Phantom Belief Propagation for Fast Approximative Inference

Thomas Fabricius and Lars Kai Hansen
Informatics and Mathematical Modelling
Technical University of Denmark
{tf,lkh}@eivind.imm.dtu.dk

Abstract

We propose a new set of approximations for inference in graphical models, with the Bethe and the naive mean field approximations as special instances and producing local update rules like belief propagation. We call this new message passing *phantom belief propagation*. Numerical experiments show that in densely connected graphs where the Bethe approximation fails to converge, we can find phantom belief propagation approximations that converge to solutions with free energies close to the true free energy.

1 Introduction

The junction tree algorithm [1] provides exact inference at a computational cost that typically scales exponentially with system size for general graphs [2]. Therefore tree approximations have been invoked [3] in variational optimization settings, these approximations trade accuracy for speed. Similarly, using statistical mechanics asymptotic results can be obtained by considering large systems. The algorithms that can be derived from this approach often provide good approximative inference, for instance the naive mean field [4], and modern generalizations [5],[6]. In communication applications the need for speed is extreme. The turbo decoding algorithm [7] is a breakthrough based on approximate inference making it possible to decode codes near the Shannon limit in polynomial time. The fact that loopy belief propagation works so well in these situations is not yet understood. One recent advance in this understanding was the recognition of the cost function corresponding to the loopy belief propagation algorithm, namely the Bethe free energy [8]. Using the cost function the belief propagation algorithm can easily be shown to be exact for trees. The Bethe free energy makes the connection from belief propagation to other approximative inference methods based on free energy approximations that are bounded by the true free energy [3]. These methods make use of an approximative joint distribution defined on a tree. Loopy belief propagation is expected to be more accurate the more the graph resembles a tree. In densely connected graphs loopy belief propagation has not proven viable, here approaches as naive mean field with various corrections work better. Fully connected graphs are often found in communication applications, for instance in code division multiple access (CDMA) systems [9]. The problem of separating the users bit sequences can be seen as an independent component problem using a binary source prior possibly solved by using mean field methods [10].

2 Approximate inference

We will consider joint distributions defined over a set of variables $X = \{x_k\}_{k\in\kappa}$ with pairwise interactions, well aware that all joint distributions can be transformed into this form. The full set of all distinct pairs over N variables will be denoted η_N. The set of pairs corresponding to the actual joint distribution will be denoted η which is a subset of η_N. So the joint distribution can be written $P(X) = \frac{1}{Z}\prod_{(i,j)\in\eta}\Psi_{ij}(x_i,x_j)\prod_{k\in\kappa}\Psi_k(x_k)$ where the $\Psi_{ij}(x_i,x_j)$ and $\Psi_k(x_k)$ are non-negative functions called potentials and Z is the normalizing constant. The sets κ and η can be used to define what we will denote the corresponding graph (κ,η). The nodes of the graph is given by κ and the edges by η. The problem we want to solve can now be seen as calculating the normalization $Z = \int\mu(dX)\prod_{(i,j)\in\eta}\Psi_{ij}(x_i,x_j)\prod_{k\in\kappa}\Psi_k(x_k)$ where μ is a measure. Finding the normalizing constant is in general exponentially hard. One way to derive approximations is to consider the following functional

$$F[B(X)] = \int\mu(dX)B(X)(\log(B(X) - \log(\prod_{(i,j)\in\eta}\Psi_{ij}(x_i,x_j)\prod_{k\in\kappa}\Psi_k(x_k))) \geq -\log(Z) \tag{1}$$

where the last inequality follows from Jensens inequality if $B(X)$ is a normalized joint distributions over X. We see right away that $F[P(X)] = -\log(Z)$. So one approach is to restrict $B(X)$ to the subset of normalized joint distribution over X and minimise the functional within this class. We will restrict the function class of $B(X)$ a bit further, so that $B(X)$ factors in a way that at most contains pairwise factors that individually are joint distributions, i.e.

