Marginal Inference in Continuous Markov Random Fields Using Mixtures

Yuanzhen Guo, Hao Xiong, Nicholas Ruozzi
University of Texas at Dallas
800 W. Campbell Road
Richardson, TX 75080
{yuanzhen.guo, hao.xiong, nicholas.ruozzi}@utdallas.edu

Abstract

Exact marginal inference in continuous graphical models is computationally challenging outside of a few special cases. Existing work on approximate inference has focused on approximately computing the messages as part of the loopy belief propagation algorithm either via sampling methods or moment matching relaxations. In this work, we present an alternative family of approximations that, instead of approximating the messages, approximates the beliefs in the continuous Bethe free energy using mixture distributions. We show that these types of approximations can be combined with numerical quadrature to yield algorithms with both theoretical guarantees on the quality of the approximation and significantly better practical performance in a variety of applications that are challenging for current state-of-the-art methods.

Introduction

Graphical models provide a flexible framework that can be used to represent joint probability distributions in a variety of application areas. The primary tool for approximate inference in these models is a local message-passing algorithm known as belief propagation (BP) and its myriad variants. In this work, we will be interested in designing algorithms for the marginal inference task (computing the partition function and/or marginal distributions) in continuous graphical models. Such inference tasks arise naturally in a variety of applications such as optical flow estimation (Fleet and Weiss 2006), depth estimation, scientific modeling, etc. As in the discrete case, inference in continuous and mixed models is an NP-hard problem that is only known to admit efficient algorithms in special cases, e.g., Gaussian graphical models (Malioutov, Johnson, and Willsky 2006) (Ruozzi and Tatikonda 2013). As a result, approximate inference algorithms such as loopy BP, and its variants, are typically employed in practice. As the focus of this work is on inference in graphical models, we will assume that the potential functions are known in advance or have already been learned from data using some other method, e.g., Chow-Liu trees (Chow and Liu 1968) or kernel BP (Song et al. 2011). Given the potential functions, our aim is then to approximately compute marginals and/or the partition function of the model.

Much of the work on inference in models with continuous variables has focused on extensions of the BP algorithm. However, efficient computation and representation of the BP messages turns out to be a challenging problem involving a variety of trade-offs. There are two primary approaches to computing the BP message updates in the continuous case. First, one class of methods approximate the messages, e.g., by using Gaussian mixtures (Sudderth et al. 2003), particles (Ihler and McAllester 2009), finite terms of orthogonal series expansions (Noorshams and Wainwright 2013), or kernels (Song et al. 2011). The BP message updates are then reformulated as expectations, and the expectations are calculated using either a sampling scheme or an inner product, in the case of the orthogonal series and kernel methods. Alternatively, the message updates can be approximated via moment matching: The expectation propagation algorithm approximates the messages using an exponential family and computes approximate BP message updates via a moment matching procedure (Minka 2001). In general, these iterative message-passing algorithms are not guaranteed to converge on arbitrary graphs with arbitrary potentials. In practice, convergent methods can be designed by exploiting the connection between fixed points of BP/EP and local optima of the Bethe free energy optimization problem. This objective is defined over locally consistent pseudomarginals and can be used to design convergent methods based on gradient ascent (Welling and Teh 2001; Heskes and Zoeter 2002). Here, we move the approximations from the messages to the pseudomarginals and design convergent methods for approximate inference over pseudomarginals that are mixtures of Gaussians. As the Bethe free energy objective requires the computation of expectations of the model potentials with respect to the pseudomarginals, our method will make use of Gauss-Hermite quadrature, a classical numerical integration method (Golub and Welsch 1969).

Our proposed approach has many advantages over existing methods in terms of practical performance, scalability, and flexibility:

- The approximation does not need to form products of mixture distributions such as those required by nonparametric BP (Sudderth et al. 2003), but it still gains the modeling advantages that Gaussian mixtures possess.
- The gradient optimization methods proposed here can be

vectorized to take advantage of modern GPUs. Even without this, our implementation requires a fraction of the time per iteration as compared to particle methods.

- With additional restrictions, the result of the inference procedure will not only yield a collection of local marginals but also a global mixture model. As a result, every marginal distribution, including conditionals, is essentially approximated using one inference pass. Message-passing algorithms would need to be rerun given evidence and not all marginal distributions can be easily extracted from the converged messages.
- Our approach easily extends to hybrid models, i.e., models with both discrete and continuous variables, with very little additional work.
- Gaussian quadrature methods come with strong theoretical guarantees and error bounds. In particular, the integral approximations can be made exact for any model in which the potential functions are log-polynomial.
- In practice, gradient methods yield convergent algorithms while message-passing algorithms, even in the exact case, can have significant difficulty converging.

We provide support for our claimed advantages from both a theoretical and a practical perspective: We apply our method to a variety of problems arising from real and synthetic data sets, in each case demonstrating the superior performance of our approach for the marginal inference task. Our experimental results suggest that Gaussian mixtures can produce pseudomarginals that more accurately reflect the ground truth (e.g., multimodal marginals) "per particle" than approximate message-passing schemes. Finally, if the goal is actually to approximate a high-dimensional integral, i.e., compute $\log Z$, the variational approach is far superior to approximations in the message domain. This is a result of the constraints in the variational approach that yield proper pseudomarginals and closed form estimates of the marginals as opposed to particle sampling methods, whose continuous marginals only approximately have these guarantees.

