11), (2.12) and (2.13), this immediately yields:

Corollary 2.8 *For binary variables with pairwise interactions and probability distribution (2.5), BP converges to a unique fixed point (irrespective of the initial messages), if the spectral radius of the $\#(\mathcal{D}) \times \#(\mathcal{D})$ matrix*

$$A_{i\to j,k\to l} := \tanh|J_{ij}|\, \delta_{il} \mathbf{1}_{\partial i \setminus j}(k)$$

is strictly smaller than 1. □

The calculation of the spectral norm of the (sparse) matrix A can be done using standard numerical techniques in linear algebra.

Any matrix norm of A is actually an upper bound on the spectral radius $\rho(A)$, since for any eigenvalue λ of A with eigenvector v we have $|\lambda|\,\|v\| = \|\lambda v\| = \|Av\| \leq \|A\|\,\|v\|$, hence $\rho(A) \leq \|A\|$. This implies that no norm in Lemma 2.3 will result in a sharper condition than Corollary 2.8, hence the title of this section.

Further, for a given matrix A and some $\epsilon > 0$, there exists a vector norm $\|\cdot\|$ such that the induced matrix norm of A satisfies $\rho(A) \leq \|A\| \leq \rho(A) + \epsilon$; see [Deutsch, 1975] for a constructive proof. Thus *for given* A one can approximate $\rho(A)$ arbitrarily close by induced matrix norms. This immediately gives a result on the convergence rate of BP (in case $\rho(A) < 1$): for any $\epsilon > 0$, there exists a norm-induced metric such that the linear rate of contraction of BP with respect to that metric is bounded from above by $\rho(A) + \epsilon$.

One might think that there is a shorter proof of Corollary 2.8: it seems quite plausible intuitively that in general, for a continuously differentiable $f : \mathbb{R}^m \to \mathbb{R}^m$, iterating f will converge to a unique fixed point if

$$\sup_{x\in\mathbb{R}^m} \rho(f'(x)) < 1.$$

However, this conjecture (which has been open for a long time) has been shown to be true in two dimensions but false in higher dimensions [Cima *et al.*, 1997].

2.3.5 Improved bound for strong local evidence

Empirically, it is known that the presence of strong local fields (i.e., single-variable factors which are far from uniform) often improves the convergence of BP. However, our results so far are completely independent of the parameters $(\theta_i)_{i\in\mathcal{V}}$ that measure the strength of the local evidence. By proceeding more carefully than we have done above, the results can easily be improved in such a way that local evidence is taken into account.

Consider the quantity $B_{i\to j}$ defined in (2.12). We have bounded this quantity by noting that $\sup_{\nu\in V}|B_{i\to j}(\nu)| = 1$. Note that for all BP updates (except for the first one), the argument ν (the incoming messages) is in $f(V)$, which can be considerably smaller than the complete vector space V. Thus, after the first BP update, we can use

$$\begin{aligned}\sup_{\nu\in f(V)}|B_{i\to j}(\nu)| &= \sup_{\nu\in f(V)}\frac{1-\tanh^2\left(\theta_i+\sum_{k\in\partial i\setminus j}\nu_{k\to i}\right)}{1-\tanh^2\left(\nu'_{i\to j}(\nu)\right)}\\ &= \sup_{\nu\in f(V)}\frac{1-\tanh^2(\eta_{i\setminus j})}{1-\tanh^2(J_{ij})\tanh^2(\eta_{i\setminus j})}\end{aligned}$$

where we used (2.7) and defined the *cavity field*

$$\eta_{i\setminus j}(\nu) := \theta_i + \sum_{k\in\partial i\setminus j}\nu_{k\to i}. \tag{2.17}$$

The function

$$x \mapsto \frac{1-\tanh^2 x}{1-\tanh^2 J_{ij}\cdot\tanh^2 x}$$

is strictly decreasing for $x \geq 0$ and symmetric around $x = 0$, thus, defining

$$\eta^{(*)}_{i\setminus j} := \inf_{\nu\in f(V)}\left|\eta_{i\setminus j}(\nu)\right|, \tag{2.18}$$

we obtain

$$\sup_{\nu\in f(V)}|B_{i\to j}(\nu)| = \frac{1-\tanh^2\left(\eta^{(*)}_{i\setminus j}\right)}{1-\tanh^2(J_{ij})\tanh^2\left(\eta^{(*)}_{i\setminus j}\right)}.$$

Now, from (2.7) we derive that

$$\{\nu_{k\to i} : \nu\in f(V)\} = (-|J_{ki}|, |J_{ki}|),$$

hence

$$\{\eta_{i\setminus j}(\nu) : \nu\in f(V)\} = \left(\eta^{(-)}_{i\setminus j}, \eta^{(+)}_{i\setminus j}\right)$$

where we defined

$$\eta^{(\pm)}_{i\setminus j} := \theta_i \pm \sum_{k\in\partial i\setminus j}|J_{ki}|.$$

We conclude that $\eta^{(*)}_{i\setminus j}$ is simply the distance between 0 and the interval $\left(\eta^{(-)}_{i\setminus j}, \eta^{(+)}_{i\setminus j}\right)$, i.e.,

$$\eta^{(*)}_{i\setminus j} = \begin{cases} \left|\eta^{(+)}_{i\setminus j}\right| & \text{if } \eta^{(+)}_{i\setminus j} < 0 \\ \eta^{(-)}_{i\setminus j} & \text{if } \eta^{(-)}_{i\setminus j} > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Thus the element $A_{i\to j,k\to i}$ (for $i \in \partial j, k \in \partial i \setminus j$) of the matrix A defined in Corollary 2.8 can be replaced by

$$\tanh|J_{ij}| \frac{1-\tanh^2\left(\eta^{(*)}_{i\setminus j}\right)}{1-\tanh^2(J_{ij})\tanh^2\left(\eta^{(*)}_{i\setminus j}\right)} = \frac{\tanh\left(|J_{ij}| - \eta^{(*)}_{i\setminus j}\right) + \tanh\left(|J_{ij}| + \eta^{(*)}_{i\setminus j}\right)}{2}$$

which is generally smaller than $\tanh|J_{ij}|$ and thus gives a tighter bound.

This trick can be repeated arbitrarily often: assume that $m \geq 0$ BP updates have been done already, which means that it suffices to take the supremum of $|B_{i\to j}(\nu)|$ over $\nu \in f^m(V)$. Define for all $i \to j \in \mathcal{D}$ and all $t = 0, 1, \ldots, m$:

$$\underline{\eta}^{(t)}_{i\setminus j} := \inf\{\eta_{i\setminus j}(\nu) : \nu \in f^t(V)\}, \tag{2.19}$$

$$\overline{\eta}^{(t)}_{i\setminus j} := \sup\{\eta_{i\setminus j}(\nu) : \nu \in f^t(V)\}, \tag{2.20}$$

and define the intervals

$$\mathcal{H}^{(t)}_{i\setminus j} := [\underline{\eta}^{(t)}_{i\setminus j}, \overline{\eta}^{(t)}_{i\setminus j}]. \tag{2.21}$$

Specifically, for $t = 0$ we have $\underline{\eta}^{(0)}_{i\setminus j} = -\infty$ and $\overline{\eta}^{(0)}_{i\setminus j} = \infty$, which means that

$$\mathcal{H}^{(0)}_{i\setminus j} = \mathbb{R}. \tag{2.22}$$

Using (2.7) and (2.17), we obtain the following recursive relations for the intervals (where we use interval arithmetic defined in the obvious way):

$$\mathcal{H}^{(t+1)}_{i\setminus j} = \theta_i + \sum_{k\in\partial i\setminus j} \tanh^{-1}\left(\tanh J_{ki} \tanh \mathcal{H}^{(t)}_{k\setminus i}\right). \tag{2.23}$$

Using this recursion relation, one can calculate $\mathcal{H}^{(m)}_{i\setminus j}$ and define $\eta^{(*)}_{i\setminus j}$ as the distance (in absolute value) of the interval $\mathcal{H}^{(m)}_{i\setminus j}$ to 0:

$$\eta^{(*)}_{i\setminus j} = \begin{cases} \left|\overline{\eta}^{(m)}_{i\setminus j}\right| & \text{if } \overline{\eta}^{(m)}_{i\setminus j} < 0 \\ \underline{\eta}^{(m)}_{i\setminus j} & \text{if } \underline{\eta}^{(m)}_{i\setminus j} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.24}$$

Thus by replacing the matrix A in Corollary 2.8 by

$$A_{i\to j,k\to l} = \frac{\tanh\left(|J_{ij}| - \eta^{(*)}_{i\setminus j}\right) + \tanh\left(|J_{ij}| + \eta^{(*)}_{i\setminus j}\right)}{2} \delta_{il} \mathbf{1}_{\partial i\setminus j}(k), \tag{2.25}$$

we obtain stronger results that improve as m increases:

Corollary 2.9 *Let $m \in \mathbb{N}$. For binary variables with pairwise interactions and probability distribution (2.5), BP converges to a unique fixed point (irrespective of the initial messages) if the spectral radius of the $\#(\mathcal{D}) \times \#(\mathcal{D})$ matrix defined in (2.25) (with $\eta^{(*)}_{i\backslash j}$ defined in equations (2.21)–(2.24)) is strictly smaller than 1.* □

2.4 General case

In the general case, when the domains $\mathcal{X}_i$ are arbitrarily large (but finite), we do not know of a natural parameterization of the messages that automatically takes care of the invariance of the messages under scaling (like (2.6) does in the binary case). Instead of handling the scale invariance by the parameterization and using standard norms and metrics, it seems easier to take a simple parameterization and to change the norms and metrics in such a way that they are insensitive to the (irrelevant) extra degrees of freedom arising from the scale invariance. This is actually the key insight in extending the previous results beyond the binary case: once one sees how to do this, the rest follows in a (more or less) straightforward way.

A related important point is to reparameterize the messages: a natural parameterization for our analysis is now in terms of logarithms of messages $\lambda_{I\to i} := \log \mu_{I\to i}$. The BP update equations (2.4) can be written in terms of the log-messages as:

$$\lambda'_{I\to i}(x_i) = \log \sum_{x_{N_I\backslash i}} \psi_I(x_{N_I}) h_{I\backslash i}(x_{N_I\backslash i}), \tag{2.26}$$

where we dropped the normalization and defined the "cavity field"

$$h_{I\backslash i}(x_{N_I\backslash i}) := \exp\left(\sum_{j\in N_I\backslash i} \sum_{J\in N_j\backslash I} \lambda_{J\to j}(x_j) \right). \tag{2.27}$$

Each log-message $\lambda_{I\to i}$ is a vector in the vector space $V_{I\to i} := \mathbb{R}^{\mathcal{X}_i}$; we will use Greek letters as indices for the components, e.g., $\lambda_{I\to i;\alpha} := \lambda_{I\to i}(\alpha)$ with $\alpha \in \mathcal{X}_i$. We will call everything that concerns individual vector spaces $V_{I\to i}$ *local* and define the *global* vector space V as the direct sum of the local vector spaces:

$$V := \bigoplus_{i\in\mathcal{V}, I\in N_i} V_{I\to i}.$$

The parallel BP update is the mapping $f : V \to V$, written out in components in (2.26) and (2.27).

Note that the invariance of the message $\mu_{I\to i}$ under scaling amounts to invariance of the log-message $\lambda_{I\to i}$ under translation. More formally, defining linear subspaces

$$W_{I\to i} := \{\lambda \in V_{I\to i} : \lambda_\alpha = \lambda_{\alpha'} \text{ for all } \alpha, \alpha' \in \mathcal{X}_i\} \tag{2.28}$$

and their direct sum

$$W := \bigoplus_{i\in\mathcal{V}, I\in N_i} W_{I\to i} \subseteq V,$$

the invariance amounts to the observation that

$$f(\lambda + w) - f(\lambda) \in W \qquad \text{for all } \lambda \in V,\, w \in W.$$

Since $\lambda + w$ and λ are equivalent for our purposes, we want our measures of distance in V to reflect this equivalence. Therefore we will "divide out" the equivalence relation and work in the quotient space V/W, which is the topic of the next subsection.

2.4.1 Quotient spaces

Let V be a finite-dimensional vector space. Let W be a linear subspace of V. We can consider the *quotient space* $V/W := \{v + W : v \in V\}$, where $v + W := \{v + w : w \in W\}$. Defining addition and scalar multiplication on the quotient space in the natural way, the quotient space is again a vector space.[7] We will denote its elements as $\overline{v} := v + W$. Note that the projection $\pi : V \to V/W : v \mapsto \overline{v}$ is linear.

Let $\|\cdot\|$ be any vector norm on V. It induces a *quotient norm* on V/W, defined by

$$\|\overline{v}\| := \inf_{w \in W} \|v + w\| , \tag{2.29}$$

which is indeed a norm, as one easily checks. The quotient norm in turn induces the *quotient metric* $d(\overline{v_1}, \overline{v_2}) := \|\overline{v_2} - \overline{v_1}\|$ on V/W. The metric space $(V/W, d)$ is complete (since any finite-dimensional normed vector space is complete).

Let $f : V \to V$ be a (possibly nonlinear) mapping with the following symmetry:

$$f(v + w) - f(v) \in W \qquad \text{for all } v \in V,\, w \in W. \tag{2.30}$$

We can then unambiguously define the quotient mapping

$$\overline{f} : V/W \to V/W : \overline{v} \mapsto \overline{f(v)},$$

which yields the following commutative diagram:

$$\begin{array}{ccc} V & \xrightarrow{\;f\;} & V \\ \downarrow{\scriptstyle \pi} & & \downarrow{\scriptstyle \pi} \\ V/W & \xrightarrow{\;\overline{f}\;} & V/W \end{array} \qquad \pi \circ f = \overline{f} \circ \pi$$

For a linear mapping $A : V \to V$, condition (2.30) amounts to $AW \subseteq W$, i.e., A should leave W invariant; we can then unambiguously define the quotient mapping $\overline{A} : V/W \to V/W : \overline{v} \mapsto \overline{Av}$.

If $f : V \to V$ is differentiable and satisfies (2.30), the symmetry property (2.30) implies that $f'(x)W \subseteq W$, hence we can define $\overline{f'(x)} : V/W \to V/W$. The operation of taking derivatives is compatible with projecting onto the quotient space. Indeed, by using the chain rule and the identity $\pi \circ f = \overline{f} \circ \pi$, one finds that the derivative

[7] Indeed, we have a null vector $0 + W$, addition $(v_1 + W) + (v_2 + W) := (v_1 + v_2) + W$ for $v_1, v_2 \in V$ and scalar multiplication $\lambda(v + W) := (\lambda v) + W$ for $\lambda \in \mathbb{R}, v \in V$.

of the induced mapping $\overline{f} : V/W \to V/W$ at $\overline{x}$ equals the induced derivative of f at x:

$$\overline{f}'(\overline{x}) = \overline{f'(x)} \qquad \text{for all } x \in V. \tag{2.31}$$

By Lemma 2.3, $\overline{f}$ is a contraction with respect to the quotient norm if

$$\sup_{\overline{x} \in V/W} \left\| \overline{f}'(\overline{x}) \right\| < 1.$$

Using (2.29) and (2.31), this condition can be written more explicitly as:

$$\sup_{x \in V} \sup_{\substack{v \in V, \\ \|v\| \leq 1}} \inf_{w \in W} \|f'(x) \cdot v + w\| < 1.$$

2.4.2 Constructing a norm on V

Whereas in the binary case, each message $\nu_{i \to j}$ was parameterized by a single real number, the messages are now $\#(\mathcal{X}_i)$-dimensional vectors $\lambda_{I \to i}$ (with components $\lambda_{I \to i;\alpha}$ indexed by $\alpha \in \mathcal{X}_i$). In extending the ℓ_1-norm that proved to be useful in the binary case to the more general case, we have the freedom to choose the "local" part of the generalized ℓ_1-norm. Here we show how to construct such a generalization of the ℓ_1-norm and its properties; for a more detailed account of the construction, see Appendix 2.A.

The "global" vector space V is the direct sum of the "local" subspaces $V_{I \to i}$. Suppose that for each subspace $V_{I \to i}$, we have a local norm $\|\cdot\|_{I \to i}$. A natural generalization of the ℓ_1-norm in the binary case is the following global norm on V:

$$\|\lambda\| := \sum_{I \to i} \|\lambda_{I \to i}\|_{I \to i} . \tag{2.32}$$

It is easy to check that this is indeed a norm on V.

Each subspace $V_{I \to i}$ has a 1-dimensional subspace $W_{I \to i}$ defined in (2.28) and the local norm on $V_{I \to i}$ induces a local quotient norm on the quotient space $V_{I \to i}/W_{I \to i}$. The global norm (2.32) on V induces a global quotient norm on V/W, which is simply the sum of the local quotient norms (see (2.57)):

$$\left\|\overline{\lambda}\right\| = \sum_{I \to i} \left\|\overline{\lambda_{I \to i}}\right\|_{I \to i} . \tag{2.33}$$

Let $\lambda \in V$. The derivative $f'(\lambda)$ of $f : V \to V$ at λ is a linear mapping $f'(\lambda) : V \to V$ satisfying $f'(\lambda)W \subseteq W$. It projects down to a linear mapping $\overline{f'(\lambda)} : V/W \to V/W$. The matrix norm of $\overline{f'(\lambda)}$ induced by the quotient norm (2.33) is given by (see (2.58)):

$$\left\|\overline{f'(\lambda)}\right\| = \max_{J \to j} \sum_{I \to i} \left\|\overline{\left(f'(\lambda)\right)_{I \to i, J \to j}}\right\|_{I \to i}^{J \to j} \tag{2.34}$$

where the local quotient matrix norm of the "block" $\big(f'(\lambda)\big)_{I\to i,J\to j}$ is given by (see (2.59)):

$$\Big\|\overline{\big(f'(\lambda)\big)_{I\to i,J\to j}}\Big\|_{I\to i}^{J\to j} = \sup_{\substack{v\in V_{J\to j},\\ \|v\|_{J\to j}\leq 1}} \Big\|\overline{\big(f'(\lambda)\big)_{I\to i,J\to j} v}\Big\|_{I\to i}. \tag{2.35}$$

The derivative of the (unnormalized) parallel BP update (2.26) is easily calculated:

$$\frac{\partial\lambda'_{I\to i}(x_i)}{\partial\lambda_{J\to j}(y_j)} = \mathbf{1}_{N_j\setminus I}(J)\mathbf{1}_{N_I\setminus i}(j)\frac{\sum_{x_{N_I\setminus i}}\psi_I(x_i,x_j,x_{N_I\setminus\{i,j\}})\delta_{x_j y_j}h_{I\setminus i}(x_{N_I\setminus i})}{\sum_{x_{N_I\setminus i}}\psi_I(x_i,x_{N_I\setminus i})h_{I\setminus i}(x_{N_I\setminus i})}. \tag{2.36}$$

Taking the global quotient norm (2.34) of (2.36) yields:

$$\Big\|\overline{f'(\lambda)}\Big\| = \max_{J\to j}\sum_{I\to i}\mathbf{1}_{N_j\setminus I}(J)\mathbf{1}_{N_I\setminus i}(j)B_{I\to i,J\to j}\big(h_{I\setminus i}(\lambda)\big), \tag{2.37}$$

where

$$B_{I\to i,J\to j}\big(h_{I\setminus i}(\lambda)\big) := \left\|\overline{\frac{\sum_{x_{N_I\setminus\{i,j\}}}\psi_I\, h_{I\setminus i}(\lambda)}{\sum_{x_{N_I\setminus i}}\psi_I\, h_{I\setminus i}(\lambda)}}\right\|_{I\to i}^{J\to j}. \tag{2.38}$$

Note that $B_{I\to i,J\to j}$ depends on λ via the dependence of $h_{I\setminus i}$ on λ (see (2.27)). We will for the moment simplify matters by assuming that λ can be any vector in V, and later discuss the more careful estimate (where $\lambda\in f^m(V)$):

$$\sup_{\lambda\in V} B_{I\to i,J\to j}\big(h_{I\setminus i}(\lambda)\big) \leq \sup_{h_{I\setminus i}>0} B_{I\to i,J\to j}(h_{I\setminus i}). \tag{2.39}$$

Defining the matrix A by the expression on the r.h.s. and using (2.35) and (2.29), we obtain:

$$A_{I\to i,J\to j} := \sup_{h_{I\setminus i}>0} B_{I\to i,J\to j}(h_{I\setminus i}) = \sup_{h_{I\setminus i}>0}\sup_{\substack{v\in V_{J\to j}\\ \|v\|_{J\to j}\leq 1}}\inf_{w\in W_{I\to i}}\left\|\frac{\sum_{x_j}v(x_j)\sum_{x_{N_I\setminus\{i,j\}}}\psi_I\, h_{I\setminus i}}{\sum_{x_{N_I\setminus i}}\psi_I\, h_{I\setminus i}} - w\right\|_{I\to i} \tag{2.40}$$

for $I\to i$ and $J\to j$ such that $j\in N_I\setminus i$ and $J\in N_j\setminus I$. Surprisingly, it turns out that we can calculate (2.40) analytically if we take all local norms to be ℓ_∞ norms. We have also tried the ℓ_2 norm and the ℓ_1 norm as local norms, but were unable to calculate expression (2.40) analytically in these cases. Numerical calculations turned out to be difficult because of the nested suprema.

2.4.3 Local ℓ_∞ norms

Take for each local norm $\|\cdot\|_{I\to i}$ the ℓ_∞ norm on $V_{I\to i}=\mathbb{R}^{\mathcal{X}_i}$. The local subspace $W_{I\to i}$ is spanned by the vector $\mathbf{1}:=(1,1,\ldots,1)\in\mathbb{R}^{\mathcal{X}_i}$. The local quotient norm of

a vector $v \in V_{I\to i}$ is thus given by

$$\|\overline{v}\|_{I\to i} = \|\overline{v}\|_\infty = \inf_{w\in\mathbb{R}} \|v + w\mathbf{1}\|_\infty = \frac{1}{2} \sup_{\alpha,\alpha'\in\mathcal{X}_i} |v_\alpha - v_{\alpha'}|. \tag{2.41}$$

For a linear mapping $A : V_{J\to j} \to V_{I\to i}$ that satisfies $AW_{J\to j} \subseteq W_{I\to i}$, the induced quotient matrix norm (2.35) is given by

$$\begin{aligned}\left\|\overline{A}\right\|_{I\to i}^{J\to j} &= \sup_{\substack{v\in V_{J\to j},\\ \|v\|_\infty \le 1}} \left\|\overline{Av}\right\|_\infty = \sup_{\substack{v\in V_{J\to j},\\ \|v\|_\infty \le 1}} \frac{1}{2} \sup_{\alpha,\alpha'\in\mathcal{X}_i} \left| \sum_{\beta\in\mathcal{X}_j} (A_{\alpha\beta} - A_{\alpha'\beta})v_\beta \right| \\ &= \frac{1}{2} \sup_{\alpha,\alpha'\in\mathcal{X}_i} \sum_{\beta\in\mathcal{X}_j} |A_{\alpha\beta} - A_{\alpha'\beta}| \end{aligned} \tag{2.42}$$

For the moment, fix $I \to i$ and $J \to j$ (such that $j \in N_I \setminus i$ and $J \in N_j \setminus I$). To lighten the notation, we will drop the corresponding subscripts and in addition, we will replace the arguments by Greek subscripts, where we let α correspond to x_i, β to x_j and γ to $x_{N_I\setminus\{i,j\}}$. For example, we write $h_{I\setminus i}(x_{N_I\setminus i})$ as $h_{\beta\gamma}$ and $\psi_I(x_i, x_j, x_{N_I\setminus\{i,j\}})$ as $\psi_{\alpha\beta\gamma}$. Using (2.42), we can write (2.40) as

$$\sup_{h>0} \frac{1}{2} \sup_{\alpha,\alpha'} \sum_\beta \left| \frac{\sum_\gamma \psi_{\alpha\beta\gamma} h_{\beta\gamma}}{\sum_\beta \sum_\gamma \psi_{\alpha\beta\gamma} h_{\beta\gamma}} - \frac{\sum_\gamma \psi_{\alpha'\beta\gamma} h_{\beta\gamma}}{\sum_\beta \sum_\gamma \psi_{\alpha'\beta\gamma} h_{\beta\gamma}} \right|.$$

Interchanging the two suprema, fixing (for the moment) α and α', defining $\tilde{\psi}_{\beta\gamma} := \psi_{\alpha\beta\gamma}/\psi_{\alpha'\beta\gamma}$ and $\tilde{h}_{\beta\gamma} := h_{\beta\gamma}\psi_{\alpha'\beta\gamma}$, noting that we can assume (without loss of generality) that $\tilde{h}$ is normalized in ℓ_1 sense, the previous expression (apart from the $\frac{1}{2}\sup_{\alpha,\alpha'}$) simplifies to

$$\sup_{\substack{\tilde{h}>0,\\ \|\tilde{h}\|_1=1}} \sum_\beta \left| \sum_\gamma \tilde{h}_{\beta\gamma} \left(\frac{\tilde{\psi}_{\beta\gamma}}{\sum_\beta \sum_\gamma \tilde{\psi}_{\beta\gamma}\tilde{h}_{\beta\gamma}} - 1 \right) \right|. \tag{2.43}$$

In Appendix 2.B we show that (2.43) equals

$$2 \sup_{\beta\neq\beta'} \sup_{\gamma,\gamma'} \tanh\left(\frac{1}{4} \log \frac{\tilde{\psi}_{\beta\gamma}}{\tilde{\psi}_{\beta'\gamma'}} \right). \tag{2.44}$$

We conclude that if we take all local norms to be the ℓ_∞ norms, then $A_{I\to i, J\to j}$ equals

$$N(\psi_I, i, j) := \sup_{\alpha\neq\alpha'} \sup_{\beta\neq\beta'} \sup_{\gamma,\gamma'} \tanh\left(\frac{1}{4} \log \frac{\psi_{I;\alpha\beta\gamma}}{\psi_{I;\alpha'\beta\gamma}} \frac{\psi_{I;\alpha'\beta'\gamma'}}{\psi_{I;\alpha\beta'\gamma'}} \right), \tag{2.45}$$

which is defined for $i, j \in N_I$ with $i \neq j$ and where $\psi_{I;\alpha\beta\gamma}$ is shorthand for $\psi_I(x_i = \alpha, x_j = \beta, x_{N_I\setminus\{i,j\}} = \gamma)$; see figure 2.2 for an illustration.

Now combining (2.37), (2.39) and (2.45), we finally obtain:

$$\left\|\overline{f'(\overline{\lambda})}\right\| = \left\|\overline{f'(\lambda)}\right\| \le \max_{J\to j} \sum_{I\in N_j\setminus J} \sum_{i\in N_I\setminus j} N(\psi_I, i, j).$$

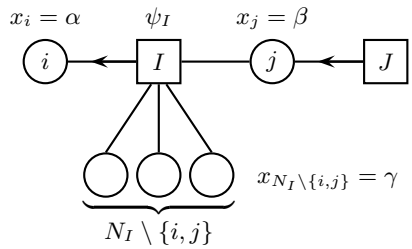


Figure 2.2: Part of the factor graph relevant in expressions (2.45), (2.46) and (2.47). Here $i, j \in N_I$ with $i \neq j$, and $J \in N_j \setminus I$.

Applying Lemma 2.3 now yields that $\overline{f}$ is a contraction with respect to the quotient norm on V/W if the right-hand side is strictly smaller than 1.

Consider the mapping $\eta : V/W \to V$ that maps $\overline{\lambda}$ to the *normalized* $\lambda \in V$, i.e., such that $\|\exp \lambda_{I \to i}\|_1 = 1$ for all components $I \to i$. If we take for f the ℓ_1-normalized BP update (in the log-domain), the following diagram commutes:

$$\begin{array}{ccc} V & \xrightarrow{f} & V \\ \downarrow{\scriptstyle \pi} & & \uparrow{\scriptstyle \eta} \\ V/W & \xrightarrow{\overline{f}} & V/W \end{array} \qquad f = \eta \circ \overline{f} \circ \pi.$$

Since both π and η are continuous, and $f^N = \eta \circ \overline{f}^N \circ \pi$ because $\pi \circ \eta = 1$, we can translate convergence results for $\overline{f}$ back to similar results for f. We have proved:

Theorem 2.10 *If*

$$\max_{J \to j} \sum_{I \in N_j \setminus J} \sum_{i \in N_I \setminus j} N(\psi_I, i, j) < 1, \tag{2.46}$$

BP converges to a unique fixed point irrespective of the initial messages. □

Now we can also generalize Corollary 2.8:

Theorem 2.11 *If the spectral radius of the matrix*

$$A_{I \to i, J \to j} = \mathbf{1}_{N_j \setminus I}(J) \mathbf{1}_{N_I \setminus i}(j) N(\psi_I, i, j), \tag{2.47}$$

is strictly smaller than 1, BP converges to a unique fixed point irrespective of the initial messages.

Proof. Similar to the binary pairwise case; see Theorem 2.19 in Appendix 2.A for details. □

Note that Theorem 2.10 is a trivial consequence of Theorem 2.11, since the ℓ_1-norm is an upper bound on the spectral radius. However, to prove the latter, it seems that we have to go through all the work (and some more) needed to prove the former.

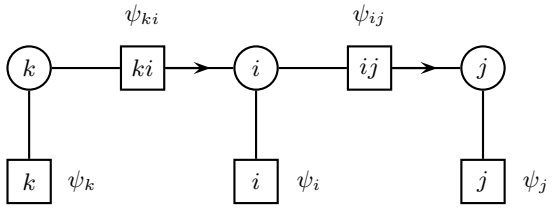


Figure 2.3: Part of the factor graph in the pairwise case relevant in (2.48) and (2.49). Here $k \in \partial i$ and $j \in \partial i \setminus k$.

2.4.4 Special cases

In this subsection we study the implications for two special cases, namely factor graphs that contain no cycles and the case of pairwise interactions.

Trees

Theorem 2.11 gives us a proof of the well-known fact that BP converges on trees (whereas Theorem 2.10 is not strong enough to prove that result):

Corollary 2.12 *If the factor graph is a tree, BP converges to a unique fixed point irrespective of the initial messages.*

Proof. The spectral radius of (2.47) is easily shown to be zero in this special case, for any choice of the potentials. □

Pairwise interactions

We formulate Theorems 2.10 and 2.11 for the special case of pairwise interactions (which corresponds to γ taking on only one value), i.e., each factor consists of either one or two variables. For a pair potential $\psi_{ij} = \psi_{ij;\alpha\beta}$, expression (2.45) simplifies to (see also figure 2.3)

$$N(\psi_{ij}) := \sup_{\alpha\neq\alpha'} \sup_{\beta\neq\beta'} \tanh\left(\frac{1}{4}\left(\log \frac{\psi_{ij;\alpha\beta}}{\psi_{ij;\alpha'\beta}} \frac{\psi_{ij;\alpha'\beta'}}{\psi_{ij;\alpha\beta'}}\right)\right). \tag{2.48}$$

Note that this quantity is invariant to "reallocation" of single-variable factors ψ_i or ψ_j to the pairwise factor ψ_{ij} (i.e., $N(\psi_{ij}) = N(\psi_{ij}\psi_i\psi_j)$). $N(\psi_{ij})$ can be regarded as a measure of the strength of the potential ψ_{ij}.

The ℓ_1-norm condition (2.46) can be written in the pairwise case as:

$$\max_{i\in\mathcal{V}} \max_{k\in\partial i} \sum_{j\in\partial i\setminus k} N(\psi_{ij}) < 1. \tag{2.49}$$

The matrix defined in (2.47), relevant for the spectral radius condition, can be replaced by the following $\#(\mathcal{D}) \times \#(\mathcal{D})$ matrix in the pairwise case:

$$A_{i\to j,k\to l} := N(\psi_{ij})\delta_{il}\mathbf{1}_{\partial i\setminus j}(k). \tag{2.50}$$

For the binary case, we reobtain our earlier results, since

$$N\big(\exp(J_{ij}x_ix_j)\big) = \tanh|J_{ij}|\,.$$

2.4.5 Factors containing zeros

Until now, we have assumed that all factors are strictly positive. In many interesting applications of Belief Propagation, this assumption is violated: the factors may assume the value zero. It is thus interesting to see if and how our results can be extended towards this more general case.

The easiest way to extend the results is by assuming that—although the factors may contain zeros—the messages are guaranteed to remain strictly positive (i.e., the log-messages remain finite) after each update.[8] Even more general extensions with milder conditions may exist, but we believe that considerably more work would be required to overcome the technical problems that arise due to messages containing zeros.

Assume that each factor ψ_I is a nonnegative function $\psi_I : \mathcal{X}_{N_I} \to [0,\infty)$. In addition, assume that all factors involving only a single variable are strictly positive. This can be assumed without loss of generality, since the single-variable factors that contain one or more zeros can simply be absorbed into multi-variable factors involving the same variable. Additionally, for each $I \in \mathcal{F}$ for which N_I contains more than one variable, assume that

$$\forall_{i\in N_I}\,\forall_{x_i\in\mathcal{X}_i}\exists_{x_{N_I\setminus i}\in\mathcal{X}_{N_I\setminus i}} : \psi_I(x_i, x_{N_I\setminus i}) > 0. \tag{2.51}$$

These conditions guarantee that strictly positive messages remain strictly positive under the update equations (2.4), as one easily checks, implying that we can still use the logarithmic parameterization of the messages and that the derivative (2.36) is still well-defined.

The expression for the potential strength (2.45) can be written in a way that is also well-defined if the potential ψ_I contains zeros:

$$N(\psi_I, i, j) := \sup_{\alpha\neq\alpha'}\sup_{\beta\neq\beta'}\sup_{\gamma,\gamma'} \frac{\sqrt{\psi_{I;\alpha\beta\gamma}\psi_{I;\alpha'\beta'\gamma'}} - \sqrt{\psi_{I;\alpha'\beta\gamma}\psi_{I;\alpha\beta'\gamma'}}}{\sqrt{\psi_{I;\alpha\beta\gamma}\psi_{I;\alpha'\beta'\gamma'}} + \sqrt{\psi_{I;\alpha'\beta\gamma}\psi_{I;\alpha\beta'\gamma'}}} \tag{2.52}$$

which is defined for $i,j \in N_I$ with $i \neq j$ and where $\psi_{I;\alpha\beta\gamma}$ is shorthand for $\psi_I(x_i = \alpha, x_j = \beta, x_{N_I\setminus\{i,j\}} = \gamma)$.

The immediate generalization of Theorem 2.11 is then as follows:

Theorem 2.13 *Under the assumptions on the potentials described above (strict positivity of single-variable factors and (2.51) for the other factors): if the spectral radius of the matrix*

$$A_{I\to i, J\to j} = \mathbf{1}_{N_j\setminus I}(J)\mathbf{1}_{N_I\setminus i}(j)N(\psi_I, i, j), \tag{2.53}$$

[8] Additionally, the initial messages are required to be strictly positive, but this requirement is easily met and is necessary for obtaining good BP results.

(with $N(\psi_I, i, j)$ defined in (2.52)) is strictly smaller than 1, BP converges to a unique fixed point irrespective of the initial messages.

Proof. Similar to the strictly positive case. The only slight subtlety occurs in Appendix 2.B where one has to take a limit of strictly positive factors converging to the desired nonnegative factor and use the continuity of the relevant expressions with respect to the factor entries to prove that the bound also holds in this limit. □

This theorem is the main result of this work; all other convergence and uniqueness theorems derived earlier, apart from Corollary 2.9, are implied by Theorem 2.13.

Example

Define, for $\epsilon \geq 0$, the ("ferromagnetic") pairwise factor $\psi(\epsilon)$ by the following matrix:

$$\psi(\epsilon) := \begin{pmatrix} 1 & \epsilon \\ \epsilon & 1 \end{pmatrix}.$$

Now consider a binary pairwise factor graph, consisting of a single loop of N binary variables, i.e., the network topology is that of a circle. Take for the $N-1$ pairwise interactions $\psi_{\{i,i+1\}}$ (for $i = 1, 2, \ldots, N-1$) the identity matrices (i.e., the above pairwise factors for $\epsilon = 0$) and take for the remaining one $\psi_{\{1,N\}} = \psi(\epsilon)$ for some $\epsilon \geq 0$. Note that the potential strength $N(\psi(\epsilon)) = \frac{1-\epsilon}{1+\epsilon}$ converges to 1 as $\epsilon \downarrow 0$. The spectral radius of the corresponding matrix $A_{I\to i,J\to j}$ can be shown to be equal to

$$\rho(A) = \left(\frac{1-\epsilon}{1+\epsilon}\right)^{1/N}$$

which is strictly smaller than 1 if and only if $\epsilon > 0$. Hence BP converges to a unique fixed point if $\epsilon > 0$. This result is sharp, since for $\epsilon = 0$, BP simply "rotates" the messages around without changing them and hence no convergence occurs (except, obviously, if the initial messages already correspond to the fixed point of uniform messages).

2.5 Comparison with other work

In this section we explore the relations of our results with previously existing work.

2.5.1 Comparison with work of Tatikonda and Jordan

In [Tatikonda and Jordan, 2002; Tatikonda, 2003], a connection is made between two seemingly different topics, namely Belief Propagation on the one hand and the theory of Gibbs measures [Georgii, 1988] on the other hand. The main result of [Tatikonda and Jordan, 2002] states that BP converges uniformly (to a unique fixed

point) if the Gibbs measure on the corresponding computation tree is unique. The *computation tree* is an "unwrapping" of the factor graph with respect to the Belief Propagation algorithm; specifically, the computation tree starting at variable $i \in \mathcal{V}$ consists of all paths starting at i that never backtrack.

This is a remarkable and beautiful result; however, the question of convergence of BP is replaced by the question of uniqueness of the Gibbs measure, which is not trivial. Fortunately, sufficient conditions for the uniqueness of the Gibbs measure exist; the most well-known are *Dobrushin's condition* and a weaker (but more easily verifiable) condition known as *Simon's condition*. In combination with the main result of [Tatikonda and Jordan, 2002], they yield directly testable sufficient conditions for convergence of BP to a unique fixed point. For reference, we will state both results in our notation below. For details, see [Tatikonda and Jordan, 2002; Tatikonda, 2003] and [Georgii, 1988]. Note that the results are valid for the case of positive factors depending on at most two variables.

BP convergence via Dobrushin's condition

Define *Dobrushin's interdependence matrix* as the $N \times N$ matrix C with entries

$$C_{ij} := \sup_{x_{\partial i \setminus j} \in \mathcal{X}_{\partial i \setminus j}} \sup_{x_j, x'_j \in \mathcal{X}_j} \frac{1}{2} \sum_{x_i \in \mathcal{X}_i} \left| \mathbb{P}(x_i \,|\, x_{\partial i \setminus j}, x_j) - \mathbb{P}(x_i \,|\, x_{\partial i \setminus j}, x'_j) \right| \tag{2.54}$$

for $j \in \partial i$ and 0 otherwise.

Theorem 2.14 *For pairwise, positive factors, BP converges to a unique fixed point if*

$$\max_{i \in \mathcal{V}} \sum_{j \in \partial i} C_{ij} < 1.$$

Proof. For a proof sketch, see [Tatikonda, 2003]. For the proof of Dobrushin's condition see [Georgii, 1988, Chapter 8]. □

We can rewrite the conditional probabilities in terms of factors:

$$\mathbb{P}(x_i \,|\, x_{\partial i \setminus j}, x_j) = \frac{\psi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k \in \partial i \setminus j} \psi_{ik}(x_i, x_k)}{\sum_{x_i} \psi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k \in \partial i \setminus j} \psi_{ik}(x_i, x_k)}.$$

Note that the complexity of the calculation of this quantity is generally exponential in the size of the neighborhood ∂i, which may prohibit practical application of Dobrushin's condition.

For the case of binary ± 1-valued variables, some elementary algebraic manipulations yield

$$\begin{aligned} C_{ij} &= \sup_{x_{\partial i \setminus j}} \frac{\sinh 2\,|J_{ij}|}{\cosh 2J_{ij} + \cosh 2\big(\theta_i + \sum_{k \in \partial i \setminus j} x_k J_{ik}\big)} \\ &= \frac{\tanh(|J_{ij}| - H_{ij}) + \tanh(|J_{ij}| + H_{ij})}{2} \end{aligned}$$

with

$$H_{ij} := \inf_{x_{\partial i \setminus j}} \left| \theta_i + \sum_{k \in \partial i \setminus j} x_k J_{ik} \right| .$$

BP convergence via Simon's condition

Simon's condition is a sufficient condition for Dobrushin's condition (see [Georgii, 1988, Proposition 8.8]). This leads to a looser, but more easily verifiable, bound:

Theorem 2.15 *For pairwise, positive factors, BP converges to a unique fixed point if*

$$\max_{i \in \mathcal{V}} \sum_{j \in \partial i} \left(\frac{1}{2} \sup_{\alpha,\alpha'} \sup_{\beta,\beta'} \log \frac{\psi_{ij;\alpha\beta}}{\psi_{ij;\alpha'\beta'}} \right) < 1.$$

□

It is not difficult to show that this bound is less tight than (2.49). Furthermore, unlike Dobrushin's condition and Corollary 2.9, it does not take into account single-variable factors.

2.5.2 Comparison with work of Ihler *et al.*

In the recent and independent work of Ihler *et al.* [2005b], a methodology was used which is very similar to the one used in this work. In particular, the same local ℓ_∞ quotient metric is used to derive sufficient conditions for BP to be a contraction. In the work presented here, the Mean Value Theorem (in the form of Lemma 2.2) is used in combination with a bound on the derivative in order to obtain a bound on the convergence rate K in (2.8). In contrast, in [Ihler *et al.*, 2005b] a direct bound on the distance of two outgoing messages is derived in terms of the distance of two different products of incoming messages [Ihler *et al.*, 2005b, Equation (13)]. This bound becomes relatively stronger as the distance of the products of incoming messages increases. This has the advantage that it can lead to stronger conclusions about the effect of finite message perturbations than would be possible with our bound, based on the Mean Value Theorem. However, for the question of *convergence*, the relevant limit turns out to be that of *infinitesimal* message perturbations, i.e., it suffices to study the derivative of the BP updates as we have done here.

In the limit of infinitesimal message perturbations, the basic bound (13) in [Ihler *et al.*, 2005b] leads to the following measure of potential strength:

$$D(\psi_{ij}) := \tanh \left(\frac{1}{2} \left(\sup_{\alpha,\alpha'} \sup_{\beta,\beta'} \log \frac{\psi_{ij;\alpha\beta}}{\psi_{ij;\alpha'\beta'}} \right) \right) .$$

Using this measure, Ihler *et. al* derive two different conditions for convergence of BP. The first one is similar to our (2.49) and the second condition is equivalent to our spectral radius result (2.50), except that in both conditions, $D(\psi_{ij})$ is used instead of $N(\psi_{ij})$. The latter condition is formulated in [Ihler *et al.*, 2005b] in terms

of the convergence properties of an iterative BP-like algorithm. The equivalence of this formulation with a formulation in terms of the spectral radius of a matrix can be seen from the fact that for any square matrix A, $\rho(A) < 1$ if and only if $\lim_{n\to\infty} A^n = 0$. However, our result also gives a contraction rate, unlike the iterative formulation in [Ihler *et al.*, 2005b].

Thus, the results in [Ihler *et al.*, 2005b] are similar to ours in the pairwise case, except for the occurrence of $D(\psi_{ij})$ instead of $N(\psi_{ij})$. It is not difficult to see that $N(\psi_{ij}) \leq D(\psi_{ij})$ for any pairwise factor ψ_{ij}; indeed, for any choice of $\alpha, \beta, \gamma, \delta$:

$$\sqrt{\psi_{ij;\alpha\gamma}\psi_{ij;\beta\delta}} \Big/ \sqrt{\psi_{ij;\beta\gamma}\psi_{ij;\alpha\delta}} \quad \leq \quad \Big(\sup_{\sigma\tau} \psi_{ij;\sigma\tau}\Big) \Big/ \Big(\inf_{\sigma\tau} \psi_{ij;\sigma\tau}\Big).$$

Thus the convergence results in [Ihler *et al.*, 2005b] are similar to, but weaker than the results derived in the present work.

After initial submission of this work, [Ihler *et al.*, 2005a] was published, which improves upon [Ihler *et al.*, 2005b] by exploiting the freedom of choice of the single-variable factors (which can be "absorbed" to an arbitrary amount by corresponding pairwise factors). This leads to an improved measure of potential strength, which turns out to be identical to our measure $N(\psi_{ij})$. Thus, for pairwise, strictly positive potentials, the results in [Ihler *et al.*, 2005a] are equivalent to the results (2.49) and (2.50) presented here. Our most general results, Theorems 2.10, 2.11 and 2.13 and Corollary 2.9, are not present in [Ihler *et al.*, 2005a].

2.5.3 Comparison with work of Heskes

A completely different methodology to obtain sufficient conditions for the uniqueness of the BP fixed point is used in [Heskes, 2004]. By studying the Bethe free energy and exploiting the relationship between properties of the Bethe free energy and the BP algorithm, conclusions are drawn about the uniqueness of the BP fixed point; however, whether uniqueness of the fixed point also implies convergence of BP seems to be an open question. We state the main result of [Heskes, 2004] in our notation below.

The following measure of potential strength is used in [Heskes, 2004]. For $I \in \mathcal{F}$, let

$$\omega_I := \sup_{x_{N_I}} \sup_{x'_{N_I}} \Big(\log \psi_I(x_{N_I}) + (\#(N_I) - 1) \log \psi_I(x'_{N_I}) - \sum_{i \in N_I} \log \psi_I(x'_{N_I \setminus i}, x_i) \Big).$$

The potential strength is then defined as $\sigma_I := 1 - e^{-\omega_I}$.