$$B(X) = \prod_{(i,j)\in\eta_b} b_{ij}(x_i,x_j)\prod_{k\in\kappa} b_k^{(1-q_k)}(x_k) \tag{2}$$

where q_k is the number of neighbors to node k in the graph (κ,η_t) and $b_k(x_k)$ is the marginal of $b_{kj}(x_k,x_j)$ or $b_{ik}(x_i,x_k)$. This factorisation only holds if the graph (κ,η_b) is a tree. There are many such factorisations, but the most common is to let $\eta_b \subseteq \eta$ i.e. the edges on the tree can also be found on the original graph (κ,η). If we write out equation 1 using equation 2 we get

$$\begin{aligned} F[B(X)] &= \sum_{(i,j)\in\eta_b} <\log b_{ij}(x_i,x_j)>_{b_{ij}(x_i,x_j)} \\ &+ \sum_{k\in\kappa}(1-q_k) <\log b_k(x_k) - \log\Psi_k(x_k)>_{b_k(x_k)} \\ &- \sum_{(i,j)\in\eta_b} <\log(\Psi_{ij}(x_i,x_j)\Psi_i(x_i)\Psi_j(x_j))>_{b_{ij}(x_i,x_j)} \\ &- \sum_{(i,j)\in\eta\setminus\eta_b} <\log(\Psi_{ij}(x_i,x_j))>_{b_{ij}(x_i,x_j)} \end{aligned} \tag{3}$$

Where $<>_{q()}$ means average with respect to $q()$. One should notice the last sum runs over all the pairs in the original graph (κ,η), not in the the graph (κ,η_b) and so contains parts with averages over $b_{ij}(x_i,x_j)$ where $(i,j)\in\eta\setminus\eta_t$ is not found directly from the factorisation in equation 2 but instead as a marginalisation over $B(X)$, which can be costly to compute. The Bethe approximation is now found using equation 3 with $\eta_b = \eta$. However, equation 3 can in general not be seen as a joint distribution over X. This implies that the last inequality in equation 1 in general doesn't hold. The only case where this is true for sure is when the graph (κ,η) of the original problem is a tree. A nice implication concerns the last term in equation 3. We find all the $b_{ij}(x_i, x_j)$, $(i,j) \in \eta = \eta_b$, directly from the factorisation in equation 2.

There are instances of graphs where the Bethe approximation has shown to produce superior results, in coding theory turbo codes [7] and Gallager codes [11]. One efficient way to minimize the Bethe free energy is by using belief propagation [8] which was introduced in [12]. One way to view belief propagation is as a local message passing scheme where each node receives messages from its neighbors according to the graph $(\kappa, \eta = \eta_b)$ which it assumes independent. This independence is only true for trees i.e. equivalent to the Bethe approximation only holding for trees.

Now we are ready to introduce the new approach. First we will generalize the above in the sence that we allow $\eta_b \subseteq \eta$ without restricting (κ, η_b) to trees. Secondly we notice that $b_{ij}(x_i, x_j) = b_i(x_i) b_j(x_j)$ for $(i,j) \in \eta \backslash \eta_b$ if the nodes i and j are disconnected according to the graph (κ, η_b). This is indeed true if (κ, η) is a tree, so this holds under the assumptions where the Bethe approximation was made. For this reason we assumes this is true for general (κ, η_b). So the last term of the free energy becomes $\sum_{(i,j)\in\eta\backslash\eta_b} < \log(\Psi_{ij}(x_i, x_j)) >_{b_i(x_i)b_j(x_j)}$. This new approximate free energy has the two special instances: the Bethe free energy $(\eta_b = \eta)$, and the naive mean field $(\eta_b = \emptyset)$. Between these two we have a plethora of free energies, with or without loops and with interesting belief propagation dynamics.