Preliminaries

A continuous, pairwise graphical model is a graph $G = (V, E)$ together with a collection of nonnegative potential functions $\phi_i : \mathbb{R} \to \mathbb{R}_{\geq 0}$ for each $i \in V$ and $\psi_{ij} : \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ for each $(i, j) \in E$. A joint probability distribution, p, factorizes with respect to G if it can be written as

$$p(x_V) = \frac{1}{Z} \prod_{i \in V} \phi_i(x_i) \prod_{(i,j) \in E} \psi_{ij}(x_i, x_j), \tag{1}$$

where Z is the normalization constant, sometimes called the partition function.

In this work, we will restrict our attention to the special case of strictly positive potential functions that are Riemann integrable over any compact subset of their domain. This is not an overly restrictive assumption as many models (e.g., Gaussians, Gaussian mixtures, Laplace distributions, etc.) easily satisfy this requirement. Our goal will be to develop methods for statistical inference in this class of distributions (e.g., computing Z or marginal distributions of p). As this task is NP-hard in general (the discrete case is a limit of the continuous case), approximations are often necessary in practice. In the remainder of this section, we review the current approaches to approximate inference in such models.

Belief Propagation

The BP algorithm, also know as the sum-product algorithm, is an iterative message-passing algorithm to approximate the marginals and partition function of graphical models. In BP, messages are passed along the edges of the graph. For each edge $(i, j) \in E$, the message passed from i to j is as follows.

$$m^t_{i \to j}(x_j) \propto \int \phi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m^{t-1}_{k \to i}(x_i) dx_i, \tag{2}$$

where $N(i)$ is the set of neighbors of node $i \in G$. Once the message-passing procedure converges, the messages are used to estimate the marginal probabilities over each variable and each pair of variables joined by an edge. This is done by constructing the following beliefs from the converged messages, m^*.

$$b_i(x_i) \propto \phi_i(x_i) \prod_{k \in N(i)} m^*_{k \to i}(x_i) \tag{3}$$

$$b_{ij}(x_i, x_j) \propto \frac{b_i(x_i) b_j(x_j) \psi_{ij}(x_i, x_j)}{m^*_{j \to i}(x_i) m^*_{i \to j}(x_j)} \tag{4}$$

The BP algorithm is exact, i.e., the beliefs are proportional to the correct marginal distributions, when the graph is a tree, but neither convergence nor correctness are guaranteed for general graphs.

The fixed point messages of BP correspond to local optima of a constrained optimization problem known as the Bethe free energy (BFE) (Yedidia, Freeman, and Weiss 2005), see Ruozzi (2017) for a more detailed discussion of the continuous case. Given a collection of normalized, nonegative beliefs that satisfy the so-called local marginalization constraints, that is,

$$\int b_i(x_i) dx_i = 1, \forall i \in V,$$

$$\int b_{ij}(x_i, x_j) dx_i = b_j(x_j), \forall (i, j) \in E, x_j \in \mathbb{R}$$

the Bethe free energy is defined as

$$F(b) = \sum_{i \in V} \mathbb{E}_{b_i}[\log \phi_i] + \sum_{(i,j) \in E} \mathbb{E}_{b_{ij}}[\log \psi_{ij}] + \sum_{i \in V} (1 - |N(i)|)\mathbf{H}(b_i) + \sum_{(i,j) \in E} \mathbf{H}(b_{ij}), \tag{5}$$

where $\mathbf{H}$ is the differential entropy. The log-Bethe partition function is obtained by maximizing F over the local marginalization constraints. While message-passing algorithms are not guaranteed to converge in general, gradient ascent based on the BFE yields convergent algorithms and can be made particularly efficient in special cases, e.g., (Welling and Teh 2001).

Though popular in the discrete case, in the continuous case, the message-passing update (2) cannot typically be computed in closed form outside of special cases (Weiss and Freeman 2001). As a result, alternative message-passing algorithms have been proposed to approximate BP. The first such approach, the nonparametric BP algorithm, approximates the messages as L-component Gaussian mixtures, and uses efficient sampling methods to compute the updates (Sudderth et al. 2003). However, the algorithm scales poorly as a function of L, $O(L^d)$ per edge per iteration where d is the maximum degree of a vertex in G. The particle belief propagation (PBP) algorithm improves upon nonparametric BP by using a collection of K particles associated with each node in the graph to approximate the BP messages instead of using Gaussian mixtures (Ihler and McAllester 2009)(Frank, Smyth, and Ihler 2009). This reduces the sampling cost to $O(K^2)$ per edge per iteration, but the constant in the big-O depends crucially on the efficiency of the sampling procedure that is being used. The expectation particle belief propagation (EPBP) algorithm is a PBP based algorithm in which the particles at each node are sampled via importance sampling (Lienart, Teh, and Doucet 2015). The proposal distributions are selected to be members of some exponential family, typically Gaussian distributions, and are computed using the same kind of moment matching updates as EP. The EPBP algorithm provides a consistent estimate of the BP messages as the number of particles increases, and the complexity of the EPBP algorithm is $O(MK)$ per edge per iteration where K is the number of particles and M is the number of samples drawn to approximate the message updates.

