Florian Kulmer, BSc B.A.(Econ.)

Self-Confident Belief Propagation

An Approach for Iterative Improvement of the Bethe Approximation

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Information and Computer Engineering

submitted to

Graz University of Technology

Supervisor

Assoc. Prof. Dipl.-Ing. Dr.mont. Franz Pernkopf
Dipl.-Ing. Christian Knoll, BSc

Signal Processing and Speech Communications Laboratory

Graz, March 2018

Abstract

Probabilistic graphical models (PGMs) are often used to represent probabilistic distributions over many random variables. Belief propagation (BP) is a prominent tool for probabilistic inference. On PGMs without loops BP performs exact inference. But PGMs often contain loops in practice; therefore, exact inference with BP is not possible anymore. Moreover, for PGMs containing loops it is not guaranteed that BP converges at all, and multiple solutions can exist, which are not necessarily accurate.
In this work we introduce self-confident belief propagation (SBP). SBP solves some problems of BP by gradually accounting for the pairwise potentials and iteratively improving the Bethe approximation. On Ising models SBP starts with neglecting the pairwise potentials, and follows a smooth solution path towards the desired solution. The solution of SBP is unique, stable and accurate. Even in the cases where BP does not converge, SBP provides a good solution. Additionally, we provide an adaption of SBP by restricting the runtime. We call this method SBP early stopping (SBP_{ES}).
We evaluate SBP on different PGMs with Ising potentials and show that SBP improves the accuracy of BP significantly. SBP is more accurate whenever BP converges, and obtains a unique, stable and accurate solution whenever BP fails to converge. SBP_{ES} does even improve the performance of SBP. Even compared to Gibbs sampling SBP and SBP_{ES} perform superior in terms of accuracy and runtime.

Kurzfassung

Probabilistische grafische Modelle (PGMs) werden häufig verwendet, um probabilistische Verteilungen über viele Zufallsvariablen darzustellen. Belief Propagation (BP) ist ein wichtiges Werkzeug für die probabilistische Inferenz auf diesen PGMs. Auf PGMs ohne Schleifen führt BP eine exakte Inferenz durch. Doch in der Praxis enthalten PGMs oft Schleifen. Daher ist die exakte Inferenz mit BP nicht mehr möglich und einige Probleme von BP treten auf. Es ist nicht garantiert, dass BP überhaupt konvergiert, und es können mehrere Lösungen existieren, die nicht unbedingt genau sind.

In dieser Arbeit präsentieren wir Self-Confident Belief Propagation (SBP). SBP löst diese Probleme von BP, indem es die Bethe-Approximation iterativ verbessert. Auf Ising-Modellen berücksichtigt SBP nach und nach die paarweisen Potentiale. Ausgehend von der Vernachlässigung der paarweisen Potentiale folgt SBP einem glatten Lösungsweg zur gewünschten Lösung. Die Lösung von SBP ist eindeutig, stabil und genau. Selbst in den Fällen, in denen BP nicht konvergiert, bietet SBP eine gute Lösung. Zusätzlich bieten wir eine Anpassung von SBP durch die Beschränkung der Laufzeit. Wir nennen diese Methode SBP Early Stopping (SBP_{ES}).

Wir evaluieren SBP auf verschiedenen PGMs mit Ising-Potentialen und zeigen, dass SBP die Genauigkeit von BP signifikant verbessert. SBP ist exakter in den Fällen, wenn BP konvergiert, und liefert eine eindeutige, stabile und genaue Lösung, wenn BP nicht konvergiert. SBP_{ES} verbessert sogar die Leistung von SBP. SBP und SBP_{ES} sind hinsichtlich Genauigkeit und Laufzeit Gibbs Sampling überlegen.

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

date	(signature)

Contents

1 Introduction **1**
1.1 Aim and Contributions . . . 1
1.2 Outline . . . 2

2 Theoretical Background **3**
2.1 Graph Theory . . . 3
2.1.1 Grid Graph . . . 3
2.1.2 Complete Graph . . . 4
2.1.3 Random Graph . . . 4
2.2 Probabilistic Graphical Models . . . 5
2.3 Inference . . . 5
2.3.1 Junction Tree Algorithm . . . 6
2.3.2 Belief Propagation . . . 7
2.3.3 Minimizing Bethe Free Energy . . . 9
2.3.4 Gibbs Sampling . . . 10
2.4 Model Specification . . . 11

3 Self-Confident Belief Propagation (SBP) **13**
3.1 Algorithm Description . . . 13
3.2 Practical Considerations . . . 15
3.2.1 Early Stopping . . . 16

4 Experiments **19**
4.1 Reference Methods . . . 19
4.2 Evaluation Criteria . . . 19
4.3 SBP Parameters . . . 20
4.4 Approximating the Minimum of the Bethe Free Energy . . . 22
4.5 Attractive Models . . . 24
4.6 General Models . . . 26
4.7 Discussion . . . 29

5 Conclusion **31**

1 Introduction

High dimensional probability distributions over many random variables are often represented with probabilistic graphical models (PGMs) in an intuitive and simply way. In various fields – including signal processing, information theory, artificial intelligence, computer vision, statistical physics, and digital communications – it is a important problem to perform inference (Frey, 1998; Koller and Friedman, 2009; MacKay, 2002). We are interested in performing inference on PGMs with many loops. While message passing algorithms, i.e. belief propagation (BP) (Pearl, 1988) or sum-product algorithm (Bishop, 2006) are exact on tree-structured models, they are not for models with loops. We distinguish between two forms: exact and approximate inference. On PGMs without any loops (i.e., tree-structured) it is simple to apply exact inference. Message-passing algorithms are a prominent tool for this task. One of them is belief propagation (BP) (Pearl, 1988), also known as sum-product algorithm (Bishop, 2006). BP passes around messages between nodes. These messages contain local information and information about neighboring nodes. BP can be applied to Bayesian networks, Markov random fields, and factor graphs.
With BP two important problems of PGMs can be solved: computing the marginal distribution, and evaluating the partition function. These problems are related to each other. In practice PGMs often contain loops. Loopy PGMs are NP-hard to solve; therefore, they can not be solved exactly in reasonable time (Cooper, 1990). We can run BP on PGMs with loops as well. This is also known as loopy BP, but we refer to it as BP for the remaining of this work. On loopy PGMs BP runs until all messages remain unchanged. We call these unchanged messages fixed points.
Obtaining the exact marginal distribution can also be considered as minimizing the Gibbs free energy $\mathcal{F}$ of the system. But minimizing $\mathcal{F}$ is not always feasible. Instead of considering all other nodes to obtain the marginal distribution, we relax the problem by only considering the neighboring nodes; thus, we approximate $\mathcal{F}$ by the Bethe free energy $\mathcal{F}_\mathcal{B}$. Applying BP on the relaxed problem is equivalent to minimizing the Bethe free energy $\mathcal{F}_\mathcal{B}$. Any fixed point of BP corresponds to a local minimum of $\mathcal{F}_\mathcal{B}$. But it is not guaranteed that the minimum of $\mathcal{F}_\mathcal{B}$ is equal to the minimum of $\mathcal{F}$. Additionally, $\mathcal{F}_\mathcal{B}$ may have multiple minima and local minima can occur as well. Therefore, applying BP on loopy PGMs can lead to multiple fixed points, and the fixed points are not necessarily accurate (Mooij and Kappen, 2007; Weiss, 2000). Another important problem of BP is that it is not guaranteed that the fixed points are stable; thus, BP does not converge at all. These problems of BP on loopy PGMs, are strongly depending on the message update-rule, initialization and other parameters.

1.1 Aim and Contributions

In this work we present self-confident belief propagation (SBP). SBP solves the problems of multiple, unstable and inaccurate fixed points of BP. The observation that strong pairwise potentials reduce accuracy and deteriorate the convergence properties (Knoll and Pernkopf, 2017) inspired us to ease the problem of inference by: solving a simple problem first and subsequently modifying the pairwise potentials to the desired values. SBP iteratively refines

the Bethe approximation and guides itself towards an accurate solution. This is achieved by solving a deterministic sequence of models by BP; this sequence converges in constant runtime to a unique fixed point, which is independent of any hyper-parameters. We show optimality of the selected fixed point for restricted models and argue why SBP is expected to obtain good fixed points for general models. Especially if BP does not converge, SBP exhibits its advantages and provides the best possible approximation of the Bethe approximation under the constraint to be stable with respect to BP.

The principle of SBP is explained in the following example: Imagine a group of people discussing about a topic. Within the group there are two different opinions and they have to find a compromise to which everyone agrees to. Everyone has its own initial beliefs, but everyone's opinion also depends on the other members of the group in a certain way. To some of them there is strong confidence, to others a weak one, or even none at all. Now consider a very chatty group. Then it will be hard to find a common group solution. Everyone's opinion will change all the time, depending on who they are talking with. This is similar to BP when it does not converge and the propagated beliefs are changing all the time. By only allowing little interactions, the own initial beliefs tend to overrule others' beliefs. Each person's opinion does not change much and the group quickly settles to a compromise. By slowly increasing the interactions the compromise may change a bit but is expected to be quickly obtained.
SBP does the same with PGMs. Every node neglects dependencies with other nodes. By increasing the pairwise potentials to the desired ones in every iteration, SBP keeps track of the BP solutions.

We evaluate SBP on grid graphs, complete graphs, and random graphs with Ising potentials and show superior performance in terms of accuracy in comparison to BP with and without damping. SBP provides a valuable tool to perform approximate inference for *easy* cases where BP provides multiple solutions and for *hard* cases where BP does not work at all, despite the existence of an accurate fixed point. Compared to Gibbs sampling, SBP achieves more accurate results in a fraction of runtime. Moreover, it is very simple to adapt BP to SBP and we hope that the ease of use lowers the hurdle for practical applications.
We further provide an adaption of SBP. By restricting the runtime of SBP we can even improve the accuracy. We call this method SBP early stopping (SBP_{ES}).

Part of this work was prepared for publication and is presented in (Knoll et al., 2018). The initial idea of SBP and the theoretical background was provided by Christian Knoll. My part covers the implementation, the practical considerations and the experiments.

1.2 Outline

This thesis is structured as follows: **Section 2** provides some background information on probabilistic graphical models and different inference methods such as junction tree algorithm, belief propagation, Gibbs sampling, and methods that minimize the Bethe free energy directly. We present our proposed algorithm SBP and its adaption SBP_{ES} as well as important properties in **Section 3**. We empirically evaluate SBP and SBP_{ES} in **Section 4**: the influence of different parameters on the performance is presented in Section 4.3 and our proposed methods are applied and compared to other methods in Section 4.4, Section 4.5, and Section 4.6. Finally we conclude this work in **Section 5**.

2 Theoretical Background

In this work we are using probabilistic graphical models (PGMs) to represent probability distributions over many random variables in an intuitive and simply readable way. PGMs are very popular in the fields of computer vision, speech and signal processing, and in the area of artificial intelligence (Pernkopf et al., 2014). There are different sorts of these models: undirected graphical models (Markov networks), directed graphical models (Bayesian networks), and factor graphs (directed and undirected). In this work we focus on undirected graphical models. We start with the definition of graphs in Section 2.1. Subsequently we define potentials and set up the PGM in Section 2.2. In Section 2.3 we present different methods for exact and approximate inference. Finally, we specifiy our model in Section 2.4.

2.1 Graph Theory

Let us consider an undirected graph $\mathcal{G} = (\mathbf{X}, \mathbf{E})$, where $\mathbf{X} = \{X_1, \dots, X_N\}$ is the set of $N = |\mathbf{X}|$ nodes, and $\mathbf{E}$ is the set of undirected edges. Two nodes X_i and X_j are joined by an edge e_{ij} if $e_{ij} \in \mathbf{E}$. The total number of edges is defined as $E = |\mathbf{E}|$. We denote the set of neighbors of X_i by $\partial(X_i) = \{X_j \in \mathbf{X} : e_{ij} \in \mathbf{E}\}$. The *degree* d_i of a node X_i is defined by the total number of neighbors $d_i = |\partial(X_i)|$. The *mean degree* $\langle d \rangle$ of a graph is equivalent to $\frac{2E}{N}$.