Theorem 2.16 *BP has a unique fixed point if there exists an "allocation matrix" X_{Ii} between factors $I \in \mathcal{F}$ and variables $i \in \mathcal{V}$ such that*

1. $X_{Ii} \geq 0 \qquad \forall I \in \mathcal{F}, \forall i \in N_I;$
2. $(1 - \sigma_I) \max_{i \in N_I} X_{Ii} + \sigma_I \sum_{i \in N_I} X_{Ii} \leq 1 \qquad \forall I \in \mathcal{F};$

3. $\sum_{I \in N_i} X_{Ii} \geq \#(N_i) - 1 \qquad \forall i \in \mathcal{V}.$

Proof. See [Heskes, 2004, Theorem 8.1]. □

The (non-)existence of such a matrix can be determined using standard linear programming techniques.

2.6 Numerical comparison of various bounds

In this subsection, we compare various bounds on binary pairwise graphical models, defined in (2.5), for various choices of the parameters. First we study the case of a completely uniform model (i.e., full connectivity, uniform couplings and uniform local fields). Then we study nonuniform couplings J_{ij}, in the absence of local fields. Finally, we take fully random models in various parameter regimes (weak/strong local fields, strong/weak ferromagnetic/spin-glass/antiferromagnetic couplings).

2.6.1 Uniform couplings, uniform local field

The fully connected Ising model consisting of N binary ± 1-valued variables with uniform couplings J and uniform local field θ is special in the sense that an exact description of the parameter region for which the Gibbs measure on the computation tree is unique is available. Using the results of Tatikonda and Jordan, this yields a strong bound on the parameter region for which BP converges to a unique fixed point. Indeed, the corresponding computation tree is a uniform Ising model on a Cayley tree of degree $N - 2$, for which (semi-)analytical expressions for the paramagnetic–ferromagnetic and paramagnetic–antiferromagnetic phase-transition boundaries are known (see [Georgii, 1988, Section 12.2]). Since the Gibbs measure is known to be unique in the paramagnetic phase, this gives an exact description of the (J, θ) region for which the Gibbs measure on the computation tree is unique, and hence a bound on BP convergence on the original model.

In figure 2.4 we have plotted various bounds on BP convergence in the (J, θ) plane for $N = 4$ (other values of N yield qualitatively similar results). The gray area (g) marks regions where the Gibbs measure on the computation tree is not unique; in the white area, the Gibbs measure is unique and hence BP is guaranteed to converge. Note that this bound is only available due to the high symmetry of the model. In [Taga and Mase, 2006b] it is shown that parallel BP does not converge in the lower (antiferromagnetic) gray region. In the upper (ferromagnetic) region on the other hand, parallel BP does converge, but it may be that the fixed point is no longer unique.

The various lines correspond to different sufficient conditions for BP convergence; the regions enclosed by two lines of the same type (i.e., the inner regions for which $|J|$ is small) mark the regions of guaranteed convergence. The lightly dotted lines

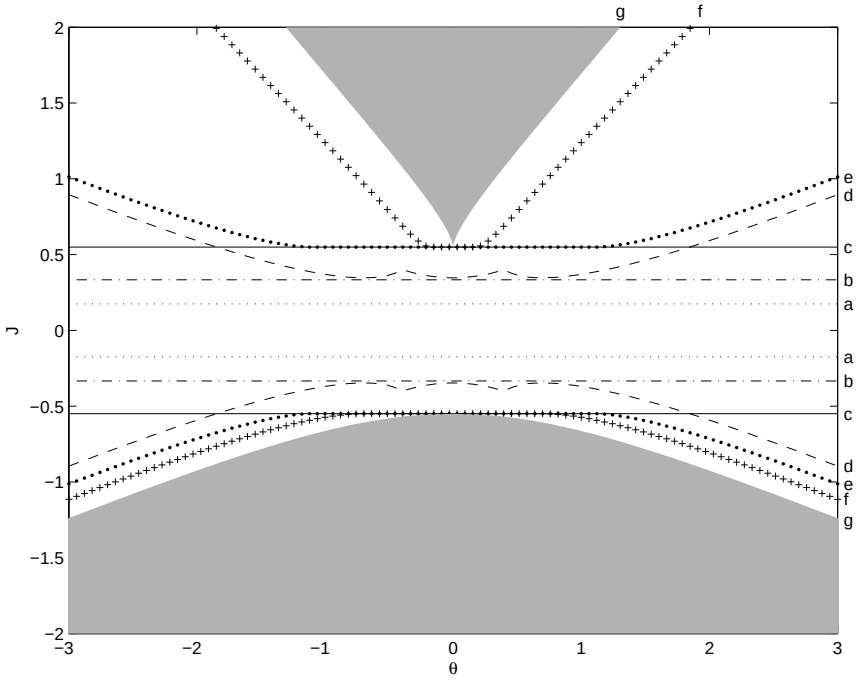


Figure 2.4: Comparison of various BP convergence bounds for the fully connected $N = 4$ binary Ising model with uniform coupling J and uniform local field θ. (a) Heskes' condition (b) Simon's condition (c) spectral radius condition (d) Dobrushin's condition (e) improved spectral radius condition for $m = 1$ (f) improved spectral radius condition for $m = 5$ (g) uniqueness of Gibbs' measure condition. See the main text (section 2.6.1) for more explanation.

(a) correspond with Heskes' condition, Theorem 2.16. The dash-dotted lines (b) correspond with Simon's condition, Theorem 2.15. The dashed lines (d) correspond with Dobrushin's condition (Theorem 2.14), which is seen to improve upon Simon's condition for $\theta \neq 0$, but is nowhere sharp. The solid lines (c) correspond with the spectral radius condition Corollary 2.8 (which coincides with the ℓ_1-norm condition Corollary 2.5 in this case and is also equivalent to the result of [Ihler *et al.*, 2005b]), which is independent of θ but is actually sharp for $\theta = 0$. The heavily dotted lines (e) correspond to Corollary 2.9 with $m = 1$, the +-shaped lines (f) to the same Corollary with $m = 5$. Both (e) and (f) are seen to coincide with (c) for small θ, but improve for large θ.

We conclude that the presence of local fields makes it more difficult to obtain sharp bounds on BP convergence; only Dobrushin's condition (Theorem 2.14) and Corollary 2.9 take into account local fields. Furthermore, in this case, our result Corollary 2.9 is stronger than the other bounds. Note that the calculation of Dobrushin's condition is exponential in the number of variables N, whereas the time

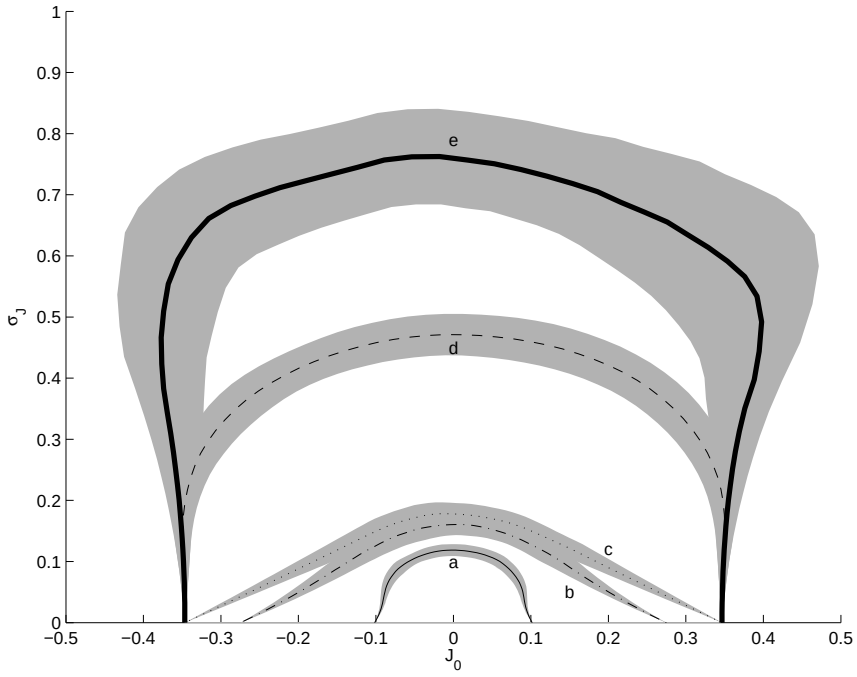


Figure 2.5: Comparison of various bounds for BP convergence for toroidal Ising model of size 10×10 with normally distributed couplings $J_{ij} \sim \mathcal{N}\left(J_0, {\sigma_J}^2\right)$ and zero local fields. (a) Heskes' condition (b) Dobrushin's condition (c) ℓ_1-norm condition (d) spectral radius condition (e) empirical convergence boundary. See the main text (section 2.6.2) for more explanation.

complexity of our bound is polynomial in N. Similar results are obtained for higher values of N.

2.6.2 Nonuniform couplings, zero local fields

We have investigated in more detail the influence of the distribution of the couplings J_{ij}, in the absence of local fields, and have also compared with the empirical convergence behavior of BP. We have taken a binary Ising model on a rectangular toroidal grid (i.e., with periodic boundary conditions) of size 10×10. The couplings were random independent normally distributed nearest-neighbor couplings $J_{ij} \sim \mathcal{N}\left(J_0, {\sigma_J}^2\right)$, the local fields were $\theta_i = 0$. Let (r_J, ϕ_J) be the polar coordinates corresponding to the Cartesian coordinates (J_0, σ_J). For various angles $\phi_J \in [0, \pi]$, we have determined the critical radius r_J for each bound. The results have been averaged over 40 instances of the model and can be found in figure 2.5. The lines correspond to the mean bounds, the gray areas are "error bars" of one standard deviation. The inner area (for which the couplings are small) bounded by each line means "convergence", either guaranteed or empirical (thus the larger

the enclosed area, the tighter the bound). From bottom to top: the thin solid line (a) corresponds with Heskes' result (Theorem 2.16), the dash-dotted line (b) with Dobrushin's condition (Theorem 2.14), the dotted line (c) corresponds with the ℓ_1-norm condition Corollary 2.5, the dashed line (d) with the spectral radius condition Corollary 2.8 and the thick solid line (e) with the empirical convergence behavior of BP.

We conclude from figure 2.5 that the spectral radius condition improves upon the ℓ_1-norm condition for nonuniform couplings and that the improvement can be quite substantial. For uniform couplings (and zero local fields), both conditions coincide and it can be proved that they are sharp [Mooij and Kappen, 2005a].

2.6.3 Fully random models

Finally, we have considered fully connected binary pairwise graphical models with completely random couplings and local fields (in various parameter regimes). We drew random couplings and local fields as follows: first, we drew i.i.d. random parameters $J_0, \sigma_J, \theta_0, \sigma_\theta$ from a normal distribution with mean 0 and variance 1. Then, for each variable i we independently drew a local field parameter $\theta_i \sim \mathcal{N}\left(\theta_0, {\sigma_\theta}^2\right)$, and for each pair $\{i, j\}$ we independently drew a coupling parameter $J_{ij} \sim \mathcal{N}\left(J_0, {\sigma_J}^2\right)$.

For the resulting graphical model, we have verified whether various sufficient conditions for BP convergence hold. If condition A holds whereas condition B does not hold, we say that A wins from B. We have counted for each ordered pair (A, B) of conditions how often A wins from B. The results (for 50000 random models consisting of $N = 4, 8$ variables) can be found in table 2.1: the number at row A, column B is the number of trials for which bound A wins from bound B. On the diagonal ($A = B$) is the total number of trials for which bound A predicts convergence. Theorem 2.14 is due to [Tatikonda, 2003], Corollary 2.8 was first published (for the binary case) in [Ihler *et al.*, 2005b] and Theorem 2.16 is due to [Heskes, 2004].

Our result Corollary 2.9 (for $m = 1$) outperforms the other bounds in each trial. For other values of N, we obtain similar results.

2.7 Discussion

In this paper we have derived sufficient conditions for convergence of BP to a unique fixed point. Our conditions are directly applicable to arbitrary factor graphs with discrete variables and nonnegative factors. This is in contrast with the sufficient conditions of Tatikonda and Jordan and with the results of Ihler, Fisher and Willsky, which were only formulated for pairwise, positive factors. We have shown cases where our results are stronger than previously known sufficient conditions.

Our numerical experiments lead us to conjecture that Corollary 2.9 is stronger than the other bounds. We have no proof for this conjecture at the moment, apart

Table 2.1: Comparison of bounds (50000 trials, for $N = 4$ and $N = 8$)

$N = 4$	Th. 2.14	Cor. 2.8	Th. 2.16	Cor. 2.9
Th. 2.14, [Tatikonda, 2003]	(5779)	170	3564	0
Cor. 2.8, [Ihler *et al.*, 2005b]	10849	(16458)	13905	0
Th. 2.16, [Heskes, 2004]	338	0	(2553)	0
Cor. 2.9, $m = 1$, this work	13820	3141	17046	(19599)

$N = 8$	Th. 2.14	Cor. 2.8	Th. 2.16	Cor. 2.9
Th. 2.14, [Tatikonda, 2003]	(668)	39	597	0
Cor. 2.8, [Ihler *et al.*, 2005b]	507	(1136)	1065	0
Th. 2.16, [Heskes, 2004]	0	0	(71)	0
Cor. 2.9, $m = 1$, this work	972	504	1569	(1640)

from the obvious fact that Corollary 2.8 is weaker than Corollary 2.9. To prove that Corollary 2.9 is stronger than Theorem 2.14 seems subtle, since it is generally not the case that $\rho(A) \leq \|C\|_\infty$, although it seems that the weaker relation $\|C\|_\infty < 1 \implies \rho(A) < 1$ does hold in general. The relation with the condition in Theorem 2.16 is not evident as well.

In the binary pairwise case, it turned out to be possible to derive sufficient conditions that take into account local evidence (Corollary 2.9). In the general case, such an improvement is possible in principle but seems to be more involved. The resulting optimization problem (essentially (2.43) with additional assumptions on h) looks difficult in general. If the variables' cardinalities and connectivities are small, the resulting optimization problem can be solved, but writing down a general solution does not appear to be trivial. The question of finding an efficient solution in the general case is left for future investigation.

The work reported here raises new questions, some of which have been (partially) answered elsewhere after the initial submission of this paper. The influence of damping the BP update equations has been considered for the binary pairwise case in [Mooij and Kappen, 2005a], where it was shown that damping has the most effect for antiferromagnetic interactions. Furthermore, it has been proved in [Mooij and Kappen, 2005a] that the bounds for BP convergence derived in the present work are sharp in the case of binary variables with (anti)ferromagnetic pairwise interactions and zero local fields, as suggested by figure 2.5. An extension of the results towards sequential update schemes has been given in [Elidan *et al.*, 2006]. Likewise, in [Taga and Mase, 2006b] it is shown that Dobrushin's condition is also valid for sequential BP.

Acknowledgments

We thank Martijn Leisink for stimulating discussions and the reviewers for their critique, which has led to a considerable improvement of the initial manuscript.

2.A Generalizing the ℓ_1-norm

Let $(V_i, \|\cdot\|_i)$ be a finite collection of normed vector spaces and let $V = \bigoplus_i V_i$ be the direct sum of the V_i. The function $\|\cdot\| : V \to \mathbb{R}$ defined by

$$\|v\| := \sum_i \|v_i\|_i \tag{2.55}$$

is a norm on V, as one easily checks. Let $A : V \to V$ be a linear mapping with "blocks" $A_{ij} : V_j \to V_i$ defined by

$$\forall v_j \in V_j : \quad Av_j = \sum_i A_{ij} v_j, \quad A_{ij} v_j \in V_i$$

for all j.

Theorem 2.17 *The matrix norm of A induced by the vector norm $\|\cdot\|$ is given by:*

$$\|A\| = \max_j \sum_i \|A_{ij}\|_i^j \tag{2.56}$$

where

$$\|A_{ij}\|_i^j := \sup_{\substack{v \in V_j, \\ \|v\|_j \leq 1}} \|A_{ij} v\|_i .$$

Proof. Let $v_k \in V_k$ such that $\|v_k\|_k = 1$. Then

$$\|Av_k\| = \left\| \sum_i A_{ik} v_k \right\| = \sum_i \|A_{ik} v_k\|_i \leq \sum_i \|A_{ik}\|_i^k \leq \max_j \sum_i \|A_{ij}\|_i^j .$$

Now let $v \in V$ such that $\|v\| = 1$. Then v can be written as the convex combination $v = \sum_k \|v_k\|_k \, \tilde{v}_k$, where

$$\tilde{v}_k := \begin{cases} \frac{v_k}{\|v_k\|_k} & \text{if } v_k \neq 0 \\ 0 & \text{if } v_k = 0. \end{cases}$$

Hence:

$$\|Av\| = \left\| \sum_k \|v_k\|_k \, A\tilde{v}_k \right\| \leq \sum_k \|v_k\|_k \, \|A\tilde{v}_k\| \leq \max_j \sum_i \|A_{ij}\|_i^j .$$

It is evident that this value is also achieved for some $v \in V$ with $\|v\| = 1$. □

An illustrative example is obtained by considering $V = \mathbb{R}^N$ to be the direct sum of N copies of $\mathbb{R}$ with the absolute value as norm; then the norm (2.55) on $\mathbb{R}^N$ is simply the ℓ_1-norm and the induced matrix norm (2.56) reduces to (2.9).

Suppose that each V_i has a linear subspace W_i. We can consider the quotient spaces V_i/W_i with quotient norms $\|\cdot\|_i$. The direct sum $W := \bigoplus_i W_i$ is itself a subspace of V, yielding a quotient space V/W. For $v \in V$ we have $\overline{v} = \sum_i \overline{v_i}$ and hence $V/W = \bigoplus_i (V_i/W_i)$. The quotient norm on V/W is simply the sum of the quotient norms on the V_i/W_i:

$$\begin{aligned} \|\overline{v}\| &:= \inf_{w \in W} \|v + w\| = \inf_{w \in W} \sum_i \|v_i + w_i\|_i \\ &= \sum_i \inf_{w_i \in W_i} \|v_i + w_i\|_i = \sum_i \|\overline{v_i}\|_i \ . \end{aligned} \tag{2.57}$$

Let $A : V \to V$ be a linear mapping such that $AW \subseteq W$. Then A induces a linear $\overline{A} : V/W \to V/W$; since $A_{ij} W_j \subseteq W_i$, each block $A_{ij} : V_j \to V_i$ induces a linear $\overline{A_{ij}} : V_j/W_j \to V_i/W_i$, and $\overline{A}$ can be regarded as consisting of the blocks $\overline{A_{ij}}$.

Corollary 2.18 *The matrix norm of $\overline{A} : V/W \to V/W$ induced by the quotient norm $\|\cdot\|$ on V/W is:*

$$\left\|\overline{A}\right\| = \max_j \sum_i \left\|\overline{A_{ij}}\right\|_i^j \tag{2.58}$$

where

$$\left\|\overline{A_{ij}}\right\|_i^j = \sup_{\substack{v \in V_j, \\ \|v\|_j \leq 1}} \left\|\overline{A_{ij} v}\right\|_i \ . \tag{2.59}$$

Proof. We can directly apply the previous Theorem to the quotient spaces to obtain (2.58); because

$$\{\overline{v} \in V_j/W_j : \|\overline{v}\|_j \leq 1\} = \overline{\{v \in V_j : \|v\|_j \leq 1\}},$$

we have:

$$\left\|\overline{A_{ij}}\right\|_i^j := \sup_{\substack{\overline{v} \in V_j/W_j \\ \|\overline{v}\|_j \leq 1}} \left\|\overline{A_{ij}}\,\overline{v}\right\|_i = \sup_{\substack{v \in V_j \\ \|v\|_j \leq 1}} \left\|\overline{A_{ij} v}\right\|_i \ .$$

□

For a linear $A : V \to V$ such that $AW \subseteq W$, we define the matrix $|A|$ with entries $|A|_{ij} := \left\|\overline{A_{ij}}\right\|_i^j$. Let A, B be two such linear mappings; then

$$\begin{aligned} |AB|_{ij} = \left\|\overline{(AB)_{ij}}\right\|_i^j &= \left\|\sum_k \overline{A_{ik} B_{kj}}\right\|_i^j \leq \sum_k \left\|\overline{A_{ik} B_{kj}}\right\|_i^j \\ &\leq \sum_k \left\|\overline{A_{ik}}\right\|_i^k \left\|\overline{B_{kj}}\right\|_k^j = \sum_k |A|_{ik} |B|_{kj} \end{aligned}$$

hence $|AB| \leq |A|\,|B|$. Note that $\||A|\|_1 = \left\|\overline{A}\right\|$. We can generalize Theorem 2.7 as follows:

Theorem 2.19 *Let $f : V \to V$ be differentiable and suppose that it satisfies (2.30). Suppose further that $|f'(v)| \leq A$ for some matrix A_{ij} (which does not depend on v) with $\rho(A) < 1$. Then for any $\overline{v} \in V/W$, the sequence $\overline{v}, \overline{f}(\overline{v}), \overline{f}^2(\overline{v}), \ldots$ obtained by iterating $\overline{f}$ converges to a unique fixed point $\overline{v}_\infty$.*

Proof. Using the chain rule, we have for any $n \in \mathbb{N}^*$ and any $v \in V$:

$$\left\|(\overline{f}^n)'(\overline{v})\right\| = \left\|\overline{(f^n)'(v)}\right\| = \left\|\overline{\prod_{i=1}^{n} f'\big(f^{i-1}(v)\big)}\right\| = \left\|\left\|\prod_{i=1}^{n} f'\big(f^{i-1}(v)\big)\right\|\right\|_1$$
$$\leq \left\|\prod_{i=1}^{n} \left|f'\big(f^{i-1}(v)\big)\right|\right\|_1 \leq \|A^n\|_1 .$$

By the Gelfand Spectral Radius Theorem, $\big(\ \|A^n\|_1\ \big)^{1/n} \to \rho(A)$ for $n \to \infty$. Choose $\epsilon > 0$ such that $\rho(A) + \epsilon < 1$. For some N, $\left\|A^N\right\|_1 \leq (\rho(A) + \epsilon)^N < 1$. Hence $\left\|(\overline{f}^N)'(\overline{v})\right\| < 1$ for all $\overline{v} \in V/W$. By Lemma 2.3, $\overline{f}^N$ is a contraction with respect to the quotient norm on V/W. Now apply Lemma 2.6. □

2.B Proof that (2.43) equals (2.44)

Let $\psi_{\beta\gamma}$ be a matrix of positive numbers. Let

$$\mathcal{H} := \{h : h_{\beta\gamma} \geq 0, \sum_\beta \sum_\gamma h_{\beta\gamma} = 1\}.$$

Define the function $g : \mathcal{H} \to \mathbb{R}$ by

$$g(h) = \sum_\beta \left| \sum_\gamma h_{\beta\gamma} \left(\frac{\psi_{\beta\gamma}}{\sum_\beta \sum_\gamma \psi_{\beta\gamma} h_{\beta\gamma}} - 1 \right) \right| .$$

Theorem 2.20

$$\sup_{h \in \mathcal{H}} g(h) = 2 \sup_{\beta \neq \beta'} \sup_{\gamma,\gamma'} \tanh \left(\frac{1}{4} \log \frac{\psi_{\beta\gamma}}{\psi_{\beta'\gamma'}} \right) .$$

Proof. First note that we can assume without loss of generality that all $\psi_{\beta\gamma}$ are different, because of continuity. Define

$$\psi_- := \inf_{\beta\gamma} \psi_{\beta\gamma}, \qquad \psi_+ := \sup_{\beta\gamma} \psi_{\beta\gamma},$$
$$\Psi := [\psi_-, \psi_+], \qquad \Psi' := \Psi \setminus \{\psi_{\beta\gamma} : \beta, \gamma\}.$$

For $\phi \in \Psi$, define

$$\mathcal{H}_\phi := \{h \in \mathcal{H} : \sum_{\beta,\gamma} \psi_{\beta\gamma} h_{\beta\gamma} = \phi\},$$

which is evidently a closed convex set. The function

$$g_\phi : \mathcal{H}_\phi \to \mathbb{R} : h \mapsto \sum_\beta \left| \sum_\gamma h_{\beta\gamma} \left(\frac{\psi_{\beta\gamma}}{\phi} - 1 \right) \right|$$

obtained by restricting g to $\mathcal{H}_\phi$ is convex. Hence it achieves its maximum on an extremal point of its domain.

Define

$$\mathcal{H}_2 := \left\{ h \in \mathcal{H} : \#\{(\beta,\gamma) : h_{\beta\gamma} > 0\} = 2 \right\}$$

as those $h \in \mathcal{H}$ with exactly two nonzero components. For $h \in \mathcal{H}_2$, define $\psi_-(h) := \inf\{\psi_{\beta\gamma} : h_{\beta\gamma} \neq 0\}$ and $\psi_+(h) := \sup\{\psi_{\beta\gamma} : h_{\beta\gamma} \neq 0\}$. Because of continuity, we can restrict ourselves to the $\phi \in \Psi'$, in which case the extremal points of $\mathcal{H}_\phi$ are precisely $\mathcal{H}_\phi^* = \mathcal{H}_\phi \cap \mathcal{H}_2$ (i.e., the extremal points have exactly two nonzero components).

Now

$$\begin{aligned} \sup_{h\in\mathcal{H}} g(h) &= \sup_{\phi\in\Psi} \sup_{h\in\mathcal{H}_\phi} g_\phi(h) = \sup_{\phi\in\Psi'} \sup_{h\in\mathcal{H}_\phi^*} g_\phi(h) \\ &= \sup_{h\in\mathcal{H}_2} \sup_{\psi_-(h)\le\phi\le\psi_+(h)} g_\phi(h) = \sup_{h\in\mathcal{H}_2} g(h). \end{aligned}$$

For those $h \in \mathcal{H}_2$ with components with different β, we can use the Lemma below. The $h \in \mathcal{H}_2$ with components with equal β are suboptimal, since the two contributions in the sum over γ in $g(h)$ have opposite sign. Hence

$$\sup_{h\in\mathcal{H}_2} g(h) = 2 \sup_{\beta\neq\beta'} \sup_{\gamma,\gamma'} \tanh\left(\frac{1}{4} \log \frac{\psi_{\beta\gamma}}{\psi_{\beta'\gamma'}} \right).$$

□

Lemma 2.21 *Let $0 < a < b$. Then*

$$\begin{aligned} &\sup_{\substack{\eta\in(0,1)^2 \\ \eta_1+\eta_2=1}} \eta_1 \left| \frac{a}{\eta_1 a + \eta_2 b} - 1 \right| + \eta_2 \left| \frac{b}{\eta_1 a + \eta_2 b} - 1 \right| \\ &= 2 \tanh\left(\frac{1}{4} \log \frac{b}{a} \right) = 2 \frac{\sqrt{b} - \sqrt{a}}{\sqrt{b} + \sqrt{a}}. \end{aligned}$$

Proof. Elementary. The easiest way to see this is to reparameterize

$$\eta = \left(\frac{e^\nu}{2\cosh\nu}, \frac{e^{-\nu}}{2\cosh\nu} \right)$$

with $\nu \in (-\infty, \infty)$. □

Chapter 3

BP and phase transitions

We analyze the local stability of the high-temperature fixed point of the Belief Propagation (BP) algorithm and how this relates to the properties of the Bethe free energy which BP tries to minimize. We focus on the case of binary variables with pairwise interactions. In particular, we state sufficient conditions for convergence of BP to a unique fixed point and show that these are sharp for purely ferromagnetic interactions. In contrast, in the purely antiferromagnetic case, the undamped parallel BP algorithm is suboptimal in the sense that the stability of the fixed point breaks down much earlier than for damped or sequential BP, and we observe that the onset of instability for those BP variants is directly related to the properties of the Bethe free energy. For spin-glass interactions, damping BP only helps slightly. We estimate analytically the temperature at which the high-temperature BP fixed point becomes unstable for random graphs with arbitrary degree distributions and random interactions.

3.1 Introduction

Techniques that were originally developed in the statistical physics of lattice models are nowadays increasingly often and successfully applied in diverse application areas such as information theory, coding theory, combinatorial optimization and machine learning. A prominent example is the Bethe-Peierls approximation [Bethe, 1935; Peierls, 1936], an extension of the ordinary Mean Field method that takes into account correlations between nearest-neighbor sites. A more general and powerful approximation scheme, which is also currently being used as a general inference tool in applications in the aforementioned areas, is the Cluster Variation Method

This chapter is based on [Mooij and Kappen, 2005a] and earlier work reported in [Mooij and Kappen, 2005c].

(CVM) [Kikuchi, 1951; Pelizzola, 2005], also called Kikuchi approximation. The CVM treats arbitrarily large clusters of sites exactly; the Bethe approximation can be seen as the simplest nontrivial case (the pair approximation) of the Cluster Variation Method.

The problems arising in the aforementioned application domains can often be reformulated as inference problems on graphical models, i.e., as the calculation of marginal probabilities of some probability distribution. Typically, this probability distribution is proportional to a product of many factors, each factor depending on only a few variables; this structure can be expressed in terms of a graph, hence the name *graphical model*. An illustrative example can be found in image restoration [Tanaka, 2002], where the 2D classical Ising model can be used to model features of monochromatic images. The pixels in the image correspond to the Ising spins, the local external fields correspond to observed, noisy pixels and the probability distribution over different images corresponds to the equilibrium Boltzmann distribution of the Ising model. The underlying graph is in this example the 2D rectangular lattice, and the interactions between the nearest neighbors correspond to factors in the probability distribution. By taking the interactions to be of the ferromagnetic type, one can obtain a smoothing filter.

In statistical physics, one is predominantly interested in the thermodynamic limit of infinitely large systems, and furthermore, in the case of disordered systems, one usually averages over a whole ensemble of such systems. In contrast, in the applications in computer science the primary interest lies in the properties of individual, finite systems—in the example above, one would be interested in individual images. Given the probability distribution, the task is then to calculate marginal probabilities, which in principle amounts to performing a summation or integral. Unfortunately, the required computation time is generally exponential in the number of variables, and the calculation quickly becomes infeasible for real-world applications.

Therefore, one is often forced to use approximative methods, such as Monte Carlo methods or "deterministic approximations". A prominent example of the latter category is the successful Belief Propagation algorithm [Pearl, 1988], which was originally developed as a fast algorithm to calculate probabilities on graphical models without loops (i.e., on trees), for which the results are exact. The same algorithm can also be applied on graphs that contain loops, in which case the results are approximate, and it is then often called Loopy Belief Propagation (LBP) to emphasize the fact that the graph may contain loops. The results can be surprisingly good, even for small graphs with many short loops, e.g., in the case of decoding error-correcting codes [McEliece *et al.*, 1998; Nishimori, 2001]. An important discovery was that the BP algorithm in fact tries to minimize the Bethe free energy (more precisely, fixed points of the BP algorithm correspond to stationary points of the Bethe free energy) [Yedidia *et al.*, 2001]. This discovery has lead to renewed interest in the Bethe approximation and related methods and to cross-fertilization between disciplines, a rather spectacular example of which is the Survey Propagation (SP)

algorithm, which is now the state-of-the-art solution method for some difficult combinatorial optimization problems [Braunstein and Zecchina, 2004]. Other examples are the generalizations of BP obtained by replacing the Bethe free energy by the more complicated Kikuchi free energy, which has resulted in algorithms that are much faster than the NIM algorithm developed originally by Kikuchi [Pelizzola, 2005].

This chapter is organized as follows. We start in section 3.2 with a brief review of the Bethe approximation and the Belief Propagation algorithm, trying to combine the two different points of view, namely the statistical physicist's perspective and the one found in machine learning and computer science. A notorious problem plaguing applications of BP is the fact that it does not always converge to a fixed point. With the aim of better understanding these convergence issues, in section 3.3 we discuss the local stability of BP fixed points, state "global" conditions for convergence towards a unique fixed point, and discuss the stability of the high-temperature Bethe free energy minimum. In section 3.4, we discuss qualitatively how these properties are related and connect them with phase transitions in the thermodynamic limit. In section 3.5, we quantify the results of the previous section by estimating the phase-transition temperatures for random graphs with random interactions.

This chapter is written primarily for statistical physicists, but we tried to make it also understandable for readers with a background in computer science, which may explain some seemingly redundant remarks.

3.2 The Bethe approximation and the BP algorithm

3.2.1 The graphical model

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected labeled graph without self-connections, defined by a set of vertices $\mathcal{V} = \{1, \ldots, N\}$ and a set of undirected edges $\mathcal{E} \subseteq \big\{\{i,j\} : 1 \leq i < j \leq N\big\}$. The *adjacency matrix* M corresponding to $\mathcal{G}$ is defined as follows: $M_{ij} = 1$ if $\{i,j\} \in \mathcal{E}$ and 0 otherwise. Denote by ∂i the set of neighbors of vertex i, and the degree (connectivity) of vertex i by $d_i := \#(\partial i) = \sum_{j \in \mathcal{V}} M_{ij}$.

To each vertex $i \in \mathcal{V}$ we associate a random variable x_i (called a "spin"), taking values in $\{-1, +1\}$. We put weights J_{ij} on the edges $\{i,j\}$: let J be a real symmetric $N \times N$ matrix that is compatible with the adjacency matrix M, i.e., $J_{ij} = 0$ if $M_{ij} = 0$. Let $\theta \in \mathbb{R}^N$ be local "fields" (local "evidence") acting on the vertices. We will study the Boltzmann distribution corresponding to the Hamiltonian

$$H = -\sum_{\{i,j\} \in \mathcal{E}} J_{ij} x_i x_j - \sum_{i \in \mathcal{V}} \theta_i x_i = -\frac{1}{2} \sum_{i,j \in \mathcal{V}} J_{ij} x_i x_j - \sum_{i \in \mathcal{V}} \theta_i x_i, \qquad (3.1)$$

i.e., the probability of the configuration $x = (x_1, \ldots, x_N) \in \{-1, +1\}^N$ is given by:

$$\mathbb{P}(x) = \frac{1}{Z} \exp\left(\beta \sum_{\{i,j\} \in \mathcal{E}} J_{ij} x_i x_j + \beta \sum_{i \in \mathcal{V}} \theta_i x_i\right) \tag{3.2}$$

with $\beta > 0$ the inverse temperature and Z a normalization constant. The problem that we would like to solve is calculating the first and second moments $\mathbb{E}(x_i)$ and $\mathbb{E}(x_i x_j)$ under this distribution. In general, this is an NP-hard problem [Cooper, 1990], so in practice we often have to settle for approximations of these quantities.

The general model class that we have described above has been the subject of numerous investigations in statistical physics. There one often takes a lattice as the underlying graph $\mathcal{G}$, or studies an ensemble of random graphs (including the fully-connected SK model as a special case). The weights J_{ij} and the local fields θ_i are often taken to be i.i.d. according to some probability distribution (a special case is where this probability distribution is a delta function—this corresponds to uniform, deterministic interactions). In these cases one can take the thermodynamic limit $N \to \infty$, which is the subject of investigation of the major part of statistical physics studies (except for the studies of "finite-size effects"). Depending on these weight distributions and on the graph structure, macroscopic order parameters can be identified that distinguish between different phases, e.g., the ferromagnetic phase for large positive weights or a spin-glass phase for weights that are distributed around zero.

The probability distribution (3.2) is a special case of the class of probability distributions over N discrete random variables $(x_i)_{i=1}^N$, with x_i taking values in some finite set $\mathcal{X}_i$, that factorize as a product of factors (often called "potentials" in computer science literature—not to be confused with the potentials in statistical physics, which are essentially the logarithms of the factors) in the following way:

$$\mathbb{P}(x) = \frac{1}{Z} \prod_{\{i,j\} \in \mathcal{E}} \psi_{ij}(x_i, x_j) \prod_{i \in \mathcal{V}} \psi_i(x_i) \tag{3.3}$$

with Z the normalization constant. These probability distributions are known in machine learning as *undirected graphical models* (in this case consisting of N nodes with pairwise potentials) or as *Markov random fields*. In fact, it is easy to see that (3.2) is equivalent to (3.3) when all variables are binary (and the factors are positive); in this case, equation (3.2) can obviously be written in the form of (3.3), but the converse also holds. In contrast with statistical physics studies, the number of variables is usually finite and one is interested in a single instance instead of the properties of an ensemble of instances.

In the following three subsections, we describe the BP algorithm and the Bethe approximation for the graphical model (3.3), and what is known about the relation between the two.

3.2.2 Bethe approximation

The calculation of properties such as marginals $\mathbb{P}(x_i)$ of the probability distribution (3.2) is an NP-hard problem [Cooper, 1990]. Only in cases with high symmetry (e.g., when all weights are equal and the field is uniform, i.e., $J_{ij} = J$ and $\theta_i = \theta$, and the graph has a high permutation symmetry, such as translation symmetry in case of a 2D rectangular lattice), or if N is small, or if the graph contains no cycles, it is possible to calculate marginals exactly. In other cases, one has to use approximate methods, such as Monte Carlo methods or "deterministic" approximation methods, the simplest of which is the well-known Mean Field method. An extension of the Mean Field method that treats pairs of neighboring spins exactly is the Bethe approximation, also known as the Bethe-Peierls approximation [Bethe, 1935; Peierls, 1936].

The Bethe approximation consists of minimizing the *Bethe free energy*, which for the factorizing probability distribution (3.3) is defined as the following functional [Yedidia *et al.*, 2001]:

$$\begin{aligned} &F_{Bethe}\big((b_i)_{i\in\mathcal{V}}, (b_{ij})_{\{i,j\}\in\mathcal{E}}\big) \\ &\quad= \sum_{\{i,j\}\in\mathcal{E}} \sum_{x_i,x_j} b_{ij}(x_i,x_j) \log \frac{b_{ij}(x_i,x_j)}{\psi_{ij}(x_i,x_j)\psi_i(x_i)\psi_j(x_j)} \\ &\quad\quad - \sum_{i\in\mathcal{V}} (d_i - 1) \sum_{x_i} b_i(x_i) \log \frac{b_i(x_i)}{\psi_i(x_i)}. \end{aligned} \tag{3.4}$$

Its arguments, called *beliefs*, are single-node (pseudo)marginals $b_i(x_i)$ and pairwise (pseudo)marginals $b_{ij}(x_i, x_j)$. The *Bethe approximation* is obtained by minimizing the Bethe free energy (3.4) with respect to the beliefs under the following nonnegativity, normalization and consistency constraints:

$$b_i(x_i) \geq 0 \qquad \forall x_i \in \mathcal{X}_i \tag{3.5a}$$

$$b_{ij}(x_i,x_j) \geq 0 \qquad \forall x_i \in \mathcal{X}_i,\ \forall x_j \in \mathcal{X}_j \tag{3.5b}$$

$$\sum_{x_i} b_i(x_i) = 1 \tag{3.5c}$$

$$\sum_{x_i} b_{ij}(x_i,x_j) = b_j(x_j) \tag{3.5d}$$

for all $i \in \mathcal{V}$, $j \in \partial i$. The beliefs that minimize F_{Bethe} under these constraints are then taken as approximations for the marginal distributions $\mathbb{P}(x_i)$ and $\mathbb{P}(x_i, x_j)$. The beliefs are the exact marginals if the underlying graph $\mathcal{G}$ contains no cycles [Baxter, 1982]. Note that the local consistency constraints do not imply global consistency of the beliefs in general, i.e., there does not always exist a probability distribution $b(x_1, \ldots, x_N)$ such that the beliefs are marginals of b.

The rationale for *minimizing* the Bethe free energy is that the Bethe free energy is an approximate Gibbs free energy with an exact energy term and an approximate entropy term (the entropy is approximated by a combination of single-node and

pairwise entropies). Minimizing the exact Gibbs free energy would recover the exact marginal distributions $\mathbb{P}(x_i)$ and $\mathbb{P}(x_i, x_j)$, but this is infeasible in general; minimizing its approximation, the Bethe free energy, gives approximations $b_i(x_i)$ and $b_{ij}(x_i, x_j)$ to the exact marginal distributions.

3.2.3 BP algorithm

A popular and efficient algorithm for obtaining the Bethe approximation is Belief Propagation (BP), also known under the names Sum-Product Algorithm [Kschischang *et al.*, 2001] and Loopy Belief Propagation [Pearl, 1988]. The adjective "loopy" is used to emphasize the fact that the graph may contain cycles, in which case the beliefs are usually only approximations of the exact marginals.

The BP algorithm consists of the iterative updating of a family of *messages* $(\mu_{i\to j})_{\{i,j\}\in\mathcal{E}}$. The new message $\mu'_{i\to j}$ that vertex i sends to its neighbor j is given in terms of all incoming messages by the following update rule [Yedidia *et al.*, 2001]:[1]

$$\mu'_{i\to j}(x_j) \propto \sum_{x_i} \psi_{ij}(x_i, x_j)\psi_i(x_i) \prod_{k\in\partial i\setminus j} \mu_{k\to i}(x_i), \tag{3.6}$$

where one usually normalizes messages such that $\sum_{x_j} \mu'_{i\to j}(x_j) = 1$. The update schedule can be chosen to be parallel ("flooding schedule"), sequential ("serial schedule") or random; the update schedule influences convergence properties.

If the messages $(\mu_{i\to j})_{\{i,j\}\in\mathcal{E}}$ converge to some fixed point $\mu^{(\infty)}$, the approximate marginal distributions (beliefs) $(b_i)_{i\in\mathcal{V}}$ and $(b_{ij})_{\{i,j\}\in\mathcal{E}}$ are calculated by:

$$b_i(x_i) \propto \psi_i(x_i) \prod_{k\in\partial i} \mu^{(\infty)}_{k\to i}(x_i), \tag{3.7}$$

$$b_{ij}(x_i, x_j) \propto \psi_{ij}(x_i, x_j)\psi_i(x_i)\psi_j(x_j) \prod_{k\in\partial i\setminus j} \mu^{(\infty)}_{k\to i}(x_i) \prod_{k\in\partial j\setminus i} \mu^{(\infty)}_{k\to j}(x_j). \tag{3.8}$$

Note that these beliefs satisfy the constraints (3.5).

Unfortunately, BP does not always converge. It can get trapped in limit cycles, or it can wander around chaotically, depending on the problem instance. This non-robust behavior hampers application of BP as a "black box" inference algorithm. Furthermore, there is some empirical evidence that if BP does not converge, the quality of the Bethe approximation (which can also be obtained by using double-loop algorithms [Heskes *et al.*, 2003] that are guaranteed to converge, but are slower than BP) is low. The analysis that we will perform in subsequent sections should be seen as first steps in obtaining a better understanding of these issues.

3.2.4 The connection between BP and the Bethe approximation

Using Lagrange multipliers, one can prove [Yedidia *et al.*, 2001] that the beliefs $b(\mu)$ corresponding to a BP fixed point μ are a stationary point of the Bethe free energy

[1] Here and in the following, if X is a set, we write $X \setminus i$ as a shorthand notation for $X \setminus \{i\}$.

under the constraints (3.5). Conversely, a set of messages μ for which the corresponding beliefs $b(\mu)$ are a stationary point of the constrained Bethe free energy, are a fixed point of BP. In other words: stationary points of the Bethe free energy correspond one-to-one to fixed points of BP.

It takes considerably more effort to prove that *(locally) stable* BP fixed points are *(local) minima* of the constrained Bethe free energy [Heskes, 2004]. The converse does not necessarily hold (as was already observed by Heskes [2004]), i.e., a minimum of the Bethe free energy need not be a stable fixed point of BP. In that case, BP cannot be used to obtain the Bethe approximation. We will see examples of this in section 3.4.

3.3 Stability analysis for binary variables

From now on, we consider the special case (3.2) for which all variables are binary. In this section, we derive conditions for the local stability of fixed points of parallel BP, in the undamped and damped cases. We state sufficient conditions for the uniqueness of the fixed point and "global" convergence properties of parallel, undamped BP. Finally, we discuss the properties of Bethe energy minima for binary variables. In section 3.4 we will study the relations between those properties. We will start with reformulating BP for the case of binary variables.

3.3.1 BP for binary variables

In the case of binary variables, we can parameterize each message $\mu_{i\to j}$ by a single real number. A canonical choice is to transform to the variables $\nu_{i\to j}$ defined by

$$\nu_{i\to j} := \tanh^{-1}\Big(\mu_{i\to j}(x_j = 1) - \mu_{i\to j}(x_j = -1)\Big). \tag{3.9}$$

The BP update equations (3.6) can be written in terms of these new messages as:

$$\tanh(\nu'_{i\to j}) = \tanh(\beta J_{ij}) \tanh(\beta \eta_{i\setminus j}), \tag{3.10}$$

where we defined the *cavity field* $\eta_{i\setminus j}$ by

$$\beta\eta_{i\setminus j} := \beta\theta_i + \sum_{k\in\partial i\setminus j} \nu_{k\to i}. \tag{3.11}$$

Our usage of the term "cavity field" corresponds to that in [Mézard and Parisi, 2001] and is motivated by the fact that $\eta_{i\setminus j}$ is the effective field that acts on spin i in the absence of spin j (under the assumption that the spins $k \in \partial i$ are independent in the absence of spin j).

The single-node beliefs $b_i(x_i)$ can be parameterized by their means ("magnetizations")

$$m_i := \mathbb{E}_{b_i}(x_i) = \sum_{x_i=\pm 1} x_i b_i(x_i), \tag{3.12}$$

and the pairwise beliefs $b_{ij}(x_i, x_j)$ can be parameterized by m_i, m_j and the second order moment ("correlation")

$$\chi_{ij} := \mathbb{E}_{b_{ij}}(x_i x_j) = \sum_{x_i=\pm 1} \sum_{x_j=\pm 1} x_i x_j b_{ij}(x_i, x_j). \tag{3.13}$$

The beliefs (3.7) and (3.8) at a fixed point ν can then be written as:

$$m_i = \tanh(\beta\eta_{i\setminus j} + \nu_{j\to i}), \tag{3.14}$$

$$\chi_{ij} = \tanh\left(\beta J_{ij} + \tanh^{-1}\left(\tanh(\beta\eta_{i\setminus j})\tanh(\beta\eta_{j\setminus i})\right)\right). \tag{3.15}$$

3.3.2 Local stability of undamped, parallel BP fixed points

For the parallel update scheme, we can consider the update mapping $F : \nu \mapsto \nu'$ written out in components in (3.10). Its derivative ("Jacobian") is given by:

$$\begin{aligned} F'(\nu) &= \frac{\partial \nu'_{i\to j}}{\partial \nu_{k\to l}} \\ &= \frac{1 - \tanh^2(\beta\eta_{i\setminus j})}{1 - \tanh^2(\beta J_{ij})\tanh^2(\beta\eta_{i\setminus j})} \tanh(\beta J_{ij})\, \mathbf{1}_{\partial i\setminus j}(k)\, \delta_{i,l} \end{aligned} \tag{3.16}$$

where $\mathbf{1}$ is the indicator function (i.e., $\mathbf{1}_X(x) = 1$ if $x \in X$ and 0 otherwise) and δ the Kronecker delta function.