Another alternative, the stochastic orthogonal series message-passing algorithm (SOSMP) approximates the BP message updates using orthogonal series expansions (Noorshams and Wainwright 2013). To keep the procedure tractable, only K basis coefficients in the expansion are maintained at each iteration. The message update is reformulated as an expectation and then approximated via standard sampling procedures (e.g., rejection sampling or importance sampling). Note that this method requires that the potential functions can be normalized in order to apply the sampling procedures (this is somewhat limiting as the model may still be, and often is, normalizable without this restriction). In addition, before computing the expectations, the messages must be projected onto the space of nonnegative functions as the basis expansion may not guarantee nonnegativity. Song et al. (2011) proposed kernel BP (KBP), a joint learning and inference procedure that represents the messages as elements of reproducing kernel Hilbert space (RKHS) so that the message updates can be represented as inner products. In this respect, it is similar to SOSMP, except that training data is used to generate a representation of the potential functions in the RKHS. KBP can also return negative beliefs, which makes it difficult to use for estimating the partition function.

The expectation propagation message-passing algorithm (EP) is a generalization of BP in which the messages are restricted to a fixed exponential family and the BP updates are approximated via a moment matching procedure (Minka 2001). The moment matching step of EP requires approximating one integral for each sufficient statistic of the exponential family used in the approximation. Like BP, the EP algorithm is not guaranteed to converge. However, EP can also be formulated as maximizing the BFE in which the marginalization constraints are replaced with simpler moment matching constraints (Heskes and Zoeter 2002). As a result, convergent formulations of EP can be obtained by optimizing the BFE approximation directly.

Finally, recent advances in Stein variational methods provide an alternative to BFE based methods (Liu, Lee, and Jordan 2016)(Wang, Zeng, and Liu 2018). Given a continuously differentiable probability distribution, p, these methods aim to find a collection of particles $x^{(1)}, \ldots, x^{(n)}$ such that $\sum_{l=1}^{n} f(x^{(l)}) \approx \mathbb{E}_p[f]$. This can be done via a gradient descent scheme known as Stein variational gradient descent (SVGD) (Liu, Lee, and Jordan 2016). Recent work has extended this method to graphical structures in an effort to take advantage of local structure in the optimization (Wang, Zeng, and Liu 2018). While this approach is similar in many ways to what is proposed here, Graphical SVGD optimizes a different objective and there is no explicit strategy as of yet to use them for marginal inference (constructing accurate marginals from the particles would require an appropriate kernel density estimation method). This makes it difficult to compare this approach with those above, though it is an interesting direction of future research to bridge this gap.

Bethe Quadrature Methods

As an alternative to the message-passing algorithms described above, we propose to approximate the BFE directly by restricting the set of allowable beliefs to a nice family from which the integrals in equation (5) can be easily approximated. Such a direct approach circumvents a number of the difficulties with the above message-passing schemes including convergence issues. In particular, the quadrature based integration methods we will employ come with strong theoretical guarantees based on the number of quadrature points used in the approximation - the quality of approximation per quadrature point is significantly better than the quality of approximation per particle. Further, the approximation will not be dictated by the beliefs that can be obtained from the potentials multiplied by approximate messages. Consider a situation in which potential functions over the individual variables may be unimodal, but the true marginals are highly multimodal. Gaussian EP can perform poorly here as it cannot return a multimodal approximation for the marginals.

First, consider restricting the allowable beliefs in the BFE to Gaussian distributions. That is, for each $i \in V$, $b_i(x_i) = \mathcal{N}(x_i; \mu_i, \delta_{ii})$ and for each $(i,j) \in E$

$$b_{ij}(x_i, x_j) = \mathcal{N}\left(x_i, x_j; \begin{bmatrix}\mu_i \\ \mu_j\end{bmatrix}, \begin{bmatrix}\delta_{ii} & \delta_{ij} \\ \delta_{ij} & \delta_{jj}\end{bmatrix}\right).$$

Each covariance matrix must be strictly positive definite in order to yield a Gaussian distribution. However, the set of all positive definite matrices does not form a closed, convex set so we relax the constraint to only require positive semidefiniteness of the covariance matrices. The log-Bethe partition function, restricted to Gaussian beliefs, is then found by maximizing (5) over the means and variances for each belief

subject to positive semidefinite constraints. Maximizing this function via gradient ascent requires computing the integrals in (5), which in general do not have closed form solutions. To make this problem more tractable we will approximate the integrals using quadrature methods. As we are integrating with respect to normal distributions, Gauss-Hermite quadrature methods (GHQs) are a reasonable choice (Golub and Welsch 1969). These quadrature methods can be thought of as deterministic sampling methods in that they approximate expectations of a function q with respect to a Gaussian distribution as a weighted sum the function q evaluated at K_Q carefully chosen quadrature points. These methods have also been applied to approximate the integrals as part of the EP algorithm (Heskes and Zoeter 2005). For a univariate function $f : \mathbb{R} \to \mathbb{R}$, GHQs approximate the expectation with respect to a normal distribution as

$$\mathbb{E}_{\mathcal{N}(\mu,\sigma^2)} f(x) \approx \sum_{k=1}^{K_Q} \frac{w_k}{\sqrt{\pi}} f\left(\sqrt{2\sigma^2} y_k + \mu\right),$$

where the w_k's and y_k's are determined by the GHQ method and are independent of the mean and variance. For the multivariate case, the integrals are approximated iteratively.

