Expectation Particle Belief Propagation

Authored by:

Arnaud Doucet
Yee Whye Teh
Thibaut Lienart

Abstract

We propose an original particle-based implementation of the Loopy Belief Propagation (LPB) algorithm for pairwise Markov Random Fields (MRF) on a continuous state space. The algorithm constructs adaptively efficient proposal distributions approximating the local beliefs at each note of the MRF. This is achieved by considering proposal distributions in the exponential family whose parameters are updated iterately in an Expectation Propagation (EP) framework. The proposed particle scheme provides consistent estimation of the LBP marginals as the number of particles increases. We demonstrate that it provides more accurate results than the Particle Belief Propagation (PBP) algorithm of Ihler and McAllester (2009) at a fraction of the computational cost and is additionally more robust empirically. The computational complexity of our algorithm at each iteration is quadratic in the number of particles. We also propose an accelerated implementation with sub-quadratic computational complexity which still provides consistent estimates of the loopy BP marginal distributions and performs almost as well as the original procedure.

1 Paper Body

Undirected Graphical Models (also known as Markov Random Fields) provide a flexible framework to represent networks of random variables and have been used in a large variety of applications in machine learning, statistics, signal processing and related fields [2]. For many applications such as tracking [3, 4], sensor networks [5, 6] or computer vision [7, 8, 9] it can be beneficial to define MRF on continuous state-spaces. Given a pairwise MRF, we are here interested in computing the marginal distributions at the nodes of the graph. A popular approach to do this is to consider the Loopy Belief Propagation (LBP) algorithm [10, 11, 2]. LBP relies on the transmission of messages between nodes. However when dealing with continuous random variables, computing these messages exactly is generally intractable. In practice, one must select a way to tractably represent these messages and a way to update these representations following the LBP algorithm. The Nonparametric Belief Propagation (NBP)

algorithm [12] represents the messages with mixtures of Gaussians while the Particle Belief Propagation (PBP) algorithm [1] uses an importance sampling approach. NBP relies on restrictive integrability conditions and does not offer consistent estimators of the LBP messages. PBP offers a way to circumvent these two issues but the implementation suggested proposes sampling from the estimated beliefs which need not be integrable. Moreover, even when they are integrable, sampling from the estimated beliefs is very expensive computationally. Practically the authors of [1] only sample approximately from those using short MCMC runs, leading to biased estimators. In our method, we consider a sequence of proposal distributions at each node from which one can sample particles at a given iteration of the LBP algorithm. The messages are then computed using importance sampling. The novelty of the approach is to propose a principled and automated way of designing a sequence of proposals in a tractable exponential family using the Expectation Prop1

agation (EP) framework [13]. The resulting algorithm, which we call Expectation Particle Belief Propagation (EPBP), does not suffer from restrictive integrability conditions and sampling is done exactly which implies that we obtain consistent estimators of the LBP messages. The method is empirically shown to yield better approximations to the LBP beliefs than the implementation suggested in [1], at a much reduced computational cost, and than EP.

2 2.1

Background Notations

We consider a pairwise MRF, i.e. a distribution over a set of p random variables indexed by a set V = {1, . . . , p}, which factorizes according to an undirected graph G = (V, E) with Y Y p(xV) ? ?u (xu) ?uv (xu , xv). (1) u?V

(u,v)?E

The random variables are assumed to take values on a continuous, possibly unbounded, space X . The positive functions ?u : X 7? R+ and ?uv : X ? X 7? R+ are respectively known as the node and edge potentials. The aim is to approximate the marginals pu (xu) for all u ? V . A popular approach is the LBP algorithm discussed earlier. This algorithm is a fixed point iteration scheme yielding approximations called the beliefs at each node [10, 2]. When the underlying graph is a tree, the resulting beliefs can be shown to be proportional to the exact marginals. This is not the case in the presence of loops in the graph. However, even in these cases, LBP has been shown to provide good approximations in a wide range of situations [14, 11]. The LBP fixed-point iteration can be written as follows at iteration t: ? Z Y ? t ? m (x) = ? (x , x)? (x) mt?1 ? v uv u v u u uv wu (xu)dxu ? w??u v , (2) Y ? ? But (xu) = ?u (xu) mtwu (xu) ? ? w??u

where ?u denotes the neighborhood of u i.e., the set of nodes {w — (w, u) ? E}, muv is known as the message from node u to node v and Bu is the belief at node u. 2.2