Direct relationships between two nodes are represented by edges, while the notion of paths describes indirect relationships across several nodes (Pernkopf et al., 2014). A sequence of nodes $\{X_1, \dots, X_n\}$, with $X_1, \dots, X_n \in \mathbf{X}$ is a *path* from X_1 to X_n if for all $i \in \{1, \dots, n-1\}$: $e_{i(i+1)} \in \mathbf{E}$. An undirected graph is connected if for any X_i, X_j there is a path between these two nodes. If any two nodes are connected by only one path it is called a tree. A set of nodes $\mathcal{C} \subseteq \mathbf{X}$ is a clique if all nodes in $\mathcal{C}$ form a complete graph (cf. Section 2.1.2), i.e., all nodes $X_i \in \mathcal{C}$ are connected by an edge. A clique is maximal if it has the largest possible size, i.e. adding another node makes it no longer complete, i.e., let $X_i \in \{\mathbf{X} \setminus \mathcal{C}\}$ then $\mathcal{C} \cup X_i$ is not complete.

In contrast to undirected graphs, directed graphs have only directed edges, and mixed graphs can have directed and undirected edges. Throughout this work we are only considering undirected graphs. We refer to them as undirected graph and graph equivalently.
In this work we only use connected graphs and distinguish between three different forms: grid graphs, complete graphs, and random graphs.

2.1.1 Grid Graph

A grid or lattice graph has the form of a regular tiling in the Euclidean space $\mathbb{R}^n$ (West, 2000). We only consider grid graphs in $\mathbb{R}^2$. An infinite large grid graph, i.e., $N = \infty$, has a mean degree $\langle d \rangle = 4$. An example of a grid graph with $N = 9$ is shown in Figure 2.1.

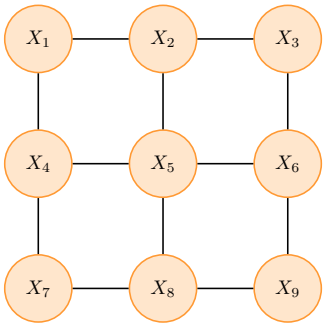

Figure 2.1: Grid graph ($N = 9$ nodes, 3×3).

2.1.2 Complete Graph

A complete or fully connected graph has an edge e_{ij} for any pair of nodes X_i, X_j: consequently $E = \frac{N \cdot (N-1)}{2}$ and $\langle d \rangle = N - 1$. A complete graph with $N = 4$ nodes is shown in Figure 2.2.

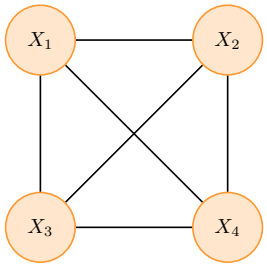

Figure 2.2: Complete graph with $N = 4$ nodes.

2.1.3 Random Graph

For a random graphs the number of nodes N is fixed, but the nodes are connected randomly. We use a Gilbert random graph $G_{N,\langle d \rangle}$ (Gilbert, 1959) where two nodes are connected with probability $P(e_{ij} \in \mathbf{E}) = \frac{\langle d \rangle}{N-1}$. While grid graphs and random graphs are always connected, random graphs are not necessarily connected. We however, consider only connected random graphs and discard any graph that is not connected. A random graph with $N = 7$ nodes and $\langle d \rangle = 2$ is shown in Figure 2.3.

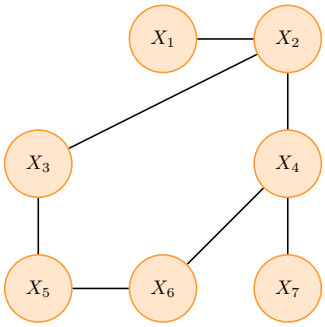

Figure 2.3: Random graph with $N = 7$ nodes and $\langle d \rangle = 2$.

2.2 Probabilistic Graphical Models

A random variable (RV) X maps from each outcome in the outcome space Ω into a value x in the event space $\mathbb{S}$, i.e., $X : \Omega \mapsto \mathbb{S}$. $P_X(X = x)$ is the probability that X takes the value $x \in \mathbb{S}$.

Let us define a probabilistic graphical model $\mathcal{U} = (\mathcal{G}, \Psi)$ where $\Psi = \{\Phi_1, \cdots, \Phi_K\}$ is the set of all K potentials. We define the set of RVs $\mathbf{X} = \{X_1, X_2, \cdots, X_N\}$, where every RV corresponds to a node in $\mathcal{G}$. We denote one specific configuration by $\mathbf{x}$ and the set of all possible configurations by $\mathcal{X}$.

Let $\Psi_C = \{\Phi_{C_1}, \cdots, \Phi_{C_L}\}$ have all potentials specified over the maximal cliques C_l of the nodes (Pearl, 1988, p.105). Then, the joint distribution factorizes to

$$P_{\mathbf{X}}(\mathbf{x}) := P_{\mathbf{X}}(\mathbf{X} = \mathbf{x}) = \frac{1}{\mathcal{Z}} \prod_{l=1}^{L} \Phi_{C_l}(\mathbf{x}_{C_l}), \tag{2.1}$$

2.3 Inference

Two important problems of inference are:
(i) obtaining the marginal distribution

$$P_{\mathbf{X_m}} = \sum_{X_i \in \{\mathbf{X} \setminus \mathbf{X_m}\}} P_{\mathbf{X}}, \tag{2.2}$$

where $\mathbf{X_m} \subset \mathbf{X}$ may be any set of RVs and the sum in (2.2) goes over all RVs except $\mathbf{X_m}$.
(ii) evaluation of the partition function $\mathcal{Z}$, which is the normalization function of the joint distribution such that $\sum_{\mathbf{x} \in \mathcal{X}} P_{\mathbf{X}}(\mathbf{x}) = 1$. The joint distribution can also be described by *Boltzmann's Law*:

$$P_{\mathbf{X}}(\mathbf{x}) = \frac{1}{\mathcal{Z}} \cdot \mathrm{e}^{-\mathcal{E}_X(\mathbf{x})}, \tag{2.3}$$

where $\mathcal{E}_X(\mathbf{x})$ is the energy of the configuration $\mathbf{x}$. It follows with (2.1) that

$$\begin{aligned} \mathcal{E}_X(\mathbf{x}) &= -\ln \prod_{l=1}^{L} \Phi_{C_l}(\mathbf{x}_{C_l}) \\ &= -\sum_{l=1}^{L} \ln \Phi_{C_l}(\mathbf{x}_{C_l}). \end{aligned} \tag{2.4}$$

We denote the average energy by $E(P_{\mathbf{X}})$ and the entropy by $S(P_{\mathbf{X}})$. Then evaluating the partition function is equivalent to minimizing the Gibbs free energy

$$\begin{aligned} \mathcal{F}(P_{\mathbf{X}}) &:= E(P_{\mathbf{X}}) - S(P_{\mathbf{X}}) \\ &= \sum_{\mathbf{x} \in \mathcal{X}} P_{\mathbf{X}}(\mathbf{x}) \cdot \mathcal{E}_X(\mathbf{x}) + \sum_{\mathbf{x} \in \mathcal{X}} P_{\mathbf{X}}(\mathbf{x}) \cdot \ln P_{\mathbf{X}}(\mathbf{x}) \end{aligned} \tag{2.5}$$

over the set of all globally realizable marginals (Wainwright et al., 2008); the system is in an equilibrium at the minimum of $\mathcal{F}$. In fact $\mathcal{Z} = \mathrm{e}^{-\mathcal{F}^*}$ where $\mathcal{F}^* = \min \mathcal{F}(P_{\mathbf{X}})$. For the remainder of this work we will only make the dependence of the Gibbs free energy on $P_{\mathbf{X}}(\mathbf{x})$ explicit if required.

Bot problems can be solved exactly (*exact inference*) for small PGMs, but are intractable as N becomes large (Cooper, 1990). Therefore, it may be necessary to resort to approximate methods (*approximate inference*).

Exact Inference

We show two different methods to perform exact inference:
(i) direct calculation of the marginals P_{X_i}: we compute the joint probability of the PGM and sum over all other RVs $\mathbf{X_m} = \{\mathbf{X} \backslash X_i\}$ according to (2.2). The time complexity of this method is $\mathcal{O}(k^N)$, where $k = |\mathbb{S}|$ is the number of states (Pernkopf et al., 2014). For tree-structured PGMs, however, one can interchange the summations and reduce the complexity to $\mathcal{O}(k^2)$ (Pernkopf et al., 2014). Note that this is equivalent to applying BP (cf. Section 2.3.2).
(ii) by using the Junction Tree algorithm (Lauritzen and Spiegelhalter, 1988): the Junction Tree algorithm can reduce the time complexity problem, by transforming the loopy PGM into a junction tree. On this junction tree BP performs exact inference (cf. Section 2.3.1).

Approximate Inference

We can reduce the time complexity of exact inference by relaxing the problem. This allows us – by accepting an error – to approximate the marginals within appropriate runtime. In this work we use the following methods for approximate inference: BP, Gibbs sampling, minimizing the Bethe free energy, and SBP.

2.3.1 Junction Tree Algorithm

The first step is to create a Junction Tree is to create an undirected graph by replacing each directed edge with an undirected one. The next step is to construct a chordal or triangulated graph by adding edges. This process is called triangulation (Koller and Friedman, 2009). A graph is called triangulated if there is no cycle of length ≥ 4 without an edge joining two non-neighboring nodes (Pernkopf et al., 2014). Triangulation can for example be done with the elimination algorithm (Koller and Friedman, 2009). An example of an triangulated graph is shown in Figure 2.4, where the edges added by the elimination algorithm are depicted as dashed.
Subsequently we must obtain the maximal cliques of the triangulated graph. The maximal cliques in the example shown in Figure 2.4 are $\{X_1, X_2, X_3\}$, $\{X_2, X_3, X_4\}$, $\{X_3, X_4, X_5\}$ and $\{X_4, X_5, X_6\}$.

Finally, the maximal cliques are connected by edges such that the intersection property holds; i.e., if a variable X_i exists in two cliques $\mathcal{C}_i$ and $\mathcal{C}_j$, it also must be in the path connecting $\mathcal{C}_i$ and $\mathcal{C}_j$. This path is sometimes referred to as separator set $\mathcal{S} = \mathcal{C}_i \cap \mathcal{C}_j$ (Pernkopf et al., 2014). The junction tree corresponding to the triangulated graph in Figure 2.4 is shown in Figure 2.5. The final step is to run BP on the junction tree. As we have mentioned in chapter 1, BP finds the exact marginals on a graph without loops. The exact marginals P_{X_i} can then be obtained by marginalizing over a clique that includes X_i.
The computational complexity of exact inference with the junction tree algorithm is exponential in the size of the largest clique (Pernkopf et al., 2014). This size is also called tree-width. Therefore, exact inference is intractable on large and dense PGMs and approximate inference algorithms are necessary.

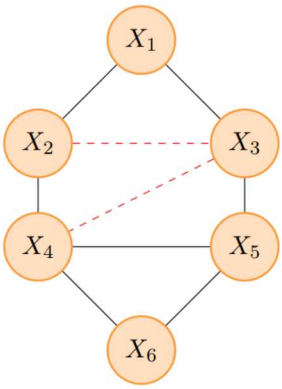

Figure 2.4: Triangulated graph; added edges are dashed (Pernkopf et al., 2014).

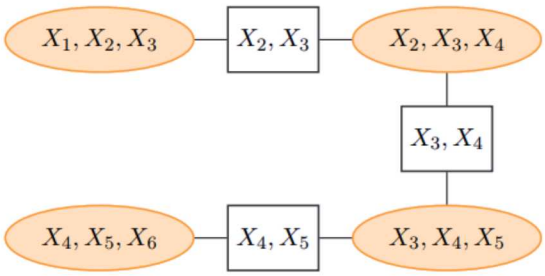

Figure 2.5: Junction Tree (Pernkopf et al., 2014).

2.3.2 Belief Propagation

On loopy graphs Belief Propagation (BP) approximates[1] the marginals by recursively exchanging local messages between random variables of the graph. After initialization the messages are exchanged between the nodes. The message from X_i to X_j for state x_j at iteration $n+1$ is given by the the following update rule:

$$\mu_{i,j}^{n+1}(x_j) \propto \sum_{x_i \in \mathbb{S}} \Phi_{X_i,X_j}(x_i, x_j) \Phi_{X_i}(x_i) \prod_{X_k \in \{\partial(X_i) \setminus X_j\}} \mu_{k,i}^{n}(x_i). \tag{2.6}$$

The initial messages are either set to a given initial value (e.g. equal probability for all states), or they are initialized randomly. In practice the messages are often normalized so that

$$\sum_{x_j \in \mathbb{S}} \mu_{i,j}^{n+1}(x_j) = 1. \tag{2.7}$$

The message update (2.6) for $\mu_{i,j}^{n+1}(x_j)$ is illustrated in Figure 2.6. The product of all incoming messages at node X_i for a certain state x_i is multiplied with the local potential $\Phi_{X_i}(x_i)$ and the pairwise potential $\Phi_{X_i,X_j}(x_i, x_j)$. The sum over all states of these products is proportional to the message $\mu_{i,j}^{n+1}(x_j)$ for iteration $n+1$.