Theorem 1 (Golub and Welsch (1969)) *For a positive integer K_Q, mean $\mu \in \mathbb{R}$, and variance $\sigma^2 \in \mathbb{R}_{>0}$, GHQ constructs $w_1, \ldots, w_{K_Q} \in \mathbb{R}$ and $y_1, \ldots, y_{K_Q} \in \mathbb{R}$ such that there exists a $\xi \in \mathbb{R}$ with*

$$\mathbb{E}_{\mathcal{N}(\mu,\sigma^2)} f(x) = \sum_{k=1}^{K_Q} \frac{w_k}{\sqrt{\pi}} f\left(\sqrt{2\sigma^2} y_k + \mu\right) + \frac{n!\sqrt{\pi}}{2^n} \frac{f^{(2K_Q)}(\xi)}{(2n)!}$$

As a consequence, using K_Q quadrature points, the approximation is exact whenever f is a polynomial of degree at most $2K_Q - 1$ in each variable separately.

The BFE over Gaussian beliefs is a non-concave optimization problem over δ and μ. Despite this, we can still use gradient methods to find local optima of this objective. In our implementation we used projected gradient ascent to keep the iterates in the space of positive semidefinite matrices. The gradient can be computed in $O(|E|K_Q^2)$ time on a single machine using the quadrature methods. This per iteration complexity looks comparable to PBP, but experimentally K_Q can be taken to be much smaller than the number of particles in PBP. The weights and quadrature points can be precomputed for a given K_Q incurring a one time computational cost at the start of the algorithm. Similar to the message-passing algorithms, the computation of the gradient can be easily parallelized across multiple machines.

The drawback of restricting the beliefs to Gaussian distributions is that the resulting beliefs are necessarily unimodal. In the case that the true beliefs are multimodal, there exist local optima that fit each of the modes separately. However, the above approach extends to the case in which the beliefs are assumed to be Gaussian mixtures. As any nonnegative, continuous function can be arbitrarily well approximated by a mixture of Gaussians, this family is very expressive, but one expects more local optima of the BFE as a result.

Like Gaussians, the family of Gaussian mixture distributions is closed under marginalization. There are two primary difficulties in extending from the Gaussian to the Gaussian mixture case. First, there is an issue of identifiability of Gaussian mixtures. Without any prior information, all $L!$ relabelings of an L-component Gaussian mixture will result in equivalent models. This can cause slow convergence if the optimization procedure bounces between these identical local optima. To discourage this behavior, the marginalization conditions are enforced by fixing a single $\mu^{(l)}$ and $\delta^{(l)}$ for each $l \in \{1, \ldots, L\}$. This fixes an ordering on the components of the mixture that is consistent across the individual beliefs. Second, although the entropy of a Gaussian distribution can be computed in closed form, the entropy of Gaussian mixtures does not admit a closed form solution, and we will need to approximate it as well. Taylor series expansions have been shown to work well in this case (Huber et al. 2008). So, we expect reasonable performance from GHQ.

In addition to the partial derivatives with respect to μ and δ, partial derivatives must also be computed for the mixture probabilities, and projected gradient methods are used to enforce the mixture and positive definite constraints. For Gaussian mixture beliefs, the gradient can be computed in $O(|E|K_Q^2 L^2)$ time on a single machine. Better per iteration complexity, e.g., $O(|E|K_Q^2 L)$, can be obtained by using stochastic gradient methods making this approach practical for large L. The entire method can be formulated using matrix-vector operations making it efficient to implement in MATLAB and on modern GPUs. In addition, K_Q can be kept small in practice as long as the log-potential functions can be well-approximated by low-degree polynomials.

Theorem 2 *For any tree-structured graphical model, every local optimum, with respect to the parameters of the beliefs, of the BFE in which the beliefs are mixtures of Gaussians yields a lower bound on the partition function assuming that sufficiently many quadrature points are used to approximate the integrals.*

Proof sketch: If the model is tree-structured, then the BFE approximation is exact when optimizing over arbitrary marginal probability distributions that satisfy the marginalization constraints, and the number of quadrature points can be chosen so that the integrals in the BFE for Gaussian mixtures are arbitrarily close to the correct answer. Note that, at the maximum of the BFE, taking the limit as the number of quadrature points tends to infinity may be necessary to achieve a lower bound.

Single-Pass Inference

Unlike the beliefs produced by typical message-passing algorithms, under certain conditions, the beliefs produced via the above variational scheme are actually the marginals of a proper joint probability distribution. In particular if we require that, for each mixture component L, the beliefs are restricted to have diagonal inverse covariance matrices, then a corresponding joint Gaussian distribution is obtained directly from the local marginals (i.e., the beliefs lie in the marginal polytope as there is some joint distribution that has these beliefs as marginals). This restriction actually simplifies the algorithm significantly: the projection step only needs to make sure that the diagonal entries are strictly positive and

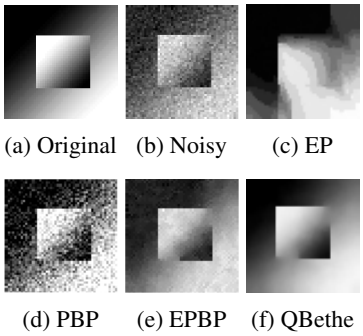
the number of parameters scales as $O(|V|L)$, independent of the size of the maximum clique in non-pairwise models. This is reminiscent of both kernel density estimation (Silverman 1986) and more recently nonparametric variational inference (Gershman, Hoffman, and Blei 2012). The joint probability distribution of the mixture beliefs is simply a mixture of independent Gaussian distributions. Given such a mixture distribution, it is easy to compute any marginal distribution over any subset of variables without computing a single sum. Similarly, even if evidence is introduced, the corresponding conditional marginal distributions are trivial to compute. The added benefit of this approach is that, in situations where repeated marginal/conditional marginal inference queries are desired, this method only requires performing approximate inference once whereas message-passing approaches would need to be rerun for each new piece of evidence or desired subset. One might expect that, in practice, L may need to be prohibitively large in order to obtain reasonable approximations - it may need to be exponential in the worst case. However, in the experimental section, we show that these types of approximations perform well when applied to potential functions that arise from real data.