Let ν be a fixed point of parallel BP. We call ν *locally stable* if starting close enough to the fixed point, BP will converge to it. A fixed point ν is locally stable if all eigenvalues of the Jacobian $F'(\nu)$ lie inside the unit circle in the complex plane [Kuznetsov, 1988]:

$$\nu \text{ is locally stable} \iff \sigma\big(F'(\nu)\big) \subseteq \{\lambda \in \mathbb{C} : |\lambda| < 1\}, \tag{3.17}$$

where $\sigma(F')$ denotes the *spectrum* (set of eigenvalues) of the matrix F'. If at least one eigenvalue lies outside the unit circle, the fixed point is unstable.

3.3.3 Local stability conditions for damped, parallel BP

The BP equations can in certain cases lead to oscillatory behavior, which may be remedied by damping the update equations. This can be done by replacing the update map $F : \nu \mapsto \nu'$ by the convex combination $F_\epsilon := (1-\epsilon)F + \epsilon I$ of F and the identity I, for damping strength $0 \le \epsilon < 1$. Fixed points of F are also fixed points of F_ϵ and vice versa. The spectrum of the local stability matrix of the damped BP update mapping becomes:

$$\sigma(F'_\epsilon(\nu)) = (1-\epsilon)\sigma(F'(\nu)) + \epsilon.$$

In words, all eigenvalues of the local stability matrix *without damping* are simply interpolated with the value 1 for damped BP. It follows that the condition for (local) stability of a fixed point ν under *arbitrarily large* damping is given by

$$\begin{aligned} &\nu \text{ is stable under } F_\epsilon \text{ for some damping strength } \epsilon \in [0, 1) \\ &\quad\iff \sigma\big(F'(\nu)\big) \subseteq \{\lambda \in \mathbb{C} : \Re\lambda < 1\}, \end{aligned} \tag{3.18}$$

i.e., all eigenvalues of $F'(\nu)$ should have real part smaller than 1.

Note that conditions (3.17) and (3.18) do not depend on the chosen parameterization of the messages. In other words, the local stability of the BP fixed points does not depend on whether one uses μ messages or ν messages, or some other parameterization, i.e., the choice made in (3.9) has no influence on the results, but it does simplify the calculations.

3.3.4 Uniqueness of BP fixed points and convergence

The foregoing conditions are local and by themselves are not strong enough for drawing conclusions about global behavior, i.e., whether or not BP will converge for any initial set of messages.

In [Mooij and Kappen, 2005b] we have derived sufficient conditions for the uniqueness of the BP fixed point and convergence of undamped, parallel BP to the unique fixed point, irrespective of the initial messages. For the binary case, our result can be stated as follows:[2]

Theorem 3.1 *If the spectral radius*[3] *of the square matrix*

$$A_{i\to j,k\to l} := \tanh(\beta\,|J_{ij}|)\delta_{i,l}\mathbf{1}_{\partial i\setminus j}(k) \tag{3.19}$$

is strictly smaller than 1, undamped parallel BP converges to a unique fixed point, irrespective of the initial messages.

Proof. See Corollary 2.8. □

Note that the matrix A, and hence the sufficient condition, depends neither on the fields θ_i, nor on the sign of the weights J_{ij}.

These conditions are sufficient, but by no means necessary, as we will see in the next section. However, for ferromagnetic interactions without local fields, they are sharp, as we will prove later on. First we discuss some properties of the Bethe free energy that we will need in section 3.4.

[2] An equivalent result but formulated in terms of an algorithm was derived independently in [Ihler *et al.*, 2005a].

[3] The spectral radius $\rho(A)$ of a matrix A is defined as $\rho(A) := \sup|\sigma(A)|$, i.e., it is the largest absolute value of the eigenvalues of A.

3.3.5 Properties of the Bethe free energy for binary variables

For the case of binary variables, the Bethe free energy (3.4) can be parameterized in terms of the means $m_i = \mathbb{E}_{b_i}(x_i)$ and correlations $\chi_{ij} = \mathbb{E}_{b_{ij}}(x_i x_j)$; it becomes:

$$\begin{aligned} F_{Bethe}(m,\chi) = & -\beta \sum_{\{i,j\}\in\mathcal{E}} J_{ij}\chi_{ij} - \beta \sum_{i\in\mathcal{V}} \theta_i m_i \\ & + \sum_{i\in\mathcal{V}} (1-d_i) \sum_{x_i=\pm 1} S\left(\frac{1+m_i x_i}{2}\right) \\ & + \sum_{\{i,j\}\in\mathcal{E}} \sum_{x_i=\pm 1} \sum_{x_j=\pm 1} S\left(\frac{1+m_i x_i + m_j x_j + x_i x_j \chi_{ij}}{4}\right) \end{aligned} \tag{3.20}$$

where $S(x) := x \log x$. The constraints (3.5) can be formulated in terms of the means and correlations as:

$$\begin{aligned} & -1 \le m_i \le 1, \\ & -1 \le \chi_{ij} \le 1, \\ & 1 + m_i\sigma + m_j\sigma' + \chi_{ij}\sigma\sigma' \ge 0 \quad \text{for all } \sigma,\sigma' = \pm 1. \end{aligned}$$

The stationary points of the Bethe free energy (3.20) are the points where the derivative of (3.20) vanishes; this yields the following equations:

$$\begin{aligned} 0 = \frac{\partial F_{Bethe}}{\partial m_i} = & -\beta\theta_i + (1-d_i)\tanh^{-1} m_i + \\ & + \tfrac{1}{4} \sum_{j\in\partial i} \log \frac{(1+m_i+m_j+\chi_{ij})(1+m_i-m_j-\chi_{ij})}{(1-m_i+m_j-\chi_{ij})(1-m_i-m_j+\chi_{ij})}. \end{aligned} \tag{3.21}$$

$$0 = \frac{\partial F_{Bethe}}{\partial \chi_{ij}} = -\beta J_{ij} + \tfrac{1}{4} \log \frac{(1+m_i+m_j+\chi_{ij})(1-m_i-m_j+\chi_{ij})}{(1+m_i-m_j-\chi_{ij})(1-m_i+m_j-\chi_{ij})}. \tag{3.22}$$

The last equation has a unique solution χ_{ij} as a function of m_i and m_j [Welling and Teh, 2001].

From now on we consider the special case of vanishing local fields (i.e., $\theta_i = 0$) in the interest of simplicity. Note that in this case, the BP update equations (3.10) have a trivial fixed point, namely $\nu_{i\to j} = 0$. The corresponding beliefs have $m_i = 0$ and $\chi_{ij} = \tanh(\beta J_{ij})$, as follows directly from (3.14) and (3.15); of course, this also follows from (3.21) and (3.22). We call this fixed point the *paramagnetic fixed point* (or the *high-temperature fixed point* to emphasize that it exists if the temperature is high enough, i.e., for β small enough).

Whether the paramagnetic stationary point of the Bethe free energy is indeed a minimum depends on whether the Hessian of F_{Bethe} is positive-definite. The

Hessian at the paramagnetic stationary point is given by:

$$\begin{aligned}
\frac{\partial^2 F_{Bethe}}{\partial m_j \partial m_i} &= \delta_{ij}\left(1+\sum_{k\in\partial i}\frac{\chi_{ik}^2}{1-\chi_{ik}^2}\right)+M_{ij}\frac{-\chi_{ij}}{1-\chi_{ij}^2} =: U_{ij}, \qquad (3.23)\\
\frac{\partial^2 F_{Bethe}}{\partial m_k \partial \chi_{ij}} &= 0,\\
\frac{\partial^2 F_{Bethe}}{\partial \chi_{kl} \partial \chi_{ij}} &= \delta_{\{i,j\},\{k,l\}}\frac{1}{1-\chi_{ij}^2}.
\end{aligned}$$

The Hessian is of block-diagonal form; the χ-block is always positive-definite, hence the Hessian is positive-definite if and only if the the m-block (U_{ij}) is positive-definite. This depends on the weights J_{ij} and on the graph structure; for β small enough (i.e., high temperature), this is indeed the case. A consequence of the positive-definiteness of the Hessian of the Bethe free energy is that the approximate covariance matrix, given by U^{-1}, is also positive-definite.

3.4 Phase transitions

In this section we discuss various phase transitions that may occur, depending on the distribution of the weights J_{ij}. We take the local fields θ_i to be zero. Our usage of the term "phase transition" is somewhat inaccurate, since we actually mean the finite-N manifestations of the phase transition in the Bethe approximation and in the dynamical behavior of the BP algorithm, instead of the common usage of the word, which refers to the $N \to \infty$ behavior of the exact probability distribution. We believe that, at least for the ferromagnetic and spin-glass phase transitions, these different notions coincide in the $N \to \infty$ limit.

3.4.1 Ferromagnetic interactions

Consider the case of purely ferromagnetic interactions, by which we mean that all interactions J_{ij} are positive. In that case, the local BP stability matrix $F'(0)$ at the trivial fixed point, given by

$$F'(0) = \tanh(\beta J_{ij})\mathbf{1}_{\partial i \setminus j}(k)\delta_{i,l} \qquad (3.24)$$

is equal to the matrix A in Theorem 3.1. For high temperature (i.e., small β), the paramagnetic fixed point is locally stable, as is evident from (3.24). Theorem 3.1 guarantees that this is the only BP fixed point and that parallel undamped BP will converge to it. When we gradually lower the temperature (i.e., increase β), at a sudden point the paramagnetic BP fixed point generally becomes unstable. This seems to hold for all graphs that have more than one cycle. By a generalization of Perron's theorem (Theorem 3.3 in the appendix), the eigenvalue of the matrix $F'(0)$ (which has positive entries) with the largest absolute value is actually positive. This property of the spectrum can be clearly seen in figure 3.1.I(a), where most

eigenvalues are distributed in a roughly circular form, except for one outlier on the positive real axis. Thus the onset of instability of the paramagnetic BP fixed point coincides with this outlier crossing the complex unit circle; the paramagnetic fixed point bifurcates and two new stable fixed points arise, describing the two ferromagnetic states. Since $A = F'(0)$, we conclude that the sufficient condition in Theorem 3.1 for convergence to a unique fixed point is sharp in this case.

At high temperature, the corresponding stationary point of the Bethe free energy is a minimum. However, as illustrated in figure 3.1.II(a), at a certain critical temperature the Hessian is no longer positive-definite. In the appendix, we prove the following theorem:

Theorem 3.2 *For $J_{ij} \geq 0$ and $\theta_i = 0$, the critical temperature at which the paramagnetic Bethe free energy minimum disappears is equal to the critical temperature at which the paramagnetic BP fixed point becomes unstable.*

Proof. See appendix. □

Beyond the transition temperature, BP converges to either of the two new fixed points describing the two ferromagnetic phases. As can be seen in figure 3.1.III(a), the number of BP iterations needed for convergence has a peak precisely at the critical temperature; far from the phase transition, BP converges rapidly to a stable fixed point.

3.4.2 Antiferromagnetic interactions

For purely antiferromagnetic interactions, i.e., all $J_{ij} < 0$, the situation is different. Again, for high temperature, the paramagnetic fixed point is the unique fixed point, is locally stable and has the complete message space as an attractor. Since the local stability matrix $F'(0)$ is exactly the same as in the ferromagnetic case, except for the minus sign (as can be seen in figure 3.1.I(b)), the local stability of the trivial fixed point is invariant under a sign change $J \mapsto -J$. Hence the paramagnetic fixed point becomes locally unstable for undamped BP exactly at the same temperature as in the ferromagnetic case (for fixed weight strengths $|J_{ij}|$). However, the spectral radius of $F'(0)$ is now determined by a negative eigenvalue. Hence in this case damping helps to some extent. Empirically, we find that changing the update scheme from parallel to sequential also helps, as illustrated by the dotted line in figure 3.1.III(b). Note that the temperature where sequential BP stops converging roughly coincides with the minimum of the smallest eigenvalue of U (compare figures 3.1.II(b) and 3.1.III(b)). This observation seems to be generic, i.e., not just a coincidence for the particular instance in figure 3.1. We have no theoretical explanation for this at the moment, but it might be possible to get such an explanation by relating U with $F'(0)$, using a technique similar to the one applied in the proof of Theorem 3.2 given in the appendix.

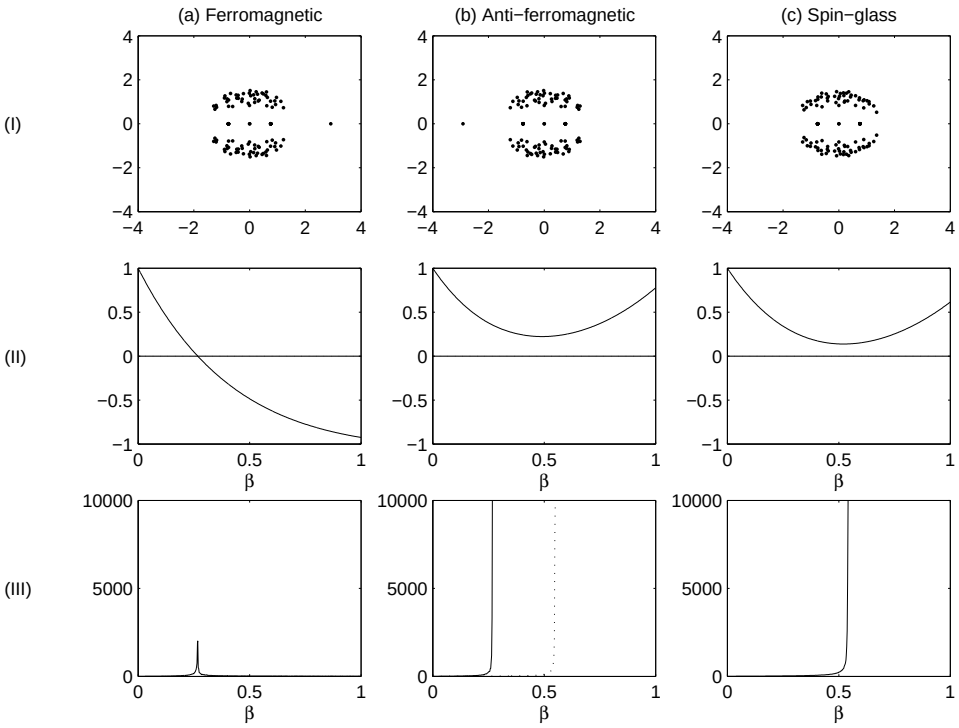


Figure 3.1: From top to bottom: (I) spectrum of the local BP stability matrix F' at the trivial fixed point $\nu = 0$, for $\beta = 1$; (II) minimal eigenvalue of $U_{ij} = \partial^2 F_{Bethe}/\partial m_i \partial m_j$ at the paramagnetic solution, as a function of inverse temperature β; (III) number of undamped, parallel BP iterations needed for convergence as a function of inverse temperature β (dotted line in antiferromagnetic case shows the number of iterations for a sequential update scheme). From left to right: (a) ferromagnetic interactions $J = M$ (b) antiferromagnetic interactions $J = -M$; (c) spin-glass interactions $J = \pm M$ with equal probability for positive or negative interaction. The underlying graph $\mathcal{G}$ is a random graph with Poissonian degree distribution, $N = 50$ and average degree $d = 4$; the local fields are zero.

3.4.3 Spin-glass interactions

Now consider spin-glass interactions, i.e., all J_{ij} are distributed around 0 such that $\langle J_{ij} \rangle \approx 0$. This case is illustrated in figure 3.1(c). Here the eigenvalues of the local stability matrix are distributed in a roughly circular form, without an outlier with a large absolute value. Note the surprising similarity between the spectra in the different cases; we have no explanation for this similarity, nor for the roughly circular form of the distribution of the majority of the eigenvalues.

Although the paramagnetic Bethe free energy minimum generally does not disappear when lowering the temperature, BP does not converge anymore once the

trivial fixed point becomes unstable, despite the possible existence of other, stable, fixed points. Neither damping nor changing the update scheme seems to help in this case. Empirically we find that the temperature at which the trivial BP fixed point becomes locally unstable roughly coincides with the temperature at which the lowest eigenvalue of U attains its minimal value [Mooij and Kappen, 2005c]. Again, we have no theoretical explanation for this observation.

3.5 Estimates of phase-transition temperatures

In this section we estimate the critical temperatures corresponding to the onset of instability of the BP paramagnetic fixed point (which we discussed qualitatively in the previous section) for a random graph with random interactions. The method is closely related to the cavity method at the replica-symmetric level (see e.g., [Mézard *et al.*, 1987; Mézard and Parisi, 2001; Wemmenhove *et al.*, 2004]). A similar analysis of the stability of the BP paramagnetic fixed point has been done by Kabashima [Kabashima, 2003]; however, the results reported in that work are limited to the case of infinite connectivity (i.e., the limit $N \to \infty$, $d \to \infty$). In this case, the results turn out to be identical to the condition of replica symmetry breaking (the "AT line") derived by de Almeida and Thouless [1978]. The analysis we present below essentially extends the analysis of Kabashima [2003] to the larger class of arbitrary degree distribution random graphs, which includes Erdős-Rényi graphs (with Poissonian degree distribution, as well as fixed degree random graphs) and power-law graphs (which have power-law degree distributions), amongst others.

3.5.1 Random graphs with arbitrary degree distributions

We consider arbitrary degree distribution random graphs [Newman *et al.*, 2001]. This class of random graphs has a prescribed expected degree distribution $\mathbb{P}(d)$; apart from that they are completely random. Given an expected degree distribution $\mathbb{P}(d)$ and the number of nodes N, a particular sample of the corresponding ensemble of random graphs can be constructed as follows: for each node i, independently draw an expected degree δ_i from the degree distribution $\mathbb{P}(d)$. Then, for each pair of nodes (i, j), independently connect them with probability $\delta_i \delta_j / \sum_i \delta_i$; the expected degree of node i is then indeed $\langle d_i \rangle = \delta_i$. We define the *average degree* $\langle d \rangle := \sum_d \mathbb{P}(d) d$ and the second moment $\langle d^2 \rangle := \sum_d \mathbb{P}(d) d^2$.

We consider the case of vanishing local fields (i.e., $\theta_i = 0$) and draw the weights J_{ij} independently from some probability distribution $\mathbb{P}(J)$. We also assume that the weights are independent of the graph structure.

3.5.2 Estimating the PA-FE transition temperature

Assume $\mathbb{P}(d)$ to be given and N to be large. Assume that v is an eigenvector with eigenvalue 1 of $F'(0)$, the Jacobian of the parallel BP update at the paramagnetic

fixed point $\nu = 0$. Using (3.16) and writing ij instead of $i \to j$ for brevity, we have:

$$v_{ij} = \sum_{kl} \left(F'(0)\right)_{ij,kl} v_{kl} = \tanh(\beta J_{ij}) \sum_{k \in \partial i \setminus j} v_{ki}. \tag{3.25}$$

Consider an arbitrary spin i; conditional on the degree d_i of that spin, we can calculate the expected value of v_{ij} as follows:

$$\begin{aligned}
\mathbb{E}\left(v_{ij} \,|\, d_i\right) &= \mathbb{E}\left(\tanh(\beta J_{ij}) \textstyle\sum_{k \in \partial i \setminus j} v_{ki} \,|\, d_i\right) && \text{(3.26a)}\\
&= \mathbb{E}\left(\tanh(\beta J_{ij})\right) \mathbb{E}\left(\textstyle\sum_{k \in \partial i \setminus j} v_{ki} \,|\, d_i\right) && \text{(3.26b)}\\
&= \langle \tanh \beta J \rangle \, (d_i - 1) \sum_{d_k} \mathbb{P}(d_k \,|\, d_i, k \in \partial i) \mathbb{E}\left(v_{ki} \,|\, d_i, d_k\right) && \text{(3.26c)}\\
&\approx \langle \tanh \beta J \rangle \, (d_i - 1) \sum_{d_k} \mathbb{P}(d_k \,|\, d_i, k \in \partial i) \mathbb{E}\left(v_{ki} \,|\, d_k\right) && \text{(3.26d)}
\end{aligned}$$

using, subsequently: (a) equation (3.25); (b) the independence of the weights from the graph structure; (c) conditioning on the degree d_k of spin k and the equivalence of the various $k \in \partial i \setminus j$; and finally, (d) neglecting the correlation between v_{ki} and d_i, given d_k. We have no formal argument for the validity of this approximation, but the result accurately describes the outcomes of numerical experiments.

For arbitrary degree distribution random graphs, the probability of d_k given the degree d_i and the fact that k is a neighbor of i is given by (see [Newman *et al.*, 2001]):

$$\mathbb{P}(d_k \,|\, d_i, k \in \partial i) = \frac{d_k \mathbb{P}(d_k)}{\langle d \rangle}. \tag{3.27}$$

Hence we obtain the relation

$$\mathbb{E}\left(v_{ij} \,|\, d_i\right) = \langle \tanh \beta J \rangle \, (d_i - 1) \sum_{d_k} \frac{d_k \mathbb{P}(d_k)}{\langle d \rangle} \mathbb{E}\left(v_{ki} \,|\, d_k\right)$$

A self-consistent nontrivial solution of these equations is

$$\mathbb{E}\left(v_{ij} \,|\, d_i\right) \propto (d_i - 1),$$

provided that

$$1 = \langle \tanh \beta J \rangle \left(\frac{\langle d^2 \rangle}{\langle d \rangle} - 1 \right). \tag{3.28}$$

This gives us the critical temperature at which the paramagnetic–ferromagnetic phase transition occurs, or in other words, where the paramagnetic BP fixed point undergoes a pitchfork bifurcation. This result is identical to the one obtained by the replica method in the replica-symmetric setting [Leone *et al.*, 2002] and to the one found by applying the cavity method [Wemmenhove *et al.*, 2004], as expected. Figure 3.2 illustrates the estimate; note that the accuracy is quite high already for low N ($N = 50$ in this case), for higher N it becomes even better.

Extending the analysis to the case of non-vanishing local fields does not appear to be straightforward, since in that case the value of the fixed point ν is not known. However, since the elements of $F'(0)$ are upper bounds for the elements of $F'(\nu)$, we can at least conclude qualitatively that in the case of non-vanishing local fields, the transition temperature will be lower.

3.5.3 The antiferromagnetic case

This is similar to the ferromagnetic case, however the eigenvalue is now -1 instead of $+1$. This yields the following equation for the transition temperature:

$$1 = \langle \tanh(-\beta J) \rangle \left(\frac{\langle d^2 \rangle}{\langle d \rangle} - 1 \right). \tag{3.29}$$

As can be seen in figure 3.2, again the prediction turns out to be quite accurate.

3.5.4 Estimating the PA-SG transition temperature

For the paramagnetic–spin-glass phase transition, we can perform a similar calculation, now assuming that v is an eigenvector with eigenvalue λ on the complex unit circle:

$$\begin{aligned}
\mathbb{E}\left(|v_{ij}|^2 \mid d_i \right) &= \mathbb{E}\left(|\tanh(\beta J_{ij})|^2 \left| \textstyle\sum_{k \in \partial i \setminus j} v_{ki} \right|^2 \mid d_i \right) \\
&= \left\langle \tanh^2(\beta J) \right\rangle \mathbb{E}\left(\left| \textstyle\sum_{k \in \partial i \setminus j} v_{ki} \right|^2 \mid d_i \right) \\
&\approx \left\langle \tanh^2(\beta J) \right\rangle \mathbb{E}\left(\textstyle\sum_{k \in \partial i \setminus j} |v_{ki}|^2 \mid d_i \right) \\
&\approx \left\langle \tanh^2(\beta J) \right\rangle (d_i - 1) \sum_{d_k} \mathbb{P}(d_k \mid d_i, k \in \partial i) \mathbb{E}\left(|v_{ki}|^2 \mid d_k \right),
\end{aligned}$$

where, in addition to the assumptions in the PA-FE case, we assumed that the correlations between the various v_{ki} can be neglected. Again, we can only motivate this assumption in that it appears to give correct results.

Using relation (3.27), we find a nontrivial self-consistent solution

$$\mathbb{E}\left(|v_{ij}|^2 \mid d_i \right) \propto (d_i - 1),$$

if the following equation holds:

$$1 = \left\langle \tanh^2(\beta J) \right\rangle \left(\frac{\langle d^2 \rangle}{\langle d \rangle} - 1 \right). \tag{3.30}$$

This result is again identical to the one obtained by the cavity method [Wemmenhove *et al.*, 2004], as expected. As illustrated in figure 3.2 (the dashed line), the accuracy is somewhat less than that of the ferromagnetic transition, but is nevertheless quite good, even for $N = 50$.

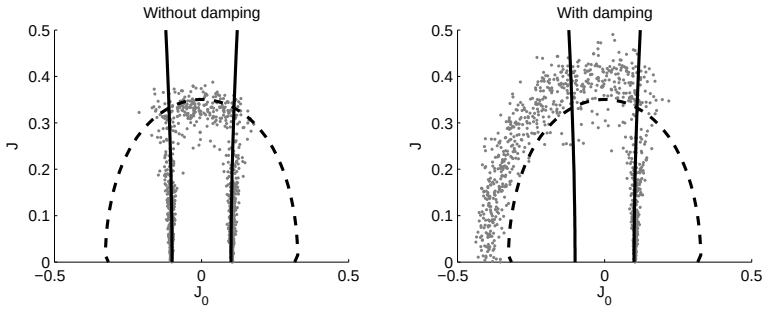


Figure 3.2: Onset of instability of the paramagnetic BP fixed point, for random graphs with $N = 50$ and a Poissonian degree distribution with $d = 10$. The weights J_{ij} are independently drawn from a Gaussian distribution with mean J_0 and variance J^2. The solid thick lines show the expected value for the (anti)ferromagnetic transitions (3.28) and (3.29), the dashed thick line for the spin-glass transition (3.30). The dots show for individual instances at which temperature the paramagnetic fixed point becomes unstable, for undamped BP (left) and for damped BP (right). The lines in the right graph (the damped case) are for reference only, they should not be interpreted as theoretical predictions, except for the ferromagnetic transition (the solid line on the right-hand side).

For completeness we would like to state that the numerical results reported in [Mooij and Kappen, 2005c], in which we numerically studied the behavior of the lowest eigenvalue of U, are accurately described by the predictions (3.28) and (3.30), which supports the hypothesis that these notions coincide in the $N \to \infty$ limit.

3.6 Conclusions

We have derived conditions for the local stability of parallel BP fixed points, both in the undamped and damped case for binary variables with pairwise interactions. We have shown how these relate to the sufficient conditions for uniqueness of the BP fixed point and convergence to this fixed point. In particular, we have shown that these sufficient conditions are sharp in the ferromagnetic case, exactly describing the pitchfork bifurcation of the paramagnetic fixed point into two ferromagnetic fixed points. For undamped BP, the local stability of the paramagnetic fixed point (for vanishing local fields) is invariant under a sign change of the interactions. For antiferromagnetic interactions, parallel undamped BP stops converging at the PA-FE transition temperature. Damping or using a sequential update scheme remedies this defect. However, although the paramagnetic minimum of the Bethe free energy does not disappear, the trivial fixed point becomes locally unstable even for damped BP at roughly the PA-SG transition temperature. Finally, for interactions that are dominantly of the spin-glass type, using damping only marginally extends the domain of convergence of BP.

We estimated the PA-FE transition temperature and the PA-SG transition temperature for arbitrary degree distribution random graphs. The results are in good agreement with numerical simulations. How this relates to the AT line is an open question and beyond the scope of this work.

We believe that the case that we have considered in detail in this work, namely vanishing local fields $\theta_i = 0$, is actually the worst-case scenario: numerically it turns out that adding local fields helps BP to converge more quickly. We have no proof for this conjecture at the moment; the local fields make an analytical analysis more difficult and we have not yet been able to extend the analysis to this more general setting. We leave the generalization to nonzero local fields as possible future work.

Acknowledgments

We thank Bastian Wemmenhove and the anonymous reviewer for valuable comments on a draft of this paper.

3.A Proof of Theorem 3.2

For a square matrix A, we write $A \geq 0$ iff all entries of A are nonnegative. $\sigma(A)$ is the set of all eigenvalues of A, $\rho(A)$ is the spectral radius of A, i.e., $\rho(A) := \max|\sigma(A)|$. We will use the following generalization of Perron's theorem:

Theorem 3.3 *If $A \geq 0$, then the spectral radius $\rho(A) \in \sigma(A)$ and there exists an associated eigenvector $v \geq 0$ such that $Av = \rho(A)v$.*

Proof. See [Meyer, 2000, p. 670]. □

Applying this theorem to the matrix A defined in (3.19), we deduce the existence of an eigenvector $v \geq 0$ with $Av = \rho(A)v$. Writing $t_{ij} := \tanh(\beta|J_{ij}|)$ and $\lambda := \rho(A)$, we derive:

$$\begin{aligned} v_{ij} &= \lambda^{-1} t_{ij} \left(\sum_{k \in \partial i} v_{ki} - v_{ji} \right) \\ &= \lambda^{-1} t_{ij} \left(\sum_{k \in \partial i} v_{ki} - \lambda^{-1} t_{ji} \left(\sum_{k \in \partial j} v_{kj} - v_{ij} \right) \right). \end{aligned}$$

Defining $V_i := \sum_{k \in \partial i} v_{ki}$, we obtain by summing over $i \in \partial j$:

$$V_j = \sum_{i \in \partial j} \lambda \frac{t_{ij}}{\lambda^2 - t_{ij} t_{ji}} V_i - \sum_{i \in \partial j} \frac{t_{ij} t_{ji}}{\lambda^2 - t_{ij} t_{ji}} V_j,$$

i.e., V is an eigenvector with eigenvalue 1 of the matrix

$$M_{ij} \frac{\rho(A) \tanh(\beta|J_{ij}|)}{\rho(A)^2 - \tanh^2(\beta|J_{ij}|)} - \delta_{ij} \sum_{k \in \partial i} \frac{\tanh^2(\beta|J_{ik}|)}{\rho(A)^2 - \tanh^2(\beta|J_{ik}|)}. \tag{3.31}$$

Now, if all J_{ij} are positive, and if $\rho(A) = 1$, this matrix is exactly $I-U$, where U_{ij} is defined in (3.23). Hence, since in this case $A = F'(0)$, the critical temperature at which the paramagnetic BP fixed point becomes unstable coincides with the matrix $I - U$ having an eigenvalue 1, or in other words U having eigenvalue 0. Thus the onset of instability of the paramagnetic BP fixed point in this case exactly coincides with the disappearance of the paramagnetic Bethe free energy minimum.

Chapter 4

Loop Corrections

We propose a method to improve approximate inference methods by correcting for the influence of loops in the graphical model. The method is a generalization and alternative implementation of a recent idea from Montanari and Rizzo [2005]. It is applicable to arbitrary factor graphs, provided that the size of the Markov blankets is not too large. It consists of two steps: (i) an approximate inference method, for example, Belief Propagation, is used to approximate cavity distributions for each variable (i.e., probability distributions on the Markov blanket of a variable for a modified graphical model in which the factors involving that variable have been removed); (ii) all cavity distributions are improved by a message-passing algorithm that cancels out approximation errors by imposing certain consistency constraints. This Loop Correction (LC) method usually gives significantly better results than the original, uncorrected, approximate inference algorithm that is used to estimate the effect of loops. Indeed, we often observe that the loop-corrected error is approximately the square of the error of the uncorrected approximate inference method. In this chapter, we compare different variants of the Loop Correction method with other approximate inference methods on a variety of graphical models, including "real world" networks, and conclude that the LC method generally obtains the most accurate results.

4.1 Introduction

In recent years, much research has been done in the field of approximate inference on graphical models. One of the goals is to obtain accurate approximations of marginal probabilities of complex probability distributions defined over many variables, using

This chapter is based on [Mooij and Kappen, 2007a], the extended version of [Mooij *et al.*, 2007].

limited computation time and memory. This research has led to a large number of approximate inference methods. Apart from sampling ("Monte Carlo") methods, there is a large number of "deterministic" approximate inference methods, such as variational methods (for example, the Mean Field method [Parisi, 1988]), and a family of algorithms that are in some way related to the highly successful Belief Propagation (BP) algorithm [Pearl, 1988]. BP is also known as the "Sum-Product Algorithm" [Kschischang *et al.*, 2001] and as "Loopy Belief Propagation" and is directly related to the Bethe approximation [Bethe, 1935; Yedidia *et al.*, 2005] from statistical physics. It is well-known that Belief Propagation yields exact results if the graphical model is a tree, or, more generally, if each connected component is a tree. If the graphical model does contain loops, BP can still yield surprisingly accurate results using little computation time. However, if the influence of loops is large, the approximate marginals calculated by BP can have large errors and the quality of the BP results may not be satisfactory.

One way to correct for the influence of short loops is to increase the cluster size of the approximation, using the Cluster Variation Method (CVM) [Pelizzola, 2005] or other region-based approximation methods [Yedidia *et al.*, 2005]. These methods are related to the Kikuchi approximation [Kikuchi, 1951], a generalization of the Bethe approximation using larger clusters. Algorithms for calculating the CVM and related region-based approximation methods are Generalized Belief Propagation (GBP) [Yedidia *et al.*, 2005] and double-loop algorithms that have guaranteed convergence [Yuille, 2002; Heskes *et al.*, 2003]. By choosing the (outer) clusters such that they subsume as many loops as possible, the BP results can be improved. However, choosing a good set of outer clusters is highly nontrivial, and in general this method will only work if the clusters do not have many intersections, or in other words, if the loops do not have many intersections [Welling *et al.*, 2005].

Another method that corrects for loops to a certain extent is TreeEP [Minka and Qi, 2004], a special case of Expectation Propagation (EP) [Minka, 2001]. TreeEP does exact inference on the base tree, a subgraph of the graphical model which has no loops, and approximates the other interactions. This corrects for the loops that consist of part of the base tree and exactly one additional factor. TreeEP yields good results if the graphical model is dominated by the base tree, which is the case in very sparse models. However, loops that consist of two or more interactions that are not part of the base tree are approximated in a similar way as in BP. Hence, for denser models, the improvement of TreeEP over BP usually diminishes.

In this chapter we propose a method that takes into account *all* the loops in the graphical model in an approximate way and therefore obtains more accurate results in many cases. Our method is a variation on the theme introduced by Montanari and Rizzo [2005]. The basic idea is to first estimate the "cavity distributions" of all variables and subsequently improve these estimates by canceling out errors using certain consistency constraints. A *cavity distribution* of some variable is the probability distribution on its Markov blanket (all its neighboring variables) for a modified graphical model, in which all factors involving that variable have been

removed. The removal of the factors breaks all the loops in which that variable takes part. This allows an approximate inference algorithm to estimate the strength of these loops in terms of effective interactions or correlations between the variables of the Markov blanket. Then, the influence of the removed factors is taken into account, which yields accurate approximations to the probability distributions of the original graphical model. Even more accuracy is obtained by imposing certain consistency relations between the cavity distributions, which results in a cancellation of errors to some extent. This error cancellation is done by a message-passing algorithm which can be interpreted as a generalization of BP in case the factor graph does not contain short loops of four nodes; indeed, assuming that the cavity distributions factorize (which they do in case there are no loops), the BP results are obtained. On the other hand, using better estimates of the effective interactions in the cavity distributions yields accurate loop-corrected results.

Although the basic idea underlying our method is very similar to that described in [Montanari and Rizzo, 2005], the alternative implementation that we propose here offers two advantages. Most importantly, it is directly applicable to arbitrary factor graphs, whereas the original method has only been formulated for the rather special case of graphical models with binary variables and pairwise factors, which excludes, for example, many interesting Bayesian networks. Furthermore, our implementation appears to be more robust and also gives improved results for relatively strong interactions, as will be shown numerically.

This chapter is organized as follows. First we explain the theory behind our proposed method and discuss the differences with the original method by Montanari and Rizzo [2005]. Then we report extensive numerical experiments regarding the quality of the approximation and the computation time, where we compare with other approximate inference methods. Finally, we discuss the results and state conclusions.

4.2 Theory

In this work, we consider graphical models such as Markov random fields and Bayesian networks. We use the general factor graph representation since it allows for formulating approximate inference algorithms in a unified way [Kschischang *et al.*, 2001]. In the next subsection, we introduce our notation and basic definitions.

4.2.1 Graphical models and factor graphs

Consider N discrete random variables $(x_i)_{i\in\mathcal{V}}$ with $\mathcal{V} := \{1,\dots,N\}$. Each variable x_i takes values in a discrete domain $\mathcal{X}_i$. We will use the following multi-index notation: for any subset $I \subseteq \mathcal{V}$, we write $x_I := (x_{i_1}, x_{i_2}, \dots, x_{i_m}) \in \mathcal{X}_I := \mathcal{X}_{i_1} \times \mathcal{X}_{i_2} \times \cdots \times \mathcal{X}_{i_m}$ if $I = \{i_1, i_2, \dots, i_m\}$ and $i_1 < i_2 < \dots i_m$. We consider a probability

distribution over $x = (x_1, \ldots, x_N)$ that can be written as a product of factors (also called "interactions") ψ_I:

$$\mathbb{P}(x) = \frac{1}{Z} \prod_{I \in \mathcal{F}} \psi_I(x_{N_I}), \qquad Z = \sum_x \prod_{I \in \mathcal{F}} \psi_I(x_{N_I}). \tag{4.1}$$

For each factor index $I \in \mathcal{F}$, there is an associated subset $N_I \subseteq \mathcal{V}$ of variable indices and the factor ψ_I is a nonnegative function $\psi_I : \mathcal{X}_{N_I} \to [0, \infty)$. For a Bayesian network, the factors are (conditional) probability tables. In case of Markov random fields, the factors are often called potentials (not to be confused with statistical physics terminology, where "potential" refers to $-\frac{1}{\beta} \log \psi_I$ instead, with β the inverse temperature). Henceforth, we will refer to a triple $\big(\mathcal{V}, \mathcal{F}, (\psi_I)_{I \in \mathcal{F}}\big)$ that satisfies the description above as a discrete *graphical model* (or *network*).

In general, the normalizing constant Z is not known and exact computation of Z is infeasible, due to the fact that the number of terms to be summed is exponential in N. Similarly, computing marginal distributions $\mathbb{P}(x_J)$ of $\mathbb{P}(x_\mathcal{V})$ for subsets of variables $J \subseteq \mathcal{V}$ is intractable in general. In this chapter, we focus on the task of accurately approximating single-variable marginals $\mathbb{P}(x_i) = \sum_{x_{\mathcal{V} \setminus \{i\}}} \mathbb{P}(x_\mathcal{V})$.

We can represent the structure of the probability distribution (4.1) using a *factor graph*. This is a bipartite graph, consisting of *variable nodes* $i \in \mathcal{V}$ and *factor nodes* $I \in \mathcal{F}$, with an edge between i and I if and only if $i \in N_I$, that is, if the factor ψ_I depends on x_i. We will represent factor nodes visually as rectangles and variable nodes as circles. See figure 4.1(a) for an example of a factor graph. The neighbors of a factor node $I \in \mathcal{F}$ are precisely the variables N_I, and the neighbors N_i of a variable node $i \in \mathcal{V}$ are the factors that depend on that variable, i.e., $N_i := \{I \in \mathcal{F} : i \in N_I\}$. Further, we define for each variable $i \in \mathcal{V}$ the set $\Delta i := \bigcup_{I \in N_i} N_I$ consisting of all variables that appear in some factor in which variable i participates, and the set $\partial i := \Delta i \setminus \{i\}$, the *Markov blanket* of i.

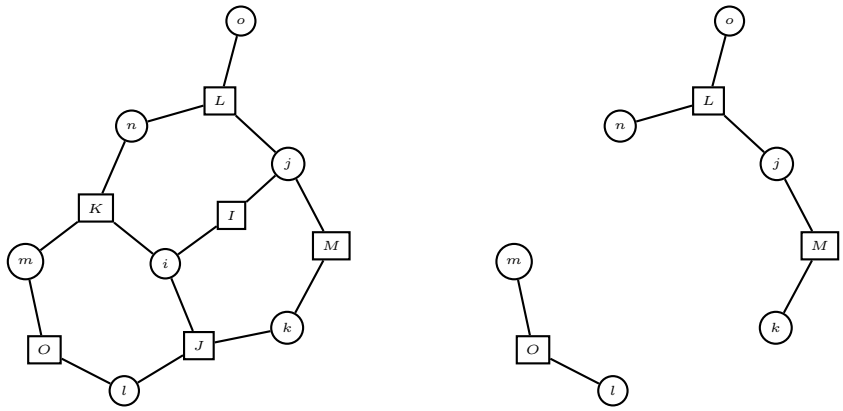
In the following, we will often abbreviate the set theoretical notation $X \setminus Y$ (i.e., all elements in X that are not in Y) by $\setminus Y$ if it is obvious from the context what the set X is. Also, we will write $X \setminus y$ instead of $X \setminus \{y\}$. Further, we will use lowercase for variable indices and uppercase for factor indices. For convenience, we will define for any subset $F \subset \mathcal{F}$ the product Ψ_F of the corresponding factors $\{\psi_I : I \in F\}$:

$$\Psi_F(x_{(\bigcup_{I \in F} N_I)}) := \prod_{I \in F} \psi_I(x_{N_I}).$$

4.2.2 Cavity networks and loop corrections

The notion of a *cavity* stems from statistical physics, where it was used originally to calculate properties of random ensembles of certain graphical models [Mézard *et al.*, 1987]. A cavity is obtained by removing one variable from the graphical model, together with all the factors in which that variable participates.

In our context, we define cavity networks as follows (see also figure 4.1):



(a) Original factor graph (b) Cavity graph of i

Figure 4.1: (a) Original factor graph, with variables nodes $\mathcal{V} = \{i, j, k, l, m, n, o\}$ and factor nodes $\mathcal{F} = \{I, J, K, L, M, O\}$ and corresponding probability distribution $\mathbb{P}(x) = \frac{1}{Z}\psi_L(x_j, x_n, x_o)\psi_I(x_i, x_j)\psi_M(x_j, x_k)\psi_K(x_i, x_m, x_n)\psi_J(x_i, x_k, x_l)\psi_O(x_l, x_m)$; (b) Factor graph corresponding to the cavity network of variable i, obtained by removing variable i and the factor nodes that contain i (i.e., I, J and K). The Markov blanket of i is $\partial i = \{j, k, l, m, n\}$. The cavity distribution $Z^{\setminus i}(x_{\partial i})$ is the (unnormalized) marginal on $x_{\partial i}$ of the probability distribution corresponding to the cavity graph (b), i.e., $Z^{\setminus i}(x_j, x_k, x_l, x_m, x_n) = \sum_{x_o} \psi_L(x_j, x_n, x_o)\psi_M(x_j, x_k)\psi_O(x_l, x_m)$.

Definition 4.1 *Given a graphical model* $\big(\mathcal{V}, \mathcal{F}, (\psi_I)_{I\in\mathcal{F}}\big)$ *and a variable* $i \in \mathcal{V}$*, the* cavity network *of variable* i *is defined as the graphical model obtained by removing* i *and all factors depending on* i*, i.e.,* $\big(\mathcal{V} \setminus i, \mathcal{F} \setminus N_i, (\psi_I)_{I\in\mathcal{F}\setminus N_i}\big)$.

The probability distribution corresponding to the cavity network of variable i is thus proportional to:

$$\Psi_{\setminus N_i}(x_{\setminus i}) = \prod_{\substack{I\in\mathcal{F}\\ i\notin I}} \psi_I(x_{N_I}).$$

Summing out all the variables, except for the neighbors ∂i of i, gives what we will call the *cavity distribution*:

Definition 4.2 *Given a graphical model* $\big(\mathcal{V}, \mathcal{F}, (\psi_I)_{I\in\mathcal{F}}\big)$ *and a variable* $i \in \mathcal{V}$*, the* cavity distribution *of* i *is*

$$Z^{\setminus i}(x_{\partial i}) := \sum_{x_{\setminus \Delta i}} \Psi_{\setminus N_i}(x_{\setminus i}). \tag{4.2}$$

Thus the cavity distribution of i is proportional to the marginal of the cavity network of i on the Markov blanket ∂i. The cavity distribution describes the *effective* interactions (or correlations) induced by the cavity network on the neighbors ∂i of

variable i. Indeed, from equations (4.1) and (4.2) and the trivial observation that $\Psi_{\mathcal{F}} = \Psi_{N_i}\Psi_{\setminus N_i}$ we conclude:

$$\mathbb{P}(x_{\Delta i}) \propto Z^{\setminus i}(x_{\partial i})\Psi_{N_i}(x_{\Delta i}). \tag{4.3}$$

Thus, given the cavity distribution $Z^{\setminus i}(x_{\partial i})$, one can calculate the marginal distribution of the original graphical model on $x_{\Delta i}$, provided that the cardinality of $\mathcal{X}_{\Delta i}$ is not too large.