We are using the same notations as in Knoll et al. (2018). Let $\underline{\mu} = \{\mu_{i,j}(x_j) : e_{ij} \in \mathbf{E}\}$ be the set of all messages; then we denote the mapping induced by (2.6) as $\underline{\mu}^{n+1} = \mathcal{BP}(\underline{\mu}^n)$. If all successive messages remain unchanged, i.e., $\underline{\mu}^{n+1} = \underline{\mu}^n$, then BP converged to a *fixed point* $\underline{\mu}^*$. We further write $\underline{\mu}^* = \mathcal{BP}^*(\underline{\mu}^0)$, where $\mathcal{BP}^*$ performs BP until convergence.
If the messages oscillate, one can try to achieve convergence by changing the scheduling (El-

[1] BP performs exact inference on tree-structured graphs.

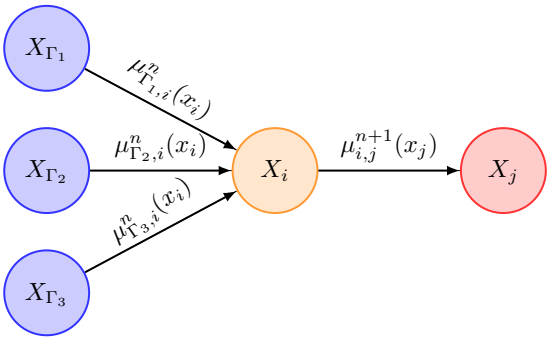

Figure 2.6: Update of message $\mu^{n+1}_{i,j}(x_j)$ from X_i to X_i. The sum over all states of the product of all incoming messages at X_i (except to the one from X_j) is sent to node X_j (Pernkopf et al., 2014).

idan et al., 2006; Knoll et al., 2015; Sutton and McCallum, 2007). In one iteration the message-update is repeated for all messages and can either be done parallel, or sequential. The former updates all messages at the same time while the latter updates only one message at the time. With sequential scheduling the message ordering can either be fixed or random. With parallel scheduling there is a higher chance of oscillating messages, whereas the sequential scheduling methods (round-robin and random) can improve this behavior.
Another method to achieve better convergence behavior is by replacing the messages with a convex combination of the last messages (Murphy et al., 1999). This method is known as damping where a damping parameter $\epsilon \in [0, 1)$ specifies $\underline{\mu}^{n+1} = (1-\epsilon)\mathcal{BP}(\underline{\mu}^n) + \epsilon\underline{\mu}^n$.

After convergence, the singleton marginals P_{X_i} and pairwise marginals P_{X_i,X_j} are approximated by the normalized products

$$\tilde{P}_{X_i}(x_i) = \frac{1}{Z_i}\Phi_{X_i}(x_i) \prod_{X_k \in \partial(X_i)} \mu^*_{k,i}(x_i), \tag{2.8}$$

$$\tilde{P}_{X_i,X_j}(x_i,x_j) = \frac{1}{Z_{ij}}\Phi_{X_i}(x_i)\Phi_{X_j}(x_j)\Phi_{X_i,X_j}(x_i,x_j) \cdot \prod_{X_k \in \{\partial(X_i)\setminus X_j\}} \mu^*_{k,i}(x_i) \cdot \prod_{X_l \in \{\partial(X_j)\setminus X_i\}} \mu^*_{l,j}(x_j), \tag{2.9}$$

where $(Z_i, Z_{ij}) \in \mathbb{R}^*_+$ guarantee that all probabilities sum to one.
BP relaxes the problem of (2.1) by considering pairwise potentials at most, so that

$$\tilde{P}_{\mathbf{X}}(\mathbf{x}) = \frac{1}{\mathcal{Z}_{\mathcal{B}}} \prod_{e_{ij} \in \mathbf{E}} \Phi_{X_i,X_j}(x_i,x_j) \prod_{i=1}^{N} \Phi_{X_i}(x_i), \tag{2.10}$$

and approximates the marginals by the pseudomarginals

$$\tilde{P}_{\mathbf{X}_{\mathcal{B}}} = \{\tilde{P}_{X_i}, \tilde{P}_{X_i,X_j} : X_i \in \mathbf{X}, e_{ij} \in \mathbf{E}\}. \tag{2.11}$$

BP may not converge to its fixed points; moreover, the fixed points are not guaranteed to be close to the exact solution. The approximated solution of BP deviates from the exact one because of over-counting of information (Weiss, 2000). We present an illustrative example to explain this in Figure 2.7. In this loopy PGM BP sends information from X_1 to X_2 ($\mu^n_{1,2}(x_i)$). X_2 processes this information and forwards it to X_3 ($\mu^n_{2,3}(x_i)$). X_3 sends this back to X_1

$(\mu^n_{3,1}(x_i))$. X_1 receives parts of the information from X_2 that it aims to neglect (2.6). Thus, the information is over-counted and distorts the solution.
Note that the behavior of BP for does not only depend on $\mathcal{G}$ (Weiss, 2000) but on the potentials Ψ as well (Knoll et al., 2017; Mooij and Kappen, 2007) (cf. Section 2.4 for further discussions).

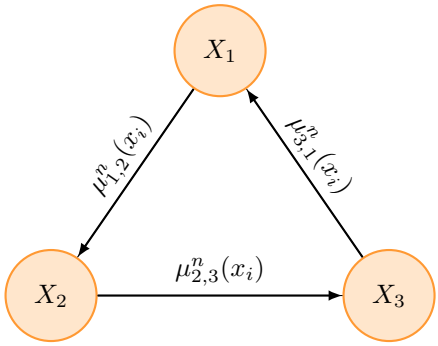

Figure 2.7: In a loopy PGM BP sends information from X_1 to X_2 $(\mu^n_{1,2}(x_i))$. X_2 processes this information and forwards it to X_3 $(\mu^n_{2,3}(x_i))$. X_3 sends this back to X_1 $(\mu^n_{3,1}(x_i))$. X_1 receives parts of the information from X_2 that it aims to neglect (2.6)*. Thus, the information is over-counted and distorts the solution.*

2.3.3 Minimizing Bethe Free Energy

The observation that fixed points of BP are in a one-to-one correspondence with stationary points of the Bethe free energy $\mathcal{F}_\mathcal{B}$ (cf. Yedidia et al. (2005)) paved the way for a better understanding of BP and provided an alternative approach by minimizing the Bethe free energy directly.

Let us consider only pairwise potentials at most as in (2.10), then we obtain the Bethe free energy

$$\mathcal{F}_\mathcal{B}(\tilde{P}_{\mathbf{X}_\mathcal{B}}) := E_\mathcal{B}(\tilde{P}_{\mathbf{X}_\mathcal{B}}) - S_\mathcal{B}(\tilde{P}_{\mathbf{X}_\mathcal{B}}) \tag{2.12}$$

with the pseudomarginals $\tilde{P}_{\mathbf{X}_\mathcal{B}}$ (2.11), where

$$E_\mathcal{B}(\tilde{P}_{\mathbf{X}_\mathcal{B}}) := -\sum_{\mathbf{x_m}:\mathbf{X_m}\in\mathbf{X}_\mathcal{B}} \tilde{P}_{\mathbf{X_m}}(\mathbf{x_m}) \cdot \ln\, \Phi_{\mathbf{X_m}}(\mathbf{x_m}) \tag{2.13}$$

is the average energy and the entropy is given by

$$S_\mathcal{B}(\tilde{P}_{\mathbf{X}_\mathcal{B}}) := -\sum_{\mathbf{x_m}:\mathbf{X_m}\in\mathbf{X}_\mathcal{B}} \tilde{P}_{\mathbf{X_m}}(\mathbf{x_m}) \cdot \ln\, \tilde{P}_{\mathbf{X_m}}(\mathbf{x_m}), \tag{2.14}$$

such that

$$\begin{aligned}\mathcal{F}_\mathcal{B}(\tilde{P}_{\mathbf{X}_\mathcal{B}}) = &\sum_{e_{ij}\in\mathbf{E}} \sum_{x_i,x_j} \tilde{P}_{X_i,X_j}(x_i,x_j) \ln \frac{\tilde{P}_{X_i,X_j}(x_i,x_j)}{\Phi_{X_i,X_j}(x_i,x_j)} \\ &- \sum_{X_i} \sum_{x_i} \tilde{P}_{X_i}(x_i) \ln \Phi_{X_i}(x_i) \\ &- \sum_{X_i} (|\partial(X_{X_i})| - 1) \sum_{x_i} \tilde{P}_{X_i}(x_i) \ln \tilde{P}_{X_i}(x_i).\end{aligned} \tag{2.15}$$

From a variational perspective BP is equivalent to minimizing $\mathcal{F}_\mathcal{B}$ (cf. Wainwright et al. (2008, pp.77)), i.e.,

$$\min(\mathcal{F}_\mathcal{B}) = \mathcal{F}_\mathcal{B}{}^* = \mathcal{F}_\mathcal{B}(\tilde{P}^*_{\mathbf{X}_\mathcal{B}}). \tag{2.16}$$

Stationary points of $\mathcal{F}_\mathcal{B}$ relate to the pseudomarginals by

$$\mathcal{F}_\mathcal{B}{}^\circ = \mathcal{F}_\mathcal{B}(\tilde{P}^\circ_{\mathbf{X}_\mathcal{B}}). \tag{2.17}$$

Note that any fixed point $\tilde{P}^\circ_{\mathbf{X}_\mathcal{B}}$ of BP corresponds to a local minimum of $\mathcal{F}_\mathcal{B}$ if BP is stable (Heskes et al., 2003); the converse, however, need not be the case, i.e., not every local minimum of $\mathcal{F}_\mathcal{B}$ corresponds to a stable fixed point (Watanabe and Fukumizu, 2009).

The correspondence between BP and $\mathcal{F}_\mathcal{B}$ was not only essential in getting a better understanding of BP, but inspired plenty methods that minimize $\mathcal{F}_\mathcal{B}$ directly (Welling and Teh, 2003; Yuille and Rangarajan, 2003). The minimization, however, is still highly non-trivial and requires good approximation methods in practice: one can further relax the problem and allow for convex surrogate free energies that provide provable convergent message passing algorithms (Globerson and Jaakkola, 2007; Hazan and Shashua, 2008; Meltzer et al., 2009; Meshi et al., 2009). Alternatively, polynomial runtime algorithms exist that approximate $\mathcal{F}_\mathcal{B}$ for restricted models: these include sparsity constraints (Shin, 2012) or require attractive models (Weller and Jebara, 2014). If both properties are fulfilled, i.e., for locally tree-like attractive models the Bethe approximation is exact and can can be optimized efficiently (Dembo et al., 2010). Note that $\mathcal{F}_\mathcal{B}$ provides an upper bound on $\mathcal{F}$ for attractive models (Ruozzi, 2013; Willsky et al., 2008).

We aim to efficiently approximate $\mathcal{F}_\mathcal{B}$ similar as in (Weller and Jebara, 2014): their approximation can be made ϵ-accurate; this, however, comes at the cost of giving up runtime guarantees for general models. Our work, on the contrary, provides an approximation in constant runtime. The approximation error, however, can not be made arbitrarily small for general models. It is worth mentioning, that both methods get rid of their respective disadvantages when restricting the models; i.e., both methods do efficiently minimize the Bethe approximation for attractive models.

2.3.4 Gibbs Sampling

Gibbs sampling is a approximate inference method that belongs to the Markov Chain Monte Carlo (MCMC) methods (Koller and Friedman, 2009). Gibbs sampling starts from an initial marginal distribution (e.g. random or equal probabilities for all states). From this distribution the next samples are sampled. Every sample depends only on the preceding one. Repeating this procedure for a large number of iterations will push the obtained marginal distribution towards the real marginal distribution.

Each sample is based on the previous sample; thus, there is a correlation between successive samples. With *thinning* (Maceachern and Berliner, 1994) we can reduce this correlation by only using, e.g., every 10th sample, while all samples in between are discarded. Using a longer chain of samples instead (e.g. 10 times longer) leads to a better approximation of the marginals (Link and Eaton, 2012).