The above strategy can also be applied to discrete MRFs and hybrid discrete/continuous MRFs with very little modification. In the simplest case, the belief associated to each discrete variable is represented as a mixture of discrete univariate probability distributions. The L^{th} component of the belief over each edge is selected to be the product of the L^{th} components of the univariate distributions of its endpoints (independent of whether they are discrete or continuous). Projected gradient descent can then be performed as before with the additional constraint that discrete univariate beliefs are nonnegative and sum to one.

Beyond Pairwise Models: High-Arity Cliques

One potential drawback of the proposed scheme is that if the model involves potential functions of arity r, then a naïve application of iterated quadrature schemes (one for each dimension as described above) for approximating the integrals in the BFE, would require a sum of size K_Q^r. This isn't surprising as it matches the scaling in the discrete cases. By way of contrast, the particle methods can determine the number of particles independent of the arity.

In practice, we may hope to approximate these integrals with alternative methods. One possible solution is to use sampling methods to approximate the integrals. In this case, sampling from a mixture of independent Gaussian distributions can be done efficiently. An alternative strategy could make use of SVGD to select good particles to approximate the appropriate expectations in the BFE. In either case, the number of samples/particles could be chosen to be independent of r, and as is the case for quadrature methods, we can hope that a small number of particles is sufficient to guarantee a good approximation in practice. With the above modifications, our approach is likely to be applicable on a wide variety of graphical models that would be computationally prohibitive for existing methods.



(a) Original (b) Noisy (c) EP

(d) PBP (e) EPBP (f) QBethe

Figure 1: Approximate denoising of the 50×50 image (b).

Experimental Results

In the following experiments, we evaluate our method on a variety of UCI datasets, an image denoising task, and a synthetic problem over a three node cycle. As the aim is to showcase performance on inference tasks, for all problems the potentials are assumed to be known in advance or have been learned using a separate method. We implemented our approach that approximates the beliefs as independent Gaussian mixtures, dubbed QBethe, using standard projected gradient ascent with a diminishing step size rule in MATLAB without parallelization. We compare against the Gaussian EP, EPBP, and PBP methods (also implemented in MATLAB) as other methods either cannot perform inference with arbitrary potentials (e.g., kernel BP) or require the potential functions to be normalizable (e.g., SOSMP). For consistency with prior work, PBP and EPBP were implemented to match Lienart, Teh, and Doucet (2015) and Ihler and McAllester (2009). For PBP, this means that the current belief is used as a proposal and MCMC is used for sampling. Initial means and particle points are sampled independently from a normal distribution with mean and standard deviation determined by data.

Approximate Inference in Chow-Liu Trees

Our aim in this section is to experimentally evaluate the performance of the proposed method for the marginal inference task in tree-structured, continuous MRFs. While we will restrict to trees so that we can evaluate the performance of each method against an accurate ground truth, we note that this problem is highly non-trivial for arbitrary continuous potential functions. This is also a best-case scenario for the particle methods as convergence of the message-updates is not problematic on trees. We will show that QBethe, despite making additional approximations, compares favorably to the particle methods on tree-structured models arising from real data.

For this set of experiments, we selected a variety of data sets from the UCI Machine Learning Repository (Dheeru and Karra Taniskidou 2017) with between 4 and 30 variables. For each data set, we first learned a Chow-Liu tree representation (Chow and Liu 1968) over the continuous variables using Parzen windows, e.g., (Ni et al. 2017)[1].

[1]The probabilities produced by this method do not necessarily marginalize to each other due to slightly different variances. We

	Average Z			Average Univariate KL divergence		
Dataset	**PBP**	**EPBP**	**QBethe**	**PBP**	**EPBP**	**QBethe**
Iris	0.20 ± 0.17	0.37 ± 0.16	0.97 ± 0.02	0.35 ± 0.33	0.25 ± 0.14	0.00 ± 0.00
B.N.	0.15 ± 0.18	0.00 ± 0.00	0.87 ± 0.01	0.62 ± 0.59	0.83 ± 0.01	0.06 ± 0.00
I.S.E.	0.00 ± 0.01	0.06 ± 0.00	0.54 ± 0.02	0.78 ± 0.37	0.30 ± 0.00	0.21 ± 0.05
Seeds	0.12 ± 0.12	0.49 ± 0.15	0.84 ± 0.05	0.29 ± 0.18	0.12 ± 0.03	0.02 ± 0.01
Yeast	0.04 ± 0.12	0.00 ± 0.00	0.67 ± 0.07	3.31 ± 3.61	1.18 ± 0.09	0.24 ± 0.05
Wdbc	0.05 ± 0.18	0.27 ± 0.20	0.21 ± 0.06	0.10 ± 0.07	0.58 ± 0.14	0.18 ± 0.19
Letter	0.00 ± 0.00	0.00 ± 0.00	0.26 ± 0.05	0.57 ± 0.26	0.73 ± 0.01	0.07 ± 0.02
Poker	0.62 ± 0.12	0.01 ± 0.00	0.63 ± 0.05	0.02 ± 0.01	0.32 ± 0.00	0.06 ± 0.01
CMSC	0.32 ± 0.08	0.47 ± 0.01	0.56 ± 0.02	0.03 ± 0.01	0.02 ± 0.00	0.02 ± 0.00

Table 1: Inference on tree-structured models. All numbers are rounded to two decimal places. In all cases, $Z = 1$.

