Message passing with l_1 penalized KL minimization

Yuan Qi ALANQI@PURDUE.EDU
Departments of Computer Science and Statistics, Purdue University, West Lafayette, IN 47907 USA

Yandong Guo GUOY@PURDUE.EDU
School of Electrical and Computer Engineering, Purdue University West Lafayette, IN 47907 USA

Abstract

Bayesian inference is often hampered by large computational expense. As a generalization of belief propagation (BP), expectation propagation (EP) approximates exact Bayesian computation with efficient message passing updates. However, when an approximation family used by EP is far from exact posterior distributions, message passing may lead to poor approximation quality and suffer from divergence. To address this issue, we propose an approximate inference method, relaxed expectation propagation (REP), based on a new divergence with a l_1 penalty. Minimizing this penalized divergence *adaptively* relaxes EP's moment matching requirement for message passing. We apply REP to Gaussian process classification and experimental results demonstrate significant improvement of REP over EP and α-divergence based power EP—in terms of algorithmic stability, estimation accuracy and predictive performance. Furthermore, we develop relaxed belief propagation (RBP), a special case of REP, to conduct inference on discrete Markov random fields (MRFs). Our results show improved estimation accuracy of RBP over BP and fractional BP when interactions between MRF nodes are strong.

1. Introduction

Bayesian learning provides a principled framework for modeling complex systems and making predictions. A critical component of Bayesian learning is the computation of posterior distributions that represent estimation uncertainty. However, the exact computation is often so expensive that it has become a bottleneck for practical applications of Bayesian learning. To reduce the computational cost, we can use message passing methods to efficiently approximate the exact posteriors. Two exemplary message passing methods are belief propagation (i.e., the sum-product algorithm) (Kschischang et al., 1998; Pearl, 1982) and expectation propagation (Minka, 2001), a generalization of BP.

Despite their wide success in various applications, BP and EP may degrade their approximation quality and diverge when the exact target distribution is far from the approximating family used by them—for example, when many samples are mislabeled for classification or variables are strongly coupled in a MRF. We can force BP and EP to converge using the CCCP algorithm (Heskes et al., 2005; Yuille, 2002). But not only are the CCCP updates slower than the message passing updates, but also the forced convergence might not be desirable—according to Minka (2001), EP diverges for a good reason, indicating a poor approximating family (or a poor energy function) used by EP. For the difficult cases, it may be too rigid to use moment matching, a natural consequence of KL minimization in BP and EP (see Section 2).

To improve both approximation quality and algorithmic stability of message passing, we propose a new approximate inference method, relaxed expectation propagation (REP). Specifically, we introduce a relaxation factor in the KL minimization and penalize it by a l_1 penalty (See Section 3). The penalized KL minimization is adaptive in moment matching: the l_1 penalty completely prunes the relaxation factor and gives the same moment matching update as in BP or EP, if the original and approximate distributions are similar; if they differ significantly (i.e., when EP struggles), the relaxation factor survives the l_1 penalty and

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

renders the original and projected distributions with different moments. To better understand REP, we also present its primal energy function in Section 3 and its dual energy function in Appendix. The primal energy function of REP has a larger feasible set than that of EP and the Bethe energy of BP, providing a higher chance for finding a better approximation.

In Section 4, we present REP inference on two important models: Gaussian process (GP) classification models and discrete MRFs. GP classification models are powerful predictive tools and have been trained by EP (Kuss & Rasmussen, 2005); MRFs are ubiquitous in scientific and engineering applications and BP is a popular choice for estimating marginal distributions in MRFs. For MRF inference, REP reduces to relaxed belief propagation (RBP). Note that we can easily adopt RBP to inference on Bayesian networks because both MRFs and Bayesian networks can be morphed into factor graph representations (Kschischang et al., 1998). In Section 5, we discuss differences between REP, power EP, and damped EP.

In Section 6, we report experimental results on synthetic and UCI benchmark datasets for GP classification. REP consistently outperforms EP, damped EP, and power EP (Minka, 2004)—in terms of algorithmic stability, estimation accuracy, and predictive performance. The MRF inference results show greatly improved estimation accuracy of RBP over BP and fractional BP (Wiegerinck & Heskes, 2003) when interactions between MRF nodes are strong.