The original reason for this new approximate free energy was to make an algorithm capable of converging when the traditional belief propagation did not, hopefully maintaining the good results when it converges. We will verify this experimentally in section 3. Another reason is that of computational complexity, with the Bethe approximation we reduce the complexity for general graphs with a state dimension d of each node from $O(d^N)$ to $O(dN^2)$ and with naive mean field to $O(dN)$, whereas Kikuchi approximations are more expensive than the Bethe approximation but they can give a significant accuracy gain [8]. In telecommunication applications $O(N^2)$ algorithms are already costly, especially if they have to be iterated. The success of turbo codes is due to the low $d = 2$ and the sparsity of the corresponding graph, hence the complexity is effectively reduced. But when detecting (potentially $N = 100$) simultaneous bits $d = 2$ in Code Division Multiple Access (CDMA) [9] with a fully connected graph, an $O(dN^2)$ algorithm is prohibitive.

2.1 Phantom belief propagation

There are several ways to minimize the new free energy for fixed η_b under the normalization and marginalization constraints of the $b_{ij}(x_i, x_j)$, $(i,j) \in \eta_b$, see [13] for an alternative to belief propagation. We here form a Lagrangian as in [14] to ensure the constraints

$$\begin{aligned} L[B(X)] = F[B(X)] &+ \sum_{(i,j)\in\eta_b} \int \mu(dx_j)\lambda_{ij}(x_j)\left(b_j(x_j) - \int \mu(dx_i) b_{ij}(x_i, x_j)\right) \\ &+ \sum_{(i,j)\in\eta_b} \int \mu(dx_i)\lambda_{ji}(x_i)\left(b_i(x_i) - \int \mu(dx_j) b_{ij}(x_i, x_j)\right) \\ + \sum_{k\in\kappa} \gamma_k \left(1 - \int \mu(dx_k) b_k(x_k)\right) &+ \sum_{(i,j)\in\eta_b} \gamma_{ij}\left(1 - \int \mu(dx_i, dx_j) b_{ij}(x_i, x_j)\right) \end{aligned} \quad (4)$$

where the measures are induced by the measure $\mu(dX)$. By functional variation we find the following implicit equations for the beliefs

$$b_{ij}(x_i, x_j) = \frac{1}{Z_{ij}} \exp(\lambda_{ij}(x_j) + \lambda_{ji}(x_i)) \Psi_{ij}(x_i, x_j) \Psi_i(x_i) \Psi_j(x_j) \quad (5)$$

and

$$
\begin{aligned}
b_k(x_k) &= \frac{1}{Z_k}\Psi_k(x_k)\exp\frac{1}{q_k-1}(\sum_{j\in N_k}\lambda_{jk}(x_k) && (6)\\
&- \sum_{(i,j)\in\eta\backslash\eta_b}\delta(i-k)\int\mu(dx_j)b_j(x_j)\log\Psi_{ij}(x_i,x_j) && (7)\\
&- \sum_{(i,j)\in\eta\backslash\eta_b}\delta(j-k)\int\mu(dx_i)b_i(x_i)\log\Psi_{ij}(x_i,x_j)) && (8)
\end{aligned}
$$

The marginalization constraints $b_k(x_k) = \int\mu(dx_l)b_{kl}(x_k,x_l)$ will be used together with

$$\lambda_{lk}(x_k) = \log\prod_{m\in N_k\backslash l} M_{mk}(x_k). \tag{9}$$

The latter is derived in [14] and leads to a local message passing scheme equivalent to belief propagation

$$M_{lk}(x_k) = c\prod_{n\in N_k^P} P_{nk}^{\frac{1}{q_k-1}}(x_k)\int\mu(dx_l)\Psi_{kl}(x_k,x_l)\Psi_l(x_l)\prod_{m\in N_l\backslash k} M_{ml}(x_l) \tag{10}$$

where we have introduced the *phantom messages* $P_{nk}(x_k)$ which come from the neighbors in the original graph (κ,η) but not in the graph (κ,η_b), N_k^P indicating the set of k's phantom neighbors. We define the phantom message by