This yields a tree $G = (V, E)$, a collection of probabilities, $p_{i \in V}$, for each variable in the model, and a collection of probabilities, $p_{(i,j) \in E}$, for each pair of variables joined by an edge in the model. We construct potential functions from these probabilities as follows: for each $i \in V$, $\phi_i(x_i) = p_i(x_i) \prod_{k \in N(i)} M_{ki}(x_i)$, and for each $(i, j) \in E$, $\psi_{ij}(x_i, x_j) = p_{ij}(x_i, x_j)/(p_i(x_i)p_j(x_j)M_{ij}(x_j)M_{ji}(x_i))$, where each message $M_{ij} : \mathbb{R} \to \mathbb{R}_{>0}$ is an arbitrary continuous function. By construction, the partition function of each of these reparameterizations is always equal to one and the exact marginal probabilities are given by the probabilities learned from the Chow-Liu tree procedure. In the simplest case, the messages are constant functions in which case PBP, which uses the univariate beliefs as proposal distributions, would be initialized with the true distributions. In order to make the inference problem slightly more realistic, we chose an edge only factorization: $M_{ij}(x_j) \triangleq p_j(x_j)^{-1/d_j}$, where d_j is the degree of node $j \in G$. In this case, $\phi_i(x_i) = 1$ for all x_i and $\psi_{ij}(x_i, x_j) = p_{ij}(x_i, x_j)/(p_i(x_i)^{1-d_i} p_j(x_j)^{1-d_j})$. Other factorizations were also considered, but we found that the performance of all of the methods to be roughly independent of the simple reparameterizations we considered.

For these experiments, QBethe was run from a random intialization with $K_Q = 4$ quadrature points and $L = 5$ mixture components. PBP and EPBP were run with 20 particles to ensure that all three methods have roughly the same per iteration complexity and use the same number of points in the integral approximations. The resulting estimates of the continuous marginals were plugged into the BFE to estimate the log-partition function. The EP algorithm was not used in these experiments as it tended to produce poor estimates of the partition function in nearly all cases. Note that in all models, the exact Z value is equal to one and KL divergence values closer to zero indicate better approximations.

Each method was run 20 times, and the average Z value and the average KL divergence values between the exact and approximate node beliefs over all variables are reported in Table 1. For EPBP and PBP, the KL divergence is calculated based on continuous beliefs obtained from the converged particles. QBethe significantly outperforms both PBP and EPBP on average on most data sets both in terms of the estimate of $\log Z$ and in terms of the KL divergence on univariate marginals. This isn't completely surprising as all methods are using the same number of points for approximate integration and QBethe selects optimal points in the sense of Theorem 1. Note also that even when the particle methods do a reasonable job of estimating the marginal distributions, they do not guarantee that the approximate beliefs marginalize to each other over the entire real line. As a result, the corresponding estimate of the partition function may be inaccurate when the approximate beliefs are plugged into the BFE. As an example consider the EPBP results for Wdbc in Table 1. While EPBP appears to have the best estimate of the partition function on average, its KL divergence is significantly worse than QBethe, which suggests that the EPBP beliefs likely did not satisfy the marginalization conditions.

ensure the marginalizatinon conditions are satisfied for purposes of this evaluation.

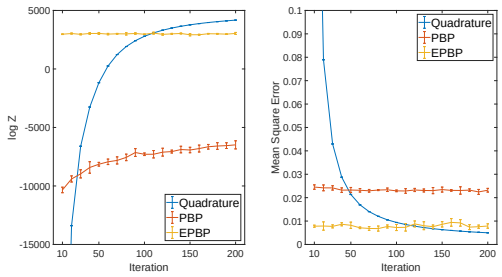


Figure 2: Estimate of the partition function and mean square error of EPBP and QBethe for the image denoising problem.

In summary, while all methods seem to return somewhat reasonable univariate KL divergences on average, accurate estimation of $\log Z$ from the corresponding continuous marginals requires converged marginals that satisfy the local marginalization conditions. PBP only has these guarantees at the converged particle points. As a result, if the aim is to compute $\log Z$, QBethe appears to be preferable in practice. Increasing the number of particles improves accuracy at significant cost, e.g., PBP requires more than 100 particles ($25\times$ slower) to be comparable, on average, to QBethe on Iris.