Our implementation of Gibbs sampling is shown in Algorithm 1. Starting with the initial sample $\mathbf{s}$, for each node the state is flipped while the states of all other nodes remain fixed. The unnormalized probability equals the product of the local potential with the pairwise potentials, where the pairwise potentials are determined by the states of the neighboring nodes. If the new value of the state has a lower unnormalized probability than the old one, it will be discarded with probability α. This procedure is repeated for every node for a given maximum number of iterations N_{Gibbs}. Subsequently the marginals can be computed from

the whole set of samples $\{\mathbf{s}_{N_{Gibbs}}\}$.

Algorithm 1: Gibbs Sampling

input : PGM $\mathcal{U}$
output: $P_{\mathbf{X}}$

initialize $\mathbf{s}$
for $k \leftarrow 1$ **to** N_{Gibbs} **do**
  for $i \leftarrow 1$ **to** N **do**
    $s_i^{new} \leftarrow \bar{s}_i^{old}$
    $P_{X_i} \leftarrow$ `UnnormalizedProbability(`$i, \mathcal{U}$`)`
    $\alpha \leftarrow$ `Min(`$1, \frac{P_{X_i}(\mathbf{X}=\mathbf{s_i^{new}})}{P_{X_i}(\mathbf{X}=\mathbf{s_i^{old}})}$`)`
    With probability α: $s_i^{old} \leftarrow s_i^{new}$
  $\{\mathbf{s}_k\} \leftarrow \mathbf{s}$
$P_{\mathbf{X}} \leftarrow$ `ComputeMarginals(`$\{\mathbf{s}_{N_{Gibbs}}\}$`)`

2.4 Model Specification

We are considering the same model as in Knoll et al. (2018). We focus on binary pairwise models in which every variable X_i takes values from $\mathbb{S} = \{-1, +1\}$. These models are known as the Ising models in physics where the variables represent magnetic dipole moments of atomic spins. Let us define couplings $J_{ij} \in \mathbb{R}$ assigned to each edge $e_{ij} \in \mathbf{E}$ and a local (magnetic) field $\theta_i \in \mathbb{R}$ acting on each variable $X_i \in \mathbf{X}$. With the couplings and fields we can define the pairwise potentials

$$\Phi_{X_i,X_j}(x_i, x_j) = \exp(\frac{1}{T} J_{ij} x_i x_j), \tag{2.18}$$

and the local potentials

$$\Phi_{X_i}(x_i) = \exp(\frac{1}{T} \theta_i x_i), \tag{2.19}$$

where T is the temperature of the system. The corresponding joint distribution of a configuration $\mathbf{x}$ is given by

$$\tilde{P}_{\mathbf{X}}(\mathbf{x}) = \frac{1}{\mathcal{Z}_{\mathcal{B}}} \exp\left(\frac{1}{T} \sum_{e_{ij} \in \mathbf{E}} J_{ij} x_i x_j + \frac{1}{T} \sum_{i=1}^{N} \theta_i x_i \right). \tag{2.20}$$

For the remainder of this thesis we assume $T = 1$ and we use J_{ij} and θ_i for altering the configurations. It is sometimes more convenient to work with the minimal parameters of this model, i.e, the mean (or magnetization)

$$m_i = \mathbb{E}(X_i) = P_{X_i}(X_i = 1) - P_{X_i}(X_i = -1). \tag{2.21}$$

The mean magnetization of a model is given by:

$$\langle m \rangle = \mathbb{E}(m_i) \tag{2.22}$$

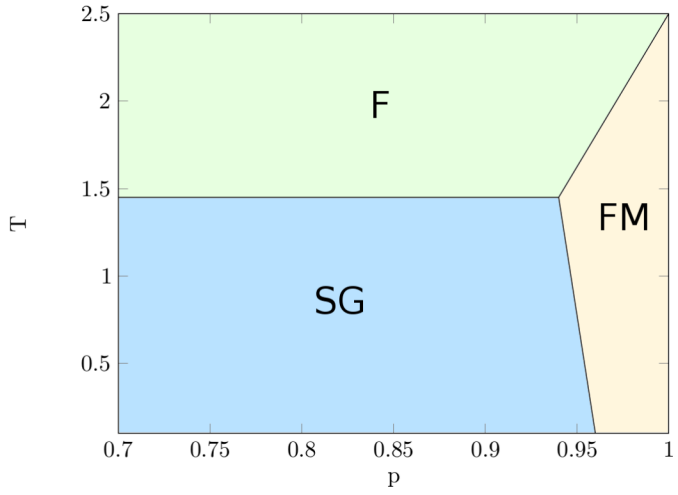

Figure 2.8: Phase diagram for a grid graph ($N = 25$ RVs, 5×5) with a common field $\theta_i = 0.1$. The couplings are drawn from $J_{ij} \in \{-1, 1\}$ with $P(J_{ij} = 1) = p$. We distinguish three different phases: Ferromagnetic with unique solution (F), ferromagnetic with multiple solutions (FM), and spin glass (SG).

We distinguish two different types of interactions: if J_{ij} is negative then the edge e_{ij} is repulsive; if J_{ij} is positive then the edge e_{ij} is attractive. We call a model $\mathcal{U}$ *attractive* if it contains only attractive edges. Note that in physics attractive models are also known as ferromagnetic models. If the model $\mathcal{U}$ also contains repulsive edges we refer to it as *general* model. In physics the general models are also known as spin glass models.

For Ising models BP does converge to a unique stable fixed point if all J_{ij} are small (relative to θ_i). For attractive models with strong couplings multiple solutions exist and BP converges to one of them. Finally, for general models with strong couplings multiple solutions may exist and BP does not converge (even if the stationary point is unique) (Knoll and Pernkopf, 2017). Note that these differences coincidence with different phases in statistical mechanics (cf. Mezard and Montanari (2009); Tatikonda and Jordan (2002); Zdeborová and Krzakala (2016)).

For general models we can distinguish between a ferromagnetic phase where BP converges to a unique fixed point (F), a ferromagnetic phase where BP converges to one of multiple fixed points (FM), and a spin glass phase (SG). The different phases depend on the field θ_i, the couplings J_{ij}, and the temperature T. The dependencies of the phases and parameters can be shown in a phase diagram (Kabashima, 2003; Saade, 2016; Zdeborová and Krzakala, 2016). A phase diagram for a grid graph ($N = 25$ RVs, 5×5) is shown in Figure 2.8. We consider $\theta_i = 0.1$ and draw the couplings from $J_{ij} \in \{-1, 1\}$ with $P(J_{ij} = 1) = p$. A low temperature and low probability of attractive edges results in the spin glass phase. In this phase BP does only converge rarely.

3 Self-Confident Belief Propagation (SBP)

In this chapter we introduce Self-Confident Belief Propagation (SBP). The underlying principle of the algorithm and a pseudocode is presented in Section 3.1. Then, we discuss the parameters of SBP and introduce SBP_{ES} in Section 3.2.

3.1 Algorithm Description[2]

The main concept is that any individual node (or RV) creates its own belief first, and their interactions with other nodes is guided by its strong initial beliefs – thus we name our proposed algorithm self-confident belief propagation.

The current understanding of BP is that strong pairwise potentials negatively influence BP and strong local potentials increase accuracy and lead to better convergence properties (Knoll and Pernkopf, 2017). Indeed, it is worth considering whether a good fixed point emerges if we start from a trivial model with independent RVs and slowly increase the potentials-strength. SBP solves this simple problem first and keeps track of the fixed point as the strength of the pairwise potentials increases. All pairwise potentials are increased by exponential scaling with $\zeta \in [0, 1]$; SBP accounts for the change and tracks the fixed point by repetitive application of BP.

The tracking of the fixed point is illustrated in Figure 3.1 for a problem where BP does not converge at all. Initially SBP obtains the marginal distribution for $\zeta = 0$ by running BP. Then, SBP estimates the marginals by successively increasing ζ and running BP. Indeed, a smooth solution path emerges and SBP is capable of tracking it. The fixed point becomes unstable for $\zeta > 0.7$. SBP stops and provides an approximate solution by using the last stable solution. In this case BP does not converge until $\zeta = 1$. Thus, the solution is not equal to a stationary point of the Bethe free energy (SBP $\neq \mathcal{F}_{\mathcal{B}}{}^{\circ}$), but the approximated marginals are already close to the exact marginals.

More formally, SBP considers an increasing length-M sequence $\{\zeta_m\}$ where $m = 1, \ldots, M$ such that $\zeta_m < \zeta_{m+1}$ and $\zeta_m \in [0, 1]$ with $\zeta_1 = 0$ and $\zeta_M = 1$. This further indexes a sequence of PGMs $\{\mathcal{U}_m\}$ that converge to the model of interest $\mathcal{U}_M = \mathcal{U}$. We further denote the fixed points of BP for $\mathcal{U}_m$ by $\underline{\mu}^*_{[m]}$. Every PGM has a set of potentials $\Psi_{[m]} = \{\Phi_{X_i,X_j}(x_i, x_j)_{[m]}, \Phi_{X_i}(x_i)_{[m]}\}$ associated, where $\Phi_{X_i}(x_i)_{[m]} = \Phi_{X_i}(x_i)$ and the pairwise potentials at index m are exponentially scaled by

$$\begin{aligned}\Phi_{X_i,X_j}(x_i, x_j)_{[m]} &= \mathrm{e}^{J_{ij}\zeta_m x_i x_j} \\ &= \Phi_{X_i,X_j}(x_i, x_j)^{\zeta_m}.\end{aligned} \tag{3.1}$$

The initialization has a major influence on the performance of BP – SBP provides favorable initialization for each instance by the fixed point of the preceding instance, i.e., SBP

[2] This section contains contributions of Christian Knoll

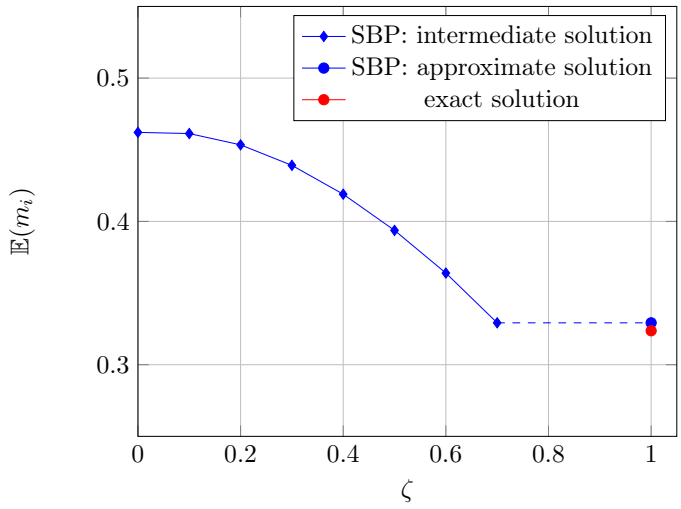

Figure 3.1: Illustrative example: SBP tracks the smooth solution path and obtains accurate marginals despite instability of the terminal fixed point.

essentially performs the composition of functions

$$\mathcal{BP}^*_{[M]}\left(\mathcal{BP}^*_{[M-1]}\left(\cdots \mathcal{BP}^*_{[1]}\left(\underline{\mu}^0_{[1]}\right)\right)\right) \tag{3.2}$$

In other words SBP is an iterative algorithm that either provides a stationary point $\mathcal{F}_\mathcal{B}{}^\circ$, or an approximation of $\mathcal{F}_\mathcal{B}{}^\circ$ if this stationary point is not stable with respect to BP. First, SBP relaxes the problem until all RVs are independent and the Bethe approximation is exact. Then, the problem is deformed into the original one by increasing ζ from zero to one. Consequently, $\mathcal{F}_\mathcal{B}$ is deformed in a continuous fashion such that the stationary point $\mathcal{F}_\mathcal{B}{}^\circ$ emerges as a continuous path. SBP keeps track of this solution with BP constantly correcting the stationary point.

SBP is (i) straightforward to implement, (ii) guaranteed to converge, and (iii) provides an accurate approximation, because of the following properties (Knoll et al., 2018):

(1) BP has a unique fixed point $\underline{\mu}^*_{[1]}$ for $\zeta_1 = 0$ that is equal to the exact solution.

(2) A smooth solution path originates from the global optimum at $\mathcal{F}_\mathcal{B}(\tilde{P}^*_{\mathcal{B}[1]})$.

(3) The solution path leads to an accurate approximation $\mathcal{F}_\mathcal{B}(\tilde{P}^\circ_{\mathbf{X}_{\mathcal{B}[m]}}) \cong \mathcal{F}_\mathcal{B}{}^*$.

(4) SBP does efficiently follow this solution path.