2. Background: EP and BP

Given observations $\mathcal{D}$, the posterior distribution of a probabilistic model with factors $\{t_i(\mathbf{w})\}_{i=1,\dots,N}$ is

$$p(\mathbf{w}|\mathcal{D}) = \frac{1}{Z}\prod_i t_i(\mathbf{w}_i). \tag{1}$$

where Z is the normalization constant and $\mathbf{w}_i$ is a subvector of $\mathbf{w}$ that is associated with the i-th factor t_i. Factors t_i are linked to the observations D. EP approximates each factor in (1):

$$q(\mathbf{w}) \propto \prod_i \tilde{t}_i(\mathbf{w}_i) \tag{2}$$

where $q(\mathbf{w})$ and $\tilde{t}_i(\mathbf{w}_i)$ approximate $p(\mathbf{w}|\mathcal{D})$ and $t_i(\mathbf{w}_i)$, respectively, and have the form of the exponential family—such as Gaussian or factorized discrete distributions. The approximation factor $\tilde{t}_i(\mathbf{w})$ is a message from the i^{th} factor t_i to variables $\mathbf{w}_i$ in a factor graph representation (Kschischang et al., 1998).

To obtain $q(\mathbf{w})$, EP refines the messages by repeating the following three steps: message deletion, belief projection, and message update on each factor. In the message deletion step, we compute the partial posterior $q^{\setminus i}(\mathbf{w})$ by removing a message $\tilde{t}_i$ from the approximate posterior $q^{\text{old}}(\mathbf{w})$: $q^{\setminus i}(\mathbf{w}) \propto q^{\text{old}}(\mathbf{w})/\tilde{t}_i(\mathbf{w}_i)$. In the projection step, we minimize the KL divergence between $\hat{p}_i(\mathbf{w}) \propto t_i(\mathbf{w}_i)q^{\setminus i}(\mathbf{w})$ and the new approximate posterior $q(\mathbf{w})$,

$$KL(\hat{p}_i||q) \tag{3}$$

such that the information from each factor is incorporated into $q(\mathbf{w})$. Finally, the message $\tilde{t}_i$ is updated via $\tilde{t}_i(\mathbf{w}_i) \propto q(\mathbf{w})/q^{\setminus i}(\mathbf{w})$. On discrete Bayesian networks or Markov random fields (MRFs), we can use a factorized approximation $q(\mathbf{w}) = \prod_j q(w_j)$ where j is the node index. Then the EP updates reduce to classical BP or sum-product updates (Minka, 2001).

Since $q(\mathbf{w})$ is in the exponential family, it has the following form

$$q(\mathbf{w}) \propto \exp(\boldsymbol{\nu}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{w}))$$

where $\boldsymbol{\phi}(\mathbf{w})$ are the features of the exponential family. Given this representation, minimizing the KL (3) amounts to the following moment matching constraint between $\hat{p}_i(\mathbf{w})$ and $q(\mathbf{w})$:

$$\int \boldsymbol{\phi}(\mathbf{w})\hat{p}_i(\mathbf{w})\mathrm{d}\mathbf{w} = \int \boldsymbol{\phi}(\mathbf{w})q(\mathbf{w})\mathrm{d}\mathbf{w}. \tag{4}$$

For BP, moment matching means $q(w_j)$ and the marginal of $\hat{p}_i(\mathbf{w})$ are matched, $q(w_j) = \sum_{\mathbf{w}_{\setminus j}} \hat{p}_i(\mathbf{w})$.

Based on moment matching, EP and BP message passing updates capture critical statistics we care about. However, when the approximating family is far from the true distribution, message passing can be too rigid, causing EP and BP to deteriorate their performance.

3. Relaxed Expectation Propagation

In this section, we present a new Bregman distance with l_1 penalty, describe the REP algorithm based on this distance, discuss choices of relaxation factors, and provide the energy function of REP.

3.1. A new divergence

To relax moment matching between $\hat{p}_i(\mathbf{w})$ and $q(\mathbf{w})$, we introduce a relaxation factor $r_i(w) \propto \exp(\boldsymbol{\eta}_i^{\mathrm{T}}\boldsymbol{\phi}(w))$ into the KL divergence and put l_1 penalty on the parameters of r_i. Specifically, we propose the following penalized divergence between $\hat{p}_i$ and q

$$KL_r(\hat{p}_i r_i||q r_i) + c|\boldsymbol{\eta}_i|_1 \tag{5}$$

where $|\boldsymbol{\eta}_i|_1$ is the l_1 norm of $\boldsymbol{\eta}_i$, the weight c controls how much the penalty we give to the relaxation, and

the KL_r divergence is defined for unnormalized distributions. It is easy to show, given r_i, KL_r is a valid Bregman distance between $\hat{p}_i$ and q. Minimizing (5) relaxes moment matching between $\hat{p}_i$ and q. This relaxation is *adaptive*: when the approximating family is significantly different from $\hat{p}_i$, the relaxation factor yields different moments for $\hat{p}_i$ and q; when $\hat{p}_i$ and q are similar, the l_1 penalty will set $\boldsymbol{\eta}_i = \mathbf{0}$ so that we obtain exact moment matching as in EP or BP.