Related work

The crux of any generic implementation of LBP for continuous state spaces is to select a way to represent the messages and design an appropriate method to

compute/approximate the message update. In Nonparametric BP (NBP) [12], the messages are represented by mixtures of Gaussians. In theory, computing the product of such messages can be done analytically but in practice this is impractical due to the exponential growth in the number of terms to consider. To circumvent this issue, the authors suggest an importance sampling approach targeting the beliefs and fitting mixtures of Gaussians to the resulting weighted particles. The computation of the update (2) is then always done over a constant number of terms. A restriction of ?vanilla? Nonparametric BP is that the messages must be finitely integrable for the message representation to make sense. This is the case if the following two conditions hold: Z Z sup ?uv (xu , xv)dxu ¡ ?, and ?u (xu)dxu ¡ ?. (3) xv

These conditions do however not hold in a number of important cases as acknowledged in [3]. For instance, the potential ?u (xu) is usually proportional to a likelihood of the form p(yu —xu) which need not be integrable in xu . Similarly, in imaging applications for example, the edge potential can encode similarity between pixels which also need not verify the integrability condition as in [15]. Further, NBP does not offer consistent estimators of the LBP messages. Particle BP (PBP) [1] offers a way to overcome the shortcomings of NBP: the authors also consider importance sampling to tackle the update of the messages but without fitting a mixture of Gaussians. 2

(i)

For a chosen proposal distribution qu on node u and a draw of N particles {xu }N i=1 ? qu (xu), the messages are represented as mixtures: m b PBP uv (xv) :=

N X

(i) ?uv ?uv (x(i) u , xv),

(i) with ?uv :=

i=1

(i) 1 ?u (xu) Y (i) m b PBP wu (xu). N qu (xu(i)) w??u v

(4)

This algorithm has the advantage that it does not require the conditions (3) to hold. The authors suggest two possible choices of sampling distributions: sampling from the local potential ?u , or sampling from the current belief estimate. The first case is only valid if ?u is integrable w.r.t. xu which, as we have mentioned earlier, might not be the case in general and the second case implies sampling from a distribution of the form Y buPBP (xu) ? ?u (xu) B m b PBP (5) wu (xu) w??u

which is a product of mixtures. As in NBP, na??ve sampling of the proposal has complexity O(N —?u —) and is thus in general too expensive to consider. Alternatively, as the authors suggest, one can run a short MCMC simulation targeting it which reduces the complexity to order O(—?u —N 2) since the buPBP point-wise, is of order O(—?u —N), and we cost of each iteration, which requires evaluating B need O(N) iterations of the MCMC simulation. The issue with this approach is that it is still computationally expensive, and it is unclear how many iterations are necessary to get N good samples. 2.3

Our contribution

In this paper, we consider the general context where the edge and node-potentials might be nonnormalizable and non-Gaussian. Our proposed method is based on PBP, as PBP is theoretically better suited than NBP since, as discussed earlier, it does not require the conditions (3) to hold, and, provided that one samples from the proposals exactly, it yields consistent estimators of the LBP messages while NBP does not. Further, the development of our method also formally shows that considering proposals close to the beliefs, as suggested by [1], is a good idea. Our core observation is that since sampling from a proposal of the form (5) using MCMC simulation is very expensive, we should consider using a more tractable proposal distribution instead. However it is important that the proposal distribution is constructed adaptively, taking into account evidence collected through the message passing itself, and we propose to achieve this by using proposal distributions lying in a tractable exponential family, and adapted using the Expectation Propagation (EP) framework [13].