For attractive models Theorem 1 guarantees that SBP provides the optimal solution (Knoll et al., 2018):

Theorem 1. *Consider an attractive model with $\theta_i \geq 0$. Then, SBP obtains pseudomarginals $\tilde{P}^\circ_{\mathbf{X}_\mathcal{B}}(\zeta_m)$ such that $m_i^\circ(\zeta_m)$ increases monotone; this solution is optimal, i.e., the mean of the approximate marginals m_i are monotonically increasing, and SBP converges to the best possible solution.*

For *general models* any fixed point that is obtained by following a smooth solution path from the origin is close to the global minimum $\mathcal{F}_\mathcal{B}{}^*$. Every other fixed point must inevitable violate some local constraints and is consequently expected to be worse with respect to approximating $\mathcal{F}_\mathcal{B}{}^*$ (Knoll et al., 2018).

3.2 Practical Considerations

The performance of SBP depends on (i) the step size, and (ii) the extrapolation of the next fixed point. The scaling factor increases in every iteration from $\zeta_1 = 0$ to $\zeta_M = 1$. We can use a constant step size $\Delta\zeta = \zeta_m - \zeta_{m-1}$ which describes the distance between to successive steps. A change of $\Delta\zeta$ could affect the performance of SBP. If SBP stops before $\zeta_m = 1$, than the solution of SBP is not equal to a stationary point of the Bethe free energy (SBP $\neq \mathcal{F}_{\mathcal{B}}{}^{\circ}$). A smaller $\Delta\zeta$ can drive SBP closer to $\zeta_m = 1$; thus, push the last converged iteration closer to $\mathcal{F}_{\mathcal{B}}{}^{\circ}$. Decreasing $\Delta\zeta$, however, will inevitable increase the total number of BP iterations. Choosing $\Delta\zeta$ too large on the other hand reduces the dependencies between successive fixed points; thus, vanishes the basic idea of SBP. The problem of finding a good step size can be overcome by using an adaptive step size, so that the step size can be kept small in areas with a big change between successive fixed points and can be increased in areas where the difference is small.
In Algorithm 2 we show the pseudocode of our implementation of an adaptive step size controller. The minimum step size is fixed with $\Delta\zeta_{init}$. The adaption of the step size is on the differences between the sequence of previous fixed points $\{\underline{\mu}^*_{[m]}\}$. K is the number of fixed points, where the differences of the magnetization between these fixed points $(\langle m\rangle(\underline{\mu}^*_{[m]}) - \langle m\rangle(\underline{\mu}^*_{[m-k]}))$ is less than a given threshold. The step size $\Delta\zeta$ is adapted using following rule:

$$\Delta\zeta = \sum_{k=1}^{K} k \cdot \Delta\zeta_{init}. \tag{3.3}$$

Algorithm 2: Adaptive Step Size Controller

input : Fixed point messages $\{\underline{\mu}^*_{[m]}\}$, $\Delta\zeta_{init}$, m
output: $\Delta\zeta$

```
Δζ ← Δζ_init
threshold ← 1 · 10^−3
k ← 1
while (⟨m⟩(μ*_[m]) − ⟨m⟩(μ*_[m−k])) < threshold do
    k ← k + 1
    Δζ ← Δζ + Δζ_init · k
```

SBP initializes every iteration with the fixed point of the preceding instance (3.2). This initialization can be expanded with the extrapolation of the next fixed point. By the smooth change of the fixed points with an increase of the pairwise potentials (cf. property (2)) we can use the preceding fixed points $\underline{\mu}^*_{[m-1]}, \underline{\mu}^*_{[m-2]}, \ldots, \underline{\mu}^*_{[m-k]}$ to extrapolate the initialization:

$$\underline{\mu}^0_{[m]} = f(\underline{\mu}^*_{[m-1]}, \underline{\mu}^*_{[m-2]}, \ldots, \underline{\mu}^*_{[m-k]}). \tag{3.4}$$

Our implementation of SBP is shown as pseudocode in Algorithm 3. The maximum number of iterations for BP is denoted by $N_{BP} = 1 \cdot 10^3$ and $\underline{\mu}^0$ is initialized randomly. Setting ζ_{m+1} is either done with a fixed step size $\Delta\zeta$, where $\zeta_{m+1} = \zeta_m + \Delta\zeta$, or an adaptive step size controller (*adaptive stepsize* = 1), where $\Delta\zeta$ is a function of the previous fixed points. ExtrapolateMessages applies cubic spline extrapolation to estimate the next fixed point (cf. Section 3.2).

Algorithm 3: Self-Confident Belief Propagation (SBP)

input : Graph $G = (\mathbf{X}, \mathbf{E})$, Potentials Ψ
output: Fixed point messages $\underline{\mu}^*$

initialization $\underline{\mu}^0_{[1]} \leftarrow \underline{\mu}^0$
$m \leftarrow 1$
$\Delta\zeta_{init} \leftarrow 0.1$
$\zeta_1 \leftarrow 0$
while $\zeta \leq 1$ **do**
  $\Psi(\zeta_m) \leftarrow$ `ScalePotentials`(Ψ, ζ_m)
  $(\underline{\mu}, iterations) \leftarrow$ `BP`$(\underline{\mu}^0_{[m]}, \Psi(\zeta_m), N_{BP})$
  if $iterations < N_{BP}$ **then**
    $\underline{\mu}^*_{[m]} \leftarrow \underline{\mu}$
  else
    break
  if *adaptive stepsize* **then**
    $\zeta_{m+1} \leftarrow \zeta_m +$ `AdaptiveStepSize`$(\{\underline{\mu}^*_{[m]}\}, \Delta\zeta_{init}, m)$
  else
    $\zeta_{m+1} \leftarrow \zeta_m + \Delta\zeta_{init}$
  $\underline{\mu}^0_{[m+1]} \leftarrow$ `ExtrapolateMessages`$(\{\underline{\mu}^*_{[m]}\},\{\zeta_m\})$
  $m \leftarrow m + 1$
$\underline{\mu}^* \leftarrow \underline{\mu}^*_{[m-1]}$

3.2.1 Early Stopping

Additionally to step size and extrapolation there is another important practical consideration for SBP. Property (3) shows that the error of SBP (w.r.t. $\mathcal{F}_{\mathcal{B}}{}^{\circ}$) decreases with the number of iterations. It follows from the decrease of the error over iterations that the error also decreases over the total number of BP iterations. While accepting a certain error, we can set a maximum number of BP iterations; thus, restrict the runtime of SBP. We call this BP iteration budget. If this budget is consumed, SBP uses the last converged iteration as final result. We can call such a method SBP early stopping (SBP_{ES}). Note that the error reduces rapidly in the first iterations by only using a small budget. Whereas a major part of the budget is consumed for increasing the accuracy only slightly (cf. Section 4.4).
We expect that the MSE w.r.t. the exact solution increases for the last iterations of SBP, because BP is over-counting information (cf. Section 2.3.2).
Using an iteration budget changes the implementation of the SBP algorithm slightly. In Algorithm 4 we can see the pseudocode of SBP_{ES}. The only difference is that the algorithm stops once the budget of iterations is consumed.

Algorithm 4: Self-Confident Belief Propagation with Early Stopping (SBP_{ES})

input : Graph $G = (\mathbf{X}, \mathbf{E})$, Potentials Ψ
output: Fixed point messages $\underline{\mu}^*$

initialization $\underline{\mu}^0_{[1]} \leftarrow \underline{\mu}^0$
$m \leftarrow 1$
$\Delta\zeta_{init} \leftarrow 0.1$
$\zeta_1 \leftarrow 0$
while $\zeta \leq 1$ **do**
  $\Psi(\zeta_m) \leftarrow$ `ScalePotentials(`Ψ, ζ_m`)`
  $(\underline{\mu}, iterations) \leftarrow$ `BP(`$\underline{\mu}^0_{[m]}, \Psi(\zeta_m), N_{Budget}$`)`
  $N_{Budget} \leftarrow N_{Budget} - iterations$
  if $N_{Budget} > 0$ **then**
    $\underline{\mu}^*_{[m]} \leftarrow \underline{\mu}$
  else
    break
  if *adaptive stepsize* **then**
    $\zeta_{m+1} \leftarrow \zeta_m +$ `AdaptiveStepSize(`$\{\underline{\mu}^*_{[m]}\}, \Delta\zeta_{init}, m$`)`
  else
    $\zeta_{m+1} \leftarrow \zeta_m + \Delta\zeta_{init}$
  $\underline{\mu}^0_{[m+1]} \leftarrow$ `ExtrapolateMessages(`$\{\underline{\mu}^*_{[m]}\}, \{\zeta_m\}$`)`
  $m \leftarrow m + 1$
$\underline{\mu}^* \leftarrow \underline{\mu}^*_{[m-1]}$

4 Experiments

We apply and evaluate SBP in various experiments[3] and determine if SBP can improve the performance of BP. Additionally we evaluate the performance of SBP_{ES}. We run SBP and SBP_{ES} on grid graphs, complete graphs, and random graphs with different potentials where we distinguish between attractive and general models. In Section 4.2 we discuss the reference methods and we show the evaluation criteria in Section 4.1. In Section 4.3 we try to find good parameter settings for SBP which we will use for the experiments in Section 4.4, Section 4.5 and Section 4.6.

4.1 Reference Methods

We evaluate the performance of SBP, BP with and without damping, Gibbs sampling, and minimizing the Bethe free energy and compare it to the Junction Tree algorithm (exact) (cf. Section 2.3).
We only consider graphs where exact inference can be achieved in reasonable time. This allows us to compare the performance of SBP the exact solution.
The aim of this work is to improve BP. Therefore BP is our main reference method of all proposed approximate inference algorithms. For the experiments we use BP with a maximum number of 10^3 iterations. We use random scheduling to increase the convergence behavior of BP. Additionally we use BP with damping (BP_{D}). The weighted message updates need more BP iterations to converge. Because of the slower convergence rate of BP_{D} we set the maximum number of BP iterations for BP_{D} to 10^4. In all our experiments we set ϵ to 0.9. This is a large value, but also necessary to achieve better convergence behavior.
For our experiments we run Gibbs sampling for $N_{Gibbs} = 10^5$ iterations. We limit the number of Gibbs iterations, because even so it takes long time to compute the marginals.
For minimizing the Bethe free energy we use (Weller and Jebara, 2014) and adapt the parameter ϵ to the given PGM. We try to keep ϵ as small as possible to get the best achievable approximation in a given time. Therefore we start with $\epsilon = 0.01$. If we choose ϵ too small, it leads to a long runtime; thus, we stop the algorithm and restart it with a larger ϵ. Depending on the graph and the potentials it is not always possible to keep ϵ as small as desired. Therefore it is not guaranteed that the approximated minimum is close to the exact minimum of $\mathcal{F}_{\mathcal{B}}$.

4.2 Evaluation Criteria

Now we are describing the evaluation criteria. We use three criteria to evaluate SBP.
First, the accuracy is evaluated by averaging the mean squared error (MSE) between the approximate and the exact marginals over all N nodes. For binary RVs we can apply symmetry

[3] The experiments in Sections 4.4, 4.5, and 4.6 except the results of SBP_{ES} are from Knoll et al. (2018)

properties of the probability mass function $P_{X_i}(X_i = 1) = 1 - P_{X_i}(X_i = -1)$, so that:

$$\text{MSE} = \frac{2}{N} \sum_{i=1}^{N} |P_{X_i}(X_i = 1) - \tilde{P}_{X_i}(X_i = 1)|^2, \tag{4.1}$$

where we obtain the exact marginals P_{X_i} with the Junction Tree algorithm (cf. Section 2.3.1). Second, we define $\text{MSE}_{\mathcal{B}}$ as the MSE between the marginals of SBP and the marginals obtained by minimizing $\mathcal{F}_{\mathcal{B}}$, where $\mathcal{F}_{\mathcal{B}}{}^*$ is approximated with (Weller and Jebara, 2014) (cf. Section 2.3.3).
Third, we evaluate the runtime behavior of SBP. Therefore we compare the runtime of all methods by counting the overall BP iterations. We also report the number of iterations for Gibbs sampling by assuming that a Gibbs sampling iteration takes equally long as a BP iteration. This is a valid assumption, because in Gibbs sampling each node makes use of all neighbors to compute the acceptance-probability of the next sample. Each iteration of Gibbs sampling is similar in runtime to the BP message update.
Every experiment is executed $L = 100$ times with random potentials. We consider the problems of the existence of multiply fixed points and non-convergence by randomly initializing BP 100 times for each these L models and run it with and without damping. This procedure allows us to find most of the stable fixed points. The reported error and the runtime are averaged over all converged runs BP^* and BP^*_{D}. Note that we consider BP as converged if at least a single initialization did converge.
On the contrary, we average the error and the runtime over all L models for SBP (SBP_{all}), SBP_{ES} (SBP_{ES-all}), Gibbs sampling (Gibbs_{all}), and for minimization of the Bethe approximation ($\mathcal{F}^*_{\mathcal{B}all}$)

4.3 SBP Parameters

Besides the standard parameters of BP (e.g. maximum number of iterations, scheduling, damping), SBP has mainly two additional parameters. Step size $\Delta\zeta$ and extrapolation of the next fixed points (cf. Section 3.2). For SBP_{ES} we further need to define the budget of BP iterations. In this section we are analyzing the effects of these parameters on accuracy and runtime.