3.2. Algorithm

By iteratively minimizing the penalized divergence (5), we obtain the following REP algorithm:

1. Initialize $q(\mathbf{w}) = 1$ and all the messages $\tilde{t}_i(\mathbf{w}) = 1$.
2. Repeat until all $\tilde{t}_i(\mathbf{w})$ converge: Pick a factor i.
   - **Message deletion:** Based on the current factor $\tilde{t}_i$ and q^{old}, calculate the partial belief $q^{\backslash i} \propto q^{\text{old}}(\mathbf{w})/\tilde{t}_i(\mathbf{w})$.
   - **Belief projection:** Minimize (5) over $q(\mathbf{w})$ and $r_i(\mathbf{w})$ to incorporate information from the factor t_i into the new belief q.
   - **Message update:** Update the message based on the new belief: $\tilde{t}_i(\mathbf{w}) \propto q(\mathbf{w})/q^{\backslash i}(\mathbf{w})$.

For simplicity, we drop the subscript of $\mathbf{w}$ in the i-th factor. Note that when some factors are in the exponential family and have the same feature form $\boldsymbol{\phi}(\mathbf{w})$ as $q(\mathbf{w})$, we can absorb them directly into $q(\mathbf{w})$ in initialization without iterative updates.

3.3. Choice of relaxation factors

For the relaxation factor $r_i(\mathbf{w}) = \exp(\boldsymbol{\eta}_i^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{w}))$, we want to parameterize $\boldsymbol{\eta}_i$ to make the minimization of (5) easy. Clearly there are many choices available. A convenient one is to set $\boldsymbol{\eta}_i$ to be a scaled version of the parameters of the old message $\tilde{t}_i$. It does not cause double-counting of factors, because r_i appears in both sides of the penalized divergence (5) and the new posterior q does not include r_i. With this choice, we can compute (5) analytically, making it easy for joint minimization over q and r_i.

3.4. Energy function

To gain further insight into REP, we derive its energy functions. The primal energy function is

$$\min_{\boldsymbol{\eta}_i, \hat{p}_i} \max_{q} \sum_i \frac{1}{\hat{Z}_i} \int_{\mathbf{w}} \hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) \log \frac{\hat{p}_i(\mathbf{w})}{\hat{Z}_i t_i(\mathbf{w}) p(\mathbf{w})} - (n-1)\frac{1}{Z_q}\int_{\mathbf{w}} q(\mathbf{w}) r_i(\mathbf{w}) \log \frac{q(\mathbf{w})}{Z_q p(\mathbf{w})} + c\sum_i |\boldsymbol{\eta}_i| \tag{6}$$

subject to

$$\frac{1}{\hat{Z}_i}\int_{\mathbf{w}} \phi(\mathbf{w})\hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) d\mathbf{w} = \frac{1}{Z_q}\int_{\mathbf{w}} \phi(\mathbf{w}) q(\mathbf{w}) r_i(\mathbf{w}) d\mathbf{w} \tag{7}$$

where $\int_{\mathbf{w}} \hat{p}_i(\mathbf{w})d\mathbf{w} = 1$, $\int_{\mathbf{w}} q(\mathbf{w})d\mathbf{w} = 1$, $\hat{Z}_i = \int_{\mathbf{w}} \hat{p}_i(\mathbf{w}) r_i(\mathbf{w}) d\mathbf{w}$, and $Z_q = \int_{\mathbf{w}} q(\mathbf{w}) r_i(\mathbf{w}) d\mathbf{w}$.

Based on a KL duality bound, we obtain the dual form of the energy function (See Appendix for details). Setting the gradient of the dual function to zero gives the fixed-point updates. If we set the relaxation factor as a scaled version of the old messages as disccused in Section 3.3, we only need to slightly modify (6) (See Appendix for details). The fixed-point updates do not guarantee convergence like the classical EP updates. However, by relaxing the moment matching requirement between $\hat{p}_i(\mathbf{w})$ and $q(\mathbf{w})$, the new updates are much more robust than classical EP updates. In our experiments, while EP diverged many times on difficult datasets, the new algorithm did *not* diverge once.