3

Expectation Particle Belief Propagation

Our aim is to address the issue of selecting the proposals in the PBP algorithm. We suggest using exponential family distributions as the proposals on a node for computational efficiency reasons, with parameters chosen adaptively based on current estimates of beliefs and EP. Each step of our algorithm involves both a projection onto the exponential family as in EP, as well as a particle approximation of the LBP message, hence we will refer to our method as Expectation Particle Belief Propagation or EPBP for short. For each pair of adjacent nodes u and v, we will use muv (xv) to denote the exact (but unavailable) LBP message from u to v, m b uv (xv) to denote the particle approximation of muv , and ?uv an exponential family projection of m b uv . In addition, let ??u denote an exponential family projection of the node potential ?u . We will consider approximations consisting of N particles. In the following, we will derive the form of our particle approximated message m b uv (xv), along with the choice of the proposal distribution qu (xu) used to construct m b uv . Our starting point is the edge-wise belief over xu and xv , given the incoming particle approximated messages, Y Y buv (xu , xv) ? ?uv (xu , xv)?u (xu)?v (xv) B m b wu (xu) m b ?v (xv). (6) w??u v

???v u

buv (xv), The exact LBP message muv (xv) can be derived by computing the marginal distribution B and constructing muv (xv) such that buv (xv) ? muv (xv)M cvu (xv), B 3

(7)

cvu (xv) = ?v (xv) Q where M b ?v (xv) is the (particle approximated) pre-message from v to ???v u m u. It is easy to see that the resulting message is as expected, Z Y muv (xv) ? ?uv (xu , xv)?u (xu) m b wu (xu)dxu . (8) w??u v

Since the above exact LBP belief and message are intractable in our scenario of interest, the idea buv (xu , xv) instead. Consider a proposal distribution is to use an importance sampler targeting B of the form qu (xu)qv (xv). Since xu and xv are independent under the proposal, we can draw (i) (j) N N independent

samples, say {xu }N i=1 and {xv }j=1 , from qu and qv respectively. We can then approximate the belief using a N ? N cross product of the particles, N (j) X buv (x(i) B u , xv) buv (xu , xv) ? 1 B ?(x(i) ,x(j)) (xu , xv) u v N 2 i,j=1 qu (xu(i))qv (x(j) v) (i) (j) (i) c (j) Q (i) N b wu (xu) vu (xv) 1 X ?uv (xu , xv)?u (xu)M w??u v m ? 2 ?(x(i) ,x(j)) (xu , xv) (i) (j) u v N i,j=1 qu (xu)qv (xv)

(9)

buv (xv), Marginalizing onto xv , we have the following particle approximation to B N (j) (j) c b uv (xv)M 1 Xm vu (xv) b ?x(j) (xv) Buv (xv) ? (j) v N j=1 qv (xv)

(10)

where the particle approximated message m b uv (xv) from u to v has the form of the message representation in the PBP algorithm (4). buv . To determine sensible proposal distributions, we can find qu and qv that are close to the target B buv kqu qv) as the measure of closeness, the optimal qu required for the Using the KL divergence KL(B u to v message is the node belief, Y buv (xu) ? ?u (xu) B m b wu (xu) (11) w??u

thus supporting the claim in [1] that a good proposal to use is the current estimate of the node belief. As pointed out in Section 2, it is computationally inefficient to use the particle approximated node belief as the proposal distribution. An idea is to use a tractable exponential family distribution for qu instead, say Y qu (xu) ? ??u (xu) ?wu (xu) (12) w??u

where ??u and ?wu are exponential family approximations of ?u and m b wu respectively. In Section 4 we use a Gaussian family, but we are not limited to this. Using the framework of expectation propogation (EP) [13], we can iteratively find good exponential family approximations as follows. w For each w ? ?u , to update the ?wu , we form the cavity distribution qu ? qu /?wu and the correw + sponding tilted distribution m b wu qu . The updated ?wu is the exponential family factor minimising the KL divergence,

h i

+ ?wu = arg min KL m b wu (xu)quw (xu) ?(xu)quw (xu) . (13) ??exp.fam.