$$P_{nk}(x_k) = \exp\int\mu(dx_n)b_n(x_n)\log\Psi_{nk}(x_n,x_k) \tag{11}$$

The case where $q_k = 1$ needs special attention. In that case node k only has one neighbor, say l. We need not send any messages towards the k^{th} node since it is only going to send out messages to node l (its only neighbor) by combining messages $M_{mk}(x_k)$ for $m \in (N_k\backslash l) = \emptyset$. This holds both for normal belief propagation and our modified updates, and so $q_k = 1$ will never be a problem. For a node with one neigbohr $q_k = 1$ we should calculate b_k by noticing

$$
\begin{aligned}
\lambda_{lk}(x_k) &= \gamma_k + \sum_{(i,j)\in\eta\backslash\eta_b}\delta(i-k)\int\mu(dx_j)b_j(x_j)\log\Psi_{ij}(x_i,x_j)\\
&+ \sum_{(i,j)\in\eta\backslash\eta_b}\delta(j-k)\int\mu(dx_i)b_i(x_i)\log\Psi_{ij}(x_i,x_j) && (12)
\end{aligned}
$$

where l is k's only neighbor according to (κ,η_b). We can use this result and equation 9 to obtain

$$b_k(x_k) = \frac{1}{Z_k}\Psi_k(x_k)\prod_{n\in N_k^P} P_{nk}(x_k)\int\mu(dx_l)\Psi_{kl}(x_k,x_l)\Psi_l(x_l)\prod_{m\in N_l\backslash k} M_{ml}(x_l) \tag{13}$$

Putting this together we obtain the following **algorithm:**

1. Initialise all the messages $M_{i,j}(x_j)$, $M_{j,i}(x_i)$ for $(i,j)\in\eta_b$.
2. Initialise all the phantom messages $P_{nk}(x_k)$, $P_{kn}(x_n)$ for $(n,k)\in\eta\backslash\eta_b$.
3. Iterate equation 10 for all $(l,k)\in\eta_b$ and $(k,l)\in\eta_b$.
4. Calculate the $b_k(x_k)$ for $k\in\kappa$ according to 8 if $q_k \neq 1$; else use 13.
5. Calculate the phantom messages $P_{nk}(x_k)$, $P_{kn}(x_n)$ for $(n,k)\in\eta\backslash\eta_b$ according to 11.
6. Repeat from step 3 until convergence.

3 Experiments

In the following we consider a system where standard belief propagation typically will not converge. We consider a fully connected spin model with N spins i.e. each spin has $N-1$ neighbors and takes the values $[-1, 1]$. We choose the potentials as $\Psi_{ij}(x_i, x_j) = \exp x_i x_j J_{ij}$, $(i,j) \in \eta = \eta_N$ and $\Psi_k(x_k) = \exp x_k h_k$, $k \in \kappa$ and sampled the weights J_{ij}, $(i,j) \in \eta = \eta_N$ from zero mean unit variance normal distribution and the evidences or biases $h_k, k \in \kappa$ from a zero mean normal distribution with standard deviation of 0.01, i.e. the graph is strongly connected.

We ran belief propagation for 100 iteration on 50 such models with $N = 10$ and calculated the exact partition function Z. If belief propagation did not converge (this happened in 37 out of the 50 trials) we deleted edges in the graph (κ, η) with a probability of 0.1 to get the graph (κ, η_b) and ran phantom belief propagation for 100 iterations. If this did not converge we deleted edges from the graph (κ, η) with a probability of 0.2 etc. In this way we got a free energy where phantom belief propagation converged. All the runs were made with the messages, ordinary as phantom, initialized uniformly on the interval $[0, 1]$. The results can be seen on figure 1.

To justify that belief propagation did not converge (convergence is very easy to detect) due to bad initial conditions we tried different initialization seeds, which did not change the conclusion justifying that the Bethe energy ansatz is not well applicable in these dense models with short loops. Another point supporting this hypothesis is the fact that in most cases where belief propagation diverged the Bethe free energy oscillated with an average below $-\log Z$ whereas when it converged it was close to $-\log Z$, typically larger as would be the case if the joint approximative distribution actually was a distribution.