In practice, exact cavity distributions are not known, and the only way to proceed is to use approximate cavity distributions. Given some approximate inference method (e.g., BP), there are two ways to calculate $\mathbb{P}(x_{\Delta i})$: either use the method to approximate $\mathbb{P}(x_{\Delta i})$ directly, or use the method to approximate $Z^{\setminus i}(x_{\partial i})$ and use equation (4.3) to obtain an approximation to $\mathbb{P}(x_{\Delta i})$. The latter approach generally gives more accurate results, since the complexity of the cavity network is less than that of the original network. In particular, the cavity network of variable i contains no loops involving that variable, since all factors in which i participates have been removed (e.g., the loop $i - J - l - O - m - K - i$ in the original network, figure 4.1(a), is not present in the cavity network, figure 4.1(b)). Thus the latter approach to calculating $\mathbb{P}(x_{\Delta i})$ takes into account loops involving variable i, although in an approximate way. It does not, however, take into account the other loops in the original graphical model. The basic idea of the loop correction approach of Montanari and Rizzo [2005] is to use the latter approach for all variables in the network, but to adjust the approximate cavity distributions in order to cancel out approximation errors before (4.3) is used to obtain the final approximate marginals. This approach takes into account *all* the loops in the original network, in an approximate way.

This basic idea can be implemented in several ways. Here we propose an implementation which we will show to have certain advantages over the original implementation proposed in [Montanari and Rizzo, 2005]. In particular, it is directly applicable to arbitrary factor graphs with variables taking an arbitrary (discrete) number of values and factors that may contain zeros and depend on an arbitrary number of variables. In the remaining subsections, we will first discuss our proposed implementation in detail. In section 4.2.6 we will discuss differences with the original approach.

4.2.3 Combining approximate cavity distributions to cancel out errors

Suppose that we have obtained an initial approximation $\zeta_0^{\setminus i}(x_{\partial i})$ of the (exact) cavity distribution $Z^{\setminus i}(x_{\partial i})$, for each $i \in \mathcal{V}$. Let $i \in \mathcal{V}$ and consider the approximation error of the cavity distribution of i, that is, the exact cavity distribution of i divided by its approximation:

$$\frac{Z^{\setminus i}(x_{\partial i})}{\zeta_0^{\setminus i}(x_{\partial i})}.$$

In general, this is an arbitrary function of the variables $x_{\partial i}$. However, we will *approximate* the error as a product of factors defined on small subsets of ∂i in the following way:

$$\frac{Z^{\backslash i}(x_{\partial i})}{\zeta_0^{\backslash i}(x_{\partial i})} \approx \prod_{I \in N_i} \phi_I^{\backslash i}(x_{N_I \backslash i}).$$

Thus we assume that the approximation error lies near a submanifold parameterized by the error factors $(\phi_I^{\backslash i}(x_{N_I \backslash i}))_{I \in N_i}$. If we were able to calculate these error factors, we could improve our initial approximation $\zeta_0^{\backslash i}(x_{\partial i})$ by replacing it with the product

$$\zeta^{\backslash i}(x_{\partial i}) := \zeta_0^{\backslash i}(x_{\partial i}) \prod_{I \in N_i} \phi_I^{\backslash i}(x_{N_I \backslash i}) \approx Z^{\backslash i}(x_{\partial i}). \tag{4.4}$$

Using (4.3), this would then yield an improved approximation of $\mathbb{P}(x_{\Delta i})$.

It turns out that the error factors can indeed be calculated by exploiting the redundancy of the information in the initial cavity approximations $(\zeta_0^{\backslash i})_{i \in \mathcal{V}}$. The fact that all $\zeta^{\backslash i}$ provide approximations to marginals of the *same* probability distribution $\mathbb{P}(x)$ via (4.3) can be used to obtain consistency constraints. The number of constraints obtained in this way is usually enough to solve for the unknown error factors $(\phi_I^{\backslash i}(x_{N_I \backslash i}))_{i \in \mathcal{V}, I \in N_i}$.

Here we propose the following consistency constraints. Let $Y \in \mathcal{F}$, $i \in N_Y$ and $j \in N_Y$ with $i \neq j$ (see also figure 4.2). Consider the graphical model $\big(\mathcal{V}, \mathcal{F} \setminus Y, (\psi_I)_{I \in \mathcal{F} \setminus Y}\big)$ that is obtained from the original graphical model by removing factor ψ_Y. The product of all factors (except ψ_Y) obviously satisfies:

$$\Psi_{\backslash Y} = \Psi_{N_i \backslash Y} \Psi_{\backslash N_i} = \Psi_{N_j \backslash Y} \Psi_{\backslash N_j}.$$

Using (4.2) and summing over all x_k for $k \notin N_Y \backslash i$, we obtain the following equation, which holds for the exact cavity distributions $Z^{\backslash i}$ and $Z^{\backslash j}$:

$$\sum_{x_i} \sum_{x_{\Delta i \backslash Y}} \Psi_{N_i \backslash Y} Z^{\backslash i} = \sum_{x_i} \sum_{x_{\Delta j \backslash Y}} \Psi_{N_j \backslash Y} Z^{\backslash j}.$$

By substituting our basic assumption (4.4) on both sides and pulling the factor $\phi_Y^{\backslash i}(x_{N_Y \backslash i})$ in the l.h.s. through the summation, we obtain:

$$\phi_Y^{\backslash i} \sum_{x_i} \sum_{x_{\Delta i \backslash Y}} \Psi_{N_i \backslash Y} \zeta_0^{\backslash i} \prod_{I \in N_i \backslash Y} \phi_I^{\backslash i} = \sum_{x_i} \sum_{x_{\Delta j \backslash Y}} \Psi_{N_j \backslash Y} \zeta_0^{\backslash j} \prod_{J \in N_j} \phi_J^{\backslash j}.$$

Since this should hold for each $j \in N_Y \setminus i$, we can take the geometric mean of the r.h.s. over all $j \in N_Y \setminus i$. After rearranging, this yields:

$$\phi_Y^{\backslash i} = \frac{\left(\prod_{j \in N_Y \backslash i} \sum_{x_i} \sum_{x_{\Delta j \backslash Y}} \Psi_{N_j \backslash Y} \zeta_0^{\backslash j} \prod_{J \in N_j} \phi_J^{\backslash j} \right)^{1/\#(N_Y \backslash i)}}{\sum_{x_i} \sum_{x_{\Delta i \backslash Y}} \Psi_{N_i \backslash Y} \zeta_0^{\backslash i} \prod_{I \in N_i \backslash Y} \phi_I^{\backslash i}} \qquad \forall_{i \in \mathcal{V}} \forall_{Y \in N_i}. \tag{4.5}$$

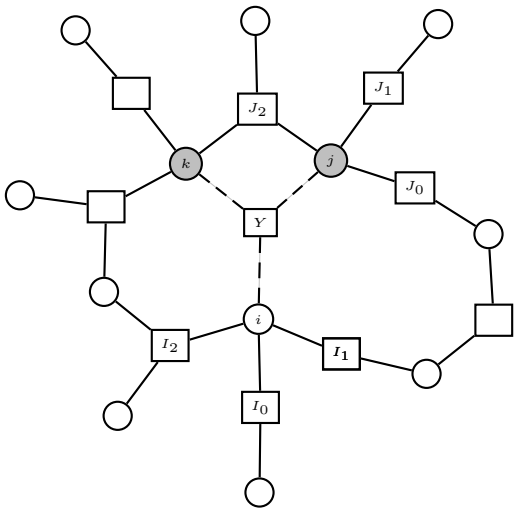


Figure 4.2: Part of the factor graph, illustrating the derivation of (4.5). The two gray variable nodes correspond to $N_Y \setminus i = \{j, k\}$.

Note that the numerator is an approximation of the joint marginal of the modified graphical model $\big(\mathcal{V}, \mathcal{F} \setminus Y, (\psi_I)_{I \in \mathcal{F} \setminus Y}\big)$ on the variables $N_Y \setminus i$.

Solving the consistency equations (4.5) simultaneously for the error factors $(\phi_I^{\setminus i})_{i \in \mathcal{V}, I \in N_i}$ can be done using a simple fixed point iteration algorithm, for example, Algorithm 4.1. The input consists of the initial approximations $(\zeta_0^{\setminus i})_{i \in \mathcal{V}}$ to the cavity distributions. It calculates the error factors that satisfy (4.5) by fixed point iteration and from the fixed point, it calculates improved approximations of the cavity distributions $(\zeta^{\setminus i})_{i \in \mathcal{V}}$ using equation (4.4).[1] From the improved cavity distributions, the loop-corrected approximations to the single-variable marginals of the original probability distribution (4.1) can be calculated as follows:

$$\mathbb{P}(x_i) \approx b_i(x_i) \propto \sum_{x_{\partial i}} \Psi_{N_i}(x_{\Delta i}) \zeta^{\setminus i}(x_{\partial i}), \tag{4.6}$$

where the factor ψ_Y is now included. Algorithm 4.1 uses a sequential update scheme, but other update schemes are possible (e.g., random sequential or parallel). In practice, the fixed sequential update scheme often converges without the need for damping.

Alternatively, one can formulate Algorithm 4.1 in terms of the "beliefs"

$$Q_i(x_{\Delta i}) \propto \Psi_{N_i}(x_{\Delta i}) \zeta_0^{\setminus i}(x_{\partial i}) \prod_{I \in N_i} \phi_I^{\setminus i}(x_{N_I \setminus i}) = \Psi_{N_i}(x_{\Delta i}) \zeta^{\setminus i}(x_{\partial i}). \tag{4.7}$$

[1] Alternatively, one could formulate the updates directly in terms of the cavity distributions $\{\zeta^{\setminus i}\}$.

Algorithm 4.1 Loop Correction Algorithm

Input: initial approximate cavity distributions $(\zeta_0^{\setminus i})_{i\in\mathcal{V}}$
Output: improved approximate cavity distributions $(\zeta^{\setminus i})_{i\in\mathcal{V}}$

1: **repeat**
2: **for all** $i \in \mathcal{V}$ **do**
3: **for all** $Y \in N_i$ **do**
4: $$\phi_Y^{\setminus i}(x_{N_Y\setminus i}) \leftarrow \frac{\left(\prod_{j\in N_Y\setminus i}\sum_{x_i}\sum_{x_{\Delta j\setminus Y}} \Psi_{N_j\setminus Y}\zeta_0^{\setminus j}\prod_{J\in N_j}\phi_J^{\setminus j}\right)^{1/\#(N_Y\setminus i)}}{\sum_{x_i}\sum_{x_{\Delta i\setminus Y}} \Psi_{N_i\setminus Y}\zeta_0^{\setminus i}\prod_{I\in N_i\setminus Y}\phi_I^{\setminus i}}$$
5: **end for**
6: **end for**
7: **until** convergence
8: **for all** $i \in \mathcal{V}$ **do**
9: $\zeta^{\setminus i}(x_{\partial i}) \leftarrow \zeta_0^{\setminus i}(x_{\partial i})\prod_{I\in N_i}\phi_I^{\setminus i}(x_{N_I\setminus i})$
10: **end for**

As one easily verifies, the update equation

$$Q_i \leftarrow Q_i \frac{\prod_{j\in N_Y\setminus i}\left(\sum_{x_{\Delta j\setminus(N_Y\setminus i)}} Q_j\,\psi_Y^{-1}\right)^{1/\#(N_Y\setminus i)}}{\sum_{x_{\Delta i\setminus(N_Y\setminus i)}} Q_i\,\psi_Y^{-1}}$$

is equivalent to line 4 of Algorithm 4.1. Intuitively, the update improves the approximate distribution Q_i on Δi by replacing its marginal on $N_Y \setminus i$ (in the absence of ψ_Y) by a more accurate approximation of this marginal, namely the numerator. Written in this form, the algorithm is reminiscent of iterative proportional fitting (IPF). However, contrary to IPF, the desired marginals are also updated at each iteration. Note that after convergence, the large beliefs $Q_i(x_{\Delta i})$ need not be consistent, that is, in general $\sum_{x_{\Delta i\setminus J}} Q_i \neq \sum_{x_{\Delta j\setminus J}} Q_j$ for $i, j \in \mathcal{V}$, $J \subseteq \Delta i \cap \Delta j$.

4.2.4 A special case: factorized cavity distributions

In the previous subsection we have discussed how to improve approximations of cavity distributions. We now discuss what happens when we use the simplest possible initial approximations $(\zeta_0^{\setminus i})_{i\in\mathcal{V}}$, namely constant functions, in Algorithm 4.1. This amounts to the assumption that no loops are present. We will show that if the factor graph does not contain short loops consisting of four nodes, fixed points of the standard BP algorithm are also fixed points of Algorithm 4.1. In this sense, Algorithm 4.1 can be considered to be a generalization of the BP algorithm. In

fact, this holds even if the initial approximations factorize in a certain way, as will be shown below.

If all factors involve at most two variables, one can easily arrange for the factor graph to have no loops of four nodes. See figure 4.1(a) for an example of a factor graph which has no loops of four nodes. The factor graph depicted in figure 4.2 does have a loop of four nodes: $k - Y - j - J_2 - k$.

Theorem 4.3 *If the factor graph corresponding to (4.1) has no loops of exactly four nodes, and all initial approximate cavity distributions factorize in the following way:*

$$\zeta_0^{\setminus i}(x_{\partial i}) = \prod_{I \in N_i} \xi_I^{\setminus i}(x_{N_I \setminus i}) \qquad \forall i \in \mathcal{V}, \tag{4.8}$$

then fixed points of the BP algorithm can be mapped to fixed points of Algorithm 4.1. Furthermore, the corresponding variable and factor marginals obtained from (4.7) are identical to the BP beliefs.

Proof. Note that replacing the initial cavity approximations by

$$\zeta_0^{\setminus i}(x_{\partial i}) \mapsto \zeta_0^{\setminus i}(x_{\partial i}) \prod_{I \in N_i} \epsilon_I^{\setminus i}(x_{N_I \setminus i})$$

for arbitrary positive functions $\epsilon_I^{\setminus i}(x_{N_I \setminus i})$ does not change the beliefs (4.7) corresponding to the fixed points of (4.5). Thus, without loss of generality, we can assume $\zeta_0^{\setminus i}(x_{\partial i}) = 1$ for all $i \in \mathcal{V}$. The BP update equations are [Kschischang *et al.*, 2001]:

$$\begin{aligned} \mu_{j \to I}(x_j) &\propto \prod_{J \in N_j \setminus I} \mu_{J \to j}(x_j) && j \in \mathcal{V}, I \in N_j, \\ \mu_{I \to i}(x_i) &\propto \sum_{x_{N_I \setminus i}} \psi_I(x_{N_I}) \prod_{j \in N_I \setminus i} \mu_{j \to I}(x_j) && I \in \mathcal{F}, i \in N_I \end{aligned} \tag{4.9}$$

in terms of messages $\big(\mu_{J \to j}(x_j)\big)_{j \in \mathcal{V}, J \in N_j}$ and $\big(\mu_{j \to J}(x_j)\big)_{j \in \mathcal{V}, J \in N_j}$. Assume that the messages μ are a fixed point of (4.9) and take the *Ansatz*

$$\phi_I^{\setminus i}(x_{N_I \setminus i}) = \prod_{k \in N_I \setminus i} \mu_{k \to I}(x_k) \qquad \text{for } i \in \mathcal{V}, I \in N_i.$$

Then, for $i \in \mathcal{V}$, $Y \in N_i$, $j \in N_Y \setminus i$, we can write out part of the numerator of (4.5) as follows:

$$\begin{aligned} \sum_{x_i} \sum_{x_{\Delta j \setminus Y}} \Psi_{N_j \setminus Y} \zeta_0^{\setminus j} \prod_{J \in N_j} \phi_J^{\setminus j} &= \sum_{x_i} \sum_{x_{\Delta j \setminus Y}} \phi_Y^{\setminus j} \prod_{J \in N_j \setminus Y} \psi_J \phi_J^{\setminus j} \\ &= \sum_{x_i} \left(\prod_{k \in N_Y \setminus j} \mu_{k \to Y} \right) \prod_{J \in N_j \setminus Y} \sum_{x_{N_J \setminus j}} \psi_J \prod_{k \in N_J \setminus j} \mu_{k \to J} \\ &= \sum_{x_i} \left(\prod_{k \in N_Y \setminus j} \mu_{k \to Y} \right) \mu_{j \to Y} = \sum_{x_i} \prod_{k \in N_Y} \mu_{k \to Y} \\ &\propto \prod_{k \in N_Y \setminus i} \mu_{k \to Y} = \phi_Y^{\setminus i}, \end{aligned}$$

where we used the BP update equations (4.9) and rearranged the summations and products using the assumption that the factor graph has no loops of four nodes. Thus, the numerator of the r.h.s. of (4.5) is simply $\phi_Y^{\backslash i}$. Using a similar calculation, one can derive that the denominator of the r.h.s. of (4.5) is constant, and hence (4.5) is valid (up to an irrelevant constant).

For $Y \in \mathcal{F}$, $i \in N_Y$, the marginal on x_Y of the belief (4.7) can be written in a similar way:

$$\begin{aligned}
\sum_{x_{\Delta i \backslash Y}} Q_i &\propto \sum_{x_{\Delta i \backslash Y}} \Psi_{N_i} \prod_{I \in N_i} \phi_I^{\backslash i} = \sum_{x_{\Delta i \backslash Y}} \prod_{I \in N_i} \psi_I \prod_{k \in N_I \backslash i} \mu_{k \to I} \\
&= \psi_Y \left(\prod_{k \in N_Y \backslash i} \mu_{k \to Y} \right) \prod_{I \in N_i \backslash Y} \sum_{x_{N_I \backslash i}} \psi_I \prod_{k \in N_I \backslash i} \mu_{k \to I} \\
&= \psi_Y \left(\prod_{k \in N_Y \backslash i} \mu_{k \to Y} \right) \prod_{I \in N_i \backslash Y} \mu_{I \to i} = \psi_Y \left(\prod_{k \in N_Y \backslash i} \mu_{k \to Y} \right) \mu_{i \to Y} \\
&= \psi_Y \prod_{k \in N_Y} \mu_{k \to Y},
\end{aligned}$$

which is proportional to the BP belief $b_Y(x_Y)$ on x_Y. Hence, also the single-variable marginal b_i defined in (4.6) corresponds to the BP single-variable belief, since both are marginals of b_Y for $Y \in N_i$. □

If the factor graph does contain loops of four nodes, we usually observe that the fixed point of Algorithm 4.1 coincides with the solution of the "minimal" CVM approximation when using factorized initial cavity approximations as in (4.8). The minimal CVM approximation uses all maximal factors as outer clusters (a *maximal* factor is a factor defined on a domain which is not a strict subset of the domain of another factor). In that case, the factor beliefs found by Algorithm 4.1 are consistent, that is, $\sum_{x_{\Delta i \backslash N_Y}} Q_i = \sum_{x_{\Delta j \backslash N_Y}} Q_j$ for $i, j \in N_Y$, and are identical to the minimal CVM factor beliefs. In particular, this holds for all the graphical models used in section 4.3.[2]

4.2.5 Obtaining initial approximate cavity distributions

There is no principled way to obtain the initial approximations $(\zeta_0^{\backslash i}(x_{\partial i}))_{i \in \mathcal{V}}$ to the cavity distributions. In the previous subsection, we investigated the results of applying the LC algorithm on factorizing initial cavity approximations. More sophisticated approximations that do take into account the effect of loops can significantly enhance the accuracy of the final result. Here, we will describe one method, which

[2] In a draft version of this work [Mooij and Kappen, 2006], we conjectured that the result of Algorithm 4.1, when initialized with factorizing initial cavity approximations, would *always* coincide with the minimal CVM approximation. This conjecture no longer stands because we have found a counter example.

uses BP on clamped cavity networks. This method captures all interactions in the cavity distribution of i in an approximate way and can lead to very accurate results. Instead of BP, any other approximate inference method that gives an approximation of the normalizing constant Z in (4.1) can be used, such as Mean Field, TreeEP [Minka and Qi, 2004], a double-loop version of BP [Heskes *et al.*, 2003] which has guaranteed convergence towards a minimum of the Bethe free energy, or some variant of GBP [Yedidia *et al.*, 2005]. One could also choose the method for each cavity separately, trading accuracy versus computation time. We focus on BP because it is a very fast and often relatively accurate algorithm.

Let $i \in \mathcal{V}$ and consider the cavity network of i. For each possible state of $x_{\partial i}$, run BP on the cavity network clamped to that state $x_{\partial i}$ and calculate the corresponding Bethe free energy $F^{\backslash i}_{Bethe}(x_{\partial i})$ [Yedidia *et al.*, 2005]. Then, take the following initial approximate cavity distribution:

$$\zeta_0^{\backslash i}(x_{\partial i}) \propto e^{-F^{\backslash i}_{Bethe}(x_{\partial i})}.$$

This procedure is exponential in the size of ∂i: it uses $\prod_{j \in \partial i} \#(\mathcal{X}_j)$ BP runs. However, many networks encountered in applications are relatively sparse and have limited cavity size and the computational cost may be acceptable.

This particular way of obtaining initial cavity distributions has the following interesting property: in case the factor graph contains only a single loop and assuming that the fixed point is unique, the final beliefs (4.7) resulting from Algorithm 4.1 are exact. This can be shown using an argument similar to that given in [Montanari and Rizzo, 2005]. Suppose that the graphical model contains exactly one loop and let $i \in \mathcal{V}$. First, consider the case that i is part of the loop; removing i will break the loop and the remaining cavity network will be singly connected. The cavity distribution approximated by BP will thus be exact. Now if i is not part of the loop, removing i will divide the network into several connected components, one for each neighbor of i. This implies that the cavity distribution calculated by BP contains no higher-order interactions, that is, $\zeta_0^{\backslash i}$ is exact modulo single-variable interactions. Because the final beliefs (4.7) are invariant under perturbation of the $\zeta_0^{\backslash i}$ by single-variable interactions, the final beliefs calculated by Algorithm 4.1 are exact if the fixed point is unique.

If all interactions are pairwise and each variable is binary and has exactly $\#(\partial i) = d$ neighbors, the time complexity of the resulting "Loop-Corrected BP" (LCBP) algorithm is given by $\mathcal{O}(N2^d E I_{BP} + Nd2^{d+1} I_{LC})$, where E is the number of edges in the factor graph, I_{BP} is the average number of iterations of BP on a clamped cavity network and I_{LC} is the number of iterations needed to obtain convergence in Algorithm 4.1.

4.2.6 Differences with the original implementation

As mentioned before, the idea of estimating the cavity distributions and imposing certain consistency relations amongst them has been first presented in [Montanari

and Rizzo, 2005]. In its simplest form (i.e., the so-called first-order correction), the implementation of that basic idea as proposed by Montanari and Rizzo [2005] differs from our proposed implementation in the following aspects.

First, the original method described by Montanari and Rizzo [2005] is only formulated for the rather special case of binary variables and pairwise interactions. In contrast, our method is formulated in a general way that makes it applicable to factor graphs with variables having more than two possible values and factors consisting of more than two variables. Also, factors may contain zeros. The generality that our implementation offers is important for many practical applications. In the rest of this section, we will assume that the graphical model (4.1) belongs to the special class of models with binary variables with pairwise interactions, allowing further comparison of both implementations.

An important difference is that Montanari and Rizzo [2005] suggest to deform the initial approximate cavity distributions by altering certain *cumulants* (also called "connected correlations"), instead of altering certain interactions. In general, for a set $V \subseteq \mathcal{V}$ of ± 1-valued random variables $(x_i)_{i \in V}$, one can define for any subset $A \subseteq V$ the *moment*

$$\mathcal{M}_A := \sum_{x_V} \mathbb{P}(x_V) \prod_{j \in A} x_j.$$

The moments $(\mathcal{M}_A)_{A \subseteq V}$ are a particular parameterization of the probability distribution $\mathbb{P}(x_V)$. An alternative parameterization is given in terms of the cumulants. The *(joint) cumulants* $(\mathcal{C}_A)_{A \subseteq V}$ are certain polynomials of the moments, defined implicitly by the following equations:

$$\mathcal{M}_A = \sum_{\mathcal{B} \in \mathrm{Part}(A)} \prod_{E \in \mathcal{B}} \mathcal{C}_E$$

where $\mathrm{Part}(A)$ is the set of partitions of A.[3] In particular, $\mathcal{C}_i = \mathcal{M}_i$ and $\mathcal{C}_{ij} = \mathcal{M}_{ij} - \mathcal{M}_i \mathcal{M}_j$ for all $i, j \in V$ with $i \neq j$. Montanari and Rizzo [2005] propose to approximate the cavity distributions by estimating the pair cumulants and assuming higher-order cumulants to be zero. Then, the singleton cumulants (i.e., the single-variable marginals) are altered, keeping higher-order cumulants fixed, in such a way as to impose consistency of the single-variable marginals, in the absence of interactions shared by two neighboring cavities. We refer the reader to Appendix 4.A for a more detailed description of the implementation in terms of cumulants suggested by Montanari and Rizzo [2005].

The assumption suggested in [Montanari and Rizzo, 2005] that higher-order cumulants are zero is the most important difference with our method, which instead takes into account effective interactions in the cavity distribution of *all* orders. In principle, the cumulant parameterization also allows for taking into account higher-order cumulants, but this would not be very efficient due to the combinatorics needed for handling the partitions.

[3] For a set X, a *partition* of X is a nonempty set Y such that each $Z \in Y$ is a nonempty subset of X and $\bigcup Y = X$.

A minor difference lies in the method to obtain initial approximations to the cavity distributions. Montanari and Rizzo [2005] propose to use BP in combination with linear response theory to obtain the initial pairwise cumulants. This difference is not very important, since one could also use BP on clamped cavity networks instead, which turns out to give almost identical results.

As we will show in section 4.3, our method of altering interactions appears to be more robust and still works in regimes with strong interactions, whereas the cumulant implementation suffers from convergence problems for strong interactions.

Montanari and Rizzo [2005] also derive a linearized version of their cumulant-based scheme (by expanding up to first order in terms of the pairwise cumulants, see Appendix 4.A) which is quadratic in the size of the cavity. This linearized, cumulant-based version is currently the only one that can be applied to networks with large Markov blankets (cavities), that is, where the maximum number of states $\max_{i \in \mathcal{V}} \#(\mathcal{X}_{\Delta i})$ is large, provided that all variables are binary and interactions are pairwise.

4.3 Numerical experiments

We have performed various numerical experiments to compare the quality of the results and the computation time of the following approximate inference methods:

MF Mean Field, with a random sequential update scheme and no damping.

BP Belief Bropagation. We have used the recently proposed update scheme [Elidan *et al.*, 2006], which converges also for difficult problems without the need for damping.

TreeEP TreeEP [Minka and Qi, 2004], without damping. We generalized the method of choosing the base tree described in [Minka and Qi, 2004] to multiple variable factors as follows: when estimating the mutual information between x_i and x_j, we take the product of the marginals on $\{i, j\}$ of all the factors that involve x_i and/or x_j. Other generalizations of TreeEP to higher-order factors are possible (e.g., by clustering variables), but it is not clear how to do this in general in an optimal way.

LCBP ("Loop-Corrected Belief Propagation") Algorithm 4.1, where the approximate cavities are initialized according to the description in section 4.2.5.

LCBP-Cum The original cumulant-based loop correction scheme by Montanari and Rizzo [2005], using Response Propagation (also known as Linear Response) to approximate the initial pairwise cavity cumulants. The full update equations (4.14) are used and higher-order cumulants are assumed to vanish. For strong interactions, the update equations (4.14) often yield values for the $\mathcal{M}_j^{\setminus i}$ outside of the valid interval $[-1, 1]$. In this case, we project these values

back into the valid interval in the hope that the method will converge to a valid result, which it sometimes does.

LCBP-Cum-Lin Similar to LCBP-Cum, but instead of the full update equations (4.14), the linearized update equations (4.15) are used.

CVM-Min A double-loop implementation [Heskes *et al.*, 2003] of the minimal CVM approximation, which uses (maximal) factors as outer clusters.

CVM-Δ A double-loop implementation of CVM using the sets $(\Delta i)_{i\in\mathcal{V}}$ as outer clusters. These are the same sets of variables as used by LCBP (c.f. (4.7)) and therefore it is interesting to compare both algorithms.

CVM-Loop$k$ A double-loop implementation of CVM, using as outer clusters all (maximal) factors together with all loops in the factor graph that consist of up to k different variables (for $k = 3, 4, 5, 6, 8$).

We have used a double-loop implementation of CVM instead of GBP because the former is guaranteed to converge to a local minimum of the Kikuchi free energy [Heskes *et al.*, 2003] without damping, whereas the latter often only converges with strong damping, or does not converge at all, even for arbitrary strong damping. Furthermore, even if damped GBP would converge, the problem is that the optimal damping constant is not known *a priori*, thus requiring one or more trial runs with different damping constants, until a suitable one is found. Using too much damping slows down convergence, whereas a certain amount of damping is required to obtain convergence in the first place. Therefore, even if (damped) GBP would converge, we would not expect it to be much faster than a double-loop implementation because of the computational cost of finding the optimal damping constant.

To be able to assess the errors of the various approximate methods, we have only considered problems for which exact inference (using a standard JunctionTree method) was still feasible.

For each approximate inference method, we report the maximum ℓ_∞ error of the approximate single-variable marginals b_i, calculated as follows:

$$\text{Error} := \max_{i\in\mathcal{V}} \max_{x_i\in\mathcal{X}_i} |b_i(x_i) - \mathbb{P}(x_i)|$$

where $\mathbb{P}(x_i)$ is the exact marginal calculated using the JunctionTree method.

The computation time was measured as CPU time in seconds on a 2.4 GHz AMD Opteron 64bits processor with 4 GB memory. The timings should be seen as indicative because we have not spent equal amounts of effort optimizing each method.[4]

We consider an iterative method to be "converged" after T time steps if for each variable $i \in \mathcal{V}$, the ℓ_∞ distance between the approximate probability distributions of that variable at time step T and $T + 1$ is less than $\epsilon = 10^{-9}$.

[4]Our C++ implementation of various approximate inference algorithms is free/open source software and can be downloaded from `http://www.mbfys.ru.nl/~jorism/libDAI`.

We have studied four different model classes: (i) random graphs of uniform degree with pairwise interactions and binary variables; (ii) random factor graphs with binary variables and factor nodes of uniform degree $k = 3$; (iii) the ALARM network, which has variables taking on more than two possible values and factors consisting of more than two variables; (iv) PROMEDAS networks, which have binary variables but factors consisting of more than two variables. For more extensive experiments, see [Mooij and Kappen, 2006].

4.3.1 Random regular graphs with binary variables

We have compared various approximate inference methods on random graphs consisting of N binary (± 1-valued) variables, having only pairwise interactions, where each variable has the same degree $\#(\partial i) = d$. In this case, the probability distribution (4.1) can be written in the following way:

$$\mathbb{P}(x) = \frac{1}{Z} \exp\left(\sum_{i\in\mathcal{V}} \theta_i x_i + \frac{1}{2}\sum_{i\in\mathcal{V}}\sum_{j\in\partial i} J_{ij} x_i x_j\right).$$

The parameters $(\theta_i)_{i\in\mathcal{V}}$ are called the *local fields*; the parameters $(J_{ij} = J_{ji})_{i\in\mathcal{V}, j\in\partial i}$ are called the *couplings*. The graph structure and the parameters θ and J were drawn randomly for each instance. The local fields $\{\theta_i\}$ were drawn independently from a $\mathcal{N}\left(0, (\beta\Theta)^2\right)$ distribution (i.e., a normal distribution with mean 0 and standard deviation $\beta\Theta$). For the couplings $\{J_{ij}\}$, we took mixed ("spin-glass") couplings, drawn independently from a normal distribution

$$J_{ij} \sim \mathcal{N}\left(0, \left(\beta \tanh^{-1}\frac{1}{\sqrt{d-1}}\right)^2\right).$$

The constant β (called "inverse temperature" in statistical physics) controls the overall interaction strength and thereby the difficulty of the inference problem, larger β corresponding usually to more difficult problems. The constant Θ controls the relative strength of the local fields, where larger Θ result in easier inference problems. The particular d-dependent scaling of the couplings is used in order to obtain roughly d-independent behavior. For $\Theta = 0$ and for $\beta \approx 1$, a phase transition occurs in the limit of $N \to \infty$, going from an easy "paramagnetic" phase for $\beta < 1$ to a complicated "spin-glass" phase for $\beta > 1$.[5]

We have also done experiments with positive ("attractive" or "ferromagnetic") couplings, but the conclusions from these experiments did not differ significantly from those using mixed couplings [Mooij and Kappen, 2006]. Therefore we do not report those experiments here.

[5] More precisely, the PA-SG phase transition occurs at $\Theta = 0$ and $(d-1) = \langle \tanh^2(\beta J_{ij})\rangle$, where $\langle\cdot\rangle$ is the average over all J_{ij} [Mooij and Kappen, 2005a]. What happens for $\Theta > 0$ is not known, to the best of our knowledge.

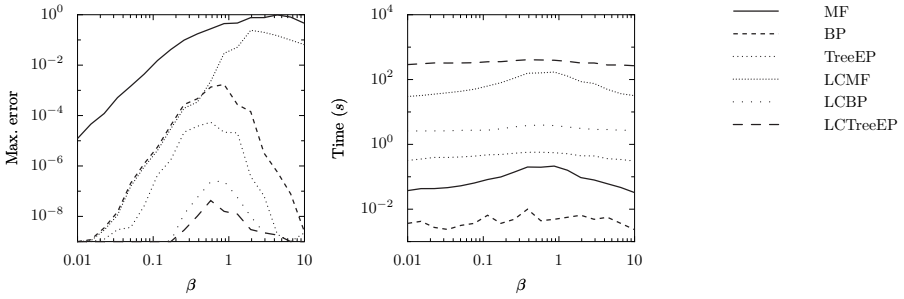


Figure 4.3: Error (left) and computation time (right) as a function of interaction strength for various approximate inference methods (MF, BP, TreeEP) and their loop-corrected versions (LCMF, LCBP, LCTreeEP). The averages (calculated in the logarithmic domain) were computed from the results for 16 randomly generated instances of $(N = 100, d = 3)$ regular random graphs with strong local fields $\Theta = 2$.

$N = 100$, $d = 3$, strong local fields ($\Theta = 2$)

We have studied various approximate inference methods on regular random graphs of low degree $d = 3$, consisting of $N = 100$ variables, with relatively strong local fields of strength $\Theta = 2$. We have considered various overall interaction strengths β between 0.01 and 10. For each value of β, we have used 16 random instances. On each instance, we have run various approximate inference algorithms.

Figure 4.3 shows results for MF, BP and TreeEP, and their loop-corrected versions, LCMF, LCBP and LCTreeEP. The loop-corrected versions are the result of Algorithm 4.1, initialized with approximate cavity distributions obtained by the procedure described in section 4.2.5 (using MF, BP, and TreeEP in the role of BP). Note that the Loop Correction method significantly reduces the error in each case. In fact, on average the loop-corrected error is approximately given by the square of the uncorrected error, as is apparent from the scatter plots in figure 4.4. BP is the fastest of the uncorrected methods and TreeEP is the most accurate but also the slowest uncorrected method. MF is both slower and less accurate than BP. Unsurprisingly, the loop-corrected methods show similar relative performance behaviors. Because BP is very fast and relatively accurate, we focus on LCBP in the rest of this chapter. Note further that although the graph is rather sparse, the improvement of LCBP over BP is significantly higher than the improvement of TreeEP over BP.

In figures 4.5 and 4.6 we compare the different implementations of the Loop Correction method on the same instances as used before. For small values of β, LCBP-Cum and LCBP-Cum-Lin both converge and yield high quality results, and the error introduced by the linearization is relatively small. However, for larger values of β, both methods get more and more convergence problems, although for the few cases where they do converge, they still yield accurate results. At $\beta \approx 10$, both methods have completely stopped converging. The error introduced by the linearization increases for larger values of β. The computation times of LCBP-

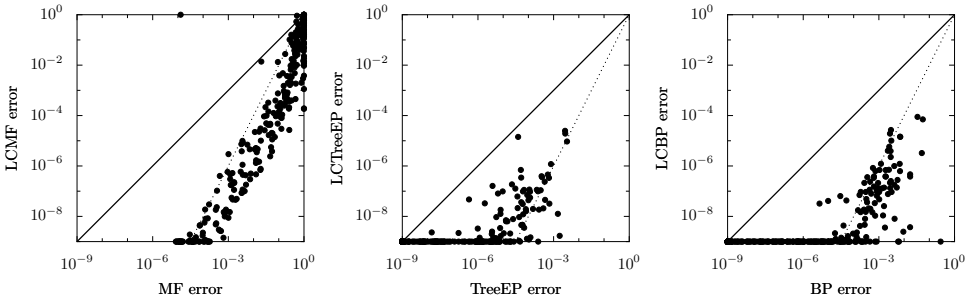


Figure 4.4: Pairwise comparisons of errors of uncorrected and loop-corrected methods, for the same instances as in figure 4.3. The solid lines correspond with $y = x$, the dotted lines with $y = x^2$. Only the cases have been plotted for which both approximate inference methods have converged. Saturation of errors around 10^{-9} is an artifact due to the convergence criterion.

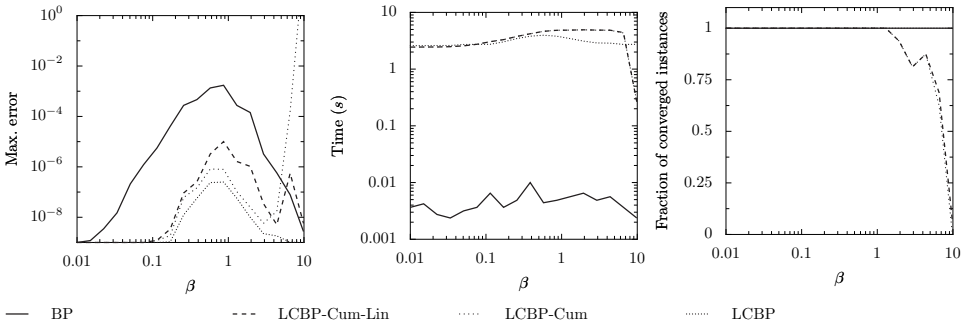


Figure 4.5: For the same instances as in figure 4.3: average error (left), average computation time (center) and fraction of converged instances (right) as a function of interaction strength β for various variants of the LC method. The averages of errors and computation time were calculated from the converged instances only. The average computation time and fraction of converged instances for LCBP-Cum and LCBP-Cum-Lin are difficult to distinguish, because they are (almost) identical.

Cum, LCBP-Cum-Lin and LCBP do not differ substantially in the regime where all methods converge. However, the quality of the LCBP results is higher than that of the cumulant-based methods. This is mainly due to the fact that LCBP also takes into account effective triple interactions in the initial estimates of the approximate cavity distributions.

We speculate that the reason for the break-down of LCBP-Cum and LCBP-Cum-Lin for strong interactions is due to the choice of cumulants instead of interactions. Indeed, consider two random variables x_i and x_j with fixed pair interaction $\exp(Jx_ix_j)$. By altering the singleton interactions $\exp(\theta_i x_i)$ and $\exp(\theta_j x_j)$, one can obtain any desired marginals of x_i and x_j. However, a fixed pair cumulant $\mathcal{C}_{ij} = \mathbb{E}(x_ix_j) - \mathbb{E}(x_i)\,\mathbb{E}(x_j)$ imposes a constraint on the range of possible expec-

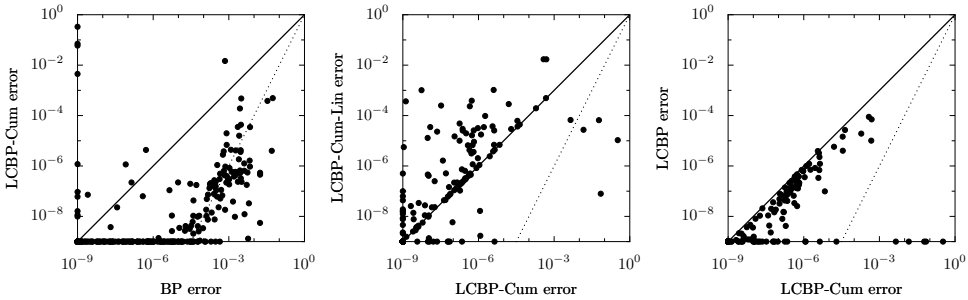


Figure 4.6: Pairwise comparisons of errors of various methods for the same instances as in figure 4.3. Only the cases have been plotted for which both approximate inference methods converged.

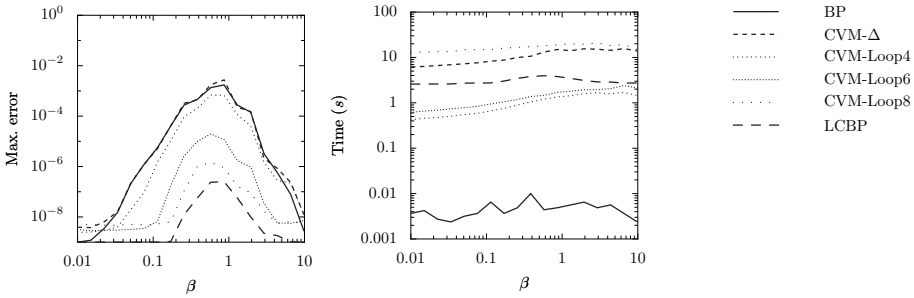


Figure 4.7: Average errors (left) and computation times (right) for various CVM methods (and LCBP, for reference) on the same instances as in figure 4.3. All methods converged on all instances.

tation values $\mathbb{E}(x_i)$ and $\mathbb{E}(x_j)$ (hence on the single-variable marginals of x_i and x_j); the freedom of choice in these marginals becomes less as the pair cumulant becomes stronger. We believe that something similar happens for LCBP-Cum (and LCBP-Cum-Lin): for strong interactions, the approximate pair cumulants in the cavity are strong, and even tiny errors can lead to inconsistencies which prevent convergence.

The results of the CVM approach to loop correction are shown in figures 4.7 and 4.8. The CVM-Loop methods, with clusters reflecting the short loops present in the factor graph, do indeed improve on BP. Furthermore, as expected, the use of larger clusters (that subsume longer loops) improves the results, although computation time quickly increases. CVM-Loop3 (not plotted) turned out not to give any improvement over BP, simply because there were (almost) no loops of 3 variables present. The most accurate CVM method, CVM-Loop8, needs more computation time than LCBP, whereas it yields inferior results.[6]

[6]The CVM errors are often seen to saturate around 10^{-8}, which indicates that the slow convergence of the CVM double-loop algorithm in these cases requires a stricter convergence criterion.

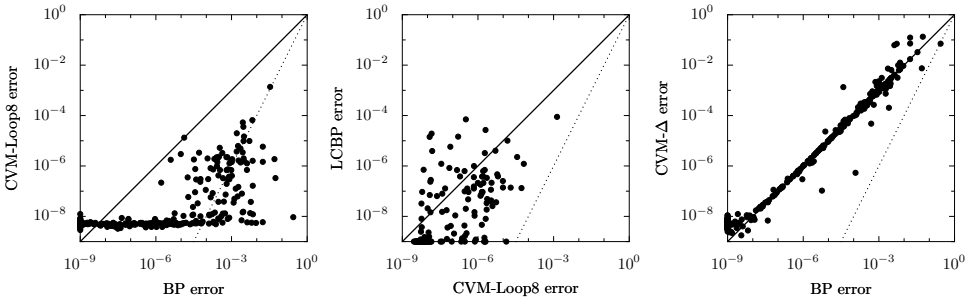


Figure 4.8: Pairwise comparisons of errors for various methods for the same instances as in figure 4.3.

In addition to the CVM-Loop methods, we compared with the CVM-Δ method, which uses $(\Delta i)_{i\in\mathcal{V}}$ as outer clusters. These clusters subsume the clusters used implicitly by BP (which are simply the pairwise factors) and therefore one would naively expect that the CVM-Δ approximation yields better results. Surprisingly however, the quality of CVM-Δ is *similar* to that of BP, although its computation time is enormous. This illustrates that simply using larger clusters for CVM does not always lead to better results. Furthermore, we conclude that although LCBP and CVM-Δ use identical clusters to approximate the target probability distribution, the nature of both approximations is very different.

Weak local fields ($\Theta = 0.2$)

We have done the same experiments also for weak local fields ($\Theta = 0.2$), with the other parameters unaltered (i.e., $N = 100$, $d = 3$). The picture roughly remains the same, apart from the following differences. First, the influence of the phase transition is more pronounced; many methods have severe convergence problems around $\beta = 1$. Second, the negative effect of linearization on the error (LCBP-Cum-Lin compared to LCBP-Cum) is smaller.

Larger degree ($d = 6$)

To study the influence of the degree $d = \#(\partial i)$, we have done additional experiments for $d = 6$. We took strong local fields ($\Theta = 2$). We had to reduce the number of variables to $N = 50$, because exact inference was infeasible for larger values of N due to quickly increasing treewidth. The results are shown in figure 4.9. As in the previous experiments, BP is the fastest and least accurate method, whereas LCBP yields the most accurate results, even for high β. Again we see that the LCBP error is approximately the square of the BP error and that LCBP gives better results than LCBP-Cum, but needs more computation time.