Geometrically, the update projects the tilted distribution onto the exponential family manifold. The optimal solution requires computing the moments of the tilted distribution through numeriw cal quadrature, and selecting ?wu so that ?wu qu matches the moments of the tilted distribution. In our scenario the moment computation can be performed crudely on a small number of evaluation points since it only concerns the updating of the importance sampling proposal. If an optimal ? in the exponential family does not exist, e.g. in the Gaussian case that the optimal ? has a negative variance, we simply revert ?wu to its previous value [13]. An analogous update is used for ??u . In the above derivation, the expectation propagation steps for each incoming message into u and for the node potential are performed first, to fit the proposal to the current estimated belief at u, before 4

it is used to draw N particles, which can then be used to form the particle approximated messages from u to each of its neighbours. Alternatively, once

each particle approximated message m b uv (xv) is formed, we can update its exponential family projection ?uv (xv) immediately. This alternative scheme is described in Algorithm 1. Algorithm 1 Node update (i)

1: sample {xu } ? qu (?) (i) Q (i) bu (x(i) 2: compute B b wu (xu) u) = ?u (xu) w??u m 3: for v ? ?u do cuv (x(i) b (i) b vu (xu(i)) 4: compute M u) := Bu (xu)/m (i) (i) cuv (x(i) 5: compute the normalized weights wuv ? M u)/qu (xu) PN (i) (i) 6: update the estimator of the outgoing message m b uv (xv) = i=1 wuv ?uv (xu , xv) ?

+ compute the cavity distribution qv ? qv /??v , get ??v in the exponential family such that ? + ? + + ??v qv approximates ?v qv , update qv ? ??v and let ??v ? ??v u + 8: compute the cavity distribution qv ? qv /?uv , get ?uv in the exponential family such that u + u + + qv approximates m b uv qv , update qv ? ?uv and let ?uv ? ?uv ?uv 9: end for

7:

3.1

Computational complexity and sub-quadratic implementation

Each EP projection step costs O(N) computations since the message m b wu is a mixture of N components (see (4)). Drawing N particles from the exponential family proposal qu costs O(N). The step with highest computational complexity is in evaluating the particle weights in (4). Indeed, evaluating the mixture representation of a message on a single point is O(N), and we need to compute this for each of N particles. Similarly, evaluating the estimator of the belief on N sampling points at node u requires O(—?u —N 2). This can be reduced since the algorithm still provides consistent estimators if we consider the evaluation of unbiased estimators of the messages instead. Since the messages PN i i have the form m b uv (xv) = i=1 wuv ?uv (xv), we can follow a method presented in [16] where ? M i }N one draws M indices {i‘ }‘=1 from a multinomial with weights {wuv i=1 and evaluates the correi? ‘ sponding M components ?uv . This reduces the cost of the evaluation of the beliefs to O(—?u —M N) which leads to an overall sub-quadratic complexity if M is o(N). We show in the next section how it compares to the quadratic implementation when M = O(log N).

4

Experiments

We investigate the performance of our method on MRFs for two simple graphs. This allows us to compare the performance of EPBP to the performance of PBP in depth. We also illustrate the behavior of the sub-quadratic version of EPBP. Finally we show that EPBP provides good results in a simple denoising application. 4.1

Comparison with PBP

We start by comparing EPBP to PBP as implemented by Ihler et al. on a 3 ? 3 grid (figure 1) with random variables taking values on R. The node and edge potentials are selected such that the marginals are multimodal, non-Gaussian and skewed with

?u (xu) = ?1 N (xu ? yu ; ?2, 1) + ?2 G(xu ? yu ; 2, 1.3) , (14) ?uv (xu , xv) = L(xu ? xv ; 0, 2) where yu denotes the observation at node u,

N (x; ?, ?) ? exp(?x2 /2? 2) (density of a Normal distribution), G(x; ?, ?) ? exp(?(x??)/? +exp(?(x??)/?)) (density of a Gumbel distribution) and L(x; ?, ?) ? exp(?—x ? ?—/?) (density of a Laplace distribution). The parameters ?1 and ?2 are respectively set to 0.6 and 0.4. We compare the two methods after 20 LBP iterations.1 1 The scheduling used alternates between the classical orderings: top-down-left-right, left-right-top-down, down-up-right-left and right-left-down-up. One ?LBP iteration? implies that all nodes have been updated once.