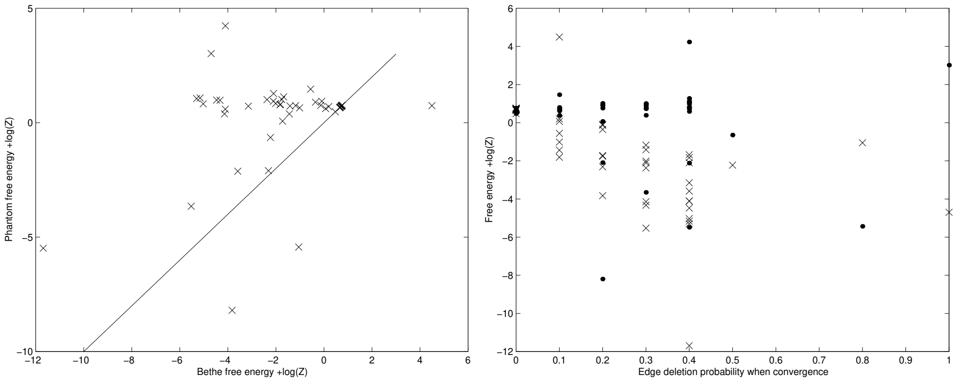

Figure 1: The average Bethe free energy $+\log Z$ is plotted against the converged phantom free energy $+\log Z$ in the left panel. We can see that the phantom free energy $+\log Z$ tend to clutter just above zero. In the right panel, the free energies are plotted against the deletion probability at which phantom belief propagation converged, phantom free energy is marked by x and Bethe by dots.

The phantom belief update with a low edge deletion probability also did not converge as we would expect and showed an oscillating free energy with an average above that of the Bethe free energy. In all our runs we found that the free energies in general were higher as we deleted more and more edges. This can possibly be understood as follows: when we delete edges, the graph (κ, η_b) becomes more and more tree-like, for which the free energy is bounded from below by $-\log Z$. When we have deleted all the edges, we have the naive mean field approximation which don't exploit any graphical structure and hence in general has the highest corresponding free energy. In all cases where normal belief propagation diverged there was an intermediate free energy where phantom belief propagation converged. In general it converged to free energis with values above $-\log Z$, closer than the diverging Bethe free energy. In 3 cases out of 50 the intermediate free energy converged below the average of the oscillating Bethe free energy. Again indicating that the solution maybe could be interpreted as a true joint, this is indeed the case when the deleted graph becomes a tree.

If the above is true in general, we could run belief propagation. If it converges, the Bethe free energy is often larger than $-\log Z$ and a good approximation. If no convergence is found we could prune away edges in a way that changes the average free energy by the smallest amount, knowing that this gives us a higher free energy. We repeat that until phantom belief propagation converges. Here we deleted edges at random with a probability. On figure 2 we see a typical run on the spin model described in the begining of this section. As seen we could for this case follow the proposed scheme. The model that converged first according to the scheme, is seen to be close to zero i.e. close to the true $-\log Z$. The corresponding graph (κ, η_b) is also shown. Only 11 edges had to be deleted compared to the total number of edges in the original graph, $N(N-1)/2 = 45$, for phantom belief propagation to converge. Even though we got free energies by the phantom belief updates that