However, we also note the following differences with the case of low degree ($d = 3$). The relative improvement of TreeEP over BP has decreased. This could

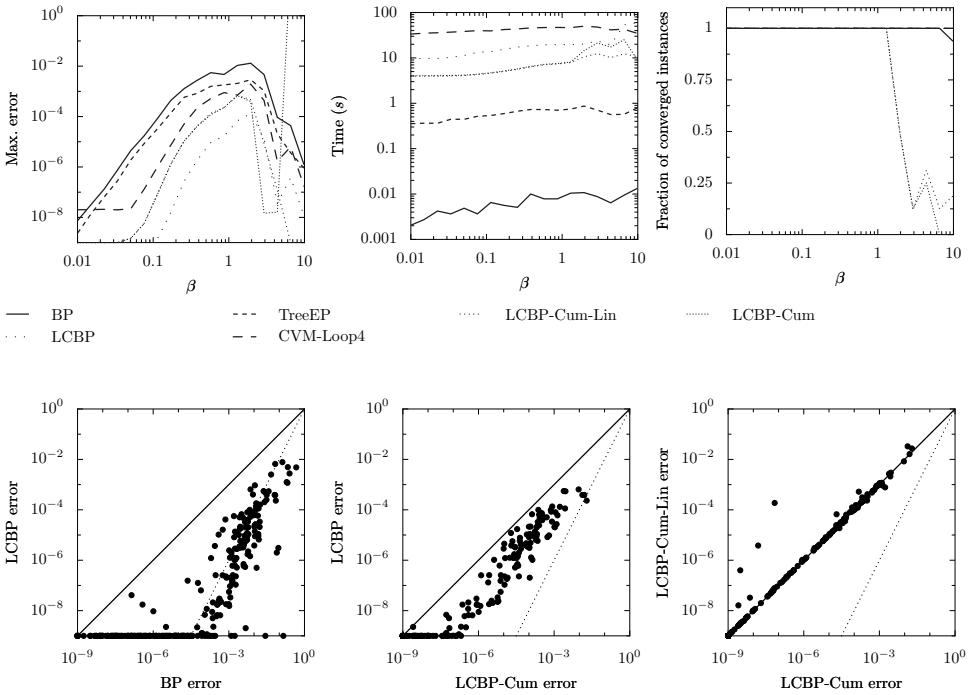


Figure 4.9: Selected results for $(N = 50, d = 6)$ regular random graphs with strong local fields $\Theta = 2$. The averaged results for LCBP-Cum and LCBP-Cum-Lin nearly coincide for $\beta \lesssim 1$.

have been expected, because in denser networks, the effect of taking out a tree becomes less.

Further, the relative improvement of CVM-Loop4 over BP has increased, probably because there are more short loops present. On the other hand, computation time of CVM-Loop4 has also increased and it is the slowest of all methods. We decided to abort the calculations for CVM-Loop6 and CVM-Loop8, because computation time was prohibitive due to the enormous amount of short loops present. We conclude that the CVM-Loop approach to loop correction is not very efficient if there are many loops present.

Surprisingly, the results of LCBP-Cum-Lin are now very similar in quality to the results of LCBP-Cum, except for a few isolated cases (presumably on the edge of the convergence region).

Scaling with N

We have investigated how computation time and error scale with the number of variables N, for fixed $\beta = 0.1$, $\Theta = 2$ and $d = 6$. We used a machine with more memory (16 GB) to be able to do exact inference without swapping also for $N = 60$.

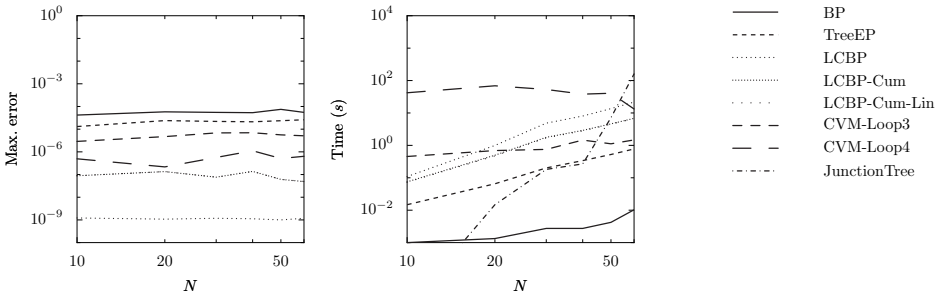


Figure 4.10: Error (left) and computation time (right) as a function of N (the number of variables), for random graphs with uniform degree $d = 6$, $\beta = 0.1$ and $\Theta = 2$. Points are averages over 16 randomly generated instances. Each method converged on all instances. The results for LCBP-Cum and LCBP-Cum-Lin coincide.

The results are shown in figure 4.10. The error of each method is approximately constant.

BP computation time should scale approximately linearly in N, which is difficult to see in this plot. LCBP variants are expected to scale quadratic in N (since d is fixed) which we have verified by checking the slopes of corresponding lines in the plot for large values of N. The computation time of CVM-Loop3 and CVM-Loop4 seems to be approximately constant, probably because the large number of overlaps of short loops for small values of N causes difficulties. The computation time of the exact JunctionTree method quickly increases due to increasing treewidth; for $N = 60$ it is already ten times larger than the computation time of the slowest approximate inference method.

We conclude that for large N, exact inference is infeasible, whereas LCBP still yields very accurate results using moderate computation time.

Scaling with d

It is also interesting to see how various methods scale with d, the variable degree, which is directly related to the cavity size. We have done experiments for random graphs of size $N = 24$ with fixed $\beta = 0.1$ and $\Theta = 2$ for different values of d between 3 and 23. The results can be found in figure 4.11. We aborted the calculations of the slower methods (LCBP, LCBP-Cum, CVM-Loop3) at $d = 15$.

Due to the particular dependence of the interaction strength on d, the errors of most methods depend only slightly on d. TreeEP is an exception: for larger d, the relative improvement of TreeEP over BP diminishes, and the TreeEP error approaches the BP error. CVM-Loop3 gives better quality, but needs relatively much computation time and becomes very slow for large d due to the large increase in the number of loops of 3 variables. LCBP is the most accurate method, but becomes very slow for large d. LCBP-Cum is less accurate and becomes slower than LCBP for large d, because of the additional overhead of the combinatorics needed to

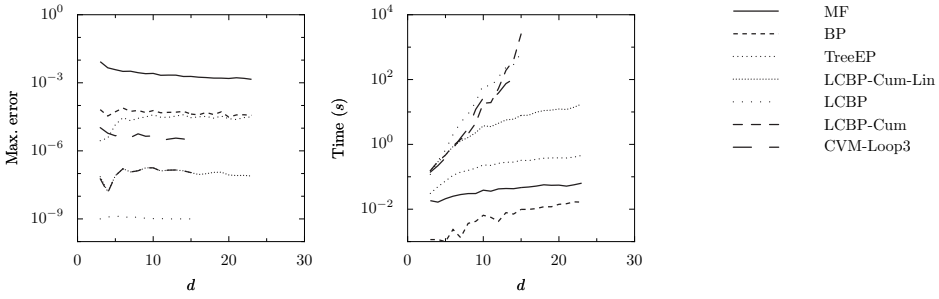


Figure 4.11: Error (left) and computation time (right) as a function of variable degree d for regular random graphs of $N = 24$ variables for $\beta = 0.1$ and $\Theta = 2$. Points are averages over 16 randomly generated instances. Each method converged on all instances. Errors of LCBP-Cum and LCBP-Cum-Lin coincide for $d \leq 15$; for $d > 15$, LCBP-Cum became too slow.

perform the update equations. The accuracy of LCBP-Cum-Lin is indistinguishable from that of LCBP-Cum, although it needs significantly less computation time.

Overall, we conclude from section 4.3.1 that for these binary, pairwise graphical models, LCBP is the best method for obtaining high accuracy marginals if the graphs are sparse, LCBP-Cum-Lin is the best method if the graphs are dense and LCBP-Cum shows no clear advantages over either method.

4.3.2 Multi-variable factors

We now go beyond pairwise interactions and study a class of random factor graphs with binary variables and uniform factor degree $\#(N_I) = k$ (for all $I \in \mathcal{F}$) with $k > 2$. The number of variables is N and the number of factors is M. The factor graphs are constructed by starting from an empty graphical model $(\mathcal{V}, \emptyset, \emptyset)$ and adding M random factors, where each factor is obtained in the following way: a subset $I = \{I_1, \ldots, I_k\} \subseteq \mathcal{V}$ of k different variables is drawn; a vector of 2^k independent random numbers $(J_I(x_I))_{x_I \in \mathcal{X}_I}$ is drawn from a $\mathcal{N}\left(0, \beta^2\right)$ distribution; the factor $\psi_I(x_{N_I}) := \exp J_I(x_I)$ is added to the graphical model. We only use those constructed factor graphs that are connected.[7] The parameter β again controls the interaction strength.

We have done experiments for $(N = 50, M = 50, k = 3)$ for various values of β between 0.01 and 2. For each value of β, we have used 16 random instances. For higher values of β, computation times increased quickly and convergence became problematic for BP, TreeEP and LCBP. This is probably related to the effects of a phase transition. The results are shown in figure 4.12.

Looking at the error and the computation time in figure 4.12, the following

[7]The reason that we require the factor graph to be connected is that not all our approximate inference method implementations currently support connected factor graphs that consist of more than one connected component.

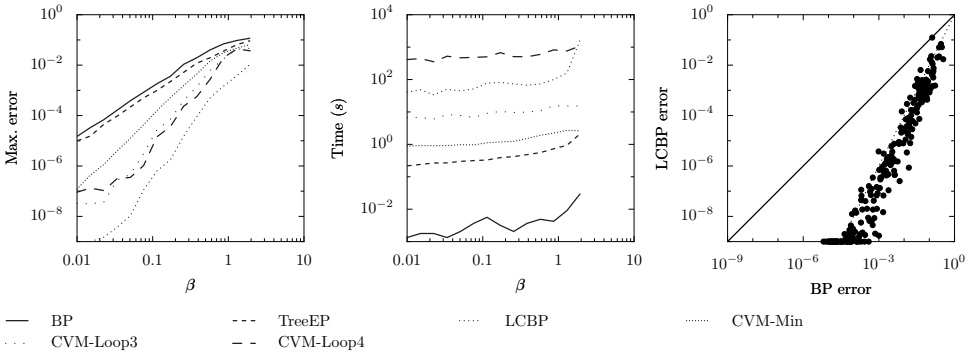


Figure 4.12: Results for $(N = 50, M = 50, k = 3)$ random factor graphs.

ranking can be made, where accuracy and computation time both increase: BP, TreeEP, CVM-Min, CVM-Loop3, LCBP. CVM-Loop4 uses more computation time than LCBP but gives worse results. LCBP-Cum and LCBP-Cum-Lin are not available due to the fact that the factors involve more than two variables. Note that the improvement of TreeEP over BP is rather small. Further, note that the LCBP error is again approximately given by the square of the BP error.

4.3.3 ALARM network

The ALARM network[8] is a well-known Bayesian network consisting of 37 variables (some of which can take on more than two possible values) and 37 factors (many of which involve more than two variables). In addition to the usual approximate inference methods, we have compared with GBP-Min, a GBP implementation of the minimal CVM approximation that uses maximal factors as outer clusters. The results are reported in table 4.1.[9]

The accuracy of GBP-Min (and CVM-Min) is almost identical to that of BP for this graphical model; GBP-Min converges without damping and is faster than CVM-Min. On the other hand, TreeEP significantly improves the BP result in roughly the same time as GBP-Min needs. Simply enlarging the cluster size (CVM-Δ) slightly deteriorates the quality of the results and also causes an enormous increase of computation time. The quality of the CVM-Loop results is roughly comparable to that of TreeEP. Surprisingly, increasing the loop depth beyond 4 deteriorates the quality of the results and results in an explosion of computation time. We conclude that the CVM-Loop method is not a very good approach to correcting loops in this case. LCBP uses considerable computation time, but yields errors

[8]The ALARM network can be downloaded from `http://compbio.cs.huji.ac.il/Repository/Datasets/alarm/alarm.dsc`.

[9]In [Mooij *et al.*, 2007], we also report experimental results for the ALARM network. In that work, we used another update rule for LCBP, which explains the different error obtained there $(5.4 \cdot 10^{-04})$. The update rule (4.5) used in the present work generally yields better results for higher-order interactions, whereas for pairwise interactions, both update rules are equivalent.

Method	Time (s)	Error
BP	0.00	$2.026 \cdot 10^{-01}$
TreeEP	0.21	$3.931 \cdot 10^{-02}$
GBP-Min	0.18	$2.031 \cdot 10^{-01}$
CVM-Min	1.13	$2.031 \cdot 10^{-01}$
CVM-Δ	280.67	$2.233 \cdot 10^{-01}$
CVM-Loop3	1.19	$4.547 \cdot 10^{-02}$
CVM-Loop4	154.97	$3.515 \cdot 10^{-02}$
CVM-Loop5	1802.83	$5.316 \cdot 10^{-02}$
CVM-Loop6	84912.70	$5.752 \cdot 10^{-02}$
LCBP	23.67	$3.412 \cdot 10^{-05}$

Table 4.1: Results for the ALARM network

that are approximately 10^4 times smaller than BP errors. The cumulant-based loop correction methods are not available, due to the presence of factors involving more than two variables and variables that can take more than two values.

4.3.4 PROMEDAS networks

In this subsection, we study the performance of LCBP on another "real world" example, the PROMEDAS medical diagnostic network [Wiegerinck *et al.*, 1999]. The diagnostic model in PROMEDAS is based on a Bayesian network. The global architecture of this network is similar to QMR-DT [Shwe *et al.*, 1991]. It consists of a diagnosis layer that is connected to a layer with findings.[10] Diagnoses (diseases) are modeled as *a priori* independent binary variables causing a set of symptoms (findings), which constitute the bottom layer. The PROMEDAS network currently consists of approximately 2000 diagnoses and 1000 findings.

The interaction between diagnoses and findings is modeled with a noisy-OR structure. The conditional probability of the finding given the parents is modeled by $m + 1$ real numbers, m of which represent the probabilities that the finding is caused by one of the diseases and one that the finding is not caused by any of the parents.

The noisy-OR conditional probability tables with m parents can be naively stored in a table of size 2^m. This is problematic for the PROMEDAS networks since findings that are affected by more than 30 diseases are not uncommon in the PROMEDAS network. We use an efficient implementation of noisy-OR relations as proposed by Takikawa and D'Ambrosio [1999] to reduce the size of these tables.

[10] In addition, there is a layer of variables, such as age and gender, that may affect the prior probabilities of the diagnoses. Since these variables are always clamped for each patient case, they merely change the prior disease probabilities and are irrelevant for our current considerations.

The trick is to introduce dummy variables s and to make use of the property

$$\mathrm{OR}(x|y_1, y_2, y_3) = \sum_s \mathrm{OR}(x|y_1, s)\mathrm{OR}(s|y_2, y_3).$$

The factors on the right hand side involve at most 3 variables instead of the initial 4 (left). Repeated application of this formula reduces all factors to triple interactions or smaller.

When a patient case is presented to PROMEDAS, a subset of the findings will be clamped and the rest will be unclamped. If our goal is to compute the marginal probabilities of the diagnostic variables only, the unclamped findings and the diagnoses that are not related to any of the clamped findings can be summed out of the network as a preprocessing step. The clamped findings cause an effective interaction between their parents. However, the noisy-OR structure is such that when the finding is clamped to a negative value, the effective interaction factorizes over its parents. Thus, findings can be clamped to negative values without additional computation cost [Jaakkola and Jordan, 1999].

The complexity of the problem now depends on the set of findings that is given as input. The more findings are clamped to a positive value, the larger the remaining network of disease variables and the more complex the inference task. Especially in cases where findings share more than one common possible diagnosis, and consequently loops occur, the model can become complex.

We use the PROMEDAS model to generate virtual patient data by first clamping one of the disease variables to be positive and then clamping each finding to its positive value with probability equal to the conditional distribution of the finding, given the positive disease. The union of all positive findings thus obtained constitute one patient case. For each patient case, the corresponding truncated graphical model is generated. The number of disease nodes in this truncated graph is typically quite large.

The results can be found in figures 4.13 and 4.14. Surprisingly, neither TreeEP nor any of the CVM methods gives substantial improvements over BP. TreeEP even gives worse results compared to BP. The CVM-Min and CVM-Loop3 results appear to be almost identical to the BP results. CVM-Loop4 manages to improve over BP in a few cases. Increased loop depth ($k = 5, 6$) results in worse quality in many cases and also in an enormous increase in computation time.

LCBP, on the other hand, is the only method that gives a significant improvement over BP, in each case. Considering all patient cases, LCBP corrects the BP error with more than one order of magnitude in half of the cases for which BP was not already exact. The improvement obtained by LCBP has its price: the computation time of LCBP is rather large compared to that of BP, as shown in figure 4.14. In many cases, this is due to a few rather large cavities. The cumulant-based loop correction methods are not available, due to the presence of factors involving more than two variables.

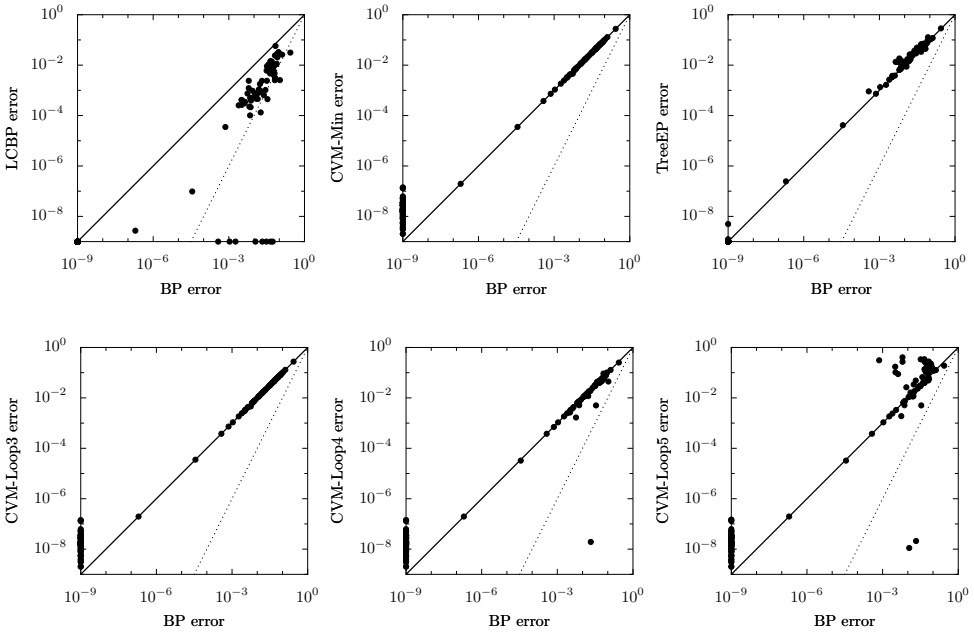


Figure 4.13: Scatter plots of errors for Promedas instances.

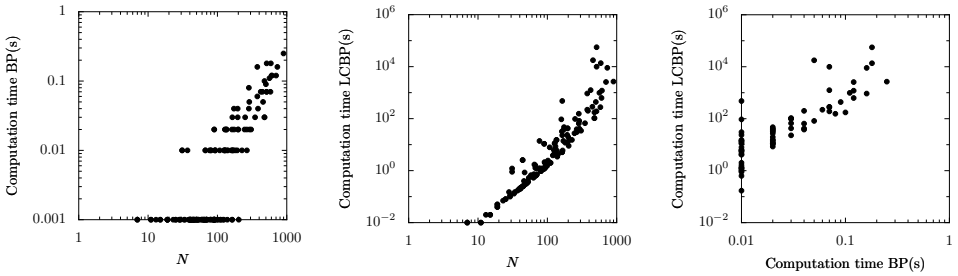


Figure 4.14: Computation time (in seconds) for Promedas instances: (left) BP computation time vs. N; (center) LCBP computation time vs. N; (right) LCBP vs. BP.

4.4 Discussion and conclusion

We have proposed a method to improve the quality of the single-variable marginals calculated by an approximate inference method (e.g., BP) by correcting for the influence of loops in the factor graph. We have proved that the method is a generalization of BP if the initial approximate cavity distributions factorize and the factor graph does not contain short loops of exactly four nodes. If the factor graph does contain such short loops, we observe in many cases that the method reduces to the minimal CVM approximation if one applies it on factorized initial approximate cavity distributions. If, on the other hand, the LC method is applied in combination with BP estimates of the effective cavity interactions, we have seen that the

loop-corrected error is approximately the square of the uncorrected BP error. Similar observations have been made for Loop-Corrected MF and TreeEP. For practical purposes, we suggest to apply loop corrections to BP ("LCBP"), because the loop correction approach requires many runs of the approximate inference method and BP is well suited for this job because of its speed. We have compared the performance of LCBP with other approximate inference methods that (partially) correct for the presence of loops. In most cases, LCBP turned out to be the most accurate method (with the notable exception of LCTreeEP, which is also considerably more expensive). LCBP still works for relatively strong interactions, in contrast with LCBP-Cum and LCBP-Cum-Lin.

On sparse factor graphs, TreeEP can obtain significant improvements over BP by correcting for loops that consist of part of the base tree and one additional interaction, using little computation time. However, for denser graphs, we observed that the difference between the quality of TreeEP and BP marginals diminishes. For both sparse and dense graphs, LCBP obtained more accurate results than TreeEP, although the computation time quickly increases for denser graphs.

We have seen that the CVM-Loop approximation, which uses small loops as outer clusters, can also provide accurate results, provided that the number of short loops is not too large and the number of intersections of clusters is limited. However, the computation time becomes prohibitive in many cases. In order to obtain the same accuracy as LCBP, the CVM-Loop approach usually needs significantly more computation time. This behavior is also seen on "real world" instances such as the ALARM network and PROMEDAS test cases. There may exist other cluster choices that give better results for the CVM approximation, but no general method for obtaining "good" cluster choices seems to be known (although for some special cases, for example, 2D grids, very good choices exist). However, Welling *et al.* [2005] give some heuristics for deciding whether a given CVM cluster choice is a "good" one. Another method that may provide good cluster choices is IJGP(i), proposed in [Dechter *et al.*, 2002]. We have not yet done an experimental comparison of LCBP with IJGP(i).

We have also compared the performance of LCBP with the original implementations proposed by Montanari and Rizzo [2005] (LCBP-Cum and LCBP-Cum-Lin) on the limited class of binary pairwise models. The original implementations work with cumulants instead of interactions and we believe that this explains the observed convergence difficulties of LCBP-Cum and LCBP-Cum-Lin in the regime of strong interactions. On sparse graphs, LCBP obtained better accuracy than LCBP-Cum and LCBP-Cum-Lin, using approximately similar computation time. This is mainly due to the fact that LCBP estimates the higher-order effective interactions in the cavity distributions. On dense graphs, both LCBP and LCBP-Cum become computationally infeasible. The linearized version LCBP-Cum-Lin, which is still applicable in these cases, performed surprisingly well, often obtaining similar accuracy as LCBP-Cum. For random graphs with high degree d (i.e., large Markov blankets), it turned out to be the most accurate of the applicable approximate in-

ference methods. It is rather fortunate that the negative effect of the linearization error on the accuracy of the result becomes smaller as the degree increases, since it is precisely for high degree where one needs the linearization because of performance issues.

In the experiments reported here, the standard JunctionTree method was almost always faster than LCBP. The reason is that we have intentionally selected experiments for which exact inference is still feasible, in order to be able to compare the quality of various approximate inference methods. However, as implied by figure 4.10, there is no reason to expect that LCBP will suddenly give inaccurate results when exact inference is no longer feasible. Thus we suggest that LCBP may be used to obtain accurate marginal estimates in cases where exact inference is impossible because of high treewidth. As illustrated in figure 4.10, the computation time of LCBP scales very different from that of the JunctionTree method: whereas the latter is exponential in treewidth, LCBP is exponential in the size of the Markov blankets.

The fact that computation time of LCBP (in its current form) scales exponentially with the size of the Markov blankets can be a severe limitation in practice. Many real world Bayesian networks have large Markov blankets, prohibiting application of LCBP. The linear cumulant-based implementation LCBP-Cum-Lin does not suffer from this problem, as it is quadratic in the size of the Markov blankets. Unfortunately, this particular implementation can in its current form only be applied to graphical models that consist of binary variables and factors that involve at most two variables (which excludes any interesting Bayesian network, for example). Furthermore, problems may arise if some factors contain zeros. For general application of loop correction methods, it will be of paramount importance to derive an implementation that combines the generality of LCBP with the speed of LCBP-Cum-Lin. This topic will be left for future research. The work presented here provides some intuition that may be helpful for constructing a general and fast loop correction method that is applicable to arbitrary factor graphs that can have large Markov blankets.

Another important direction for future research would be to find an extension of the loop correction framework that also gives a loop-corrected approximation of the normalization constant Z in (4.1). Additionally, and possibly related to that, it would be desirable to find an approximate "free energy", a function of the beliefs, whose stationary points coincide with the fixed points of Algorithm 4.1. This can be done for many approximate inference methods (MF, BP, CVM, EP) so it is natural to expect that the LC algorithm can also be seen as a minimization procedure of a certain approximate free energy. Despite some efforts, we have not yet been able to find such a free energy.

Recently, other loop correction approaches to the Bethe approximation have been proposed in the statistical physics literature [Parisi and Slanina, 2006; Chertkov and Chernyak, 2006b]. In particular, Chertkov and Chernyak [2006b] have derived a series expansion of the *exact* normalizing constant Z in terms of the BP

solution. The first term of the series is precisely the Bethe free energy evaluated at the BP fixed point. The number of terms in the series is finite, but can be very large, even larger than the number of total states of the graphical model. Each term is associated with a "generalized loop", which is a subgraph of the factor graph for which each node has at least connectivity two. By truncating the series, it is possible to obtain approximate solutions that improve on BP by taking into account a subset of all generalized loops [Gómez *et al.*, 2007; Chertkov and Chernyak, 2006a]. Summarizing, the approach to loop corrections by Chertkov and Chernyak [2006b] takes a subset of loops into account in an exact way, whereas the loop correction approach presented in this chapter takes all loops into account in an approximate way. More experiments should be done to compare both approaches.

Summarizing, we have proposed a method to correct approximate inference methods for the influence of loops in the factor graph. We have shown that it can obtain very accurate results, also on real world graphical models, outperforming existing approximate inference methods in terms of quality, robustness or applicability. We have shown that it can be applied to problems for which exact inference is infeasible. The rather large computation time required is an issue which deserves further consideration; it may be possible to use additional approximations on top of the loop correction framework that trade quality for computation time.

Acknowledgments

We thank Bastian Wemmenhove for stimulating discussions and for providing the PROMEDAS test cases.

4.A Original approach by Montanari and Rizzo (2005)

For completeness, we describe the implementation based on cumulants as originally proposed by Montanari and Rizzo [2005]. The approach can be applied in recursive fashion. Here we will only discuss the first recursion level.

Consider a graphical model which has only binary (± 1-valued) variables and factors that involve at most two variables. The corresponding probability distribution can be parameterized in terms of the local fields $(\theta_i)_{i\in\mathcal{V}}$ and the couplings $(J_{ij} = J_{ji})_{i\in\mathcal{V}, j\in\partial i}$:

$$\mathbb{P}(x) = \frac{1}{Z} \exp\left(\sum_{i\in\mathcal{V}} \theta_i x_i + \frac{1}{2} \sum_{i\in\mathcal{V}} \sum_{j\in\partial i} J_{ij} x_i x_j \right).$$

Let $i \in \mathcal{V}$ and consider the corresponding cavity network of i. For $A \subseteq \partial i$, the cavity *moment* $\mathcal{M}_A^{\backslash i}$ is defined as the following expectation value under the cavity

distribution:

$$\mathcal{M}_A^{\setminus i} := \frac{\sum_{x_{\partial i}} Z^{\setminus i}(x_{\partial i}) \prod_{j \in A} x_j}{\sum_{x_{\partial i}} Z^{\setminus i}(x_{\partial i})},$$

where we will not explicitly distinguish between approximate and exact quantities, following the physicists' tradition.[11] The cavity *cumulants* (also called "connected correlations") $\mathcal{C}_A^{\setminus i}$ are related to the moments in the following way:

$$\mathcal{M}_A^{\setminus i} = \sum_{\mathcal{B} \in \mathrm{Part}(A)} \prod_{E \in \mathcal{B}} \mathcal{C}_E^{\setminus i}$$

where $\mathrm{Part}(A)$ is the set of partitions of A.

We introduce some notation: we define for $A \subseteq \partial i$:

$$t_{iA} := \prod_{k \in A} \tanh J_{ik}.$$

Further, for a set X, we denote the even subsets of X as $\mathcal{P}_+(X) := \{Y \subseteq X : \#(Y) \text{ is even}\}$ and the odd subsets of X as $\mathcal{P}_-(X) := \{Y \subseteq X : \#(Y) \text{ is odd}\}$.

Using standard algebraic manipulations, one can show that for $j \in \partial i$, the expectation value of x_j in the absence of the interaction $\psi_{ij} = \exp(J_{ij} x_i x_j)$ can be expressed in terms of cavity moments of i as follows:

$$\frac{\sum_{A \in \mathcal{P}_+(\partial i \setminus j)} t_{iA} \mathcal{M}_{A \cup j}^{\setminus i} + \tanh \theta_i \sum_{A \in \mathcal{P}_-(\partial i \setminus j)} t_{iA} \mathcal{M}_{A \cup j}^{\setminus i}}{\sum_{A \in \mathcal{P}_+(\partial i \setminus j)} t_{iA} \mathcal{M}_A^{\setminus i} + \tanh \theta_i \sum_{A \in \mathcal{P}_-(\partial i \setminus j)} t_{iA} \mathcal{M}_A^{\setminus i}}. \tag{4.10}$$

On the other hand, the same expectation value can also be expressed in terms of cavity moments of j as follows:

$$\frac{\tanh \theta_j \sum_{B \in \mathcal{P}_+(\partial j \setminus i)} t_{jB} \mathcal{M}_B^{\setminus j} + \sum_{B \in \mathcal{P}_-(\partial j \setminus i)} t_{jB} \mathcal{M}_B^{\setminus j}}{\sum_{B \in \mathcal{P}_+(\partial j \setminus i)} t_{jB} \mathcal{M}_B^{\setminus j} + \tanh \theta_j \sum_{B \in \mathcal{P}_-(\partial j \setminus i)} t_{jB} \mathcal{M}_B^{\setminus j}}. \tag{4.11}$$

The consistency equations are now given by equating (4.10) to (4.11) for all $i \in \mathcal{V}$, $j \in \partial i$.

The expectation value of x_i (in the presence of all interactions) can be similarly expressed in terms of cavity moments of i:

$$\mathcal{M}_i := \sum_{x_i = \pm 1} \mathbb{P}(x_i) x_i = \frac{\tanh \theta_i \sum_{A \in \mathcal{P}_+(\partial i)} t_{iA} \mathcal{M}_A^{\setminus i} + \sum_{A \in \mathcal{P}_-(\partial i)} t_{iA} \mathcal{M}_A^{\setminus i}}{\sum_{A \in \mathcal{P}_+(\partial i)} t_{iA} \mathcal{M}_A^{\setminus i} + \tanh \theta_i \sum_{A \in \mathcal{P}_-(\partial i)} t_{iA} \mathcal{M}_A^{\setminus i}}. \tag{4.12}$$

[11] In [Montanari and Rizzo, 2005], the notation $\tilde{C}_{\mathcal{A}}^{(i)}$ is used for the cavity moment $\mathcal{M}_A^{\setminus i}$.

4.A.1 Neglecting higher-order cumulants

Montanari and Rizzo proceed by neglecting cavity cumulants $\mathcal{C}_A^{\setminus i}$ with $\#(A) > 2$. Denote by $\mathrm{Part}_2(A)$ the set of all partitions of A into subsets which have cardinality 2 at most. Thus, neglecting higher-order cavity cumulants amounts to the following approximation:

$$\mathcal{M}_A^{\setminus i} \approx \sum_{\mathcal{B}\in \mathrm{Part}_2(A)} \prod_{E\in\mathcal{B}} \mathcal{C}_E^{\setminus i}. \tag{4.13}$$

By some algebraic manipulations, one can express the consistency equation (4.10) = (4.11) in this approximation as follows:

$$\begin{aligned}\mathcal{M}_j^{\setminus i} = {} & \frac{\tanh\theta_j \sum_{B\in\mathcal{P}_+(\partial j\setminus i)} t_{jB}\mathcal{M}_B^{\setminus j} + \sum_{B\in\mathcal{P}_-(\partial j\setminus i)} t_{jB}\mathcal{M}_B^{\setminus j}}{\sum_{B\in\mathcal{P}_+(\partial j\setminus i)} t_{jB}\mathcal{M}_B^{\setminus j} + \tanh\theta_j \sum_{B\in\mathcal{P}_-(\partial j\setminus i)} t_{jB}\mathcal{M}_B^{\setminus j}} \\ & - \sum_{k\in\partial i\setminus j} t_{ik}\mathcal{C}_{jk}^{\setminus i} \frac{\tanh\theta_i \sum_{A\in\mathcal{P}_+(\partial i\setminus\{j,k\})} t_{iA}\mathcal{M}_A^{\setminus i} + \sum_{A\in\mathcal{P}_-(\partial i\setminus\{j,k\})} t_{iA}\mathcal{M}_A^{\setminus i}}{\sum_{A\in\mathcal{P}_+(\partial i\setminus j)} t_{iA}\mathcal{M}_A^{\setminus i} + \tanh\theta_i \sum_{A\in\mathcal{P}_-(\partial i\setminus j)} t_{iA}\mathcal{M}_A^{\setminus i}}.\end{aligned} \tag{4.14}$$

One can use (4.13) to write (4.14) in terms of the singleton cumulants $(\mathcal{M}_j^{\setminus i})_{i\in\mathcal{V},j\in\partial i}$ and the pair cumulants $(\mathcal{C}_{jk}^{\setminus i})_{i\in\mathcal{V},j\in\partial i,k\in\partial i\setminus j}$. Given (estimates of) the pair cumulants, the consistency equations (4.14) are thus fixed point equations in the singleton cumulants. The procedure is now:

- Estimate the pair cumulants $(\mathcal{C}_{jk}^{\setminus i})_{i\in\mathcal{V},j\in\partial i,k\in\partial i\setminus j}$ using BP in combination with linear response (called "Response Propagation" in [Montanari and Rizzo, 2005]).
- Calculate the fixed point $(\mathcal{M}_j^{\setminus i})_{i\in\mathcal{V},j\in\partial i}$ of (4.14) using the estimated pair cumulants.
- Use (4.12) in combination with (4.13) to calculate the final expectation values $(\mathcal{M}_j)_{j\in\mathcal{V}}$ using the estimated pair cumulants and the fixed point of (4.14).

4.A.2 Linearized version

The update equations can be linearized by expanding up to first order in the pair cumulants $\mathcal{C}_{jk}^{\setminus i}$. This yields the following linearized consistency equation [Montanari and Rizzo, 2005]:

$$\mathcal{M}_j^{\setminus i} = T_i^{\setminus j} - \sum_{l\in\partial i\setminus j} \Omega_{j,l}^{\setminus i} t_{il}\mathcal{C}_{jl}^{\setminus i} + \sum_{\substack{l_1,l_2\in\partial j\setminus i \\ l_1<l_2}} \Gamma_{i,l_1l_2}^{\setminus j} t_{jl_1}t_{jl_2}\mathcal{C}_{l_1l_2}^{\setminus j} \tag{4.15}$$

where

$$T_A^{\setminus i} := \tanh\left(\theta_i + \sum_{k\in\partial i\setminus A} \tanh^{-1}(t_{ik}\mathcal{M}_k^{\setminus i})\right),$$
$$\Omega_{j,l}^{\setminus i} := \frac{T_{jl}^{\setminus i}}{1 + t_{il}M_l^{\setminus i}T_{jl}^{\setminus i}},$$
$$\Gamma_{i,l_1l_2}^{\setminus j} := \frac{T_{il_1l_2}^{\setminus j} - T_i^{\setminus j}}{1 + t_{jl_1}t_{jl_2}\mathcal{M}_{l_1}^{\setminus j}\mathcal{M}_{l_2}^{\setminus j} + t_{jl_1}\mathcal{M}_{l_1}^{\setminus j}T_{il_1l_2}^{\setminus j} + t_{jl_2}\mathcal{M}_{l_2}^{\setminus j}T_{il_1l_2}^{\setminus j}}.$$

The final magnetizations (4.12) are, up to first order in the pair cumulants:

$$\mathcal{M}_j = T^{\setminus j} + \sum_{\substack{l_1,l_2\in\partial j\\ l_1<l_2}} \Gamma_{l_1l_2}^{\setminus j} t_{jl_1}t_{jl_2}\mathcal{C}_{l_1l_2}^{\setminus j} + \mathcal{O}(C^2)$$

where

$$\Gamma_{l_1l_2}^{\setminus j} := \frac{T_{l_1l_2}^{\setminus j} - T^{\setminus j}}{1 + t_{jl_1}t_{jl_2}M_{l_1}^{\setminus j}\mathcal{M}_{l_2}^{\setminus j} + t_{jl_1}\mathcal{M}_{l_1}^{\setminus j}T_{l_1l_2}^{\setminus j} + t_{jl_2}\mathcal{M}_{l_2}^{\setminus j}T_{l_1l_2}^{\setminus j}}.$$

Chapter 5

Novel bounds on marginal probabilities

We derive two related novel bounds on single-variable marginal probability distributions in factor graphs with discrete variables. The first method propagates bounds over a subtree of the factor graph rooted in the variable, and the second method propagates bounds over the self-avoiding walk tree starting at the variable. By construction, both methods not only bound the exact marginal probability distribution of a variable, but also its approximate Belief Propagation marginal ("belief"). Thus, apart from providing a practical means to calculate bounds on marginals, our contribution also lies in an increased understanding of the error made by Belief Propagation. Empirically, we show that our bounds often outperform existing bounds in terms of accuracy and/or computation time. We also show that our bounds can yield nontrivial results for medical diagnosis inference problems.

5.1 Introduction

Graphical models are used in many different fields. A fundamental problem in the application of graphical models is that exact inference is NP-hard [Cooper, 1990]. In recent years, much research has focused on approximate inference techniques, such as sampling methods and deterministic approximation methods, e.g., Belief Propagation (BP) [Pearl, 1988]. Although the approximations obtained by these methods can be very accurate, there are only few guarantees on the error of the approximation, and often it is not known (without comparing with the exact solution) how accurate an approximate result is. Thus it is desirable to calculate, in

The material in this chapter has been submitted to the Journal of Machine Learning Research. A preprint is available as [Mooij and Kappen, 2008].

addition to the approximate results, tight bounds on the approximation error. Existing methods to calculate bounds on marginals include [Tatikonda, 2003; Leisink and Kappen, 2003; Taga and Mase, 2006a; Ihler, 2007]. Also, upper bounds on the partition sum, e.g., [Jaakkola and Jordan, 1996; Wainwright *et al.*, 2005], can be combined with lower bounds on the partition sum, such as the well-known mean field bound or higher-order lower bounds [Leisink and Kappen, 2001], to obtain bounds on marginals.

In this chapter, we derive novel bounds on exact single-variable marginals in factor graphs. The original motivation for this work was to better understand and quantify the BP error. This has lead to bounds which are at the same time bounds for the exact single-variable marginals as well as for the BP beliefs. A particularly nice feature of the bounds is that their computational cost is relatively low, provided that the number of possible values of each variable in the factor graph is small. Unfortunately, the computation time is exponential in the number of possible values of the variables, which limits application to factor graphs in which each variable has a low number of possible values. On these factor graphs however, our bounds perform exceedingly well and we show empirically that they outperform the state-of-the-art in a variety of factor graphs, including real-world problems arising in medical diagnosis.

This chapter is organized as follows. In the next section, we derive our novel bounds. In section 5.3, we discuss related work. In section 5.4 we present experimental results. We conclude with conclusions and a discussion in section 5.5.

5.2 Theory

In this work, we consider graphical models such as Markov random fields and Bayesian networks. We use the unifying factor graph representation [Kschischang *et al.*, 2001]. In the first subsection, we introduce our notation and some basic definitions concerning factor graphs. Then, we shortly remind the reader of some basic facts about convexity. After that, we introduce some notation and concepts for measures on subsets of variables. We proceed with a subsection that considers the interplay between convexity and the operations of normalization and multiplication. In the next subsection, we introduce "(smallest bounding) boxes" that will be used to describe sets of measures in a convenient way. Then, we formulate the basic lemma that will be used to obtain bounds on marginals. We illustrate the basic lemma with two simple examples. Then we formulate our first result, an algorithm for propagating boxes over a subtree of the factor graph, which results in a bound on the marginal of the root variable of the subtree. In the last subsection, we show how one can go deeper into the computation tree and derive our second result, an algorithm for propagating boxes over self-avoiding walk trees. The result of that algorithm is a bound on the marginal of the root variable (starting point) of the self-avoiding walk tree. For the special case where all factors in the factor graph depend on two variables at most ("pairwise interactions"), our first result

is equivalent to a truncation of the second one. This is not true for higher-order interactions, however.

5.2.1 Factor graphs

Let $\mathcal{V} := \{1, \ldots, N\}$ and consider N discrete random variables $(x_i)_{i\in\mathcal{V}}$. Each variable x_i takes values in a discrete domain $\mathcal{X}_i$. We will frequently use the following multi-index notation. Let $A = \{i_1, i_2, \ldots, i_m\} \subseteq \mathcal{V}$ with $i_1 < i_2 < \ldots i_m$. We write $\mathcal{X}_A := \mathcal{X}_{i_1} \times \mathcal{X}_{i_2} \times \cdots \times \mathcal{X}_{i_m}$ and for any family $(Y_i)_{i\in B}$ with $A \subseteq B \subseteq \mathcal{V}$, we write $Y_A := (Y_{i_1}, Y_{i_2}, \ldots, Y_{i_m})$.

We consider a probability distribution over $x = (x_1, \ldots, x_N) \in \mathcal{X}_\mathcal{V}$ that can be written as a product of factors (also called "interactions") $(\psi_I)_{I\in\mathcal{F}}$:

$$\mathbb{P}(x) = \frac{1}{Z} \prod_{I\in\mathcal{F}} \psi_I(x_{N_I}), \qquad Z = \sum_{x\in\mathcal{X}_\mathcal{V}} \prod_{I\in\mathcal{F}} \psi_I(x_{N_I}). \tag{5.1}$$

For each factor index $I \in \mathcal{F}$, there is an associated subset $N_I \subseteq \mathcal{V}$ of variable indices and the factor ψ_I is a nonnegative function $\psi_I : \mathcal{X}_{N_I} \to [0, \infty)$. For a Bayesian network, the factors are (conditional) probability tables. In case of Markov random fields, the factors are often called potentials.[1] In the following, we will use lowercase for variable indices and uppercase for factor indices.

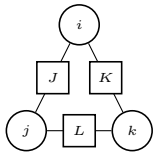
In general, the normalizing constant Z is not known and exact computation of Z is infeasible, due to the fact that the number of terms to be summed is exponential in N. Similarly, computing marginal distributions $\mathbb{P}(x_A)$ for subsets of variables $A \subseteq \mathcal{V}$ is intractable in general. In this chapter, we focus on the task of obtaining rigorous bounds on single-variable marginals $\mathbb{P}(x_i) = \sum_{x_{\mathcal{V}\setminus\{i\}}} \mathbb{P}(x)$.

We can represent the structure of the probability distribution (5.1) using a *factor graph* $(\mathcal{V}, \mathcal{F}, \mathcal{E})$. This is a bipartite graph, consisting of *variable nodes* $i \in \mathcal{V}$, *factor nodes* $I \in \mathcal{F}$, and *edges* $e \in \mathcal{E}$, with an edge $\{i, I\}$ between $i \in \mathcal{V}$ and $I \in \mathcal{F}$ if and only if the factor ψ_I depends on x_i (i.e., if $i \in N_I$). We will represent factor nodes visually as rectangles and variable nodes as circles. Figure 5.1 shows a simple example of a factor graph and the corresponding probability distribution. The set of neighbors of a factor node I is precisely N_I; similarly, we denote the set of neighbors of a variable node i by $N_i := \{I \in \mathcal{F} : i \in N_I\}$. Further, we define for each variable $i \in \mathcal{V}$ the set $\Delta i := \bigcup N_i$ consisting of all variables that appear in some factor in which variable i participates, and the set $\partial i := \Delta i \setminus \{i\}$, the *Markov blanket* of i.

We will assume throughout this chapter that the factor graph corresponding to (5.1) is connected. Furthermore, we will assume that

$$\forall I \in \mathcal{F}\ \forall i \in N_I\ \forall x_{N_I\setminus\{i\}} \in \mathcal{X}_{N_I\setminus\{i\}} : \sum_{x_i\in\mathcal{X}_i} \psi_I(x_i, x_{N_I\setminus\{i\}}) > 0.$$

[1] Not to be confused with statistical physics terminology, where "potential" refers to $-\frac{1}{\beta}\log\psi_I$ instead, with β the inverse temperature.



$$\mathbb{P}(x_i, x_j, x_k) = \tfrac{1}{Z}\psi_J(x_i, x_j)\psi_K(x_i, x_k)\psi_L(x_j, x_k)$$

$$Z = \sum_{x_i \in \mathcal{X}_i} \sum_{x_j \in \mathcal{X}_j} \sum_{x_k \in \mathcal{X}_k} \psi_J(x_i, x_j)\psi_K(x_i, x_k)\psi_L(x_j, x_k)$$

Figure 5.1: Example of a factor graph with three variable nodes (i, j, k), represented as circles, and three factor nodes (J, K, L), represented as rectangles. The corresponding random variables are x_i, x_j, x_k; the corresponding factors are $\psi_J : \mathcal{X}_i \times \mathcal{X}_j \to [0, \infty)$, $\psi_K : \mathcal{X}_i \times \mathcal{X}_k \to [0, \infty)$ and $\psi_L : \mathcal{X}_j \times \mathcal{X}_k \to [0, \infty)$. The corresponding probability distribution $\mathbb{P}(x)$ is written out on the right.

This will prevent technical problems regarding normalization later on.[2]

One final remark concerning notation: we will sometimes abbreviate $\{i\}$ as i if no confusion can arise.