Image Denoising

As a second application, we consider a simple image denoising problem (Lienart, Teh, and Doucet 2015). Of particular note is that the edge potentials in this model are not integrable by themselves. The aim of this experiment is to demonstrate that the proposed method is practical on medium sized models with thousands of variables, and it still outperforms the particle methods in terms of maximizing the BFE in this setting. For the denoising task, the input is a

50×50 image that has been corrupted with Gaussian noise. The model is a 50×50 grid graph in which each pixel in the image corresponds to a node in the graph and neighboring pixels in the image are joined by an edge. For this problem, the node and edge potentials are selected to bias node u to take the value in the noisy image y_u such that neighboring nodes are encouraged to have the same denoised value subject to a cutoff: $\phi_u(x_u) = \mathcal{N}(x_u - y_u; 0, 0.01)$ and $\psi_{uv}(x_u, x_v) = \mathcal{L}^\lambda(x_u - x_v; 0, 0.03)$, where $\mathcal{L}^\lambda(x; \mu, \nu)$ is a truncated Laplace distribution.

$$\mathcal{L}^\lambda(x; \mu, \nu) = \begin{cases} \mathcal{L}(x; \mu, \nu), & |x| \leq \lambda \\ \mathcal{L}(\lambda; \mu, \nu), & otherwise \end{cases}.$$

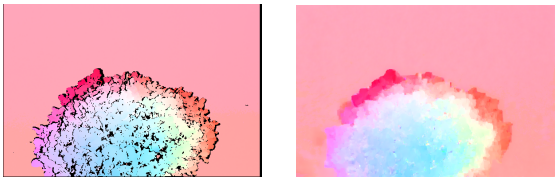
In our experiments, λ was set to 0.2 to maintain consistency with prior work (Lienart, Teh, and Doucet 2015). The number of particles for the sampling methods was set to 100. QBethe was run with $L = 1$ and three quadrature points. For all algorithms, the mean value of the approximate node marginals was selected as the denoised value for the corresponding pixel. PBP is slow to converge on the model - we tried different proposals, but were unable to improve convergence or quality dramatically. Figure 1 shows the results in which the values for all pixels were scaled into the interval $[0, 1]$ and plotted as a grayscale image. QBethe and EPBP both produce reasonable looking denoisings while EP produces a poor estimate of the ground truth, suggesting that the true marginal distributions are possibly multimodal and that EP converges to a poor local optimum. Figure 2 compares the per iteration quality of the particle methods and QBethe in terms of the value of the log-partition function and the mean-squared error (MSE). While QBethe starts at a significantly worse solution, it quickly surpasses EPBP and PBP both in terms of approximation of the log-partition function and in terms of MSE. While more iterations of QBethe are necessary to achieve comparable performance, note that QBethe can be implemented extremely efficiently using matrix-vector operations in MATLAB. The average per iteration complexities for each method are quite different: QBethe (.02s), EPBP (50s), PBP (305s). As a result, even though one iteration of EPBP yields a reasonable solution, QBethe can perform 2500 iterations in the same amount of time. While all methods can be sped up with parallel processing, QBethe is likely to be much more practical on medium/large scale problems.

Message Passing Convergence Issues

One advantage of the variational approach is that it does not suffer from the types of convergence issues that arise even for standard BP. Using a simple model on a 3-cycle, we demonstrate that convergence issues can result in poor estimates of the partition function with the continuous message-passing algorithms while the variational approach is unaffected.For this experiment, the node and edge potentials are chosen so that the true univariate marginal distributions are trimodal with three well-separated peaks:

$$\phi_u(x_u) = e^{-.1|x_u|}, \ \psi_{uv}(x_u, x_v) = (f_{10} + f_{-10})(x_u, x_v)$$
$$f_a(x_u, x_v) = e^{-.1(x_u - a)^2 - .1(x_v + a)^2}$$

We examined the estimate of the partition function produced by QBethe, with different numbers of mixture components using three quadrature points, and the particle methods for varying numbers of particles. We computed the average log-partition function using 50 trials of 150 iterations of each method. For reference, $\log Z = -16.17$ for this model. QBethe returns accurate approximations of the log-partition function even with a small number of mixture components, and as L increases, QBethe returns better answers on average: $-17.94 \pm 0, -17.56 \pm 0.36, -17.32 \pm 0.42, -17.07 \pm 0.45, -16.88 \pm 0.41, -16.78 \pm 0.42$, for $L = 1, \ldots, 6$ respectively. Selecting more mixture components increases the chance that the algorithm converges to an approximation that correctly identifies all three peaks. Contrast this with the estimates of $\log Z$ generated by PBP where the partition function estimates can be orders of magnitude more extreme: $-47.34 \pm 38.03, -25.57 \pm 31.05, -22.24 \pm 19.05, -20.64 \pm 19.34, -10.61 \pm 12.62$, for $M \in \{5, 25, 50, 75, 100\}$. Although the variance does appear to decrease as the number of particles increases, it is still over 12 orders of magnitude with 100 particles. Closer inspection shows that the messages produced by PBP do not appear to be converging for this model. Changing to a rejection sampling procedure using a Gaussian proposal reduced the variance, but did not appear to result in convergence. The average KL-divergence of PBP is also poor: 154 even with 100 particles. EPBP produces similarly poor estimates of $\log Z$ and an average KL-divergence of 147 indepedent of the number of particles.



(a) Ground Truth (b) QBethe

Figure 3: Performance of QBethe on an optical flow estimation task.