First, we analyze the effects of step size and extrapolation method on accuracy on a grid graph ($N = 25$ RVs, 5×5) with attractive and repulsive edges., where $\theta_i = 0.3$ and pairwise potentials are sampled $J_{ij} \in \{-5, 5\}$ with equal probability. For extrapolation of the next fixed point we evaluate (i) reusing the preceding fixed point (3.4) (*none*), (ii) linear extrapolation (*linear*) and, (iii) spline extrapolation with different orders (*spline 1 to 4*).
The MSE for a fixed step size $\Delta\zeta = \frac{1}{M-1}$ is presented for different extrapolation methods in Figure 4.1. SBP never converges for $\zeta_m = 1$ (i.e., $\text{SBP}_{ES} \neq \mathcal{F}_{\mathcal{B}}{}^\circ$). A step size smaller than 0.1 ($M \geq 11$) does not increase the accuracy. It is the same with different extrapolation methods. For a sufficient small step size, there is no effect of the extrapolation method on the accuracy.

Second, we analyze the runtime effects of different extrapolation methods. On an easier problem, where SBP always converges for $\zeta_m = 1$ (i.e., $\text{SBP}_{ES} = \mathcal{F}_{\mathcal{B}}{}^\circ$). We consider grid graphs ($N = 9$ RVs, 3×3) with (i) attractive edges ($J_{ij} = 2$) and, (ii) repulsive edges ($J_{ij} = -2$). For the total runtime we sum up the runtimes of (i) and (ii). With $M = 21$, we have enough SBP iterations, to see bigger differences between the methods. The results are shown in Table 4.1. SBP needs longer to converge with repulsive edges than with attractive

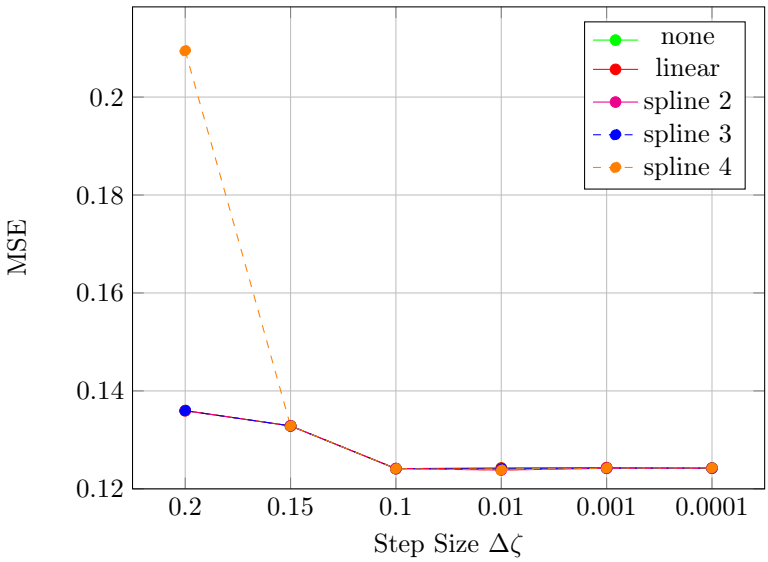

Figure 4.1: MSE depending on step size and extrapolation method for a grid graph ($N = 25$ RVs, 25×25) with attractive and repulsive edges. SBP never converges for $\zeta_M = 1$ (i.e., $SBP_{ES} \neq \mathcal{F}_{\mathcal{B}}{}^{\circ}$). For $\Delta\zeta \leq 0.15$ here is no influence of the extrapolation methods on the accuracy. Therefore the curves representing the different methods are covering each other.

edges. Repulsive edges make the problem harder to solve for BP.
There is no difference between *linear* and *spline 1* in runtime expressed in BP iterations. Both methods provide the same extrapolation of fixed points, but they differ in computation time (*linear* is about 7 times faster than *spline 1*). We can see that spline extrapolations are computationally more complex, but they estimate the next fixed point more accurately. This reduces the total number of BP iterations and furthermore, the total runtime of SBP expressed in ms. Cubic spline extrapolation (*spline 3*) provides the best runtime performance. Despite high calculation effort, the extrapolated fixed points are already close enough, such that BP only needs a few iterations to converge.

Table 4.1: Runtime performance of SBP with different extrapolation methods on a grid graph ($N = 9$ RVs, 3×3).

Extrapolation Method	Runtime in BP Iterations			Runtime in ms	
	Repulsive	Attractive	Total	Total	Extrapolation
none	324	239	563	210.397	0.044
linear	291	210	501	193.461	3.377
spline 1	291	210	501	215.213	23.867
spline 2	268	188	456	200.040	24.647
spline 3	267	184	**451**	**190.606**	24.575
spline 4	275	178	453	194.629	25.389

We are interested in obtaining a good overall size of iteration budget for SBP_{ES}. Therefore we evaluated the MSE over the total number of BP iterations. We use three different graphs: grid graph ($N = 100$ RVs, 10×10), complete graph ($N = 10$ RVs), and random graph ($N = 10$ RVs with $\langle d \rangle = 3$) with $\theta_i = 0.4$. The pairwise potentials are sampled from $J_{ij} \in \{-1, +1\}$ with equal probability.

We can see the MSE over number of BP iterations in Figure 4.2. For all graphs there is a minimum of the MSE at about 60 BP iterations. With an increase of BP iterations the MSE starts to rise again. Therefore we need to find a budget which runs SBP long enough to get close to the minimum of the MSE. The minimum of the MSE usually lies between 40 - 80 BP iterations. For attractive models the MSE is monotonically decreasing, but beyond 50 BP iterations the MSE changes only slightly. Setting a budget of 70 BP iterations gives us confidence to catch most of these minima. Therefore we assume that this budget leads to the best overall performance.

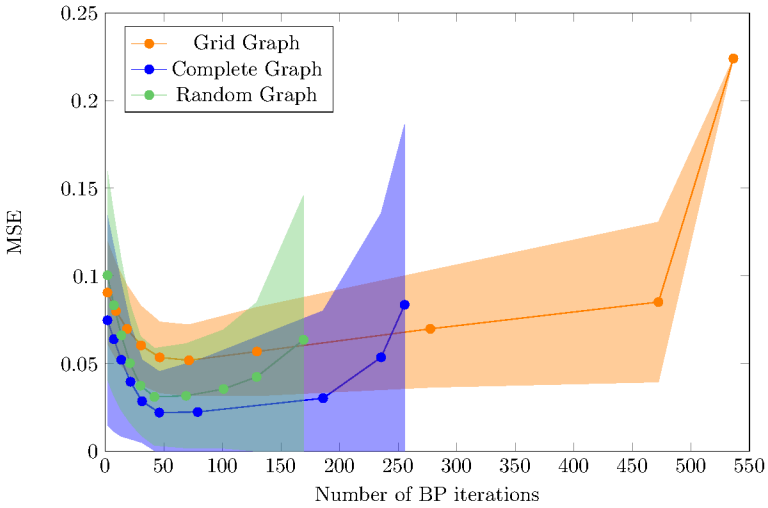

Figure 4.2: MSE over number of BP iterations for grid graphs ($N = 100$ RVs, 10×10), complete graphs ($N = 10$ RVs), and random graphs ($N = 10$ RVs with $\langle d \rangle = 3$). We use common local potentials ($\theta_i = 0.4$). The pairwise potentials are sampled from $J_{ij} \in \{-1, +1\}$ with equal probability. Shaded areas represent the standard deviations

The parameters of SBP do not impact the accuracy, but we achieve the best runtime performance – on average – with an adaptive step size with $\Delta\zeta_{init} = 0.1$, and cubic spline extrapolation. For SBP_{ES} we use a BP iteration budget of 70. We further observed that SBP_{ES} performs better with a lower threshold for the adaptive step size controller; the MSE increases if the budget is too large and SBP_{ES} already overreached the minimum of MSE. We expect that there is only a small change between the fixed points; this eventually leads to a larger step and therefore increases the MSE even more. A lower threshold for the adaptive step size controller keeps these steps smaller.

4.4 Approximating the Minimum of the Bethe Free Energy

We evaluate how good SBP approximates the minimum of the Bethe free energy $\mathcal{F}_{\mathcal{B}}{}^*$. We consider PGMs where it is not guaranteed that BP converges for $\zeta_M = 1$ (i.e., $\text{SBP}_{ES} \neq \mathcal{F}_{\mathcal{B}}{}^{\circ}$). On a grid graph ($N = 25$ RVs, 5×5) all local potentials are defined by $\theta_i = \theta$ and the model contains attractive and repulsive edges that are sampled from $J_{ij} \in \{-1, 1\}$ with equal probability. For these graphs $\mathcal{F}_{\mathcal{B}}{}^*$ can still be approximated reasonable well by (Weller and Jebara, 2014).
The results are summarized in Table 4.2 and show that SBP approximates $\mathcal{F}_{\mathcal{B}}{}^*$ within accuracy of our reference method ($\text{MSE}_{\mathcal{B}}$). We further report for how many models SBP obtains a stationary point ($\mathcal{F}_{\mathcal{B}}{}^{\circ}(\zeta_M)$ equals SBP); although SBP frequently stops for $\zeta_m < 1$ it does still obtain accurate fixed points. SBP_{ES} shows a similar behavior. The number of obtained

Table 4.2: Grid graph ($N = 25$ RVs, 5×5) and $J_{ij} \in \{-1, 1\}$ for different values of θ. We evaluate MSE to the exact solution and $\text{MSE}_{\mathcal{B}}$ to the Bethe approximation.

		$\theta = 0$	$\theta = 0.1$	$\theta = 0.4$
$\text{MSE}_{\mathcal{B}}$	SBP_{all}	0.036	0.037	0.022
	SBP_{ES-all}	0.036	0.034	0.040
$\mathcal{F}_{\mathcal{B}}{}^{\circ}(\zeta_M)$ equals	SBP	100	10	23
	SBP_{ES}	100	0	0
MSE	BP^*	0.338	0.251	0.102
	BP^*_{D}	0.226	0.198	0.066
	SBP_{all}	**0.000**	0.029	0.047
	SBP_{ES-all}	**0.000**	**0.008**	**0.037**
	Gibbs_{all}	0.001	0.016	0.064
	$\mathcal{F}^*_{\mathcal{B}_{all}}$	0.036	0.042	0.069
Percentage of converged runs	BP^*	5	11	26
	BP^*_{D}	11	16	69
BP Iterations	BP^*	40	52	84
	BP^*_{D}	1370	1449	1735
	SBP_{all}	5	182	146
	SBP_{ES-all}	5	57	60
	Gibbs_{all}	10^5	10^5	10^5

stationary points is even lower than with SBP, but early stopping has only a minor impact on the accuracy of the Bethe approximation. This suggests that stability of BP breaks down only close to $\mathcal{F}_{\mathcal{B}}{}^{\circ}$ (cf. property (3)). Moreover, by looking at the MSE we observe that SBP also accurately approximates the exact solution. SBP_{ES} provides even more accurate results.

We further examine the dependence of MSE and $\text{MSE}_{\mathcal{B}}$ on ζ_m and present the evolution over the cumulative iterations. We use a grid graph ($N = 25$ RVs, 5×5). All local potentials are defined by $\theta_i = 0.4$ and the model contains attractive and repulsive edges that are sampled from $J_{ij} \in \{-1, 1\}$ with equal probability. The results are shown in Figure 4.3. Note that $\text{MSE}_{\mathcal{B}}$ (blue) decreases monotonically with every iteration. This empirically verifies that SBP proceeds along a well-behaved solution path (cf. property (2)). The actual intent, however, is to approximate the exact marginals. Therefore we present the MSE (orange). We can see that the MSE starts to rise beyond a certain number of BP iterations. The minimum of MSE lies somewhere around 70 BP iterations. This is similar to the observations in Figure 4.2.
The behavior of over-counting explains the increase of the MSE with ζ_m close to 1 (cf. Section 2.3.2). The early stopping mechanism of SBP_{ES} stops before the approximated solution deviates from the exact because of over-counting information. Therefore, SBP_{ES} does not only reduce runtime, but even increases the accuracy compared to the exact solution.