5.2.2 Convexity

Let V be a real vector space. For T elements $(v_t)_{t=1,\dots,T}$ of V and T nonnegative numbers $(\lambda_t)_{t=1,\dots,T}$ with $\sum_{t=1}^{T} \lambda_t = 1$, we call $\sum_{t=1}^{T} \lambda_t v_t$ a *convex combination* of the $(v_t)_{t=1,\dots,T}$ with weights $(\lambda_t)_{t=1,\dots,T}$. A subset $X \subseteq V$ is called *convex* if for all $x_1, x_2 \in X$ and all $\lambda \in [0, 1]$, the convex combination $\lambda x_1 + (1 - \lambda)x_2 \in X$. An *extreme point* of a convex set X is an element $x \in X$ which cannot be written as a (nontrivial) convex combination of two different points in X. In other words, $x \in X$ is an extreme point of X if and only if for all $\lambda \in (0, 1)$ and all $x_1, x_2 \in X$, $x = \lambda x_1 + (1 - \lambda)x_2$ implies $x_1 = x_2$. We denote the set of extreme points of a convex set X by $\mathrm{Ext}\,(X)$. For a subset Y of the vector space V, we define the *convex hull of* Y to be the smallest convex set $X \subseteq V$ with $Y \subseteq X$; we denote the convex hull of Y as $\mathrm{Hull}\,(Y)$.

5.2.3 Measures and operators

For $A \subseteq \mathcal{V}$, define $\mathcal{M}_A := [0, \infty)^{\mathcal{X}_A}$, i.e., $\mathcal{M}_A$ is the set of nonnegative functions on $\mathcal{X}_A$. $\mathcal{M}_A$ can be identified with the set of finite measures on $\mathcal{X}_A$. We will simply call the elements of $\mathcal{M}_A$ "measures on A". We also define $\mathcal{M}_A^* := \mathcal{M}_A \setminus \{0\}$. We will denote $\mathcal{M} := \bigcup_{A \subseteq \mathcal{V}} \mathcal{M}_A$ and $\mathcal{M}^* := \bigcup_{A \subseteq \mathcal{V}} \mathcal{M}_A^*$.

Adding two measures $\Psi, \Phi \in \mathcal{M}_A$ results in the measure $\Psi + \Phi$ in $\mathcal{M}_A$. For $A, B \subseteq \mathcal{V}$, we can multiply an element of $\mathcal{M}_A$ with an element of $\mathcal{M}_B$ to obtain an element of $\mathcal{M}_{A \cup B}$; a special case is multiplication with a scalar. Note that there is a natural embedding of $\mathcal{M}_A$ in $\mathcal{M}_B$ for $A \subseteq B \subseteq \mathcal{V}$ obtained by multiplying an element $\Psi \in \mathcal{M}_A$ by $\mathbf{1}_{B \setminus A} \in \mathcal{M}_{B \setminus A}$, the constant function with value 1 on $\mathcal{X}_{B \setminus A}$.

[2] This condition ensures that if one runs Belief Propagation on the factor graph, the messages will always remain nonzero, provided that the initial messages are nonzero.

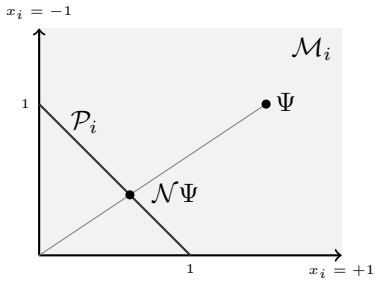


Figure 5.2: Illustration of some concepts in the simple case of a binary random variable $x_i \in \mathcal{X}_i = \{\pm 1\}$ and the subset $A = \{i\}$. A measure $\Psi \in \mathcal{M}_i$ can be identified with a point in the quarter plane as indicated in the figure. A normalized measure can be obtained by scaling Ψ; the result $\mathcal{N}\Psi$ is contained in the simplex $\mathcal{P}_i$, a lower-dimensional submanifold of $\mathcal{M}_i$.

Another important operation is the partial summation: given $A \subseteq B \subseteq \mathcal{V}$ and $\Psi \in \mathcal{M}_B$, define $\sum_{x_A} \Psi$ to be the measure in $\mathcal{M}_{B\setminus A}$ that satisfies

$$\left(\sum_{x_A} \Psi\right)(x_{B\setminus A}) = \sum_{x_A \in \mathcal{X}_A} \Psi(x_A, x_{B\setminus A}) \qquad \forall x_{B\setminus A} \in \mathcal{X}_{B\setminus A}.$$

Also, defining $A' = B\setminus A$, we will sometimes write this measure as $\sum_{x_{\setminus A'}} \Psi$, which is an abbreviation of $\sum_{x_{B\setminus A'}} \Psi$. This notation does not make explicit which variables are summed over (which depends on the measure that is being partially summed), although it shows which variables remain after summation.

In the following, we will implicitly define operations on *sets* of measures by applying the operation on elements of these sets and taking the set of the resulting measures; e.g., if we have two subsets $\Xi_A \subseteq \mathcal{M}_A$ and $\Xi_B \subseteq \mathcal{M}_B$ for $A, B \subseteq \mathcal{V}$, we define the product of the sets Ξ_A and Ξ_B to be the set of the products of elements of Ξ_A and Ξ_B, i.e., $\Xi_A\Xi_B := \{\Psi_A\Psi_B : \Psi_A \in \Xi_A, \Psi_B \in \Xi_B\}$.

In figure 5.2, the simple case of a binary random variable x_i and the subset $A = \{i\}$ is illustrated. Note that in this case, a measure $\Psi \in \mathcal{M}_i$ can be identified with a point in the quarter plane $[0, \infty) \times [0, \infty)$.

We will define $\mathcal{Q}_A$ to be the set of completely factorized measures on A, i.e.,

$$\mathcal{Q}_A := \prod_{a\in A} \mathcal{M}_{\{a\}} = \left\{ \prod_{a\in A} \Psi_a : \Psi_a \in \mathcal{M}_{\{a\}} \text{ for each } a \in A \right\}.$$

Note that $\mathcal{M}_A$ is the convex hull of $\mathcal{Q}_A$. Indeed, we can write each measure $\Psi \in \mathcal{M}_A$ as a convex combination of measures in $\mathcal{Q}_A$; let $Z := \sum_{x_A} \Psi$ and note that

$$\Psi(x) = \sum_{y\in\mathcal{X}_A} \frac{\Psi(y)}{Z} \left(Z\delta_y(x)\right) \qquad \forall x \in \mathcal{X}_A.$$

For any $y \in \mathcal{X}_A$, the Kronecker delta function $\delta_y \in \mathcal{M}_A$ (which is 1 if its argument is equal to y and 0 otherwise) is an element of $\mathcal{Q}_A$ because $\delta_y(x) = \prod_{a \in A} \delta_{y_a}(x_a)$. We denote $\mathcal{Q}_A^* := \mathcal{Q}_A \setminus \{0\}$.

We define the *partition sum operator* $\mathcal{Z} : \mathcal{M} \to [0, \infty)$ which calculates the partition sum (normalization constant) of a measure, i.e.,

$$\mathcal{Z}\Psi := \sum_{x_A \in \mathcal{X}_A} \Psi(x_A) \qquad \text{for } \Psi \in \mathcal{M}_A,\ A \subseteq \mathcal{V}.$$

We denote with $\mathcal{P}_A$ the set of probability measures on A, i.e., $\mathcal{P}_A = \{\Psi \in \mathcal{M}_A : \mathcal{Z}\Psi = 1\}$, and define $\mathcal{P} := \bigcup_{A \subseteq \mathcal{V}} \mathcal{P}_A$. The set $\mathcal{P}_A$ is called a *simplex* (see also figure 5.2). Note that a simplex is convex; the simplex $\mathcal{P}_A$ has precisely $\#(\mathcal{X}_A)$ extreme points, each of which corresponds to putting all probability mass on one of the possible values of x_A.

Define the *normalization operator* $\mathcal{N} : \mathcal{M}^* \to \mathcal{P}$ which normalizes a measure, i.e.,

$$\mathcal{N}\Psi := \frac{1}{\mathcal{Z}\Psi}\Psi \qquad \text{for } \Psi \in \mathcal{M}^*.$$

Note that $\mathcal{Z} \circ \mathcal{N} = 1$. Figure 5.2 illustrates the normalization of a measure in a simple case.

5.2.4 Convex sets of measures

To calculate marginals of subsets of variables in some factor graph, several operations performed on measures are relevant: normalization, taking products of measures, and summing over subsets of variables. In this section we study the interplay between convexity and these operations; this will turn out to be useful later on, because our bounds make use of convex sets of measures that are propagated over the factor graph.

The interplay between normalization and convexity is described by the following Lemma, which is illustrated in figure 5.3.

Lemma 5.1 *Let $A \subseteq \mathcal{V}$, $T \in \mathbb{N}^*$ and let $(\xi_t)_{t=1,\ldots,T}$ be elements of $\mathcal{M}_A^*$. Each convex combination of the normalized measures $(\mathcal{N}\xi_t)_{t=1,\ldots,T}$ can be written as a normalized convex combination of the measures $(\xi_t)_{t=1,\ldots,T}$ (which has different weights in general), and vice versa.*

Proof. Let $(\lambda_t)_{t=1,\ldots,T}$ be nonnegative numbers with $\sum_{t=1}^T \lambda_t = 1$. Then

$$\mathcal{Z}\left(\sum_{t=1}^T \lambda_t \mathcal{N}\xi_t\right) = \sum_{t=1}^T \lambda_t \mathcal{Z}\mathcal{N}\xi_t = 1,$$

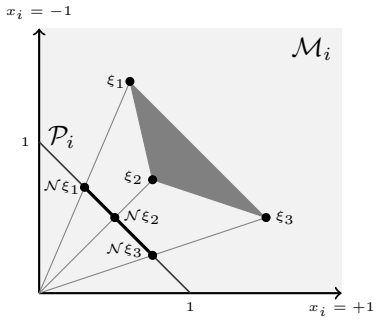


Figure 5.3: Any convex combination of $\mathcal{N}\xi_1$, $\mathcal{N}\xi_2$ and $\mathcal{N}\xi_3$ can be written as a normalized convex combination of ξ_1, ξ_2 and ξ_3. Vice versa, normalizing a convex combination of ξ_1, ξ_2 and ξ_3 yields a convex combination of $\mathcal{N}\xi_1, \mathcal{N}\xi_2$ and $\mathcal{N}\xi_3$.

therefore

$$\sum_{t=1}^{T} \lambda_t(\mathcal{N}\xi_t) = \mathcal{N}\left(\sum_{t=1}^{T} \lambda_t(\mathcal{N}\xi_t)\right) = \mathcal{N}\left(\sum_{t=1}^{T} \frac{\lambda_t}{\mathcal{Z}\xi_t}\xi_t\right)$$
$$= \mathcal{N}\left(\sum_{t=1}^{T} \frac{\frac{\lambda_t}{\mathcal{Z}\xi_t}}{\sum_{s=1}^{T}\frac{\lambda_s}{\mathcal{Z}\xi_s}}\ \xi_t\right),$$

which is the result of applying the normalization operator to a convex combination of the elements $(\xi_t)_{t=1,\ldots,T}$.

Vice versa, let $(\mu_t)_{t=1,\ldots,T}$ be nonnegative numbers with $\sum_{t=1}^{T} \mu_t = 1$. Then

$$\mathcal{N}\left(\sum_{t=1}^{T} \mu_t\xi_t\right) = \sum_{t=1}^{T} \frac{\mu_t}{Z}\xi_t$$

where

$$Z := \mathcal{Z}\left(\sum_{t=1}^{T} \mu_t\xi_t\right) = \sum_{t=1}^{T} \mu_t\mathcal{Z}\xi_t = \sum_{t=1}^{T} \mu_t Z_t$$

where we defined $Z_t := \mathcal{Z}\xi_t$ for all $t = 1, \ldots, T$. Thus

$$\mathcal{N}\left(\sum_{t=1}^{T} \mu_t\xi_t\right) = \sum_{t=1}^{T} \frac{\mu_t}{\sum_{s=1}^{T}\mu_s Z_s}\xi_t = \sum_{t=1}^{T} \frac{\mu_t Z_t}{\sum_{s=1}^{T}\mu_s Z_s}\mathcal{N}\xi_t,$$

which is a convex combination of the normalized measures $(\mathcal{N}\xi_t)_{t=1,\ldots,T}$. □

The following lemma concerns the interplay between convexity and taking products; it says that if we take the product of convex sets of measures on different spaces, the resulting set is contained in the convex hull of the product of the extreme points of the convex sets. We have not made a picture corresponding to this lemma because the simplest nontrivial case would require at least four dimensions.

Lemma 5.2 *Let $T \in \mathbb{N}^*$ and $(A_t)_{t=1,\dots,T}$ be a family of mutually disjoint subsets of $\mathcal{V}$. For each $t = 1, \dots, T$, let $\Xi_t \subseteq \mathcal{M}_{A_t}$ be convex with a finite number of extreme points. Then:*

$$\prod_{t=1}^{T} \Xi_t \subseteq \text{Hull}\left(\prod_{t=1}^{T} \text{Ext}\,\Xi_t\right),$$

Proof. Let $\Psi_t \in \Xi_t$ for each $t = 1, \dots, T$. For each t, Ψ_t can be written as a convex combination

$$\Psi_t = \sum_{\xi_t \in \text{Ext}\,(\Xi_t)} \lambda_{t;\xi_t} \xi_t, \qquad \sum_{\xi_t \in \text{Ext}\,(\Xi_t)} \lambda_{t;\xi_t} = 1, \qquad \forall\, \xi_t \in \text{Ext}\,(\Xi_t) : \lambda_{t;\xi_t} \geq 0.$$

Therefore the product $\prod_{t=1}^{T} \Psi_t$ is also a convex combination:

$$\begin{aligned}
\prod_{t=1}^{T} \Psi_t &= \prod_{t=1}^{T} \left(\sum_{\xi_t \in \text{Ext}\,(\Xi_t)} \lambda_{t;\xi_t} \xi_t \right) \\
&= \sum_{\xi_1 \in \text{Ext}\,(\Xi_1)} \sum_{\xi_2 \in \text{Ext}\,(\Xi_2)} \cdots \sum_{\xi_T \in \text{Ext}\,(\Xi_T)} \left(\prod_{t=1}^{T} \lambda_{t;\xi_t} \right) \left(\prod_{t=1}^{T} \xi_t \right) \\
&\in \text{Hull}\left(\prod_{t=1}^{T} \text{Ext}\,\Xi_t \right).
\end{aligned}$$

□

5.2.5 Boxes and smallest bounding boxes

In this subsection, we define "(smallest bounding) boxes", certain convex sets of measures that will play a central role in our bounds, and study some of their properties.

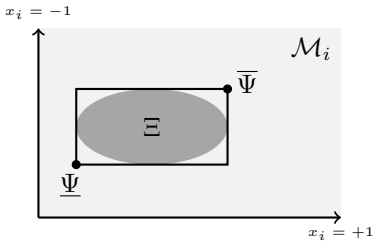
Definition 5.3 *Let $A \subseteq \mathcal{V}$. For $\underline{\Psi} \in \mathcal{M}_A$ and $\overline{\Psi} \in \mathcal{M}_A$ with $\underline{\Psi} \leq \overline{\Psi}$, we define the* box *between the lower bound $\underline{\Psi}$ and the upper bound $\overline{\Psi}$ by*

$$\mathcal{B}_A\left(\underline{\Psi}, \overline{\Psi}\right) := \{\Psi \in \mathcal{M}_A : \underline{\Psi} \leq \Psi \leq \overline{\Psi}\}.$$

The inequalities should be interpreted pointwise, e.g., $\underline{\Psi} \leq \Psi$ means $\underline{\Psi}(x) \leq \Psi(x)$ for all $x \in \mathcal{X}_A$. Note that a box is convex; indeed, its extreme points are the "corners" of which there are $2^{\#(\mathcal{X}_A)}$.

Definition 5.4 *Let $A \subseteq \mathcal{V}$ and $\Xi \subseteq \mathcal{M}_A$ be bounded (i.e., $\Xi \leq \Psi$ for some $\Psi \in \mathcal{M}_A$). The* smallest bounding box *for Ξ is defined as $\mathcal{B}(\Xi) := \mathcal{B}_A\left(\underline{\Psi}, \overline{\Psi}\right)$, where $\underline{\Psi}, \overline{\Psi} \in \mathcal{M}_A$ are given by*

$$\begin{aligned}
\underline{\Psi}(x_A) &:= \inf\{\Psi(x_A) : \Psi \in \Xi\} \qquad \forall x_a \in \mathcal{X}_A, \\
\overline{\Psi}(x_A) &:= \sup\{\Psi(x_A) : \Psi \in \Xi\} \qquad \forall x_a \in \mathcal{X}_A.
\end{aligned}$$



$$\mathcal{B}(\Xi) = \mathcal{B}_i\left(\underline{\Psi}, \overline{\Psi}\right)$$

Figure 5.4: The smallest bounding box $\mathcal{B}(\Xi)$ for Ξ is given by the box $\mathcal{B}_i\left(\underline{\Psi}, \overline{\Psi}\right)$ with lower bound $\underline{\Psi}$ and upper bound $\overline{\Psi}$.

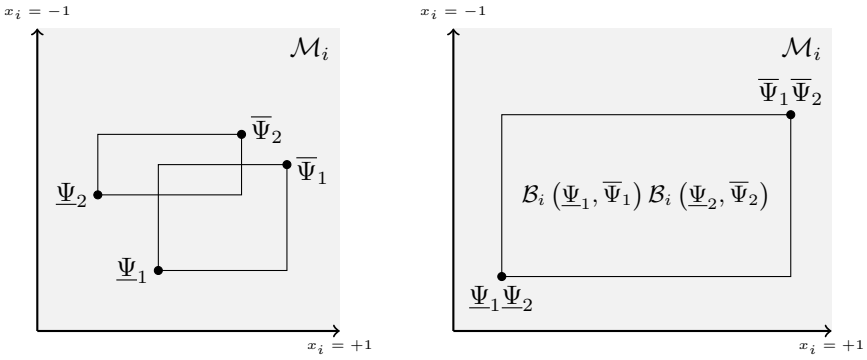


Figure 5.5: Multiplication of two boxes on the same variable set $A = \{i\}$.

Figure 5.4 illustrates this concept. Note that $\mathcal{B}(\Xi) = \mathcal{B}(\text{Hull}(\Xi))$. Therefore, if Ξ is convex, the smallest bounding box for Ξ depends only on the extreme points $\text{Ext}(\Xi)$, i.e., $\mathcal{B}(\Xi) = \mathcal{B}(\text{Ext}(\Xi))$.

The product of several boxes on the same subset A of variables can be easily calculated as follows (see also figure 5.5).

Lemma 5.5 *Let $A \subseteq \mathcal{V}$, $T \in \mathbb{N}^*$ and for each $t = 1, \ldots, T$, let $\underline{\Psi}_t, \overline{\Psi}_t \in \mathcal{M}_A$ such that $\underline{\Psi}_t \leq \overline{\Psi}_t$. Then*

$$\prod_{t=1}^{T} \mathcal{B}_A\left(\underline{\Psi}_t, \overline{\Psi}_t\right) = \mathcal{B}_A\left(\prod_{t=1}^{T} \underline{\Psi}_t, \prod_{t=1}^{T} \overline{\Psi}_t\right),$$

i.e., the product of the boxes is again a box, with as lower bound the product of the lower bounds of the boxes and as upper bound the product of the upper bounds of the boxes.

Proof. We prove the case $T = 2$; the general case follows by induction. We show that

$$\mathcal{B}_A\left(\underline{\Psi}_1, \overline{\Psi}_1\right)\mathcal{B}_A\left(\underline{\Psi}_2, \overline{\Psi}_2\right) = \mathcal{B}_A\left(\underline{\Psi}_1\,\underline{\Psi}_2, \overline{\Psi}_1\,\overline{\Psi}_2\right).$$

That is, for $\Phi \in \mathcal{M}_A$ we have to show that

$$\underline{\Psi}_1(x)\underline{\Psi}_2(x) \leq \Phi(x) \leq \overline{\Psi}_1(x)\overline{\Psi}_2(x) \qquad \forall x \in \mathcal{X}_A$$

if and only if there exist $\Phi_1, \Phi_2 \in \mathcal{M}_A$ such that:

$$\begin{aligned}
&\Phi(x) = \Phi_1(x)\Phi_2(x) && \forall x \in \mathcal{X}_A;\\
&\underline{\Psi}_1(x) \leq \Phi_1(x) \leq \overline{\Psi}_1(x) && \forall x \in \mathcal{X}_A;\\
&\underline{\Psi}_2(x) \leq \Phi_2(x) \leq \overline{\Psi}_2(x) && \forall x \in \mathcal{X}_A.
\end{aligned}$$

Note that the problem "decouples" for the various possible values of $x \in \mathcal{X}_A$ so that we can treat each component (indexed by $x \in \mathcal{X}_A$) seperately. That is, the problem reduces to showing that

$$[a, b] \cdot [c, d] = [ac, bd]$$

for $0 \leq a \leq b$ and $0 \leq c \leq d$ (take $a = \underline{\Psi}_1(x)$, $b = \overline{\Psi}_1(x)$, $c = \underline{\Psi}_2(x)$ and $d = \overline{\Psi}_2(x)$). In other words, we have to show that $y \in [ac, bd]$ if and only if there exist $y_1 \in [a, b]$, $y_2 \in [c, d]$ with $y = y_1 y_2$. For the less trivial part of this assertion, it is easily verified that choosing y_1 and y_2 according to the following table:

Condition	y_1	y_2
$bc \leq y, b > 0$	b	$\frac{y}{b}$
$b = 0$	0	c
$bc \geq y, c > 0$	$\frac{y}{c}$	c
$bc \geq y, c = 0$	b	0

does the job. □

In general, the product of several boxes is not a box itself. Indeed, let $i, j \in \mathcal{V}$ be two different variable indices. Then $\mathcal{B}_i\left(\underline{\Psi}_i, \overline{\Psi}_i\right)\mathcal{B}_j\left(\underline{\Psi}_j, \overline{\Psi}_j\right)$ contains only factorizing measures, whereas $\mathcal{B}_{\{i,j\}}\left(\underline{\Psi}_i\underline{\Psi}_j, \overline{\Psi}_i\overline{\Psi}_j\right)$ is not a subset of $\mathcal{Q}_{\{i,j\}}$ in general. However, we do have the following identity:

Lemma 5.6 *Let $T \in \mathbb{N}^*$ and for each $t = 1, \ldots, T$, let $A_t \subseteq \mathcal{V}$ and $\underline{\Psi}_t, \overline{\Psi}_t \in \mathcal{M}_{A_t}$ such that $\underline{\Psi}_t \leq \overline{\Psi}_t$. Then*

$$\mathcal{B}\left(\prod_{t=1}^{T}\mathcal{B}_{A_t}\left(\underline{\Psi}_t, \overline{\Psi}_t\right)\right) = \mathcal{B}_{\left(\bigcup_{t=1}^{T} A_t\right)}\left(\prod_{t=1}^{T}\underline{\Psi}_t, \prod_{t=1}^{T}\overline{\Psi}_t\right).$$

Proof. Straightforward, using the definitions. □

5.2.6 The basic lemma

After defining the elementary concepts, we can proceed with the basic lemma. Given the definitions introduced before, the basic lemma is easy to formulate. It is illustrated in figure 5.6.

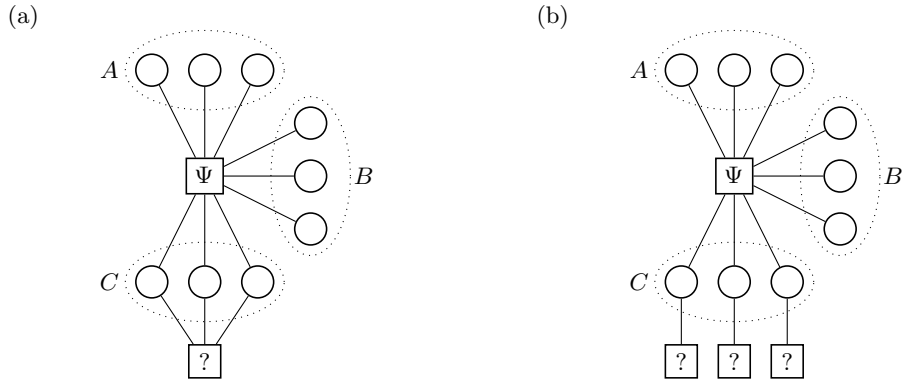


Figure 5.6: The basic lemma: the smallest bounding box enclosing the set of possible marginals of x_A is identical in cases (a) and (b), if we are allowed to put arbitrary factors on the factor nodes marked with question marks.

Lemma 5.7 *Let $A, B, C \subseteq \mathcal{V}$ be mutually disjoint subsets of variables. Let $\Psi \in \mathcal{M}_{A\cup B\cup C}$ such that for each $x_C \in \mathcal{X}_C$,*

$$\sum_{x_{A\cup B}} \Psi > 0.$$

Then:

$$\mathcal{B}\left(\mathcal{N}\left(\sum_{x_B,x_C} \Psi \mathcal{M}_C^*\right)\right) = \mathcal{B}\left(\mathcal{N}\left(\sum_{x_B,x_C} \Psi \mathcal{Q}_C^*\right)\right).$$

Proof. Note that $\mathcal{M}_C^*$ is the convex hull of $\mathcal{Q}_C^*$. Furthermore, the multiplication with Ψ and the summation over x_B, x_C preserves convex combinations, as does the normalization operation (see Lemma 5.1). Therefore,

$$\mathcal{N}\left(\sum_{x_B,x_C} \Psi \mathcal{M}_C^*\right) \subseteq \text{Hull}\left(\mathcal{N}\left(\sum_{x_B,x_C} \Psi \mathcal{Q}_C^*\right)\right)$$

from which the lemma follows. □

The positivity condition is a technical condition, which in our experience is fulfilled for most practically relevant factor graphs.

5.2.7 Examples

Before proceeding to the first main result, we first illustrate for a simple case how the basic lemma can be employed to obtain bounds on marginals. We show two bounds for the marginal of the variable x_i in the factor graph in figure 5.7(a).

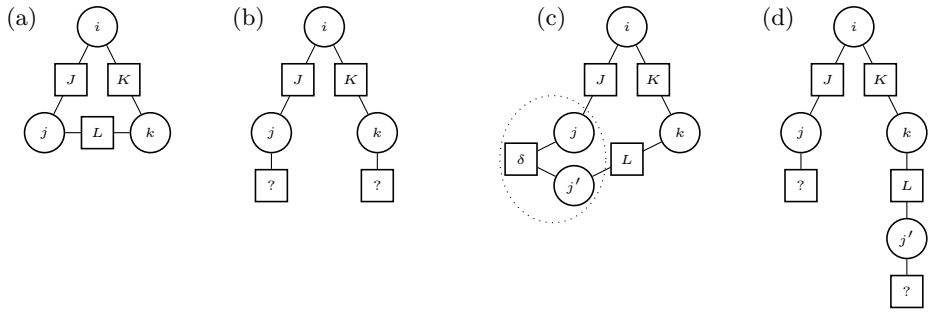


Figure 5.7: (a) Factor graph; (b) Illustration of the bound on $\mathbb{P}(x_i)$ corresponding to Example I; (c) Cloning node j by adding a new variable j' and a factor $\psi_\delta(x_j, x_{j'}) = \delta_{x_j}(x_{j'})$; (d) Illustration of the improved bound on $\mathbb{P}(x_i)$, corresponding to Example (II), based on (c).

Example I

First, note that the marginal of x_i satisfies

$$\mathbb{P}(x_i) = \mathcal{N}\left(\sum_{x_j}\sum_{x_k}\psi_J\psi_K\psi_L\right) \in \mathcal{N}\left(\sum_{x_j}\sum_{x_k}\psi_J\psi_K\mathcal{M}^*_{\{j,k\}}\right).$$

because, obviously, $\psi_L \in \mathcal{M}^*_{\{j,k\}}$. Now, applying the basic lemma with $A = \{i\}$, $B = \emptyset$, $C = \{j,k\}$ and $\Psi = \psi_J\psi_K$, we obtain

$$\mathbb{P}(x_i) \in \mathcal{B}\left(\mathcal{N}\left(\sum_{x_j}\sum_{x_k}\psi_J\psi_K\mathcal{Q}^*_{\{j,k\}}\right)\right).$$

Applying the distributive law, we conclude

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\Big(\sum_{x_j}\psi_J\mathcal{M}^*_j\Big)\Big(\sum_{x_k}\psi_K\mathcal{M}^*_k\Big)\right),$$

which certainly implies

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\mathcal{BN}\left(\sum_{x_j}\psi_J\mathcal{M}^*_j\right)\cdot\mathcal{BN}\left(\sum_{x_k}\psi_K\mathcal{M}^*_k\right)\right).$$

This is illustrated in figure 5.7(b), which should be read as "What can we say about the range of $\mathbb{P}(x_i)$ when the factors corresponding to the nodes marked with question marks are arbitrary?" Because of the various occurrences of the normalization operator, we can restrict ourselves to normalized measures on the question-marked factor nodes:

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\mathcal{BN}\left(\sum_{x_j}\psi_J\mathcal{P}_j\right)\cdot\mathcal{BN}\left(\sum_{x_k}\psi_K\mathcal{P}_k\right)\right).$$

Now it may seem that this smallest bounding box would be difficult to compute, because in principle one would have to compute all the measures in the sets $\mathcal{N}\sum_{x_j}\psi_J\mathcal{P}_j$ and $\mathcal{N}\sum_{x_k}\psi_K\mathcal{P}_k$. Fortunately, we only need to compute the extreme points of these sets, because the mapping

$$\mathcal{M}^*_{\{j\}} \to \mathcal{M}^*_{\{i\}} : \psi \mapsto \mathcal{N}\sum_{x_j}\psi_J\psi$$

maps convex combinations into convex combinations (and similarly for the other mapping, involving ψ_K). Since smallest bounding boxes only depend on extreme points, we conclude that

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\mathcal{BN}\left(\sum_{x_j}\psi_J \mathrm{Ext}\,\mathcal{P}_j\right)\cdot\mathcal{BN}\left(\sum_{x_k}\psi_K \mathrm{Ext}\,\mathcal{P}_k\right)\right)$$

which can be calculated efficiently if the number of possible values of each variable is small.

Example II

We can improve this bound by using another trick: cloning variables. The idea is to first clone the variable x_j by adding a new variable $x_{j'}$ that is constrained to take the same value as x_j. In terms of the factor graph, we add a variable node j' and a factor node δ, connected to variable nodes j and j', with corresponding factor $\psi_\delta(x_j, x_{j'}) := \delta_{x_j}(x_{j'})$; see also figure 5.7(c). Clearly, the marginal of x_i satisfies:

$$\begin{aligned}\mathbb{P}(x_i) &= \mathcal{N}\left(\sum_{x_j}\sum_{x_k}\psi_J\psi_K\psi_L\right)\\ &= \mathcal{N}\left(\sum_{x_j}\sum_{x_{j'}}\sum_{x_k}\psi_J\psi_K\psi_L\delta_{x_j}(x_{j'})\right)\end{aligned}$$

where it should be noted that in the first line, ψ_L is shorthand for $\psi_L(x_j, x_k)$ but in the second line it is meant as shorthand for $\psi_L(x_{j'}, x_k)$. Noting that $\psi_\delta \in \mathcal{M}^*_{\{j,j'\}}$ and applying the basic lemma with $C = \{j, j'\}$ yields:

$$\begin{aligned}\mathbb{P}(x_i) &\in \mathcal{N}\left(\sum_{x_j}\sum_{x_{j'}}\sum_{x_k}\psi_J\psi_K\psi_L\mathcal{M}^*_{\{j,j'\}}\right)\\ &\in \mathcal{BN}\left(\sum_{x_j}\sum_{x_{j'}}\sum_{x_k}\psi_J\psi_K\psi_L\mathcal{Q}^*_{\{j,j'\}}\right).\end{aligned}$$

Applying the distributive law, we obtain (see also figure 5.7(d)):

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\left(\sum_{x_j}\psi_J\mathcal{M}^*_{\{j\}}\right)\left(\sum_{x_k}\psi_K\sum_{x_{j'}}\psi_L\mathcal{M}^*_{\{j'\}}\right)\right),$$

from which we conclude

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\mathcal{BN}\left(\sum_{x_j}\psi_J\mathcal{P}_{\{j\}}\right)\mathcal{BN}\left(\sum_{x_k}\psi_K\mathcal{BN}\left(\sum_{x_{j'}}\psi_L\mathcal{P}_{\{j'\}}\right)\right)\right).$$

This can again be calculated efficiently by considering only extreme points.

As a more concrete example, take all variables as binary and take for each factor $\psi = \left(\begin{smallmatrix}1 & 2\\ 2 & 1\end{smallmatrix}\right)$. Then the first bound (Example I) yields:

$$\mathbb{P}(x_i) \in \mathcal{B}_i\left(\left(\begin{smallmatrix}1/5\\ 1/5\end{smallmatrix}\right), \left(\begin{smallmatrix}4/5\\ 4/5\end{smallmatrix}\right)\right),$$

whereas the second, tighter, bound (Example II) gives:

$$\mathbb{P}(x_i) \in \mathcal{B}_i\left(\left(\begin{smallmatrix}2/7\\ 2/7\end{smallmatrix}\right), \left(\begin{smallmatrix}5/7\\ 5/7\end{smallmatrix}\right)\right).$$

Obviously, the exact marginal is

$$\mathbb{P}(x_i) = \left(\begin{smallmatrix}1/2\\ 1/2\end{smallmatrix}\right).$$

5.2.8 Propagation of boxes over a subtree

We now formulate a message passing algorithm that resembles Belief Propagation. However, instead of propagating measures, it propagates boxes (or simplices) of measures; furthermore, it is only applied to a subtree of the factor graph, propagating boxes from the leaves towards a root node, instead of propagating iteratively over the whole factor graph several times. The resulting "belief" at the root node is a box that bounds the exact marginal of the root node. The choice of the subtree is arbitrary; different choices lead to different bounds in general. We illustrate the algorithm using the example that we have studied before (see figure 5.8).

Definition 5.8 *Let $(\mathcal{V}, \mathcal{F}, \mathcal{E})$ be a factor graph. We call the bipartite graph (V, F, E) a* subtree of $(\mathcal{V}, \mathcal{F}, \mathcal{E})$ with root i *if $i \in V \subseteq \mathcal{V}$, $F \subseteq \mathcal{F}$, $E \subseteq \mathcal{E}$ such that (V, F, E) is a tree with root i and for all $\{j, J\} \in E$, $j \in V$ and $J \in F$ (i.e., there are no "loose edges").*[3]

An illustration of a factor graph and a possible subtree is given in figure 5.8(a)-(b). We denote the parent of $j \in V$ according to (V, F, E) by $\mathrm{par}(j)$ and similarly, we denote the parent of $J \in F$ by $\mathrm{par}(J)$. In the following, we will use the topology of the *original* factor graph $(\mathcal{V}, \mathcal{F}, \mathcal{E})$ whenever we refer to neighbors of variables or factors.

Each edge of the subtree will carry one message, oriented such that it "flows" towards the root node. In addition, we define messages entering the subtree for

[3]Note that this corresponds to the notion of subtree of a bipartite graph; for a subtree of a factor graph, one sometimes imposes the additional constraint that for all factors $J \in F$, all its connecting edges $\{J, j\}$ with $j \in N_J$ have to be in E; here we do not impose this additional constraint.

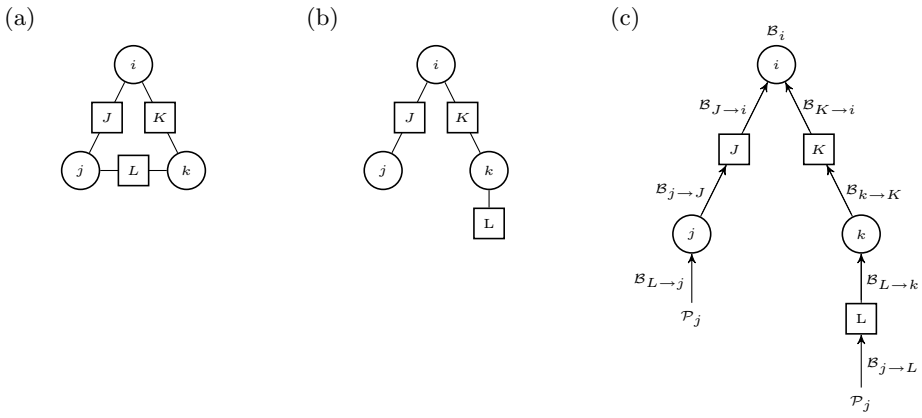


Figure 5.8: Box propagation algorithm corresponding to Example II: (a) Factor graph $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$; (b) a possible subtree (V, F, E) of G; (c) propagating sets of measures (boxes or simplices) on the subtree leading to a bound $\mathcal{B}_i$ on the marginal probability of x_i in $\mathcal{G}$.

all "missing" edges in the subtree. Because of the bipartite character of the factor graph, we can distinguish between two types of messages: messages $\mathcal{B}_{J\to j} \subseteq \mathcal{M}_j$ sent to a variable $j \in V$ from a neighboring factor $J \in N_j$, and messages $\mathcal{B}_{j\to J} \subseteq \mathcal{M}_j$ sent to a factor $J \in F$ from a neighboring variable $j \in N_J$.

The messages entering the subtree are all defined to be simplices; more precisely, we define the incoming messages

$$\begin{aligned} \mathcal{B}_{j\to J} &= \mathcal{P}_j \qquad J \in F,\, \{j, J\} \in \mathcal{E} \setminus E \\ \mathcal{B}_{J\to j} &= \mathcal{P}_j \qquad j \in V,\, \{j, J\} \in \mathcal{E} \setminus E. \end{aligned}$$

We propagate messages towards the root i of the tree using the following update rules (note the similarity with the BP update rules). The message sent from a variable $j \in V$ to its parent $J = \mathrm{par}(j) \in F$ is defined as

$$\mathcal{B}_{j\to J} = \begin{cases} \displaystyle\prod_{K \in N_j \setminus J} \mathcal{B}_{K\to j} & \text{if all incoming } \mathcal{B}_{K\to j} \text{ are boxes} \\ \mathcal{P}_j & \text{if at least one of the } \mathcal{B}_{K\to j} \text{ is the simplex } \mathcal{P}_j, \end{cases}$$

where the product of the boxes can be calculated using Lemma 5.5. The message sent from a factor $J \in F$ to its parent $k = \mathrm{par}(J) \in V$ is defined as

$$\mathcal{B}_{J\to k} = \mathcal{BN}\left(\sum_{x_{N_J \setminus k}} \psi_J \prod_{l \in N_J \setminus k} \mathcal{B}_{l\to J} \right). \tag{5.2}$$

This smallest bounding box can be calculated using the following Corollary of Lemma 5.2:

Corollary 5.9

$$\mathcal{BN}\left(\sum_{x_{N_J\setminus k}} \psi_J \prod_{l\in N_J\setminus k} \mathcal{B}_{l\to J}\right) = \mathcal{BN}\left(\sum_{x_{N_J\setminus k}} \psi_J \prod_{l\in N_J\setminus k} \operatorname{Ext}\mathcal{B}_{l\to J}\right)$$

Proof. By Lemma 5.2,

$$\prod_{l\in N_J\setminus k} \mathcal{B}_{l\to J} \subseteq \operatorname{Hull}\left(\prod_{l\in N_J\setminus k} \operatorname{Ext}\mathcal{B}_{l\to J}\right).$$

Because the multiplication with ψ_J and the summation over $x_{N_J\setminus k}$ preserves convex combinations, as does the normalization (see Lemma 5.1), the statement follows. □

The final "belief" $\mathcal{B}_i$ at the root node i is calculated by

$$\mathcal{B}_i = \begin{cases} \mathcal{BN}\left(\prod_{K\in N_j} \mathcal{B}_{K\to j}\right) & \text{if all incoming } \mathcal{B}_{K\to j} \text{ are boxes} \\ \mathcal{P}_j & \text{if at least one of the } \mathcal{B}_{K\to j} \text{ is the simplex } \mathcal{P}_j. \end{cases}$$

Note that when applying this to the case illustrated in figure 5.8, we obtain the bound that we derived earlier on ("Example II").

We can now formulate our first main result, which gives a rigorous bound on the exact single-variable marginal of the root node:

Theorem 5.10 *Let $(\mathcal{V},\mathcal{F},\mathcal{E})$ be a factor graph with corresponding probability distribution (5.1). Let $i\in\mathcal{V}$ and (V,F,E) be a subtree of $(\mathcal{V},\mathcal{F},\mathcal{E})$ with root $i\in V$. Apply the "box propagation" algorithm described above to calculate the final "belief" $\mathcal{B}_i$ on the root node i. Then $\mathbb{P}(x_i)\in\mathcal{B}_i$.*

Proof sketch. The first step consists in extending the subtree such that each factor node has the right number of neighboring variables by cloning the missing variables. The second step consists of applying the basic lemma where the set C consists of all the variable nodes of the subtree which have connecting edges in $\mathcal{E}\setminus E$, together with all the cloned variable nodes. Then we apply the distributive law, which can be done because the extended subtree has no cycles. Finally, we relax the bound by adding additional normalizations and smallest bounding boxes at each factor node in the subtree. It should now be clear that the recursive algorithm "box propagation" described above precisely calculates the smallest bounding box at the root node i that corresponds to this procedure. □

Note that a subtree of the orginal factor graph is also a subtree of the *computation tree for i* [Tatikonda and Jordan, 2002]. A computation tree is an "unwrapping" of the factor graph that has been used in analyses of the Belief Propagation algorithm. The computation tree starting at variable $i\in\mathcal{V}$ consists of all paths on the

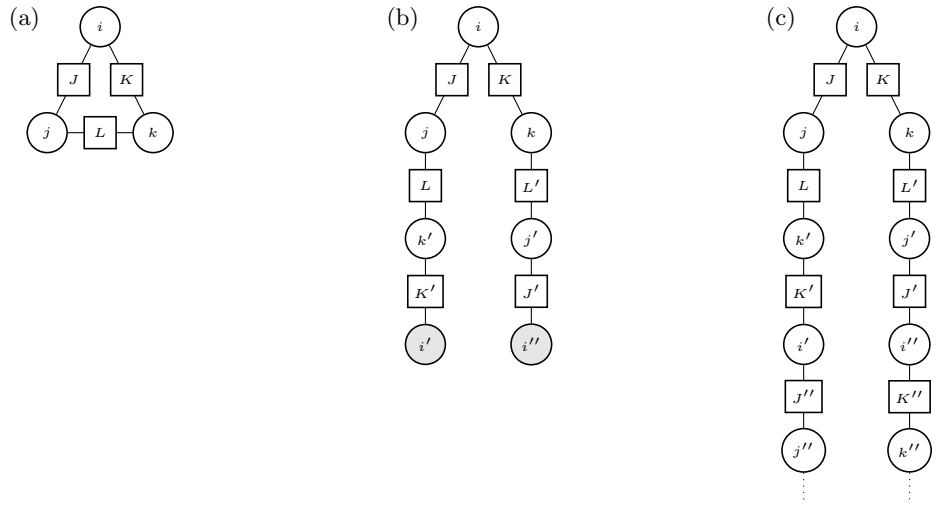


Figure 5.9: (a) Factor graph; (b) Self-avoiding walk tree with root i, with cycle-induced leaf nodes shown in gray; (c) Computation tree for i.

factor graph, starting at i, that never backtrack (see also figure 5.9(c)). This means that the bounds on the (exact) marginals that we just derived are at the same time bounds on the approximate Belief Propagation marginals (beliefs).

Corollary 5.11 *In the situation described in Theorem 5.10, the final bounding box $\mathcal{B}_i$ also bounds the (approximate) Belief Propagation marginal of the root node i, i.e., $\mathbb{P}_{BP}(x_i) \in \mathcal{B}_i$.* □

5.2.9 Bounds using self-avoiding walk trees

While writing this chapter, we became aware that a related method to obtain bounds on single-variable marginals has been proposed recently by Ihler [2007].[4] The method presented there uses a different local bound, which empirically seems to be less tight than ours, but has the advantage of being computationally less demanding if the domains of the random variables are large. On the other hand, the bound presented there does not use subtrees of the factor graph, but uses self-avoiding walk (SAW) trees instead. Since each subtree of the factor graph is a subtree of an SAW tree, this may lead to tighter bounds.

[4]Note that [Ihler, 2007, Lemma 5] contains an error: to obtain the correct expression, one has to replace δ with δ^2, i.e., the correct statement would be that

$$\frac{m(j)}{\delta^2 + (1-\delta^2)m(j)} \leq p(j) \leq \frac{\delta^2 m(j)}{1-(1-\delta^2)m(j)}$$

if $d\big(p(x)/m(x)\big) \leq \delta$ (where p and m should both be normalized).

The idea of using a self-avoiding walk tree for calculating marginal probabilities seems to be generally attributed to Weitz [2006], but can already be found in [Scott and Sokal, 2005]. In this subsection, we show how this idea can be combined with the propagation of bounding boxes. The result Theorem 5.13 will turn out to be an improvement over Theorem 5.10 in case there are only pairwise interactions, whereas in the general case, Theorem 5.10 often yields tighter bounds empirically.