5
1
4
7
2
5
8
3
6
9
1 2 4
5
3 6
7
8

Figure 1: Illustration of the grid (left) and tree (right) graphs used in the experiments. PBP as presented in [1] is implemented using the same parameters than those in an implementation code provided by the authors: the proposal on each node is the last estimated belief and sampled with a 20-step MCMC chain, the MH proposal is a normal distribution. For EPBP, the approximation of the messages are Gaussians. The ground truth is approximated by running LBP on a deterministic equally spaced mesh with 200 points. All simulations were run with Julia on a Mac with 2.5 GHz Intel Core i5 processor, our code is available online.2 Figure 2 compares the performances of both methods. The error is computed as the mean L1 error over all nodes between the estimated beliefs and the ground truth evaluated over the same deterministic mesh. One can observe that not only does PBP perform worse than EPBP but also that the error plateaus with increasing number of samples. This is because the secondampling within PBP is done approximately and hence the consistency of the estimators is lost. The speed-up offered by EPBP is very substantial (figure 4 left). Hence, although it would be possible to use more MCMC (Metropolis-Hastings) iterations within PBP to improve its performance, it would ? make the method prohibitively expensive to use. Note that for EPBP, one observes the usual 1/ N convergence of particle methods. Figure 3 compares the estimator of the beliefs obtained by the two methods for three arbitrarily picked nodes (node 1, 5 and 9 as illustrated on figure 1). The figure also illustrates the last proposals constructed with our approach and one notices that their supports match closely the support of the true beliefs. Figure 4 left illustrates how the estimated beliefs

converge as compared to the true beliefs with increasing number of iterations. One can observe that PBP converges more slowly and that the results display more variability which might be due to the MCMC runs being too short. We repeated the experiments on a tree with 8 nodes (figure 1 right) where we know that, at convergence, the beliefs computed using BP are proportional to the true marginals. The node and edge potentials are again picked such that the marginals are multimodal with

?u (xu) = ?1 N (xu ? yu ; ?2, 1) + ?2 N (xu ? yu ; 1, 0.5) , (15) ?uv (xu , xv) = L(xu ? xv ; 0, 1) with ?1 = 0.3 and ?2 = 0.7. On this example, we also show how ?pure EP? with normal distributions performs. We also try using the distributions obtained with EP as proposals for PBP (referred to as ?PBP after EP? in figures). Both methods underperform compared to EPBP as illustrated visually in Figure 5. In particular one can observe in Figure 3 that ?PBP after EP? converges slower than EPBP with increasing number of samples. 4.2

Sub-quadratic implementation and denoising application

As outlined in Section 3.1, in the implementation of EPBP one can use an unbiased estimator of the edge weights based on a draw of M components from a multinomial. The complexity of the resulting algorithm is O(M N). We apply this method to the 3 ? 3 grid example in the case where M is picked to be roughly of order log(N): i.e., for N = {10, 20, 50, 100, 200, 500}, we pick M = {5, 6, 8, 10, 11, 13}. The results are illustrated in Figure 6 where one can see that the N log N implementation compares very well to the original quadratic implementation at a much reduced cost. We apply this sub-quadratic method on a simple probabilistic model for an image denoising problem. The aim of this example is to show that the method can be applied to larger graphs and still provide good results. The model underlined is chosen to showcase the flexibility and applicability of our method in particular when the edge-potential is non-integrable. It is not claimed to be an optimal approach to image denoising.[3] The node and edge potentials are defined as follows:

?u (xu) = N (xu ? yu ; 0, 0.1) , (16) ?uv (xu , xv) = L? (xu ? xv ; 0, 0.03)

2 3

https://github.com/tlienart/EPBP. In this case in particular, an optimization-based method such as [17] is likely to yield better results.

where L? (x; ?, ?) = L(x; ?, ?) if —x— ? ? and L(?; ?, ?) otherwise. In this example we set ? = 0.2. The value assigned to each pixel of the reconstruction is the estimated mean obtained over the corresponding node (figure 7). The image has size 50 ? 50 and the simulation was run with N = 30 particles per nodes, M = 5 and 10 BP iterations taking under 2 minutes to complete. We compare it with the result obtained with EP on the same model. 10 0