Discussion

In summary, the proposed method outperforms the state-of-the-art message-passing algorithms on standard marginal inference tasks both in terms of speed and accuracy: the method scales well to larger models and does not suffer from the kinds of convergence issues that are common for message-passing algorithms on loopy graphs. To further demonstrate the scalability of the method, we applied it to optical flow estimation. For this demonstration, we extended the standard discrete formulation (Sun, Roth, and Black 2010) to the continuous case using a bicubic interpolation. Consider Hydrangea from the Middlebury Optical Flow data set (Baker et al. 2011). The flow model for this image is a 584×388 grid graph with over $200,000$ random variables. Max-product versions of the particle message-passing schemes have been applied for this problem, but they reduce the number of nodes to around 9000 using superpixels (Pacheco and Sudderth 2015). Even with this reduction, the methods require roughly 50 seconds per iteration with 10 particles. We applied our method directly on the pixel level with $L = 1$ and $K_Q = 11$ near the zero

temperature limit with a GPU implementation on an NVIDIA Tesla V100. Our method completed 5,000 iterations in just under 200 seconds (.04 sec/iteration) and returned an average end-point error of 0.273. The ground truth and our approximate result can be found in Figure 3. Given the practicality and flexibility of our approach, we plan to apply these methods to inference and learning problems in computer vision and relational models with both discrete and continuous variables, especially on models for which existing methods are difficult to apply due to scale or other limitations.

Acknowledgments

This work was supported by the DARPA Explainable Artificial Intelligence (XAI) program under contract number N66001-17-2-4032 and NSF grant III-1527312.

References

Baker, S.; Scharstein, D.; Lewis, J. P.; Roth, S.; Black, M. J.; and Szeliski, R. 2011. A database and evaluation methodology for optical flow. *International Journal of Computer Vision* 92(1):1–31.

Chow, C., and Liu, C. 1968. Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory* 14(3):462–467.

Dheeru, D., and Karra Taniskidou, E. 2017. UCI machine learning repository.

Fleet, D., and Weiss, Y. 2006. Optical flow estimation. In *Handbook of mathematical models in computer vision*. Springer. 237–257.

Frank, A.; Smyth, P.; and Ihler, A. T. 2009. Particle-based variational inference for continuous systems. In *Advances in Neural Information Processing Systems (NIPS)*, 826–834.

Gershman, S. J.; Hoffman, M. D.; and Blei, D. M. 2012. Nonparametric variational inference. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 235–242.

Golub, G. H., and Welsch, J. H. 1969. Calculation of Gauss quadrature rules. *Mathematics of computation* 23(106):221–230.

Heskes, T., and Zoeter, O. 2002. Expectation propagation for approximate inference in dynamic bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 216–223.

Heskes, T., and Zoeter, O. 2005. Gaussian quadrature based expectation propagation. In *Proceedings of the Eighth International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Huber, M. F.; Bailey, T.; Durrant-Whyte, H.; and Hanebeck, U. D. 2008. On entropy approximation for Gaussian mixture random vectors. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), IEEE International Conference on*, 181–188.

Ihler, A. T., and McAllester, D. A. 2009. Particle belief propagation. In *Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 256–263.

Lienart, T.; Teh, Y. W.; and Doucet, A. 2015. Expectation particle belief propagation. In *Advances in Neural Information Processing Systems (NIPS)*, 3609–3617.

Liu, Q.; Lee, J.; and Jordan, M. 2016. A kernelized Stein discrepancy for goodness-of-fit tests. In *International Conference on Machine Learning (ICML)*, 276–284.

Malioutov, D. M.; Johnson, J. K.; and Willsky, A. S. 2006. Walk-sums and belief propagation in Gaussian graphical models. *Journal of Machine Learning Research* 7:2031–2064.

Minka, T. P. 2001. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in Artificial Intelligence (UAI)*, 362–369.

Ni, X.; Quadrianto, N.; Wang, Y.; and Chen, C. 2017. Composing tree graphical models with persistent homology features for clustering mixed-type data. In *International Conference on Machine Learning (ICML)*, 2622–2631.

Noorshams, N., and Wainwright, M. J. 2013. Belief propagation for continuous state spaces: stochastic message-passing with quantitative guarantees. *Journal of Machine Learning Research (JMLR)* 14(1):2799–2835.

Pacheco, J., and Sudderth, E. 2015. Proteins, particles, and pseudo-max-marginals: a submodular approach. In *International Conference on Machine Learning (ICML)*, 2200–2208.

Ruozzi, N., and Tatikonda, S. 2013. Message-passing algorithms for quadratic minimization. *The Journal of Machine Learning Research* 14(1):2287–2314.

Ruozzi, N. 2017. A Lower Bound on the Partition Function of Attractive Graphical Models in the Continuous Case. In Singh, A., and Zhu, J., eds., *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54 of *Proceedings of Machine Learning Research*, 1048–1056. Fort Lauderdale, FL, USA: PMLR.

Silverman, B. W. 1986. *Density estimation for statistics and data analysis*. Chapman & Hall/CRC.

Song, L.; Gretton, A.; Bickson, D.; Low, Y.; and Guestrin, C. 2011. Kernel belief propagation. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 707–715.

Sudderth, E. B.; Ihler, A. T.; Isard, M.; Freeman, W. T.; and Willsky, A. S. 2003. Nonparametric belief propagation. In *Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society Conference on*.

Sun, D.; Roth, S.; and Black, M. J. 2010. Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2432–2439. IEEE.

Wang, D.; Zeng, Z.; and Liu, Q. 2018. Stein variational message passing for continuous graphical models. In *International Conference on Machine Learning (ICML)*.

Weiss, Y., and Freeman, W. T. 2001. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. *Neural computation* 13(10):2173–2200.

Welling, M., and Teh, Y. W. 2001. Belief optimization for binary networks: A stable alternative to loopy belief propagation. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 554–561.

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