Definition 5.12 *Let $\mathcal{G} := (\mathcal{V}, \mathcal{F}, \mathcal{E})$ be a factor graph and let $i \in \mathcal{V}$. A* self-avoiding walk (SAW) *starting at $i \in \mathcal{V}$ of length $n \in \mathbb{N}^*$ is a sequence $\alpha = (\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n) \in (\mathcal{V} \cup \mathcal{F})^n$ that*

(i) starts at $i \in \mathcal{V}$, i.e., $\alpha_1 = i$;

(ii) subsequently visits neighboring nodes in the factor graph, i.e., $\alpha_{j+1} \in N_{\alpha_j}$ for all $j = 1, 2, \ldots, n-1$;

(iii) does not backtrack, i.e., $\alpha_j \neq \alpha_{j+2}$ for all $j = 1, 2, \ldots, n-2$;

(iv) the first $n-1$ nodes are all different, i.e., $\alpha_j \neq \alpha_k$ if $j \neq k$ for $j, k \in \{1, 2, \ldots, n-1\}$.[5]

The set of all self-avoiding walks starting at $i \in \mathcal{V}$ has a natural tree structure, defined by declaring each SAW $(\alpha_1, \alpha_2, \ldots, \alpha_n, \alpha_{n+1})$ to be a child of the SAW $(\alpha_1, \alpha_2, \ldots, \alpha_n)$, for all $n \in \mathbb{N}^*$; the resulting tree is called the self-avoiding walk (SAW) tree with root $i \in \mathcal{V}$, *denoted $T_{\mathcal{G}}^{SAW}(i)$.*

Note that the name "self-avoiding walk tree" is slightly inaccurate, because the last node of a SAW may have been visited already. In general, the SAW tree can be much larger than the original factor graph. Following Ihler [2007], we call a leaf node in the SAW tree a *cycle-induced leaf node* if it contains a cycle (i.e., if its final node has been visited before in the same walk), and call it a *dead-end leaf node* otherwise. We denote the parent of node α in the SAW tree by $\mathrm{par}(\alpha)$ and we denote its children by $\mathrm{ch}(\alpha)$. The final node of a SAW $\alpha = (\alpha_1, \ldots, \alpha_n)$ is denoted by $\mathcal{G}(\alpha) = \alpha_n$. An example of a SAW tree for our running example factor graph is shown in figure 5.9(b).

Let $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$ be a factor graph and let $i \in \mathcal{V}$. We now define a propagation algorithm on the SAW tree $T_{\mathcal{G}}^{SAW}(i)$, where each node $\alpha \in T_{\mathcal{G}}^{SAW}(i)$ (except for the root i) sends a message $\mathcal{B}_{\alpha \to \mathrm{par}(\alpha)}$ to its parent node $\mathrm{par}(\alpha) \in T_{\mathcal{G}}^{SAW}(i)$. Each cycle-induced leaf node of $T_{\mathcal{G}}^{SAW}(i)$ sends a simplex to its parent node: if α is a cycle-induced leaf node, then

$$\mathcal{B}_{\alpha \to \mathrm{par}(\alpha)} = \begin{cases} \mathcal{P}_{\mathcal{G}(\alpha)} & \text{if } \mathcal{G}(\alpha) \in \mathcal{V} \\ \mathcal{P}_{\mathcal{G}(\mathrm{par}(\alpha))} & \text{if } \mathcal{G}(\alpha) \in \mathcal{F}. \end{cases} \tag{5.3}$$

[5] Note that (iii) almost follows from (iv), except for the condition that $\alpha_{n-2} \neq \alpha_n$.

All other nodes α in the SAW tree (i.e., the dead-end leaf nodes and the nodes with children, except for the root i) send a message according to the following rules. If $\mathcal{G}(\alpha) \in \mathcal{V}$,

$$\mathcal{B}_{\alpha \to \mathrm{par}(\alpha)} = \begin{cases} \prod_{\beta \in \mathrm{ch}(\alpha)} \mathcal{B}_{\beta \to \alpha} & \text{if all } \mathcal{B}_{\beta \to \alpha} \text{ are boxes} \\ \mathcal{P}_{\mathcal{G}(\alpha)} & \text{if at least one of the } \mathcal{B}_{\beta \to \alpha} \text{ is a simplex.} \end{cases} \tag{5.4}$$

On the other hand, if $\mathcal{G}(\alpha) \in \mathcal{F}$,

$$\mathcal{B}_{\alpha \to \mathrm{par}(\alpha)} = \mathcal{BN}\left(\sum_{x_{\setminus \mathcal{G}(\mathrm{par}(\alpha))}} \psi_{\mathcal{G}(\alpha)} \mathcal{B}\left(\prod_{\beta \in \mathrm{ch}(\alpha)} \mathcal{B}_{\beta \to \alpha} \right) \right). \tag{5.5}$$

The final "belief" at the root node $i \in \mathcal{V}$ is defined as:

$$\mathcal{B}_i = \begin{cases} \mathcal{BN}\left(\prod_{\beta \in \mathrm{ch}(i)} \mathcal{B}_{\beta \to i} \right) & \text{if all } \mathcal{B}_{\beta \to i} \text{ are boxes} \\ \mathcal{P}_{\mathcal{G}(i)} & \text{if at least one of the } \mathcal{B}_{\beta \to i} \text{ is a simplex.} \end{cases} \tag{5.6}$$

We will refer to this algorithm as "box propagation on the SAW tree"; it is similar to the propagation algorithm for boxes on subtrees of the factor graph that we defined earlier. However, note that whereas (5.2) bounds a sum-product assuming that incoming measures factorize, (5.5) is a looser bound that also holds if the incoming measures do not necessarily factorize. In the special case where the factor depends only on two variables, the updates (5.2) and (5.5) are identical.

Theorem 5.13 *Let $\mathcal{G} := (\mathcal{V}, \mathcal{F}, \mathcal{E})$ be a factor graph. Let $i \in \mathcal{V}$ and let $T_{\mathcal{G}}^{SAW}(i)$ be the SAW tree with root i. Then $\mathbb{P}(x_i) \in \mathcal{B}_i$, where $\mathcal{B}_i$ is the bounding box that results from propagating bounds on the SAW tree $T_{\mathcal{G}}^{SAW}(i)$ according to equations (5.3)–(5.6).*

The following lemma, illustrated in figure 5.10, plays a crucial role in the proof of the theorem. It seems to be related to the so-called "telegraph expansion" used in Weitz [2006].

Lemma 5.14 *Let $A, C \subseteq \mathcal{V}$ be two disjoint sets of variable indices and let $\Psi \in \mathcal{M}_{A \cup C}$ be a factor depending on (some of) the variables in $A \cup C$. Then:*

$$\mathcal{N}\left(\sum_{x_C} \Psi \right) \in \mathcal{B}\left(\prod_{i \in A} B_i \right)$$

where

$$B_i := \mathcal{BN}\left(\sum_{x_{A \setminus i}} \sum_{x_C} \Psi \mathcal{Q}^*_{A \setminus i} \right).$$

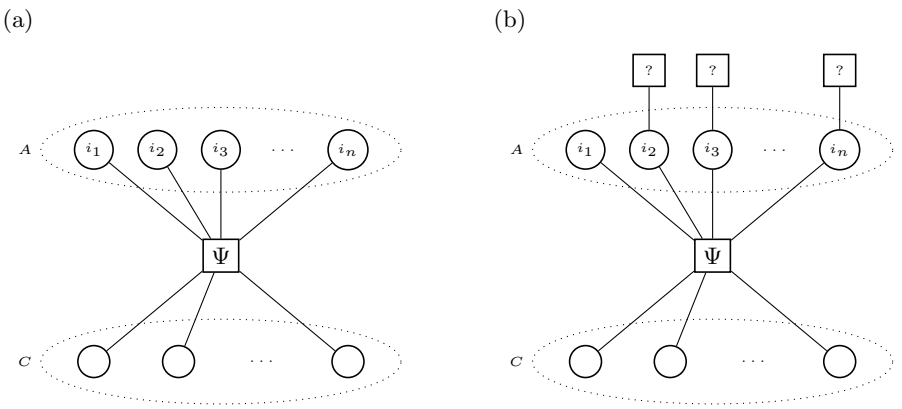


Figure 5.10: The second basic lemma: the marginal on x_A in (a) is contained in the bounding box of the product of smallest bounding boxes B_i for $i \in A$, where (b) the smallest bounding box B_i is obtained by putting arbitrary factors on the other variables in $A \setminus \{i\}$ and calculating the smallest bounding box on i, illustrated here for the case $i = i_1$.

Proof. We assume that $C = \emptyset$; the more general case then follows from this special case by replacing Ψ by $\sum_{x_C} \Psi$.

Let $A = \{i_1, i_2, \ldots, i_n\}$ and let $\underline{\Psi}_i$, $\overline{\Psi}_i$ be the lower and upper bounds corresponding to B_i, for all $i \in A$. For each $k = 1, 2, \ldots, n$, note that

$$\left(\prod_{l=1}^{k-1} \mathbf{1}_{i_l}\right)\left(\prod_{l=k+1}^{n} \delta_{x_{i_l}}\right) \in \mathcal{Q}^*_{A\setminus i_k},$$

for all $x_{\{i_{k+1},\ldots,i_n\}} \in \mathcal{X}_{\{i_{k+1},\ldots,i_n\}}$. Therefore, we obtain from the definition of B_{i_k} that

$$\forall x_A \in \mathcal{X}_A : \qquad \underline{\Psi}_{i_k} \quad \leq \quad \frac{\sum_{x_{i_{k-1}}} \cdots \sum_{x_{i_1}} \Psi}{\sum_{x_{i_k}} \sum_{x_{i_{k-1}}} \cdots \sum_{x_{i_1}} \Psi} \quad \leq \quad \overline{\Psi}_{i_k}$$

for all $k = 1, 2, \ldots, n$. Taking the product of these n inequalities yields

$$\prod_{k=1}^{n} \underline{\Psi}_{i_k} \leq \mathcal{N}\Psi \leq \prod_{k=1}^{n} \overline{\Psi}_{i_k}$$

pointwise, and therefore $\mathcal{N}\Psi \in \mathcal{B}\left(\prod_{k=1}^{n} B_{i_k}\right)$. □

The following corollary is somewhat elaborate to state, but readily follows from the previous lemma after attaching a factor I that depends on all nodes in A and one additional newly introduced node i:

Corollary 5.15 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$ be a factor graph. Let $i \in \mathcal{V}$ with exactly one neighbor in $\mathcal{F}$, say $N_i = \{I\}$. Then $\mathbb{P}(x_i) \in B_i$ where*

$$B_i = \mathcal{BN}\left(\sum_{x_{N_I \setminus i}} \psi_I \mathcal{B}\left(\prod_{k \in N_I \setminus i} B_k^{\setminus I}\right)\right) \tag{5.7}$$

and

$$B_k^{\setminus I} = \mathcal{BN}\left(\sum_{x_{\mathcal{V} \setminus \{i,k\}}} \Psi_{\mathcal{F} \setminus I} \mathcal{Q}^*_{N_I \setminus \{i,k\}}\right)$$

with

$$\Psi_{\mathcal{F} \setminus I} := \prod_{J \in \mathcal{F} \setminus I} \psi_J.$$

□

We now proceed with a sketch of the proof of Theorem 5.13, which was inspired by [Ihler, 2007].

Proof sketch of Theorem 5.13. The proof proceeds using structural induction, recursively transforming the original factor graph $\mathcal{G}$ into the SAW tree $T_{\mathcal{G}}^{SAW}(i)$, refining the bound at each step, until it becomes equivalent to the result of the message propagation algorithm on the SAW tree described above in equations (5.3)–(5.6).

Let $\mathcal{G} := (\mathcal{V}, \mathcal{F}, \mathcal{E})$ be a factor graph. Let $i \in \mathcal{V}$ and let $T_{\mathcal{G}}^{SAW}(i)$ be the SAW tree with root i. Let $\{I_1, \ldots, I_n\} = N_i$.

Suppose that $n > 1$. Consider the equivalent factor graph $\mathcal{G}'$ that is obtained by creating n copies i_n of the variable node i, where each copy i_j is only connected with the factor node I_j (for $j = 1, \ldots, n$); in addition, all copies are connected with the original variable i using the delta function $\psi_\delta := \delta(x_i, x_{i_1}, \ldots, x_{i_n})$. This step is illustrated in figure 5.11(a)–(b). Applying Corollary 5.15 to $\mathcal{G}'$ yields the following bound which follows from (5.7) because of the properties of the delta function:

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\prod_{j=1}^{n} B_{i_j}^{\setminus \delta}\right) \tag{5.8}$$

where

$$B_{i_j}^{\setminus \delta} := \mathcal{BN}\left(\sum_{x_{\setminus i_j}} \left(\prod_{J \in \mathcal{F}} \psi_J\right) \mathcal{Q}^*_{\{i_1, \ldots, i_{j-1}, i_{j+1}, \ldots, i_n\}}\right) \qquad j = 1, \ldots, n.$$

In the expression on the right-hand side, the factor ψ_{I_k} implicitly depends on i_k instead of i (for all $k = 1, \ldots, n$). This bound is represented graphically in figure 5.11(c)–(d) where the gray variable nodes correspond to simplices of single-variable factors, i.e., they are meant to be multiplied with unknown single-variable factors. Note that (5.8) corresponds precisely with (5.6).

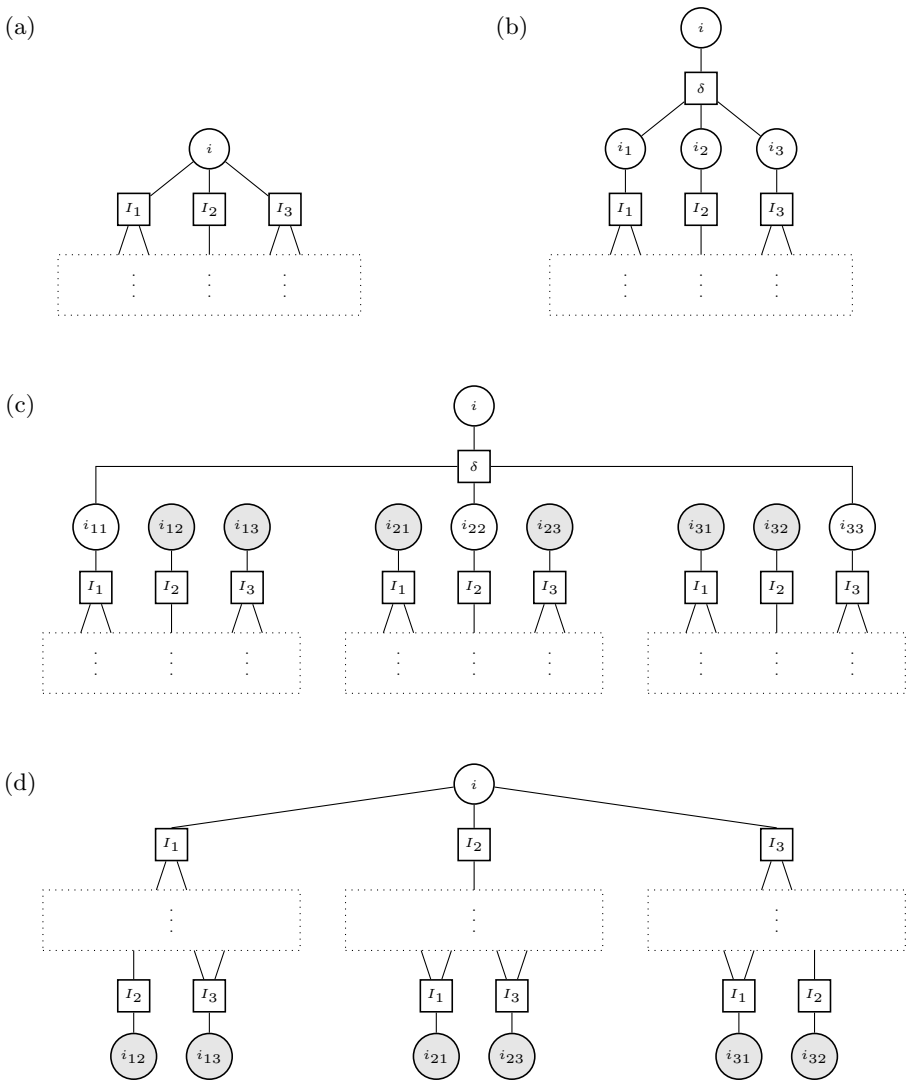


Figure 5.11: One step in the proof of Theorem 5.13: propagating bounds towards variable i in case it has more than one neighboring factor nodes $I_1, \dots, I_n$ (here, $n = 3$). Gray nodes represent added (unknown) single-variable factors. (a) Factor graph $\mathcal{G}$. (b) Equivalent factor graph $\mathcal{G}'$. (c) Result of replicating $\mathcal{G}$ n times, where in each copy $\mathcal{G}_k$ of $\mathcal{G}$, i is replaced by exactly n copies i_{kj} of i for $j = 1, \dots, n$, where i_{kj} is connected only with the factor I_j in $\mathcal{G}_k$. Then, the original variable i is connected using a delta factor with n of its copies i_{jj} for $j = 1, \dots, n$. (d) Simplification of (c) obtained by identifying i with its n copies i_{jj} for $j = 1, \dots, n$ and changing the layout slightly.

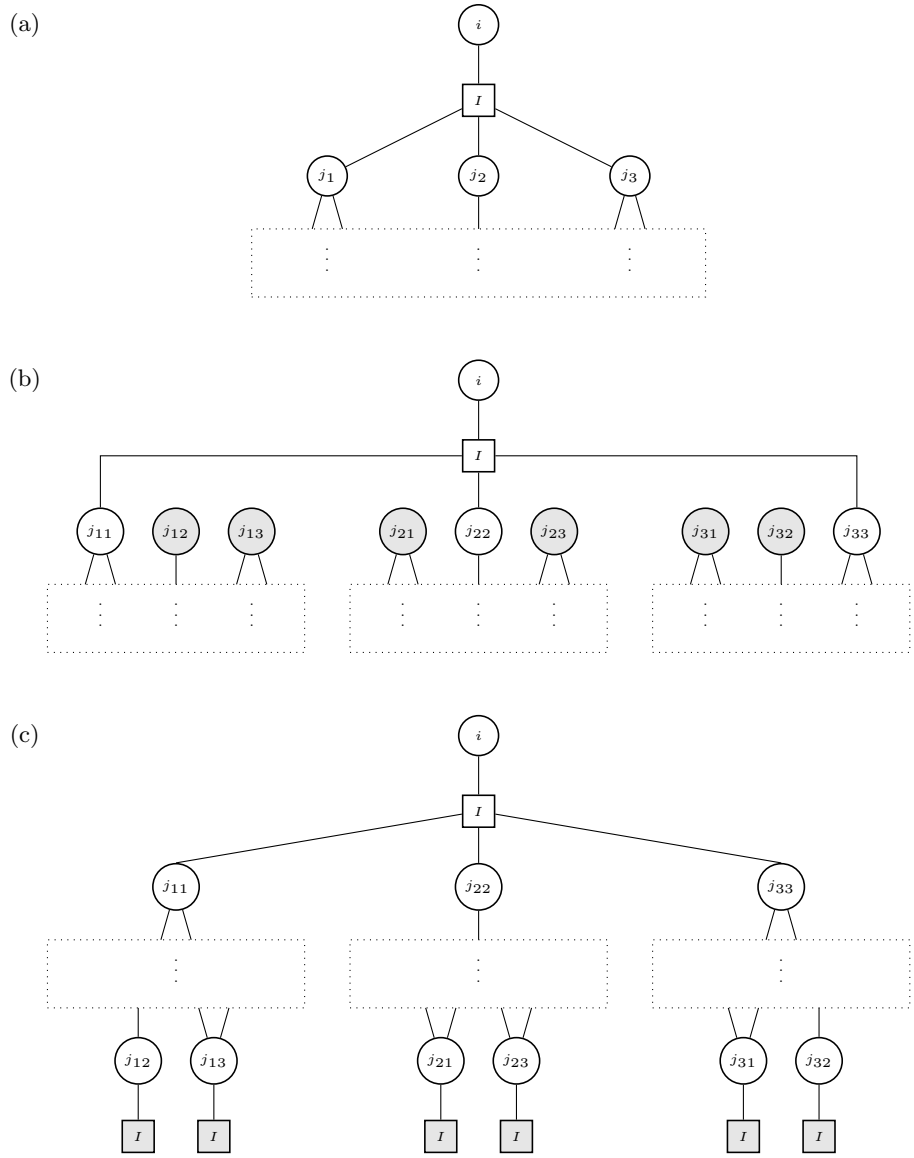


Figure 5.12: Another step in the proof of Theorem 5.13: propagating bounds towards variable i in case it has exactly one neighboring factor node I which has $m+1$ neighboring variables $\{i, j_1, \ldots, j_m\}$. (a) Factor graph $\mathcal{G}$. (b) Result of replicating $\mathcal{G} \setminus \{i, I\}$ m times and connecting the factor I with i and with copy j_{kk} of j_k for $k = 1, \ldots, m$. (c) Equivalent to (b) but with a slightly changed layout. The gray copies of I represent (unknown) single-variable factors (on their neighboring variable).

The case that $n = 1$ is simpler because there is no need to introduce the delta function. It is illustrated in figure 5.12. Let $\{I\} = N_i$ and let $\{j_1, \ldots, j_m\} = N_I \setminus i$. Applying Corollary 5.15 yields the following bound:

$$\mathbb{P}(x_i) \in \mathcal{BN}\left(\sum_{x_{N_I \setminus i}} \psi_I \mathcal{B}\left(\prod_{k=1}^{m} B_{j_k}^{\setminus I}\right)\right) \tag{5.9}$$

where

$$B_{j_k}^{\setminus I} := \mathcal{BN}\left(\sum_{x_{\setminus\{i,j_k\}}} \left(\prod_{J \in \mathcal{F} \setminus I} \psi_J\right) \mathcal{Q}^*_{\{j_1,\ldots,j_{k-1},j_{k+1},\ldots,j_m\}}\right) \qquad k = 1, \ldots, m.$$

This bound is represented graphically in figure 5.12(b)–(c) where the gray nodes correspond with simplices of single-variable factors. Note that (5.9) corresponds precisely with (5.5).

Recursively iterating the factor graph operations in figures 5.12 and 5.11, the connected component that remains in the end is precisely the SAW tree $T_{\mathcal{G}}^{SAW}(i)$; the bounds derived along the way correspond precisely with the message passing algorithm on the SAW tree described above. □

Again, the self-avoiding walk tree with root i is a subtree of the computation tree for i. This means that the bounds on the exact marginals given by Theorem 5.13 are bounds on the approximate Belief Propagation marginals (beliefs) as well.

Corollary 5.16 *In the situation described in Theorem 5.13, the final bounding box $\mathcal{B}_i$ also bounds the (approximate) Belief Propagation marginal of the root node i, i.e., $\mathbb{P}_{BP}(x_i) \in \mathcal{B}_i$.* □

5.3 Related work

There exist many other bounds on single-variable marginals. Also, bounds on the partition sum can be used to obtain bounds on single-variable marginals. For all bounds known to the authors, we will discuss how they compare with our bounds. In the following, we will denote exact marginals as $p_i(x_i) := \mathbb{P}(x_i)$ and BP marginals (beliefs) as $b_i(x_i) := \mathbb{P}_{BP}(x_i)$.

5.3.1 The Dobrushin-Tatikonda bound

Tatikonda [2003] derived a bound on the error of BP marginals using mathematical tools from Gibbs measure theory [Georgii, 1988], in particular using a result known as Dobrushin's theorem. The bounds on the error of the BP marginals can be easily translated into bounds on the exact marginals:

$$|b_i(x_i) - p_i(x_i)| \leq \epsilon \implies p_i(x_i) \in [b_i(x_i) - \epsilon, b_i(x_i) + \epsilon]$$

for all $i \in \mathcal{V}$ and $x_i \in \mathcal{X}_i$.

The Dobrushin-Tatikonda bound depends on the *girth* of the graph (the number of edges in the shortest cycle, or infinity if there is no cycle) and the properties of Dobrushin's interdependence matrix, which is a $N \times N$ matrix C. The entry C_{ij} is only nonzero if $i \in \partial j$ and in that case, the computational cost of computing its value is exponential in the size of the Markov blanket. Thus the computational complexity of the Dobrushin-Tatikonda bound is $\mathcal{O}(\max_{i \in \mathcal{V}} \#(\mathcal{X}_{\partial i}))$, plus the cost of running BP.

5.3.2 The Dobrushin-Taga-Mase bound

Inspired by the work of Tatikonda and Jordan [2002], Taga and Mase [2006a] derived another bound on the error of BP marginals, also based on Dobrushin's theorem. This bound also depends on the properties of Dobrushin's interdependence matrix and has similar computational cost. Whereas the Dobrushin-Tatikonda bound gives one bound for all variables, the Dobrushin-Taga-Mase bound gives a different bound for each variable.

5.3.3 Bound Propagation

Leisink and Kappen [2003] proposed a method called "Bound Propagation" which can be used to obtain bounds on exact marginals. The idea underlying this method is very similar to the one employed in this work, with one crucial difference. Whereas we use a cavity approach, using as basis equation

$$\mathbb{P}(x_i) \propto \sum_{x_{\partial i}} \left(\prod_{I \in N_i} \psi_I \right) \mathbb{P}^{\setminus i}(x_{\partial i}), \qquad \mathbb{P}^{\setminus i}(x_{\partial i}) \propto \sum_{x_{\mathcal{V} \setminus \Delta i}} \prod_{I \in \mathcal{F} \setminus N_i} \psi_I$$

and bound the quantity $\mathbb{P}(x_i)$ by optimizing over $\mathbb{P}^{\setminus i}(x_{\partial i})$, the basis equation employed by Bound Propagation is

$$\mathbb{P}(x_i) = \sum_{x_{\partial i}} \mathbb{P}(x_i \,|\, x_{\partial i}) \mathbb{P}(x_{\partial i})$$

and the optimization is over $\mathbb{P}(x_{\partial i})$. Unlike in our case, the computational complexity is exponential in the size of the Markov blanket, because of the required calculation of the conditional distribution $\mathbb{P}(x_i \,|\, x_{\partial i})$. On the other hand, the advantage of this approach is that a bound on $\mathbb{P}(x_j)$ for $j \in \partial i$ is also a bound on $\mathbb{P}(x_{\partial i})$, which in turn gives rise to a bound on $\mathbb{P}(x_i)$. In this way, bounds can propagate through the graphical model, eventually yielding a new (tighter) bound on $\mathbb{P}(x_{\partial i})$. Although the iteration can result in rather tight bounds, the main disadvantage of Bound Propagation is its computational cost: it is exponential in the Markov blanket and often many iterations are needed for the bounds to become tight. Indeed, for a simple tree of $N = 100$ variables, it can happen that Bound Propagation needs several minutes and still obtains very loose bounds (whereas our

bounds give the exact marginal as lower and as upper bound, i.e., they arrive at the optimally tight bound).

5.3.4 Upper and lower bounds on the partition sum

Various upper and lower bounds on the partition sum Z in (5.1) exist. An upper and a lower bound on Z can be combined to obtain bounds on marginals in the following way. First, note that the exact marginal of i satisfies

$$\mathbb{P}(x_i) = \frac{Z_i(x_i)}{Z},$$

where we defined the partition sum of the *clamped* model as follows:

$$Z_i(x_i) := \sum_{x_{\mathcal{V}\setminus\{i\}}} \prod_{I\in\mathcal{F}} \psi_I.$$

Thus, we can bound

$$\frac{Z_i^-(x_i)}{Z^+} \leq p_i(x_i) \leq \frac{Z_i^+(x_i)}{Z^-}$$

where $Z^- \leq Z \leq Z^+$ and $Z_i^-(x_i) \leq Z_i(x_i) \leq Z_i^+(x_i)$ for all $x_i \in \mathcal{X}_i$.

A well-known lower bound of the partition sum is the Mean Field bound. A tighter lower bound was derived by Leisink and Kappen [2001]. An upper bound on the log partition sum was derived by Wainwright *et al.* [2005]. Other lower and upper bounds (for the case of binary variables with pairwise interactions) have been derived by Jaakkola and Jordan [1996].

5.4 Experiments

We have done several experiments to compare the quality and computation time of various bounds empirically. For each variable in the factor graph under consideration, we calculated the *gap* for each bound $\mathcal{B}_i\left(\underline{\Psi}_i, \overline{\Psi}_i\right) \ni \mathbb{P}(x_i)$, which we define as the ℓ_0-norm $\left\|\overline{\Psi}_i - \underline{\Psi}_i\right\|_0 = \max_{x_i\in\mathcal{X}_i} \left|\overline{\Psi}_i(x_i) - \underline{\Psi}_i(x_i)\right|$.

We have used the following bounds in our comparison:

DT: Dobrushin-Tatikonda [Tatikonda, 2003, Proposition V.6].

DTM: Dobrushin-Taga-Mase [Taga and Mase, 2006a, Theorem 1].

BOUNDPROP: Bound Propagation [Leisink and Kappen, 2003], using the implementation of M. Leisink, where we chose the maximum cluster size to be $\max_{i\in\mathcal{V}} \#(\Delta i)$.

BOXPROP-SUBT: Theorem 5.10, where we used a simple breadth-first algorithm to recursively construct the subtree.

BOXPROP-SAWT: Theorem 5.13, where we truncated the SAW tree to at most 5000 nodes.

Ihler-SAWT: Ihler's bound [Ihler, 2007]. This bound has only been formulated for pairwise interactions.

Ihler-SubT: Ihler's bound [Ihler, 2007] applied on a truncated version of the SAW tree, namely on the same subtree as used in BoxProp-SubT. This bound has only been formulated for pairwise interactions.

In addition, we compared with appropriate combinations of the following bounds:

MF: Mean-field lower bound.

LK3: Third-order lower bound [Leisink and Kappen, 2001, Eq. (10)], where we took for μ_i the mean field solutions. This bound has been formulated only for the binary, pairwise case.

JJ: Refined upper bound [Jaakkola and Jordan, 1996, Section 2.2], with a greedy optimization over the parameters. This bound has been formulated only for the binary, pairwise case.

TRW: Our implementation of [Wainwright *et al.*, 2005]. This bound has been formulated only for pairwise interactions.

For reference, we calculated the Belief Propagation (BP) errors by comparing with the exact marginals, using the ℓ_0 distance as error measure.

5.4.1 Grids with binary variables

We considered a 5×5 Ising grid with binary (± 1-valued) variables, i.i.d. spin-glass nearest-neighbor interactions $J_{ij} \sim \mathcal{N}\left(0, \beta^2\right)$ and i.i.d. local fields $\theta_i \sim \mathcal{N}\left(0, \beta^2\right)$, with probability distribution

$$\mathbb{P}(x) = \frac{1}{Z} \exp\left(\sum_{i \in \mathcal{V}} \theta_i x_i + \frac{1}{2} \sum_{i \in \mathcal{V}} \sum_{j \in \partial i} J_{ij} x_i x_j\right).$$

We took one random instance of the parameters J and θ (drawn for $\beta = 1$) and scaled these parameters with the interaction strength parameter β, for which we took values in $\{10^{-2}, 10^{-1}, 1, 10\}$.

The results are shown in figure 5.13. For very weak interactions ($\beta = 10^{-2}$), BoxProp-SAWT gave the tightest bounds of all other methods, the only exception being BoundProp, which gave a somewhat tighter bound for 5 variables out of 25. For weak and moderate interactions ($\beta = 10^{-1}, 1$), BoxProp-SAWT gave the tightest bound of all methods for each variable. For strong interactions ($\beta = 10$), the results were mixed, the best methods being BoxProp-SAWT, BoundProp, MF-TRW and LK3-TRW. Of these, BoxProp-SAWT was the fastest method,

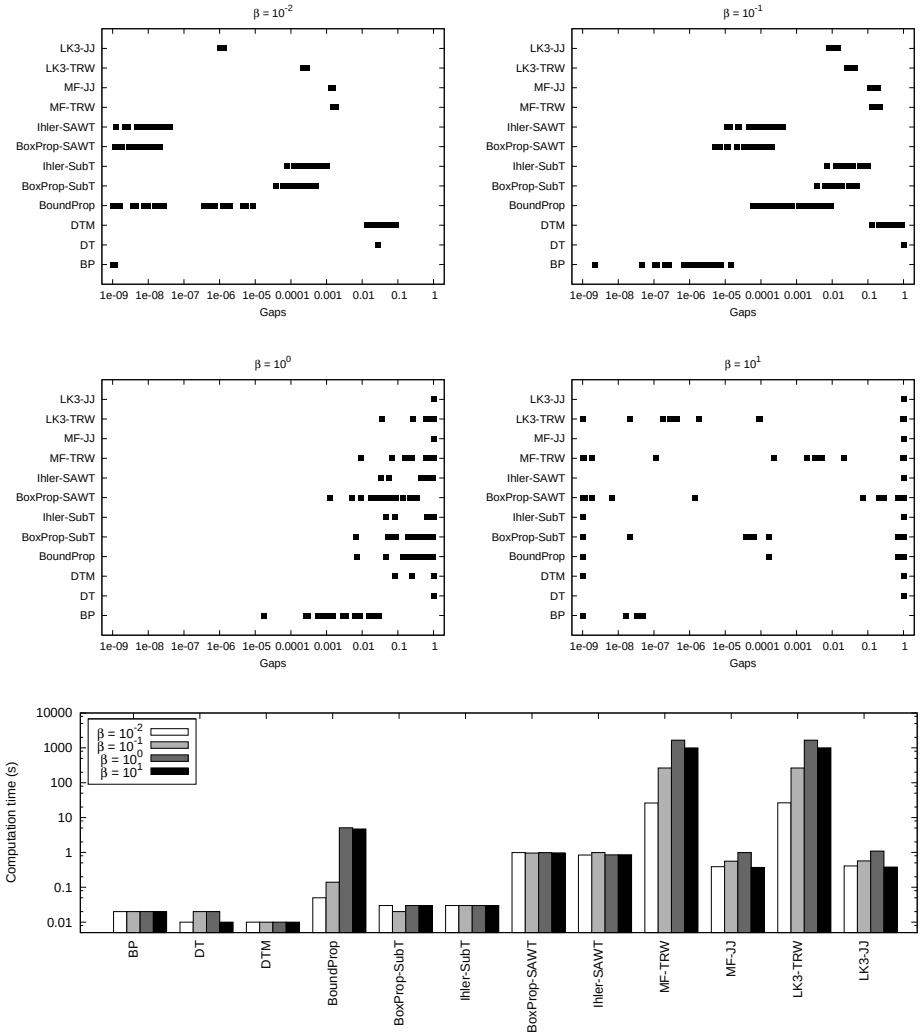


Figure 5.13: Results for grids with binary variables. The first four graphs show, for different values of the interaction strength β, the gaps of bounds on marginals calculated using different methods. Also shown for reference are the errors in the BP approximations to the same marginals. The final graph shows the total computation time for each method.

whereas the methods using TRW were the slowest.[6] For $\beta = 10$, we present scatter plots in figure 5.14 to compare the results of some methods in more detail. These

[6] We had to loosen the convergence criterion for the inner loop of TRW, otherwise it would have taken hours. Since some of the bounds are significantly tighter than the convergence criterion we used, this may suggest that one can loosen the convergence criterion for TRW even more and still obtain good results using less computation time than the results we present here. Unfortunately, it is not clear how this criterion should be chosen in an optimal way without actually trying different values and using the best one.

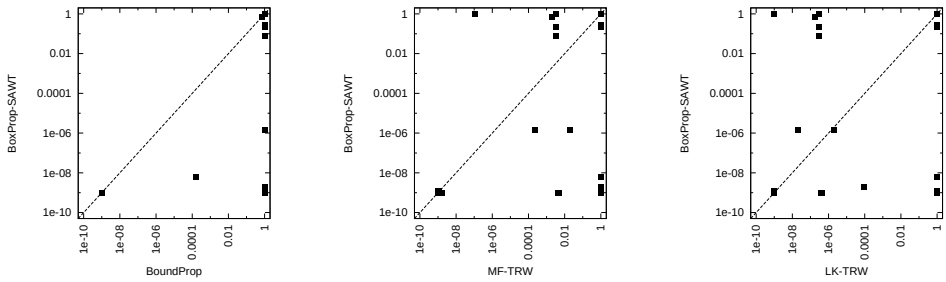


Figure 5.14: Scatter plots comparing some methods in detail for grids with binary variables for strong interactions ($\beta = 10$).

plots illustrate that the tightness of bounds can vary widely over methods and variables.

Among the methods yielding the tightest bounds, the computation time of our bounds is relatively low in general; only for low interaction strengths, BOUNDPROP is faster tahn BOXPROP-SAWT. Furthermore, the computation time of our bounds does not depend on the interaction strength, in contrast with iterative methods such as BOUNDPROP and MF-TRW, which need more iterations for increasing interaction strength (as the variables become more and more correlated). Further, as expected, BOXPROP-SUBT needs less computation time than BOXPROP-SAWT but also yields less tight bounds. Another observation is that our bounds outperform the related versions of Ihler's bounds.

5.4.2 Grids with ternary variables

To evaluate the bounds beyond the special case of binary variables, we have performed experiments on a 5×5 grid with ternary variables and pairwise factors between nearest-neighbor variables on the grid. The entries of the factors were i.i.d., drawn by taking a random number from a normal distribution $\mathcal{N}\left(0, \beta^2\right)$ with mean 0 and standard deviation β and taking the exp of that random number.

The results are shown in figure 5.15. We have not compared with bounds involving JJ or LK3 because these methods have only been formulated originally for the case of binary variables. This time, our method BOXPROP-SAWT yielded the tightest bounds for all interaction strengths and for all variables (although this is not immediately clear from the plots).

5.4.3 Medical diagnosis

We also applied the bounds on simulated PROMEDAS patient cases [Wemmenhove *et al.*, 2007]. These factor graphs have binary variables and singleton, pairwise and triple interactions (containing zeros). Two examples of such factor graphs are

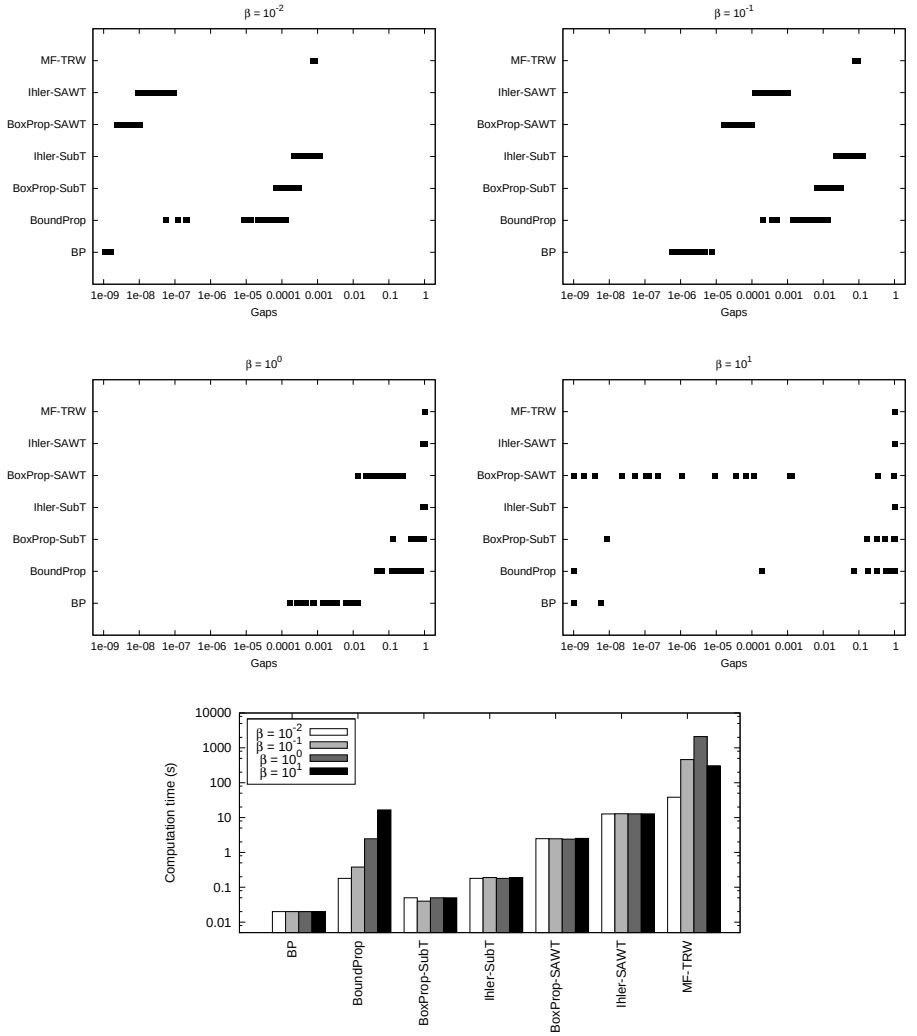


Figure 5.15: Results for grids with ternary variables. The first four graphs show, for different values of the interaction strength β, the gaps of bounds on marginals calculated using different methods. Also shown for reference are the errors in the BP approximations to the same marginals. The final graph shows the total computation time for each method.

shown in figure 5.16. Because of the triple interactions, less methods were available for comparison.

The results of various bounds for nine different, randomly generated, instances are shown in figure 5.17. The total number of variables for these nine instances was 1270. The total computation time needed for BOXPROP-SUBT was 51 s, for BOXPROP-SAWT 149 s, for BOUNDPROP more than 75000 s (we aborted the method for two instances because convergence was very slow, which explains the

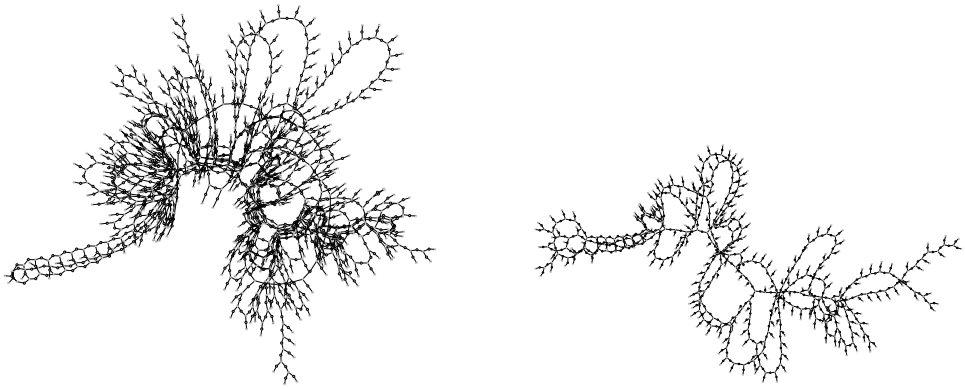


Figure 5.16: Two of the Promedas factor graphs.

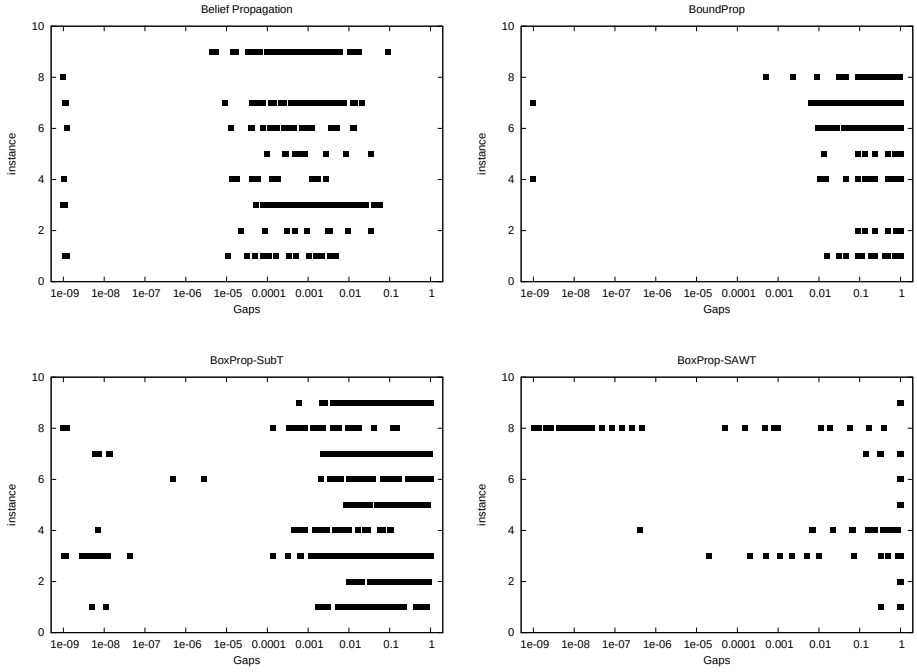


Figure 5.17: Results for nine different factor graphs corresponding to simulated Promedas patient cases. In reading order: Belief Propagation errors, BoundProp, BoxProp-SubT and BoxProp-SAWT.

missing results in the plot) and to calculate the Belief Propagation errors took 254 s. BoundProp gave the tightest bound for only 1 out of 1270 variables, BoxProp-SAWT for 5 out of 1270 variables and BoxProp-SubT gave the tightest bound for the other 1264 variables.

Interestingly, whereas for pairwise interactions, BoxProp-SAWT gives tighter

bounds than BOXPROP-SUBT, for the factor graphs considered here, the bounds calculated by BOXPROP-SAWT were generally less tight than those calculated by BOXPROP-SUBT. This is presumably due to the local bound (5.5) needed on the SAW tree, which is quite loose compared with the local bound (5.2) that assumes independent incoming bounds.

Not only does BOXPROP-SUBT give the tightest bounds for almost all variables, it is also the fastest method. Finally, note that the tightness of these bounds still varies widely depending on the instance and on the variable of interest.

5.5 Conclusion and discussion

We have derived two related novel bounds on exact single-variable marginals. Both bounds also bound the approximate Belief Propagation marginals. The bounds are calculated by propagating convex sets of measures over a subtree of the computation tree, with update equations resembling those of BP. For variables with a limited number of possible values, the bounds can be computed efficiently. Empirically, our bounds often outperform the existing state-of-the-art in that case. Although we have only shown results for factor graphs for which exact inference was still tractable (in order to be able to calculate the BP error), we would like to stress here that it is not difficult to construct factor graphs for which exact inference is no longer tractable but the bounds can still be calculated efficiently. An example are large Ising grids (of size $m \times m$ with m larger than 30). Indeed, for binary Ising grids, the computation time of the bounds (for all variables in the network) scales linearly in the number of variables, assuming that we truncate the subtrees and SAW trees to a fixed maximum size.

Whereas the results of different approximate inference methods usually cannot be combined in order to get a better estimate of marginal probabilities, for bounds one can combine different methods simply by taking the tightest bound or the intersection of the bounds. Thus it is generally a good thing to have different bounds with different properties (such as tightness and computation time).