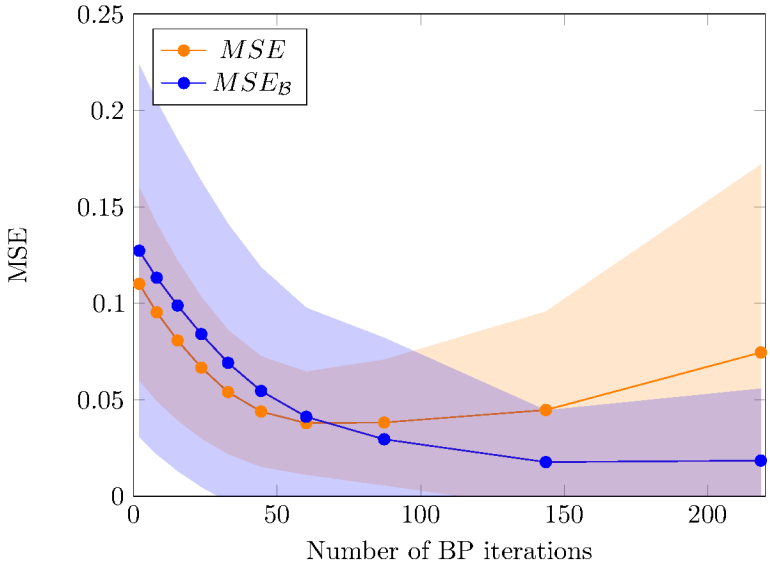

Figure 4.3: MSE (orange) and $MSE_{\mathcal{B}}$ (blue) over the cumulative iterations with standard deviation. MSE is averaged over 100 grid graphs ($N = 25$ RVs, 5×5); $\theta_i = 0.4$ and $J_{ij} \in \{-1, 1\}$.

4.5 Attractive Models

In the next experiment we restrict our models by only allowing attractive edges. For attractive models BP always converges, but multiple solutions are possible (cf. Section 2.3.2). If multiple solutions exist it depends on the initialization of BP to which one it converges. With SBP we can improve this behavior (cf. property (3)). We consider grid graphs with $N = 100$ RVs, and random and complete graphs with $N = 10$ RVs. The potentials are sampled according to $\theta_i \sim \mathcal{U}(-0.5, 0.5)$ and $J_{ij} \sim \mathcal{U}(0, \beta)$, and we generate $L = 100$ models for every value of $\beta \in [0, 5]$. We compute the MSE for every value of β and visualize the mean and the standard deviation of the MSE in Figure 4.4. Note that the MSE is not Gaussian distributed but we report the standard deviation for simplicity.
BP (purple) converges rapidly for all graphs considered. SBP (blue) slightly increases the number of iterations but does converge in fewer iterations than BP with damping (green). We observe that SBP always captures the global optimum $\mathcal{F}_{\mathcal{B}}{}^*$ (cf. Theorem 1) and consequently outperforms BP with respect to accuracy for all models considered. This becomes especially evident for models with strong couplings that exhibit multiple fixed points.
Except for the complete graph, SBP_{ES} (light-blue) is slightly better than SBP with respect to accuracy and runtime. For the complete graph the solutions of both methods are equal. In these cases SBP_{ES} does not stop earlier. Thus, SBP_{ES} requires less iterations than N_{Budget}; consequently SBP and SBP_{ES} are equivalent; i.e., $\text{SBP}_{ES} = \text{SBP} = \mathcal{F}_{\mathcal{B}}{}^{\circ}$. A smaller iteration budget would be necessary to increase the performance. The slower runtime of SBP_{ES} is caused by the lower threshold for the adaptive step size controller.

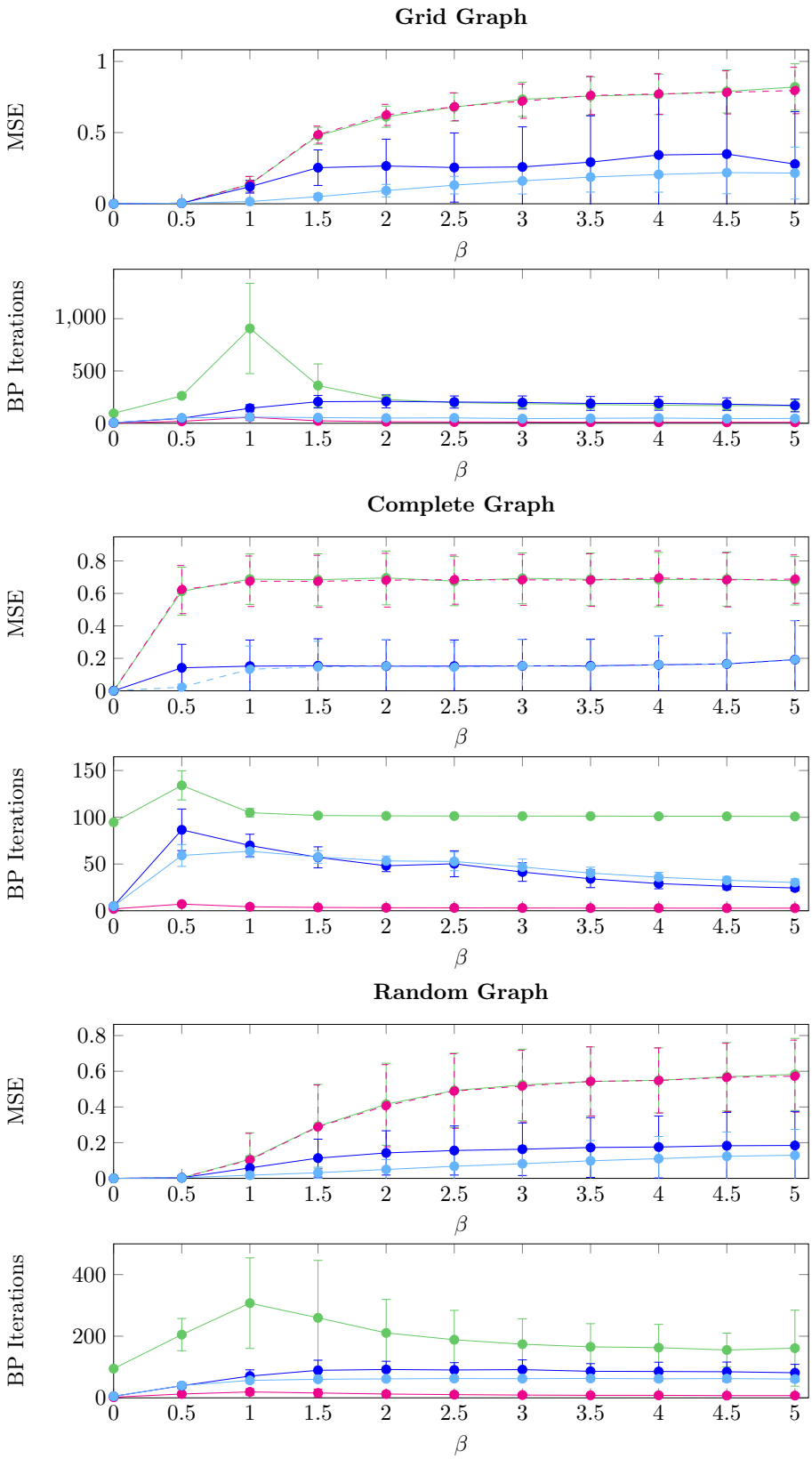

Figure 4.4: Attractive models with $\theta_i \in [-0.5, 0.5]$ and $J_{ij} \in [0, \beta]$. We compare MSE and number of iterations for: SBP_{all} (blue), SBP_{ES-all} (light-blue), BP^* (purple), and BP^*_D (green).

4.6 General Models

General models traditionally pose problems for BP and other methods that aim to minimize the Bethe approximation. We consider $\theta_i = \theta$ and draw the couplings with equal probability from $J_{ij} \in \{-1, 1\}$. The results are summarized in Table 4.3. We observe that BP and $\mathrm{BP_D}$ do not converge for most models. SBP, however, provides accurate fixed points for all models considered and stops after only a few iterations. SBP significantly outperforms BP on all graphs and achieves accuracy competitive with Gibbs sampling but requires fewer iterations in the order of 10^3. But again, $\mathrm{SBP_{ES}}$ outperforms SBP and therefore all other methods. For all settings it provides the lowest MSE and also the runtime is comparable to plain BP.

Increasing the coupling strength is expected to influence the behavior significantly; therefore we further consider $\theta_i \sim \mathcal{U}(-0.5, 0.5)$ and $J_{ij} \sim \mathcal{U}(-\beta, \beta)$. For every $\beta \in [0, 5]$ we execute $L = 100$ experiments and present the result in Figure 4.5 [4]. Note that we only show results for $\beta \leq 2$ on the grid graph because BP did only converge sporadically for models with stronger couplings. SBP (blue) requires only slightly more iterations than BP, even if we consider only models where BP converged. Again SBP outperforms BP^* (purple) and BP^*_D (green) on all graphs with respect to accuracy. The benefits of SBP become increasingly evident as the coupling strength increases.
For $\mathrm{SBP_{ES}}$(light-blue) the results are similar to the previous experiments. It outperforms SBP in terms of accuracy and runtime. The runtime of $\mathrm{SBP_{ES}}$ is equal or even faster than BP (if it converges).

[4] Note that the MSE is not Gaussian distributed but we report the standard deviation for simplicity. In some data points the difference between the mean and the standard deviation is negative. On the logarithmic scale negative values can not be represented; thus, no data point is shown.

Table 4.3: Results for general models with $J_{ij} \in \{-1, 1\}$ on grid graphs ($N = 100$ RVs, 10×10), complete graphs ($N = 10$ RVs), and random graphs ($N = 10$ RVs with $\langle d \rangle = 3$). We report the MSE with respect to the exact solution, the percentage of converged runs, and the runtime. Only converged runs are considered for BP and BP^*_D, whereas all results are considered for SBP_{all} and $Gibbs_{all}$.*

			$\theta = 0$	$\theta = 0.1$	$\theta = 0.4$
Grid Graph	MSE	BP*	-	-	0.184
		BP^*_D	0.186	0.240	0.154
		SBP_{all}	**0.000**	0.026	0.077
		SBP_{ES-all}	**0.000**	**0.013**	**0.060**
		$Gibbs_{all}$	0.001	0.037	0.120
	Percentage of converged runs	BP*	0	0	2
		BP^*_D	1	2	12
	BP Iterations	BP*	-	-	102
		BP^*_D	2711	2313	2599
		SBP_{all}	5	149	209
		SBP_{ES-all}	5	58	55
		$Gibbs_{all}$	10^5	10^5	10^5
Complete Graph	MSE	BP*	0.463	0.466	0.356
		BP^*_D	0.463	0.473	0.422
		SBP_{all}	**0.000**	0.055	0.074
		SBP_{ES-all}	**0.000**	**0.035**	**0.063**
		$Gibbs_{all}$	0.096	0.096	0.077
	Percentage of converged runs	BP*	41	42	50
		BP^*_D	41	41	50
	BP Iterations	BP*	17	17	18
		BP^*_D	211	207	234
		SBP_{all}	5	51	110
		SBP_{ES-all}	5	47	47
		$Gibbs_{all}$	10^5	10^5	10^5
Random Graph	MSE	BP*	0.252	0.202	0.101
		BP^*_D	0.128	0.116	0.083
		SBP_{all}	**0.000**	0.048	0.049
		SBP_{ES-all}	**0.000**	**0.010**	**0.032**
		$Gibbs_{all}$	0.001	0.011	0.048
	Percentage of converged runs	BP*	30	33	49
		BP^*_D	62	64	80
	BP Iterations	BP*	42	53	50
		BP^*_D	1077	1057	873
		SBP_{all}	5	149	131
		SBP_{ES-all}	5	56	61
		$Gibbs_{all}$	10^5	10^5	10^5