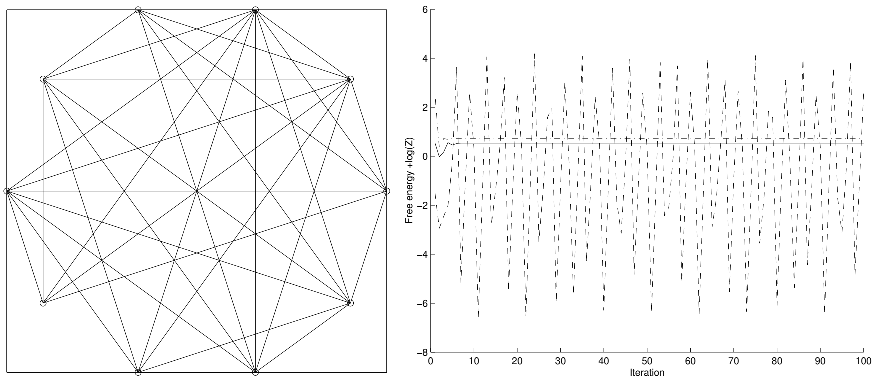

Figure 2: On the left we see the graph (κ, η_b) i.e. after deletion of edges with a probability $\rho = 0.1$. This graph correspond to the free energy $+\log Z$ on the right (full line), here we also see the Bethe free energy that diverged (dashed line) and of one deleted with a probability $\rho = 0.2$

were close to $-\log Z$ we haven't addressed the quality of the actual beliefs.

4 Conclusion

We have derived a new scheme for approximative inference that is less computationally costly than belief propagation. We call it *phantom belief propagation* due to the fact that some messages in the traditional belief propagation are substituted by a kind of phantom messages arising from deleted edges, i.e. like neural phantom messages from removed limbs. We showed empirically that it worked extremely well in densely connected graphs where belief propagation traditionally diverges. We also proposed a scheme for using this new algorithm in conjunction with traditional belief propagation.

Acknowledgments. The authors thank Niels Mørch, Ole Nørklit and Pedro Højen-Sørensen all Nokia R&D and Jan Larsen at Informatic and Mathematical Modelling. for useful discussion.

References

[1] S. Lauritzen and D. Spiegelhalter. Local computations with probabilties on graphical structures and their application to expert systems. *J. Royal Statistical society B*, 50:2154–227–268, 1988.

[2] G. Cooper. Computational complexity of probabilistic inference using Bayesian belief networks. *SIAM Journal on Computing*, 42:393–405, 1990.

[3] Lawrence K. Saul and Michael I. Jordan. Exploiting tractable substructures in intractable networks. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 486–492. The MIT Press, 1996.

[4] Carsten Peterson and James R. Anderson. A mean field theory learning algorithm for neural networks. *Complex Systems*, 1:995–1019, 1987.

[5] B. H. J. Kappen and F. B. Rodríguez. Boltzmann machine learning using mean field theory and linear response correction. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[6] Manfred Opper and Ole Winther. Mean field methods for classification with gaussian processes. In Michael S. Kearns, Sara S. Solla, and David A. Cohn., editors, *Advances in Neural Information Processing Systems*, volume 11, pages 309–315. The MIT Press, 1999.

[7] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: turbo-codes. In *Proc. IEEE International Conference on Communication (ICC)*, volume 5, pages 1064–1070. IEEE, 1993.

[8] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation (to appear). In *Advances in Neural Information Processing Systems*, volume 13. The MIT Press, 2001.

[9] S. Verdu. Computational complexity of optimum multiuser detection. *Algorithmica*, 4(3):303–312, 1989.

[10] Pedro A.d.F.R. Højen-Sørensen, Ole Winther, and Lars Kai Hansen. Mean field approaches to independent component analysis (to appear). In *Advances in Neural Information Processing Systems*, volume 13. The MIT Press, 2001.

[11] MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45, 1999.

[12] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, San Mateo, California, 1988.

[13] M. Welling and Y. W. Teh. Belief optimization for binary networks: A stable alternative to loopy belief propagation (to appear). In *Proceedings of Uncertainty in AI.* Morgan Kaufmann, 2001.

[14] J.S. Yedidia. An idiosyncratic journey beyond mean field theory. Book chapter to appear in *Advanced Mean Field Methods - Theory and Practice*, eds. D. Saad and M. Opper, MIT Press, 2000. TR 2000-27, Mitsubishi Electric Research Institution, 2000.