10 0 EPBP PBP after EP

Mean L1 error

Mean L1 error

PBP EPBP

10 -1

10 -2

10 1
10 2
10 -1
10 -2
10 3
10 1
10 2
Number of samples per node
10 3
Number of samples per node
Figure 2: (left) Comparison of the mean L1 error for PBP and EPBP for the 3 ? 3 grid example. (right) Comparison of the mean L1 error for ?PBP after EP? and EPBP for the tree example. In both cases, EPBP is more accurate for the same number of samples. 0.35
0.6
0.3
0.3
0.5
0.25
0.4
0.2
0.3
0.15
0.25
True belief Estimated belief (EPBP) Estimated belief (PBP) Proposal (EPBP)
0.2 0.15 0.1 0.05 0 -5
0
5
10
15
0.2
0.1
0.1
0.05
0 -5
0
5
10
15
0 -5
0
5
10
15
Figure 3: Comparison of the beliefs on node 1, 5 and 9 as obtained by evaluating LBP on a deterministic mesh (true belief), with PBP and with

EPBP for the 3 ? 3 grid example. The proposal used by EPBP at the last step is also illustrated. The results are obtained with N = 100 samples on each node and 20 BP iterations. One can observe visually that EPBP outperforms PBP.
4.5
4
10
3
10
2
PBP EPBP
3.5
Wall-clock time [s]
3
Mean L1 error
10 EPBP PBP
4
2.5 2 1.5 1
10 1
10 0
0.5 0
0
5
10
15
10 -1
20
Number of BP iterations
10 1
10 2
10 3
Number of samples per node

Figure 4: (left) Comparison of the convergence in L1 error with increasing number of BP iterations for the 3 ? 3 grid example when using N = 30 particles. (right) Comparison of the wall-clock time needed to perform PBP and EPBP on the 3 ? 3 grid example.

5 Discussion

We have presented an original way to design adaptively efficient and easy-to-sample-from proposals for a particle implementation of Loopy Belief Propagation. Our proposal is inspired by the Expectation Propagation framework. We have demonstrated empirically that the resulting algorithm is significantly faster and more accurate than an implementation of PBP using the estimated beliefs as proposals and sampling from them using MCMC as proposed in [1]. It is also more accurate than EP due to the nonparametric nature of the messages and offers consistent estimators of the LBP messages. A sub-quadratic version

of the method was also outlined and shown to perform almost as well as the original method on 7

mildly multi-modal models, it was also applied successfully in a simple image denoising example illustrating that the method can be applied on graphical models with several hundred nodes. We believe that our method could be applied successfully to a wide range of applications such as smoothing for Hidden Markov Models [18], tracking or computer vision [19, 20]. In future work, we will look at considering other divergences than the KL and the ?Power EP? framework [21], we will also look at encapsulating the present algorithm within a sequential Monte Carlo framework and the recent work of Naesseth et al. [22].
1.2

0.9
0.5
0.8
0.45
0.7
0.4
0.6
0.35
1 0.8

True belief Est. bel. (EPBP) Est. bel. (PBP) Est. bel. (EP) Est. bel. (PBP after EP)

0.3
0.5
0.25
0.6 0.4
0.2 0.4
0.3
0.15
0.2
0.1
0.1
0.05
0.2 0 -2
0
2
4
0 -2
6
0
2
4
6
0 -2
0
2

4
6
Figure 5: Comparison of the beliefs on node 1, 3 and 8 as obtained by evaluating LBP on a deterministic mesh, using EPBP, PBP, EP and PBP using the results of EP as proposals. This is for the tree example with N = 100 samples on each node and 20 LBP iterations. Again, one can observe visually that EPBP outperforms the other methods. 10
0
10
2
10
10
NlogN implementation Quadratic implementation
Wall-clock time [s]
Mean L1 error
NlogN implementation Quadratic implementation
-1
-2
10 1
10 2
10 1
10 0
10
10 3
Number of samples
-1
10 1
10 2
10 3
Number of samples per node
Figure 6: Comparison of the mean L1 error for PBP and EPBP on a 3 ? 3 grid (left). For the same number of samples, EPBP is more accurate. It is also faster by about two orders of magnitude (right). The simulations were run several times for the same observations to illustrate the variability of the results.