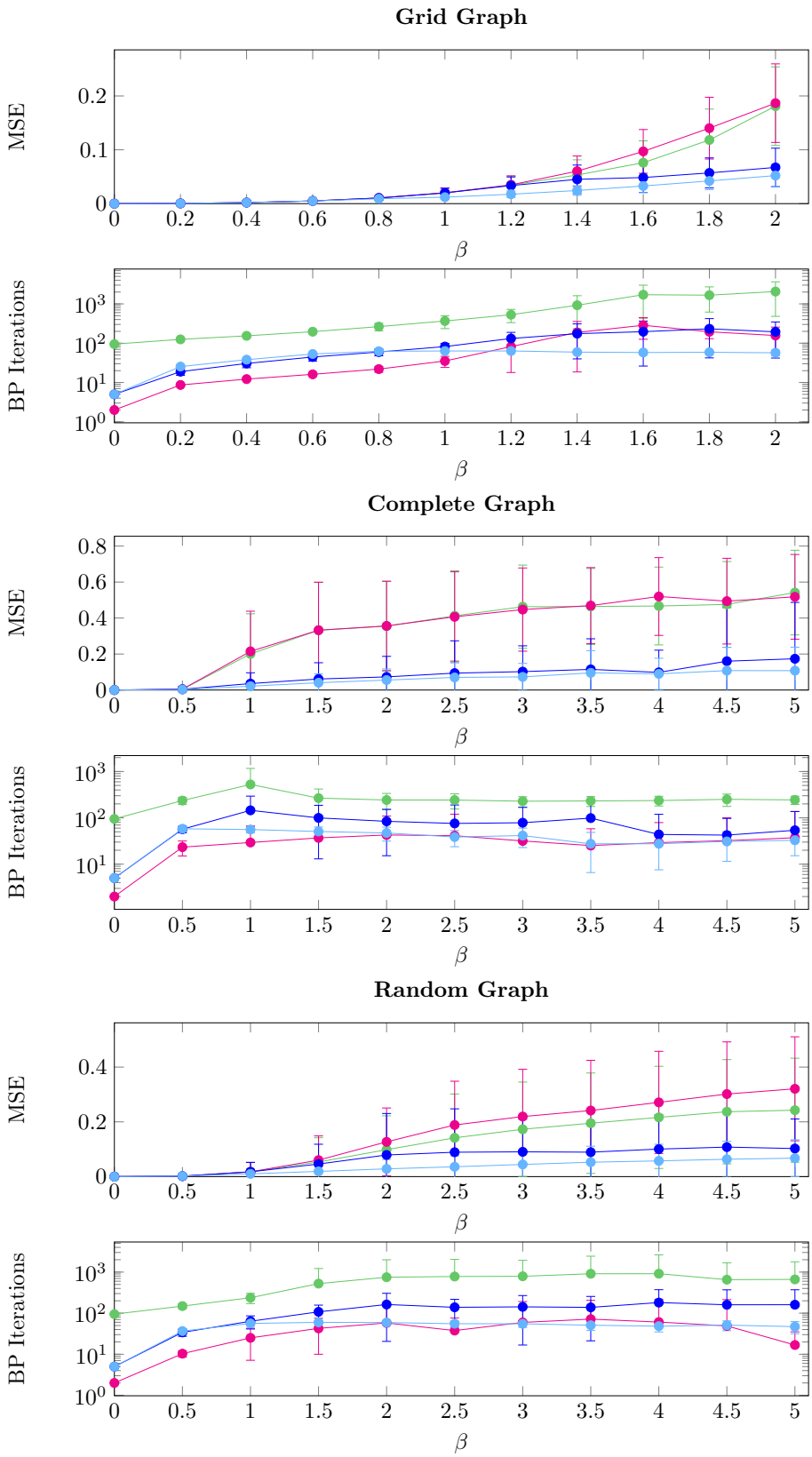

Figure 4.5: General models with $\theta_i \in [-0.5, 0.5]$ and $J_{ij} \in [-\beta, \beta]$. We compare MSE and number of iterations for: SBP_{all} (blue), SBP_{ES-all} (light-blue), BP^* (purple), and BP^*_D (green).

4.7 Discussion

In the previous experiments we have seen that SBP provides a unique solution that is superior to BP and BP_D in terms of accuracy with respect to the exact solution. For attractive models SBP always obtains the best possible solution and therefore outperforms BP in all cases with respect to accuracy. For general models BP does not converge for most models, and BP with damping only slightly improves this. In contrast, SBP still provides accurate solutions for all models. These experiments empirically verify the claim that SBP guides itself towards an accurate fixed point on general graphs with *attractive* and *repulsive* edges (cf. property (3)).

Reconsidering the phase diagram in Figure 2.8 allows us to distinguish the behavior of SBP: (i) in the ferromagnetic phase SBP obtains the same (unique) solution as BP, (ii) in the ferromagnetic phase with multiple solutions SBP obtains the best solution with respect to accuracy, and (iii) in the spin glass phase SBP obtains an accurate solution although BP fails to converge.

Compared to Gibbs sampling SBP shows a similar performance in terms of accuracy by only using a fraction of runtime.

Using a fixed iteration budget further increases the performance of SBP significantly. SBP_{ES} not only reduces the runtime, but also increases the accuracy and provides the most accurate results in all experiments.

5 Conclusion

In this work we introduced self-confident belief propagation (SBP), an iterative algorithm that approximates the marginal distributions.
While exact inference is not always feasible on loopy PGMs, approximate inference methods can provide good solutions. BP is a prominent tool to perform approximate inference (Koller and Friedman, 2009; Pernkopf et al., 2014), but the relation between convergence rate, accuracy and uniqueness of fixed points is still an open problem for general models. Fixed points may be unstable and are not necessarily unique. Even if the fixed points are stable, it is not guaranteed that they are close to the exact solution (Ihler et al., 2005; Mooij and Kappen, 2007; Weiss, 2000).
We showed how SBP overcomes these problems of BP and obtains a unique, stable and accurate solution.

We achieved this favorable behavior of SBP by exploiting the following properties: (i) a smooth solution path exists and originates from the unique fixed point that is obtained by neglecting the pairwise potentials; (ii) this solution path is well-behaved and can be tracked efficiently; (ii) the solution of SBP approximates the exact solution well and corresponds to the global optimum of the Bethe approximation for attractive models.
SBP tracks this solution path by solving a simple problem without pairwise potentials first and then gradually increases the pairwise potentials and follows a smooth solution path towards a unique, stable and accurate solution.

Additionally, we provided and discussed some practical considerations to efficiently track the solution path. We explained how a fixed budget of iterations (SBP_{ES}) enhances the accuracy with respect to the exact solution in addition to restricting the overall runtime.

Finally, we evaluated the performance of SBP on various graphs with Ising potentials. SBP significantly improves the performance of BP: the obtained marginals are consistently better than for BP with and without damping. Moreover, SBP approximates the exact marginals well on probabilistic graphical models for which BP does not converge at all. Compared to Gibbs sampling, SBP obtains similar accurate solutions in a fraction of runtime. SBP_{ES} further improves the performance of SBP, i.e., early stopping increases the accuracy and reduces the runtime.

So far we have estimated the iteration budget empirically. We aim to investigate the dependence between the number of iterations that minimize the MSE and the coupling strength in order to obtain an optimal iteration budget for a given probabilistic graphical model.
We restricted our analysis to models with binary random variables. It would be straightforward to extend SBP and allow for random variables with more than two states.

Bibliography

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc.

Cooper, G. F. (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial intelligence*, 42(2-3):393–405.

Dembo, A., Montanari, A., et al. (2010). Ising models on locally tree-like graphs. *The Annals of Applied Probability*, 20(2):565–592.

Elidan, G., McGraw, I., and Koller, D. (2006). Residual belief propagation: Informed scheduling for asynchronous message passing. In *Proceedings of UAI*, pages 165–173.

Frey, B. J. (1998). *Graphical Models for Machine Learning and Digital Communication*. MIT Press, Cambridge, MA, USA.

Gilbert, E. (1959). Random Graphs. *Annals of Mathematical Statistics*, 30(4):1141–1144.

Globerson, A. and Jaakkola, T. (2007). Convergent propagation algorithms via oriented trees. In *Proceedings of UAI*.

Hazan, T. and Shashua, A. (2008). Convergent message-passing algorithms for inference over general graphs with convex free energies. In *Proceedings of UAI*.

Heskes, T. et al. (2003). Stable fixed points of loopy belief propagation are minima of the Bethe free energy. In *NIPS*, volume 15, pages 359–366.

Ihler, A., Fisher, J., and Willsky, A. (2005). Loopy belief propagation: convergence and effects of message errors. In *Journal of Machine Learning Research*, pages 905–936.

Kabashima, Y. (2003). Propagating beliefs in spin-glass models. *Journal of the Physical Society of Japan*, 72(7):1645–1649.

Knoll, C., Kulmer, F., and Pernkopf, F. (2018). Self-Guided Belief Propagation – Iterative Enhancement of the Bethe Approximation. *Submitted to UAI*.

Knoll, C., Mehta, D., Chen, T., and Pernkopf, F. (2017). Fixed Points of Belief Propagation–An Analysis via Polynomial Homotopy Continuation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*.

Knoll, C. and Pernkopf, F. (2017). On Loopy Belief Propagation–Local Stability Analysis for Non-Vanishing Fields. In *Proceedings of UAI*.

Knoll, C., Rath, M., Tschiatschek, S., and Pernkopf, F. (2015). Message Scheduling Methods for Belief Propagation. In *Machine Learning and Knowledge Discovery in Databases*, pages 295–310. Springer.

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT press.

Lauritzen, S. and Spiegelhalter, D. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society*, pages 157–224.

Link, W. A. and Eaton, M. J. (2012). On thinning of chains in MCMC. *Methods in Ecology and Evolution*, 3(1):112–115.

Maceachern, S. N. and Berliner, L. M. (1994). Subsampling the gibbs sampler. *The American Statistician*, 48(3):188–190.

MacKay, D. J. C. (2002). *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA.

Meltzer, T., Globerson, A., and Weiss, Y. (2009). Convergent message passing algorithms: a unifying view. In *Proceedings of UAI*, pages 393–401.

Meshi, O., Jaimovich, A., Globerson, A., and Friedman, N. (2009). Convexifying the Bethe free energy. In *Proceedings of UAI*, pages 402–410.

Mezard, M. and Montanari, A. (2009). *Information, Physics, and Computation*. Oxford Univ. Press.

Mooij, J. M. and Kappen, H. J. (2007). Sufficient conditions for convergence of the sum–product algorithm. *IEEE Transactions on Information Theory*, 53(12):4422–4437.

Murphy, K., Weiss, Y., and Jordan, M. (1999). Loopy belief propagation for approximate inference: an empirical study. In *Proceedings of UAI*, pages 467–475.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers.

Pernkopf, F., Peharz, R., and Tschiatschek, S. (2014). *Introduction to Probabilistic Graphical Models*. Academic Press' Library in Signal Processing.

Ruozzi, N. (2013). Beyond log-supermodularity: lower bounds and the bethe partition function. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 546–555. AUAI Press.

Saade, A. (2016). *Spectral inference methods on sparse graphs : theory and applications*. PhD thesis, PSL Research University.

Shin, J. (2012). Complexity of Bethe Approximation. In *Proceedings of AISTATS*, pages 1037–1045.

Sutton, C. and McCallum, A. (2007). Improved dynamic schedules for belief propagation. In *Proceedings of UAI*, pages 376–383.

Tatikonda, S. C. and Jordan, M. I. (2002). Loopy belief propagation and Gibbs measures. In *Proceedings UAI*, pages 493–500. Morgan Kaufmann Publishers Inc.

Wainwright, M. J., Jordan, M. I., et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305.

Watanabe, Y. and Fukumizu, K. (2009). Graph zeta function in the Bethe free energy and loopy belief propagation. In *NIPS*, pages 2017–2025.

Weiss, Y. (2000). Correctness of local probability propagation in graphical models with loops. *Neural Comp.*, 12(1).

Weller, A. and Jebara, T. (2014). Approximating the Bethe partition function. In *Proceedings of UAI*, pages 858–867.

Welling, M. and Teh, Y. (2003). Approximate inference in Boltzmann machines. *Artificial Intelligence*, 143(1):19–50.

West, D. B. (2000). *Introduction to Graph Theory*. Prentice Hall, 2 edition.

Willsky, A. S., Sudderth, E. B., and Wainwright, M. J. (2008). Loop series and Bethe variational bounds in attractive graphical models. In *NIPS*, pages 1425–1432.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2005). Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312.

Yuille, A. and Rangarajan, A. (2003). The concave-convex procedure. *Neural Comp.*, 15(4):915–936.

Zdeborová, L. and Krzakala, F. (2016). Statistical physics of inference: Thresholds and algorithms. *Advances in Physics*, 65(5):453–552.