An advantage of our methods BOXPROP-SUBT and BOXPROP-SAWT over iterative methods like BOUNDPROP and MF-TRW is that the computation time of the iterative methods is difficult to predict (since it depends on the number of iterations needed to converge which is generally not known a priori). In contrast, the computation time needed for our bounds BOXPROP-SUBT and BOXPROP-SAWT only depends on the structure of the factor graph (and the chosen subtree) and is independent of the values of the interactions. Furthermore, by truncating the tree one can trade some tightness for computation time.

By far the slowest methods turned out to be those combining the upper bound TRW with a lower bound on the partition sum. The problem here is that TRW usually needs many iterations to converge, especially for stronger interactions where convergence rate can go down significantly. In order to prevent exceedingly long

computations, we had to hand-tune the convergence criterion of TRW according to the case at hand.

BoundProp can compete in certain cases with the bounds derived here, but more often than not it turned out to be rather slow or did not yield very tight bounds. Although BoundProp also propagates bounding boxes over measures, it does this in a slightly different way which does not exploit independence as much as our bounds. On the other hand, it can propagate bounding boxes several times, refining the bounds more and more each iteration.

Regarding the related bounds BoxProp-SubT, BoxProp-SAWT and Ihler-SAWT we can draw the following conclusions. For pairwise interactions and variables that have not too many possible values, BoxProp-SAWT is the method of choice, yielding the tightest bounds without needing too much computation time. The bounds are more accurate than the bounds produced by Ihler-SAWT due to the more precise local bound that is used; the difference is largest for strong interactions. However, the computation time of this more precise local bound is exponential in the number of possible values of the variables, whereas the local bound used in Ihler-SAWT is only polynomial in the number of possible values of the variables. Therefore, if this number is large, BoxProp-SAWT may be no longer applicable in practice, whereas Ihler-SAWT still may be applicable. If factors are present that depend on more than two variables, it seems that BoxProp-SubT is the best method to obtain tight bounds, especially if the interactions are strong. Note that it is not immediately obvious how to extend Ihler-SAWT beyond pairwise interactions, so we could not compare with that method in that case.

This work also raises some new questions and opportunities for future work. First, the bounds can be used to generalize the improved conditions for convergence of Belief Propagation that were derived in [Mooij and Kappen, 2007b] beyond the special case of binary variables with pairwise interactions. Second, it may be possible to combine the various ingredients in BoundProp, BoxProp-SubT and BoxProp-SAWT in novel ways in order to obtain even better bounds. Third, it is an interesting open question whether the bounds can be extended to continuous variables in some way. Finally, although our bounds are a step forward in quantifying the error of Belief Propagation, the actual error made by BP is often at least one order of magnitude lower than the tightness of these bounds. This is due to the fact that (loopy) BP cycles information through loops in the factor graph; this cycling apparently improves the results. The interesting and still unanswered question is why it makes sense to cycle information in this way and whether this error reduction effect can be quantified.

Acknowledgments

We thank Wim Wiegerinck for several fruitful discussions, Bastian Wemmenhove for providing the Promedas test cases, and Martijn Leisink for kindly providing his implementation of Bound Propagation.

List of Notations

General

Probabilities

Graphical models, factor graphs

$\mathcal{G}$	Graph	17
$\mathcal{V}$	Index set of random variables in graphical model	17
N	Number of random variables in graphical model	17
$\mathcal{D}$	Set of directed edges	17
$\mathcal{E}$	Set of undirected edges	18
$\mathcal{F}$	Index set of factors in graphical model	19
$i, j, k, \ldots$	Indices of random variables, i.e., elements of $\mathcal{V}$	20
x_i	Random variable with index i	17
$\mathcal{X}_i$	Domain of random variable x_i	17
N_i	Factor nodes adjacent to variable i	19
∂i	Markov blanket of variable i	19
Δi	All variables that share a factor with variable i	19
$\mathrm{ch}(\alpha)$	Children of node α	132
$\mathrm{par}(i)$	Parent(s) of node i	18
$I, J, K, \ldots$	Indices of factors, i.e., elements of $\mathcal{F}$	20
ψ_I	Factor (potential, interaction) with index I	19
N_I	Variable nodes adjacent to factor I	19
Ψ_F	Product of factors in $F \subseteq \mathcal{F}$	86
Z	Partition sum (normalizing constant) of graphical model	19
θ_i	Local field on binary variable i	33
J_{ij}	Coupling of binary variables i and j	33

Belief Propagation

$\mathcal{D}$	Indices of BP messages	22
$I \to i$	Index of BP message	22
$b_i(x_i)$	BP belief (approximate marginal) of x_i	23
$b_I(x_{N_I})$	BP belief (approximate marginal) of x_{N_I}	23

Bibliography

J. R. L. de Almeida and D. J. Thouless (**1978**). "Stability of the Sherrington-Kirkpatrick solution of a spin glass model". *J. Phys. A*, 11:983.

R. J. Baxter (**1982**). *Exactly Solved Models in Statistical Mechanics*. Academic, New York.

H. Bethe (**1935**). "Statistical theory of superlattices". *Proc. R. Soc. A*, 150:552–575.

D. Bickson, D. Dolev and Y. Weiss (**2006**). "Efficient large scale content distribution". Technical Report TR-2006-07, Leibniz Center, The Hebrew University.

A. Braunstein, M. Mézard and R. Zecchina (**2005**). "Survey propagation: an algorithm for satisfiability". *Random Structures and Algorithms*, 27(2):201–226. URL `http://dx.doi.org/10.1002/rsa.20057`

A. Braunstein and R. Zecchina (**2004**). "Survey propagation as local equilibrium equations". *Journal of Statistical Mechanics: Theory and Experiment*, 2004(06):P06007. URL `http://stacks.iop.org/1742-5468/2004/P06007`

A. Brunton and C. Shu (**2006**). "Belief propagation for panorama generation". In "Proceedings of the Third International Symposium on 3D Data Processing, Visualization and Transmission", pages 885–892. Chapel Hill, NC, USA. ISBN 0-7695-2825-2.

M. Chertkov and V. Y. Chernyak (**2006**a). "Loop calculus helps to improve belief propagation and linear programming decodings of low-density-parity-check codes". *arXiv.org*, arXiv:cs/0609154v1 [cs.IT].
URL `http://arxiv.org/abs/cs/0609154v1`

M. Chertkov and V. Y. Chernyak (**2006**b). "Loop series for discrete statistical models on graphs". *Journal of Statistical Mechanics: Theory and Experiment*, 2006(06):P06009. URL `http://stacks.iop.org/1742-5468/2006/P06009`

F. Chu, Y. Wang, D. S. Parker and C. Zaniolo (**2005**). "Data cleaning using belief propagation". In "Proceedings of the 2nd international workshop on Information quality in information systems (IQIS'05)", pages 99–104. ACM Press, New York, NY, USA. ISBN 1-59593-160-0.

A. Cima, A. van den Essen, A. Gasull, E. Hubbers and F. Manosas (**1997**). "A polynomial counterexample to the Markus-Yamabe conjecture". *Advances in Mathematics*, 131(2):453–457.

G. Cooper (**1990**). "The computational complexity of probabilistic inferences". *Artificial Intelligence*, 42(2-3):393–405.

J. Coughlan and H. Shen (**2004**). "Shape matching with belief propagation: Using dynamic quantization to accomodate occlusion and clutter". In "CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)", volume 12, page 180. IEEE Computer Society, Washington, DC, USA. ISBN 0-7695-2158-4.

J. M. Coughlan and S. J. Ferreira (**2002**). "Finding deformable shapes using loopy belief propagation". In "ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part III", pages 453–468. Springer-Verlag, London, UK. ISBN 3-540-43746-0.
URL `http://www.springerlink.com/content/a044dpw9ql1kp0r0/`

C. Crick and A. Pfeffer (**2003**). "Loopy belief propagation as a basis for communication in sensor networks". In "Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)", pages 159–16. Morgan Kaufmann, San Francisco, CA.

P. Dagum and M. Luby (**1993**). "Approximating probabilistic inference in Bayesian belief networks is NP-hard". *Artifical Intelligence*, 60(1):141–153.

P. Dangauthier (**2006**). "Sudoku and belief propagation". `http://emotion.inrialpes.fr/~dangauthier/blog/2006/03/06/sudoku-and-belief-propagation/`.

R. Dechter, K. Kask and R. Mateescu (**2002**). "Iterative join-graph propagation". In "Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)", pages 128–13. Morgan Kaufmann, San Francisco, CA.

E. Deutsch (**1975**). "On matrix norms and logarithmic norms". *Numerische Mathematik*, 24(1):49–51.

J. Dieudonné (**1969**). *Foundations of Modern Analysis*, volume 10-I of *Pure and Applied Mathematics*. Academic Press, New York.

G. Elidan, I. McGraw and D. Koller (**2006**). "Residual belief propagation: Informed scheduling for asynchronous message passing". In "Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)", Boston, Massachussetts.

P. F. Felzenszwalb and D. P. Huttenlocher (**2004**). "Efficient belief propagation for early vision". In "Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)", volume 01, pages 261–268. IEEE Computer Society, Los Alamitos, CA, USA. ISSN 1063-6919.

P. F. Felzenszwalb and D. P. Huttenlocher (**2006**). "Efficient belief propagation for early vision". *International Journal of Computer Vision*, 70(1):41–54. ISSN 0920-5691.

W. T. Freeman, T. R. Jones and E. C. Pasztor (**2002**). "Example-based super-resolution". *IEEE Computer Graphics and Applications*, 22(2):56–65.

W. T. Freeman, E. C. Pasztor and O. T. Carmichael (**2000**). "Learning low-level vision". *International Journal of Computer Vision*, 40(1):25–47. ISSN 0920-5691.

B. Frey and D. MacKay (**1997**). "A revolution: Belief propagation in graphs with cycles". In "Advances in Neural Information Processing Systems", volume 10, pages 479–485.

B. J. Frey, R. Koetter and N. Petrovic (**2002**). "Very loopy belief propagation for unwrapping phase images". In "Advances in Neural Information Processing Systems (NIPS 2001)", volume 14.

R. G. Gallager (**1963**). *Low Density Parity Check Codes*. M.I.T Press, Cambridge, Massachusetts.

J. Gao and J. Shi (**2004**). "Multiple frame motion inference using belief propagation". In "Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition 2004 (FGR 2004)", pages 875–880.

H.-O. Georgii (**1988**). *Gibbs Measures and Phase Transitions*. Walter de Gruyter, Berlin.

V. Gómez, J. M. Mooij and H. J. Kappen (**2007**). "Truncating the loop series expansion for belief propagation". *Journal of Machine Learning Research*, 8:1987–2016. URL `http://www.jmlr.org/papers/volume8/gomez07a/gomez07a.pdf`

M. D. Gupta, S. Rajaram, N. Petrovic and T. S. Huang (**2005**). "Non-parametric image super-resolution using multiple images". In "IEEE International Conference on Image Processing (ICIP 2005)", volume 2, pages 89–92.

T. Heskes (**2004**). "On the uniqueness of loopy belief propagation fixed points". *Neural Computation*, 16(11):2379–2413.

T. Heskes (**2006**). "Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies". *Journal of Artificial Intelligence Research*, 26:153–190.

T. Heskes, C. A. Albers and H. J. Kappen (**2003**). "Approximate inference and constrained optimization". In "Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)", pages 313–320. Morgan Kaufmann Publishers, San Francisco, CA.

T. Heskes, M. Opper, W. Wiegerinck, O. Winther and O. Zoeter (**2005**). "Approximate inference techniques with expectation constraints". *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11015.
URL `http://stacks.iop.org/1742-5468/2005/P11015`

A. Ihler (**2007**). "Accuracy bounds for belief propagation". In "Proceedings of the 23th Annual Conference on Uncertainty in Artificial Intelligence (UAI-07)", .

A. T. Ihler, J. W. Fisher and A. S. Willsky (**2005**a). "Loopy belief propagation: Convergence and effects of message errors". *Journal of Machine Learning Research*, 6:905–936.

A. T. Ihler, J. W. Fisher and A. S. Willsky (**2005**b). "Message errors in belief propagation". In L. K. Saul, Y. Weiss and L. Bottou, editors, "Advances in Neural Information Processing Systems 17 (NIPS*2004)", pages 609–616. MIT Press, Cambridge, MA.

A. T. Ihler, J. W. F. III, R. L. Moses and A. S. Willsky (**2005**c). "Nonparametric belief propagation for self-localization of sensor networks". *IEEE Journal on Selected Areas in Communications*, 23(4):809–819.

T. Jaakkola and M. I. Jordan (**1999**). "Variational probabilistic inference and the QMR-DT network". *Journal of Artificial Intelligence Research*, 10:291–322.
URL `http://www.jair.org/papers/paper583.html`

T. S. Jaakkola and M. Jordan (**1996**). "Recursive algorithms for approximating probabilities in graphical models". In "Proc. Conf. Neural Information Processing Systems (NIPS 9)", pages 487–493. Denver, CO.

Y. Kabashima (**2003**). "Propagating beliefs in spin glass models". *J. Phys. Soc. Japan*, 72:1645–1649.

H. Kamisetty, E. P. Xing and C. J. Langmead (**2006**). "Free energy estimates of all-atom protein structures using generalized belief propagation". Technical Report CMU-CS-06-160, Carnegie Mellon University.
URL `http://reports-archive.adm.cs.cmu.edu/anon/2006/CMU-CS-06-160.pdf`

R. Kikuchi (**1951**). "A theory of cooperative phenomena". *Phys. Rev.*, 81:988–1003.

F. R. Kschischang, B. J. Frey and H.-A. Loeliger (**2001**). "Factor graphs and the sum-product algorithm". *IEEE Trans. Inform. Theory*, 47(2):498–519.

Y. A. Kuznetsov (**1988**). *Elements of Applied Bifurcation Theory*, volume 112 of *Applied Mathematical Sciences*. Springer, New York, 2nd edition.

S. L. Lauritzen and D. J. Spiegelhalter (**1988**). "Local computations with probabilities on graphical structures and their application to expert systems". *J. Royal Statistical Society B*, 50:154–227.

M. Leisink and B. Kappen (**2003**). "Bound propagation". *Journal of Artificial Intelligence Research*, 19:139–154.

M. A. R. Leisink and H. J. Kappen (**2001**). "A tighter bound for graphical models". In L. K. Saul, Y. Weiss and L. Bottou, editors, "Advances in Neural Information Processing Systems 13 (NIPS*2000)", pages 266–272. MIT Press, Cambridge, MA.

M. Leone, A. Vázquez, A. Vespignani and R. Zecchina (**2002**). "Ferromagnetic ordering in graphs with arbitrary degree distribution". *Eur. Phys. Jour. B*, 28:191 – 197.

R. J. McEliece, D. J. C. MacKay and J.-F. Cheng (**1998**). "Turbo decoding as an instance of Pearl's 'belief propagation' algorithm". *IEEE J. Select. Areas Commun.*, 16:140–152.

C. D. Meyer (**2000**). *Matrix Analysis and Applied Linear Algebra*. SIAM, Philadelphia.

M. Mézard and G. Parisi (**2001**). "The Bethe lattice spin glass revisited". *Eur. Phys. Jour. B*, 20:217–233.

M. Mézard, G. Parisi and M. A. Virasoro (**1987**). *Spin Glass Theory and Beyond*. World Scientific, Singapore.

M. Mézard and R. Zecchina (**2002**). "Random K-satisfiability: from an analytic solution to a new efficient algorithm". *Phys. Rev. E*, 66:056126.

T. Minka (**2001**). "Expectation propagation for approximate Bayesian inference". In "Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)", pages 362–369. Morgan Kaufmann Publishers, San Francisco, CA.

T. Minka and Y. Qi (**2004**). "Tree-structured approximations by expectation propagation". In S. Thrun, L. Saul and B. Schölkopf, editors, "Advances in Neural Information Processing Systems 16", MIT Press, Cambridge, MA.

A. Montanari and T. Rizzo (**2005**). "How to compute loop corrections to the Bethe approximation". *Journal of Statistical Mechanics: Theory and Experiment*, 2005(10):P10011.
URL `http://stacks.iop.org/1742-5468/2005/P10011`

J. M. Mooij and H. J. Kappen (**2005**a). "On the properties of the Bethe approximation and loopy belief propagation on binary networks". *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11012.
URL `http://stacks.iop.org/1742-5468/2005/P11012`

J. M. Mooij and H. J. Kappen (**2005**b). "Sufficient conditions for convergence of loopy belief propagation". In F. Bacchus and T. Jaakkola, editors, "Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)", pages 396–403. AUAI Press, Corvallis, Oregon.

J. M. Mooij and H. J. Kappen (**2005**c). "Validity estimates for loopy belief propagation on binary real-world networks". In L. K. Saul, Y. Weiss and L. Bottou, editors, "Advances in Neural Information Processing Systems 17 (NIPS*2004)", pages 945–952. MIT Press, Cambridge, MA.

J. M. Mooij and H. J. Kappen (**2006**). "Loop corrections for approximate inference". *arXiv.org*, arXiv:cs/0612030v1 [cs.AI].
URL `http://arxiv.org/abs/cs/0612030v1`

J. M. Mooij and H. J. Kappen (**2007**a). "Loop corrections for approximate inference on factor graphs". *Journal of Machine Learning Research*, 8:1113–1143.
URL `http://www.jmlr.org/papers/volume8/mooij07a/mooij07a.pdf`

J. M. Mooij and H. J. Kappen (**2007**b). "Sufficient conditions for convergence of the sum-product algorithm". *IEEE Transactions on Information Theory*, 53(12):4422–4437.

J. M. Mooij and H. J. Kappen (**2008**). "Novel bounds on marginal probabilities". *arXiv.org*, arXiv:0801.3797 [math.PR]. Submitted to Journal of Machine Learning Research.
URL `http://arxiv.org/abs/0801.3797`

J. M. Mooij, B. Wemmenhove, H. J. Kappen and T. Rizzo (**2007**). "Loop corrected belief propagation". In "Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07)", volume 11.
URL `http://www.stat.umn.edu/~aistat/proceedings/start.htm`

K. Murphy, Y. Weiss and M. Jordan (**1999**). "Loopy belief propagation for approximate inference: an empirical study". In "Proc. of the Conf. on Uncertainty in AI", pages 467–475.

K. Nakanishi (**1981**). "Two- and three-spin cluster theory of spin-glasses". *Physical Review B*, 23(7).

M. E. J. Newman, S. H. Strogatz and D. J. Watts (**2001**). "Random graphs with arbitrary degree distributions and their applications". *Phys. Rev. E*, 64:026118.

H. Nishimori (**2001**). *Statistical Physics of Spin Glasses and Information Processing - an Introduction*. Oxford Press, Oxford.

M. Opper and O. Winter (**2005**). "Expectation consistent approximate inference". *Journal of Machine Learning Research*, 6:2177–2204.

G. Parisi (**1988**). *Statistical Field Theory*. Addison-Wesley, Redwood City, Ca.

G. Parisi and F. Slanina (**2006**). "Loop expansion around the Bethe-Peierls approximation for lattice models". *Journal of Statistical Mechanics: Theory and Experiment*, 2006(02):L02003.
URL `http://stacks.iop.org/1742-5468/2006/L02003`

J. Pearl (**1988**). *Probabilistic Reasoning in Intelligent systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, CA.

J. Pearl (**2000**). *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

R. E. Peierls (**1936**). "On Isings model of ferromagnetism". *Proc. Cambridge Philos. Soc.*, 32:477.

A. Pelizzola (**2005**). "Cluster variation method in statistical physics and probabilistic graphical models". *J. Phys. A: Math. Gen.*, 38:R309–R339.

N. Petrovic, I. Cohen, B. J. Frey, R. Koetter and T. S. Huang (**2001**). "Enforcing integrability for surface reconstruction algorithms using belief propagation in graphical models". *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, 01:743. ISSN 1063-6919.

D. Roth (**1996**). "On the hardness of approximate reasoning". *Artificial Intelligence*, 82(1-2):273–302.

A. D. Scott and A. D. Sokal (**2005**). "The repulsive lattice gas, the independent-set polynomial, and the lovasz local lemma". *Journal of Statistical Physics*, 118:1151–1261.

M. A. Shwe, B. Middleton, D. E. Heckerman, M. Henrion, E. J. Horvitz, H. P. Lehmann and G. F. Cooper (**1991**). "Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. I. The probabilistic model and inference algorithms". *Methods of information in Medicine*, 30(4):241–255.

E. Sudderth, A. Ihler, W. Freeman and A. Willsky (**2003**). "Nonparametric belief propagation". In "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)", .

E. Sudderth, M. Mandel, W. Freeman and A. Willsky (**2004**). "Visual hand tracking using nonparametric belief propagation". Technical Report 2603, MIT LIDS Technical Report.

J. Sun, N.-N. Zheng and H.-Y. Shum (**2003**). "Stereo matching using belief propagation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800. ISSN 0162-8828.

N. Taga and S. Mase (**2006**a). "Error bounds between marginal probabilities and beliefs of loopy belief propagation algorithm". In "MICAI", pages 186–196.
URL `http://dx.doi.org/10.1007/11925231_18`

N. Taga and S. Mase (**2006**b). "On the convergence of loopy belief propagation algorithm for different update rules". *IEICE Trans. Fundamentals*, E89-A(2):575–582.

M. Takikawa and B. D'Ambrosio (**1999**). "Multiplicative factorization of noisy-max". In "Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)", pages 622–63. Morgan Kaufmann, San Francisco, CA.

K. Tanaka (**2002**). "Statistical-mechanical approach to image processing". *Journal of Physics A: Mathematical and General*, 35(37):R81–R150.
URL `http://stacks.iop.org/0305-4470/35/R81`

M. F. Tappen and W. T. Freeman (**2003**). "Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters". In "ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision", page 900. IEEE Computer Society, Washington, DC, USA. ISBN 0-7695-1950-4.

S. C. Tatikonda (**2003**). "Convergence of the sum-product algorithm". In "Proceedings 2003 IEEE Information Theory Workshop", pages 222–225.

S. C. Tatikonda and M. I. Jordan (**2002**). "Loopy belief propagation and Gibbs measures". In "Proc. of the 18th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-02)", pages 493–500. Morgan Kaufmann Publishers, San Francisco, CA.

P. J. Villeneuve and Y. Mao (**1994**). "Lifetime probability of developing lung cancer, by smoking status, canada". *Canadian Journal of Public Health*, 85(6):385–8.

M. J. Wainwright, T. Jaakkola and A. S. Willsky (**2005**). "A new class of upper bounds on the log partition function". *IEEE Transactions on Information Theory*, 51:2313–2335.

Y. Weiss (**2000**). "Correctness of local probability propagation in graphical models with loops". *Neur. Comp.*, 12:1–41.

Y. Weiss and W. T. Freeman (**2001**). "On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs". *IEEE Transactions on Information Theory*, 47(2):736–744.

D. Weitz (**2006**). "Counting independent sets up to the tree threshold". In "Proceedings ACM symposium on Theory of Computing", page 140149. ACM.

M. Welling, T. Minka and Y. W. Teh (**2005**). "Structured region graphs: Morphing EP into GBP". In "Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)", page 609. AUAI Press, Arlington, Virginia.

M. Welling and Y. W. Teh (**2001**). "Belief optimization for binary networks: A stable alternative to loopy belief propagation". In "Proc. of the 17th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-01)", pages 554–561. Morgan Kaufmann Publishers, San Francisco, CA.

B. Wemmenhove, J. M. Mooij, W. Wiegerinck, M. Leisink, H. J. Kappen and J. P. Neijt (**2007**). "Inference in the Promedas medical expert system". In "Proceedings of the 11th Conference on Artificial Intelligence in Medicine (AIME 2007)", volume 4594 of *Lecture Notes in Computer Science*, pages 456–460. Springer. ISBN 978-3-540-73598-4.

B. Wemmenhove, T. Nikoletopoulos and J. P. L. Hatchett (**2004**). "Replica symmetry breaking in the 'small world' spin glass". *arXiv.org*, arXiv:cond-mat/0405563v1 [cond-mat.dis-nn].
URL `http://arxiv.org/abs/cond-mat/0405563v1`

W. Wiegerinck and T. Heskes (**2003**). "Fractional belief propagation". In S. T. S. Becker and K. Obermayer, editors, "Advances in Neural Information Processing Systems 15", pages 438–445. MIT Press, Cambridge, MA.

W. Wiegerinck, H. J. Kappen, E. W. M. T. ter Braak, W. J. P. P. ter Burg, M. J. Nijman, Y. L. O and J. P. Neijt (**1999**). "Approximate inference for medical diagnosis". *Pattern Recognition Letters*, 20:1231–1239.

J. S. Yedidia, W. T. Freeman and Y. Weiss (**2001**). "Generalized belief propagation". In L. K. Saul, Y. Weiss and L. Bottou, editors, "Advances in Neural Information Processing Systems 13 (NIPS*2000)", pages 689–695. MIT Press, Cambridge, MA.

J. S. Yedidia, W. T. Freeman and Y. Weiss (**2005**). "Constructing free-energy approximations and generalized belief propagation algorithms". *IEEE Transactions on Information Theory*, 51(7):2282–2312.

A. L. Yuille (**2002**). "CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation". *Neural Computation*, 14(7):1691–1722.

H. Zheng, M. Daoudi and B. Jedynak (**2004**). "From maximum entropy to belief propagation: an application to skin detection". In "Proceedings of the British Machine Vision Conference (BMVC'04)", .

Summary

The research reported in this thesis focuses on the study of existing approximation techniques for inference in graphical models and on introducing novel techiques for approximate inference. *Approximate inference* can be defined as the task of calculating approximations for certain probabilities in large, complex probabilistic models, such as Bayesian networks, Markov random fields or Ising spin systems (all of which are special cases of "graphical models"). We have focussed on the case of graphical models with variables with a finite number of possible values. Calculating these probabilities is simple in principle, but computationally hard in practice, because it requires a summation over an exponential number of terms. However, the practical relevance is enormous: application areas include genetic linkage analysis, medical diagnosis, expert systems, error correcting codes, speech recognition, computer vision and many more. Because the probabilities that one is interested in cannot always be calculated exactly (given a limited amount of computation time), one often uses approximate methods, which use less computation time but only give an approximation of the quantities of interest.

In this thesis we have tried to better understand and improve upon Belief Propagation (BP), a popular approximate inference method that performs surprisingly well on many problems. It has been rediscovered many times in different fields and is therefore known under different names: "Belief Propagation", "Loopy Belief Propagation", the "Sum-Product Algorithm" and the "Bethe-Peierls approximation". BP is an iterative fixed point algorithm that minimises the Bethe free energy. It yields exact results if the underlying graphical model has no loops. If the graphical model does have loops, the BP results are approximate but can be surprisingly accurate. However, if variables become highly dependent, the error made by BP can become significant. In some cases, BP does not even converge anymore.

The results in this thesis have contributed to a better understanding of these issues. In addition, we introduced a method that improves the accuracy of BP by taking into account the influence of loops in the graphical model. Finally, we proposed a method to calculate exact bounds on marginal probabilities, which was inspired by BP. Below we summarize the various chapters in more detail.

Convergence of BP

In chapter 2, we studied the question of convergence and uniqueness of the fixed point for parallel, undamped BP. We derived novel conditions that guarantee convergence of BP to a unique fixed point, irrespective of the initial messages. In contrast with previously existing conditions, our conditions are directly applicable to arbitrary factor graphs (with discrete variables) and are shown to be valid also in the case of factors containing zeros, under some additional conditions. For the special case of binary variables with pairwise interactions, we derived stronger results that take into account local evidence (i.e., single variable factors) and the type of pair interactions (attractive or repulsive). We showed empirically that these convergence bounds outperform all known existing bounds. This research has significantly improved our theoretical understanding of the convergence properties of BP. Extensions to other update schedules have later been provided by Elidan *et al.* [2006].

BP and phase transitions

Chapter 3 studies the local stability of the "high-temperature" (also known as "weak-interactions") fixed point of BP. The relationship with the properties of the corresponding stationary point of the Bethe free energy has been investigated in detail, focussing on the special case of binary variables with pairwise interactions and zero local fields in the interest of simplicity. This has led to conclusions about the influence of damping and alternative update schedules.

In the case of ferromagnetic (attractive) interactions, we proved that the sufficient conditions for convergence of parallel undamped BP and the uniqueness of its fixed point derived in chapter 2 are sharp. Furthermore, we showed that the use of damping would only slow down convergence to this fixed point. In contrast, for antiferromagnetic (repulsive) interactions, the fixed point of undamped parallel BP becomes unstable already for significantly lower interaction strengths than for damped BP or sequential BP. Thus in this case, the use of damping or sequential updates significantly improves the range of instances on which BP converges. In the spin-glass case, we observe that damping only slightly improves convergence of BP.

Further, we showed how one can estimate analytically the temperature (interaction strength) at which the high-temperature BP fixed point becomes unstable for random graphs with arbitrary degree distributions and random interactions, extending earlier worst-case results with some average-case results. The results provide a link between statistical physics and the properties of the BP algorithm. In particular, we conclude that the behavior of BP is closely related to the phase transitions in the underlying graphical model.

Loop corrections

After having studied the limitations of BP, we showed in chapter 4 how the accuracy of BP can be improved by taking into account the influence of loops in the graphical model. Extending a method proposed by Montanari and Rizzo [2005], we propose a novel way of generalizing the BP update equations by dropping the basic BP assumption of independence of incoming messages. We call this method the Loop Correction (LC) method. Contrary to the original implementation, our extension is applicable to arbitrary factor graphs, provided that the Markov blankets are not too large.

The basic idea behind the Loop Correction method is the following. A *cavity distribution* of some variable in a graphical model is the probability distribution on its Markov blanket for a modified graphical model, in which all factors involving that variable have been removed, thereby breaking all the loops involving that variable. The Loop Correction method consists of two steps: first, the cavity distributions of all variables are estimated (using some approximate inference method), and second, these initial estimates are improved by a message-passing algorithm, which reduces the errors in the estimated cavity distributions.

If the initial cavity approximations are taken to be uniform (or completely factorized) distributions, the Loop Correction algorithm reduces to the BP algorithm. In that sense, it can be considered to be a generalization of BP. On the other hand, if the initial cavity approximations contain the effective interactions between variables in the cavity, application of the Loop Correction method usually gives significantly better results than the original (uncorrected) approximate inference algorithm used to estimate the cavity approximations.

We report the results of an extensive experimental comparison of various approximate inference methods on a variety of graphical models, including real world networks. We found that the LC error is usually approximately the square of the error of the uncorrected approximate inference method. Furthermore, the results of LC were in most cases more accurate than those of all other approximate inference methods that we considered.

Error bounds

By further developing some of the ideas from earlier chapters, we derived rigorous bounds on the exact single-variable marginals in chapter 5. These bounds also apply, by construction, to the BP marginals (beliefs). We introduced two related methods for calculating bounds: the first one propagates bounds on a subtree of the graphical model, whereas the second one propagates bounds on the (larger) "self-avoiding walk tree". The advantage of bounds over mere approximations is that the bounds also specify the accuracy of the answer, whereas with approximate methods, the accuracy is often unknown. We showed empirically that our new bounds are competitive or even outperform existing bounds in terms of quality or

computation time. We applied the bounds to factor graphs arising in a medical diagnosis application and showed that the bounds can yield nontrivial results.

Samenvatting

In dit proefschrift worden bestaande methodes voor benaderende inferentie bestudeerd en worden nieuwe methodes hiervoor geïntroduceerd. Met *benaderende inferentie* wordt hier bedoeld het berekenen van benaderingen voor kansverdelingen in veelal grote, complexe kansmodellen, zoals Bayesiaanse netwerken, Markov velden en Ising spinsystemen (met als overkoepelende term "grafische modellen"). Hierbij hebben we ons beperkt tot het geval van grafische modellen met variabelen met een eindig aantal mogelijke waarden. Het exact berekenen van deze kansverdelingen is eenvoudig in theorie, maar kan lastig zijn in de praktijk omdat het aantal termen waarover gesommeerd moet worden in het algemeen exponentieel is in het aantal variabelen in het model. De praktische relevantie is echter enorm: er zijn legio toepassingen in bijvoorbeeld de genetica, medische diagnose, fout-corrigerende codes, spraakherkenning en de visuele herkenning van objecten. Omdat de kansen waarin men geïnteresseerd is niet altijd exact te berekenen zijn (gegeven een beperkte hoeveelheid rekentijd), zoekt men vaak de toevlucht tot benaderingstechnieken, die binnen afzienbare rekentijd een (hopelijk goede) benadering van deze kansen geven.

In dit proefschrift is met name gepoogd om Belief Propagation (BP), een populaire methode voor benaderende inferentie, die verbazingwekkend goede resultaten levert voor veel problemen, beter te begrijpen en te verbeteren. BP is herhaaldelijk opnieuw ontdekt in verscheidene vakgebieden en staat daarom onder verscheidene namen bekend ("Belief Propagation", "Loopy Belief Propagation", "Sum-Product Algorithm" en de "Bethe-Peierls benadering"). BP is een iteratief vaste punten algorithme dat de zogenaamde Bethe vrije energie minimaliseert. Het levert exacte resultaten als het onderliggende grafische model geen cykels heeft. Als dit wel het geval is, zijn de resultaten van BP slechts benaderingen, maar deze benaderingen kunnen verbazingwekkend nauwkeurig zijn. Echter, als de variabelen in het grafische model sterke afhankelijkheden vertonen, kan de fout in de BP benadering aanzienlijk zijn. In sommige gevallen convergeert BP zelfs helemaal niet meer naar een vast punt.

De resultaten in dit proefschrift hebben bijgedragen aan een beter begrip van deze materie. Verder hebben we een methode geïntroduceerd waarbij de nauwkeurigheid van de benadering van BP kan worden verbeterd door rekening te houden met de cykels in het grafische model. Tot slot hebben we, geïnspireerd door BP, een methode voorgesteld waarmee exacte ongelijkheden voor kansen kunnen worden

berekend. We vatten hieronder de verschillende hoofdstukken in meer detail samen.

Convergentie van BP

In hoofdstuk 2 hebben we de vraag van convergentie naar en uniciteit van het vaste punt bestudeerd voor een bepaalde vorm van BP (namelijk met parallelle, ongedempte updates). We hebben nieuwe condities afgeleid waaronder deze variant van BP gegarandeerd convergeert naar een uniek vast punt, onafhankelijk van de beginvoorwaarden. In contrast met eerder voorhanden zijnde condities, zijn onze condities direct toepasbaar op willekeurige factorgrafen (met discrete variabelen) en ze zijn, onder bepaalde voorwaarden, ook geldig in het extreme geval dat de factoren nullen bevatten. Voor het speciale geval van binaire variabelen met paarsgewijze interacties, hebben we sterkere resultaten afgeleid die ook rekening houden met factoren die van een enkele variabele afhangen en met het type paarinteracties (aantrekkend of afstotend). We hebben empirisch aangetoond dat deze voorwaarden sterker zijn dan alle tot op heden bekende voorwaarden. Dit onderzoek heeft een significante bijdrage geleverd aan ons theoretisch begrip van de convergentie eigenschappen van BP. Uitbreidingen naar andere varianten van BP zijn later gegeven door Elidan *et al.* [2006].

BP en fase-overgangen

Hoofdstuk 3 bestudeert de lokale stabiliteit van het "hoge temperatuur" (d.w.z. zwakke interacties) vaste punt van BP. De relatie met de eigenschappen van het betreffende stationaire punt van de Bethe vrije energie is in detail bestudeerd, waarbij we ons in het belang van de eenvoud hebben beperkt tot het speciale geval van binaire variabelen met paarsgewijze interacties en geen lokale velden. Op deze wijze hebben we conclusies kunnen trekken over de invloed van "demping" en alternatieve "update schedules".

In het geval van ferromagnetische (aantrekkende) interacties hebben we bewezen dat de voldoende voorwaarden voor convergentie van parallelle, ongedempte BP en de uniciteit van het vaste punt die zijn afgeleid in hoofdstuk 2, scherp zijn. Verder hebben we aangetoond dat het gebruik van demping alleen maar zou leiden tot tragere convergentie naar dit vaste punt. Daarentegen, voor antiferromagnetische (afstotende) interacties, wordt het vaste punt van parallelle, ongedempte BP al onstabiel voor significant lagere interactiesterktes dan voor gedempte BP of sequentiële BP. Dus in dit geval kan het gebruik van demping of sequentiële updates de klasse van instanties waarvoor BP convergeert significant vergroten. In het spin-glas geval observeren we dat demping slechts een kleine bijdrage levert aan de convergentie van BP.

Verder hebben we aangetoond hoe men analytisch de temperatuur (ofwel interactie sterkte) kan bepalen waarvoor het hoge temperatuur vaste punt van BP onstabiel wordt, voor toevallige grafen met willekeurige connectiviteitsdistributies

en toevallige interacties; hiermee worden de eerdere "worst-case" resultaten uitgebreid met "average-case" resultaten. Deze resultaten vormen een brug tussen de statistische fysica en de eigenschappen van het BP algorithme. In het bijzonder concluderen we dat het gedrag van BP sterk gerelateerd is aan de fase-overgangen in het onderliggende grafische model.

Cykel correcties

Nadat we de limitaties van BP hebben bestudeerd, hebben we in hoofdstuk 4 laten zien hoe de nauwkeurigheid van BP verbeterd kan worden door rekening te houden met de invloed van cykels in het grafische model. We stellen een nieuwe manier voor om de BP vergelijkingen te generaliseren door de basis aanname van onafhankelijkheid van binnenkomende boodschappen te laten vallen. Dit werk is een uitbreiding op en variant van het werk van Montanari and Rizzo [2005]. We noemen deze methode de "cykel correctie" ("Loop Correction", afgekort LC) methode. In tegenstelling tot de oorspronkelijke implementatie, is onze aanpak toepasbaar op algemene factorgrafen, mits de Markov dekens niet te groot zijn.

Het basis idee achter de LC methode is het volgende. Een *"cavity" distributie* van een variabele in een grafisch model is de kansverdeling op de Markov deken van diezelfde variabele voor een gewijzigd grafisch model, waaruit alle factoren die van die variabele afhangen zijn verwijderd. Dit breekt alle cykels waarvan die variabele deel uitmaakt. De LC methode bestaat uit twee stappen: ten eerste worden de cavity distributies van alle variabelen geschat (gebruik makend van een benaderende inferentie methode); ten tweede worden deze schattingen verbeterd door een "message passing" algorithme, dat de fouten in de geschatte cavity distributies reduceert.

Als de aanvankelijke cavity distributies uniform zijn (of compleet gefactoriseerd), dan reduceert het LC algorithme tot het BP algorithme. In die zin kan het worden opgevat als een generalisatie van BP. Aan de andere kant, als de aanvankelijke cavity distributies de effectieve interacties tussen variabelen in de cavity bevatten, dan leidt het toepassen van het LC algorithme meestal tot significant betere resultaten dan die, die zouden zijn verkregen uit de aanvankelijke, ongecorrigeerde, schattingen van de cavity distributies.

We rapporteren de resultaten van een uitgebreide experimentele vergelijking van verscheidene benaderende inferentie methoden voor een verscheidenheid aan grafische modellen, waaronder "real-world" netwerken. We vonden dat de fout van de LC methode meestal ongeveer het kwadraat is van de fout van de ongecorrigeerde benaderende inferentie methode. Bovendien waren de resultaten van de LC methode in de meeste gevallen nauwkeuriger dan die van alle andere benaderende inferentie methoden waarmee we hebben vergeleken.

Ongelijkheden voor (BP) marginals

In hoofdstuk 5 hebben we, door het verder ontwikkelen van de ideeën van voorgaande hoofdstukken, rigoreuze ongelijkheden afgeleid voor de exacte kansverdelingen van afzonderlijke variabelen in het grafische model. Deze ongelijkheden zijn per constructie ook van toepassing op de resultaten van BP. We introduceren twee gerelateerde methodes voor het berekenen van ongelijkheden: de eerste propageert ongelijkheden over een deelboom van de factorgraaf, de tweede over de (grotere) "self-avoiding walk tree". Het voordeel van zulke ongelijkheden boven een benadering is, dat ongelijkheden tevens de nauwkeurigheid van het antwoord specificeren, terwijl bij een benadering vaak onbekend is hoe nauwkeurig het antwoord is. We hebben empirisch laten zien dat deze ongelijkheden vaak niet onderdoen voor of zelfs beter presteren (in termen van kwaliteit of rekentijd) dan andere reeds bestaande methodes voor het berekenen van ongelijkheden. We hebben de ongelijkheden toegepast op factorgrafen die voorkomen in een medische diagnose toepassing en we hebben aangetoond dat onze methodes hiervoor niet-triviale resultaten kunnen leveren.

Publications

Journal publications

R. Tolboom, N. Dam, H. ter Meulen, J. Mooij and H. Maassen (2004). "Quantitative Imaging through a Spectrograph. 1. Principles and Theory". *Applied Optics* 43(30):5669–81.

J. M. Mooij and H. J. Kappen (2005a). "On the properties of the Bethe approximation and Loopy Belief Propagation on binary networks". *Journal of Statistical Mechanics: Theory and Experiment* P11012.

J. M. Mooij and H. J. Kappen (2007a). "Loop Corrections for Approximate Inference on Factor Graphs". *Journal of Machine Learning Research* 8(May):1113–43.

J. M. Mooij and H. J. Kappen (2007b). "Sufficient Conditions for Convergence of the Sum-Product Algorithm", *IEEE Transactions on Information Theory* 53(12):4422–37.

V. Gómez, J. M. Mooij and H. J. Kappen (2007). "Truncating the loop series expansion for Belief Propagation" *Journal of Machine Learning Research* 8(Sep):1987–2016.

Conference Proceedings

J. M. Mooij and H. J. Kappen (2005c). "Validity estimates for Loopy Belief Propagation on binary real-world networks". In "Advances in Neural Information Processing Systems 17 (NIPS*2004)", pages 945–952. MIT Press, Cambridge, MA.

J. M. Mooij and H. J. Kappen (2005b). "Sufficient conditions for convergence of Loopy Belief Propagation". In "Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)", pages 396–403. AUAI Press, Corvallis, Oregon.

J. M. Mooij, B. Wemmenhove, H. J. Kappen and T. Rizzo (2007). "Loop Corrected Belief Propagation". In "Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07)", `http://www.stat.umn.edu/~aistat/proceedings/start.htm`.

B. Wemmenhove, J. M. Mooij, W. Wiegerinck, M. Leisink, H. J. Kappen and J. P. Neijt (2007). "Inference in the PROMEDAS medical expert system". In "Proceedings of the 11th Conference on Artificial Intelligence in Medicine (AIME 2007)", volume 4594 of *Lecture Notes in Computer Science*, pages 456–460. Springer. ISBN 978-

3-540-73598-4.

Preprints and technical reports

J. M. Mooij and H. J. Kappen (2004). "Spin-glass phase transitions on real-world graphs", *arXiv:cond-mat/0408378v2*.

J. M. Mooij and H. J. Kappen (2008). "Novel Bounds on Marginal Probabilities", *arXiv:0801.3797 [math.PR]*, submitted to Journal of Machine Learning Research.

Acknowledgments

I wish to thank everyone who has contributed in some way to creating this thesis.

First of all, I would like to thank Bert for his supervision. I appreciate the freedom, trust and encouragement that he gave me to pursue what gradually became more and more my personal research interests. Although he was often very busy, Bert would always make time for me whenever I needed his help. Bert, thank you for everything you have taught me and for your guidance whenever I was uncertain about how to proceed.

I am also grateful to my colleagues at SNN and the rest of our department (the name of which I lost track of because of all the reorganisations throughout the years). Thanks for all the interesting discussions about scientific and less scientific topics, the relaxed and friendly atmosphere, and all the moral, practical and other support provided. In particular, I would like to thank Kees for being my constant and enjoyable office mate from beginning to end.

Lies, Aleta, Pallieter, Ton, and Sofie: thank you for your willingness to act as laymen and take the effort to read, understand and even provide useful feedback to the first part of the introduction of my thesis. The seemingly very difficult problem of explaining the subject of this thesis to laymen transformed into a fun challenge when I imagined that the result would finally enable you and others to understand (well, more or less) what I have been doing all these years. Everything that is still unclear is of course my own responsibility.

I am indebted to the countless people who contribute to Free and Open Source Software (FOSS): their hard, often voluntary, work for the greater cause has spared me significant amounts of time, money and frustrations while producing this thesis.

My friends and climbing mates were indispensible in helping me to temporarily forget the troubling scientific problems I was working on and to just enjoy life instead. Thanks to all of you for the good times spent together.

I am also very grateful to my parents for all their practical and moral support throughout my life; without them, this thesis would not exist.

Finally, I would like to thank Aleta for all the love, support, joy and happiness she gave me during the last two and a half years. It is because of you, Aleta, that I can say without any doubt that my time as a Ph.D. student has been the best of my life so far.

Curriculum Vitae

Joris Marten Mooij was born in Nijmegen, the Netherlands, on March 11th, 1980. At primary school he skipped two years and he went to secondary school when he was ten years old, graduating *cum laude* in eleven subjects in 1996.

He then went on to study physics at what was then called the Katholieke Universiteit Nijmegen and is now known by the name Radboud Universiteit Nijmegen. In his first year studying physics, he came into contact with "real" mathematics (as opposed to the mathematics taught at secondary school) and was so fascinated by it that he decided to follow some additional mathematics courses. After a few years he had done so much mathematics courses that he decided to obtain a degree in mathematics as well. In 2002, he obtained the M.Sc. degree in Theoretical Physics *cum laude*. One year later, he also obtained the M.Sc. degree *cum laude* in Mathematics.

He decided to stay in Nijmegen and started his Ph.D. research at the same university under supervision of Prof. Bert Kappen on the subject of approximate inference in graphical models. About four years later, after a short intermezzo as an intern at Microsoft Research (Cambridge, UK), he started working as a postdoc at the Max Planck Institute for Biological Cybernetics in Tübingen, Germany.