Figure 7: From left to right: comparison of the original (first), noisy (second) and recovered image using the sub-quadratic implementation of EPBP (third) and with EP (fourth). Acknowledgments We thank Alexander Ihler and Drew Frank for sharing their implementation of Particle Belief Propagation. TL gratefully acknowledges funding from EPSRC (grant 1379622) and the Scatcherd European scholarship scheme. YWT?s research leading to these results has received funding from EPSRC (grant EP/K009362/1) and ERC under the EU?s FP7 Programme (grant agreement no. 617411). AD?s research was supported by the EPSRC (grant EP/K000276/1, EP/K009850/1) and by AFOSR/AOARD (grant AOARD-144042). 8

2 References

[1] Alexander T. Ihler and David A. McAllester. Particle belief propagation. In Proc. 12th AISTATS, pages 256?263, 2009. [2] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. Found. and Tr. in Mach. Learn., 1(1?2):1?305, 2008. [3] Erik B. Sudderth, Alexander T. Ihler, Michael Isard, William T. Freeman, and Alan S. Willsky. Nonparametric belief propagation. Commun. ACM, 53(10):95?102, 2010. [4] Jeremy Schiff, Erik B. Sudderth, and Ken Goldberg. Nonparametric belief propagation for distributed tracking of robot networks with noisy inter-distance measurements. In IROS ?09, pages 1369?1376, 2009. [5] Alexander T. Ihler, John W. Fisher, Randolph L. Moses, and Alan S. Willsky. Nonparametric belief propagation for self-localization of sensor networks. In IEEE Sel. Ar. Comm., volume 23, pages 809?819, 2005. [6] Christopher Crick and Avi Pfeffer. Loopy belief propagation as a basis for communication in sensor networks. In Proc. 19th UAI, pages 159?166, 2003. [7] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. In IEEE Trans. Patt. An. Mach. Int., volume 25, pages 787?800, 2003. [8] Andrea Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In Proc. 18th ICPR, volume 3, pages 15?18, 2006. [9] Nima Noorshams and Martin J. Wainwright. Belief propagation for continuous state spaces: Stochastic message-passing with quantitative guarantees. JMLR, 14:2799?2835, 2013. [10] Judea Pearl. Probabilistic Reasoning in Intelligent Systems. Morgan Kaufman, 1988. [11] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Constructing free energy approximations and generalized belief propagation algorithms. MERL Technical Report, 2002. [12] Erik B. Sudderth, Alexander T. Ihler, William T. Freeman, and Alan S. Willsky. Nonparametric belief propagation. In Procs. IEEE Comp. Vis. Patt. Rec., volume 1, pages 605?612, 2003. [13] Thomas P. Minka. Expectation propagation for approximate Bayesian inference. In Proc. 17th UAI, pages 362?369, 2001. [14] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference: an empirical study. In Proc. 15th UAI, pages 467?475, 1999. [15] Mila Nikolova. Thresholding implied by truncated quadratic regularization. IEEE Trans. Sig. Proc., 48(12):3437?3450, 2000. [16] Mark Briers, Arnaud Doucet, and Sumeetpal S. Singh. Sequential auxiliary particle belief propagation. In Proc. 8th ICIF, volume 1, pages 705?711, 2005. [17] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. Physica D, 60(1):259?268, 1992. [18] M. Briers, A. Doucet, and S. Maskell. Smoothing algorithms for state-space models. Ann. Inst. Stat. Math., 62(1):61?89, 2010. [19] Erik B. Sudderth, Michael I. Mandel, William T. Freeman, and Alan S. Willsky. Visual hand tracking using nonparametric belief propagation. In Procs. IEEE Comp. Vis. Patt. Rec., 2004. [20] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. Int. Journ. Comp. Vis., 59(2), 2004. [21] Thomas P. Minka. Power EP. Technical Report MSR-TR-2004-149, 2004. [22] Christian A. Naesseth, Fredrik

Lindsten, and Thomas B. Sch?on. Sequential monte carlo for graphical models. In Proc. 27th NIPS, pages 1862?1870, 2014.
9