From the energy function perspective, we enlarge the feasible set for the energy function of EP. The min-max cost function (6) reduces to that of EP as a special case if we set $r_i(w) = 1$. As shown by (Heskes et al., 2005), the cost function of EP corresponds to the Bethe energy (Yedidia et al., 2003) that approximates the system entropy with the exact moment matching constraint. With the larger feasible set, we can potentially obtain better entropy approximation.

4. REP for GP and MRF inference

In this section, we apply the new algorithm to train binary Gaussian process classification models and to perform inference on discrete Markov random fields.

4.1. REP for Gaussian process classification

First, let us denote N independent and identically distributed samples by $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i$ is a d dimensional input and y_i is a scalar output. We assume there is a latent function f that we are modeling and a noisy realization of f at $\mathbf{x}_i$ is y_i.

We use a GP prior with zero mean over f. Its projection at the samples $\{\mathbf{x}_i\}$ defines a joint Gaussian distribution: $p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, K)$ where $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is the covariance function encoding the prior notation of smoothness. For classification, the data likelihood has the following form

$$p(y_i|f) = (1-\epsilon)\Theta(f(\mathbf{x}_i)y_i) + \epsilon\Theta(-f(\mathbf{x}_i)y_i) \tag{8}$$

where ϵ models the labeling error, $\Theta(\cdot)$ is a step function ($\Theta(a) = 1$ if $a \geq 0$, and $\Theta(a) = 0$ otherwise).

Given the GP prior and the data likelihood, the posterior process of f is

$$p(f|\mathcal{D}) \propto GP(f|0,K)\prod_{i=1}^{N} p(y_i|f(\mathbf{x}_i)). \tag{9}$$

Due to the nonlinearity in $p(y_i|f)$, $p(f|\mathcal{D})$ is not a Gaussian process and we cannot compute the parameters in the posterior process analytically. To obtain an approximation to $p(f|\mathcal{D})$, we approximate each non-Gaussian factor $p(y_i|f(\mathbf{x}_i))$ by a Gaussian factor $\tilde{t}_i(f_i) = \mathcal{N}(f_i|m_i, v_i)$. Then we obtain a Gaussian process approximation $q(f)$ to $p(f|\mathcal{D})$:

$$q(f) \propto GP(f|0,K)\prod_{i=1}^{N} \mathcal{N}(f_i|m_i, v_i). \tag{10}$$

For REP, we parameterize the relaxation factor r_i as

$$r_i(f_i) \propto \mathcal{N}(f_i|m_i, b_i^{-1}), \tag{11}$$

so that r_i shares the mean as $\tilde{t}_i$ and b_i is the only free parameter in r_i. To simplify the notation in the following presentation, we define $\tilde{t}_{i,b}(f_i) \equiv \mathcal{N}(f_i|m_{i,b}, v_{i,b}) \propto r_i(f_i)\tilde{t}_i(f_i)$. Then we have the following REP training algorithm for GP classification.

1. Initialize all $m_i = 0$, $v_i = \infty$, and $b_i = 0$. Also, initialize $h_i = 0$, $\mathbf{A} = \mathbf{K}$, and $\lambda_i = \mathbf{K}_{ii}$.
2. Until all (m_i, v_i, b_i) converge: Pick a sample i.
   (a) Remove $\tilde{t}_i$ from the approximated posterior:
   $$\lambda_i^{\backslash i} = (\frac{1}{\mathbf{A}_{ii}} - \frac{1}{v_i})^{-1} \tag{12}$$
   $$h_i^{\backslash i} = h_i + \lambda_i^{\backslash i} v_i^{-1}(h_i - m_i) \tag{13}$$
   (b) Jointly minimize the penalized divergence over q and b_i by line search on b_i (See Appendix for details). With the optimized b, we compute the new message $\tilde{t}_i$:
   - Multiple $q^{\backslash i}$ with r_i:
   $$\tilde{\lambda}_i^{\backslash i} = 1/(1/\lambda_i^{\backslash i} + b_i) \tag{14}$$
   $$\tilde{h}_i^{\backslash i} = h_i^{\backslash i} - \tilde{\lambda}_i^{\backslash i} b_i (h_i^{\backslash i} - m_i) \tag{15}$$
   - Minimize the penalized divergence to obtain $\tilde{t}_{i,b}$:
   $$\alpha = \frac{1}{\sqrt{\tilde{\lambda}_i^{\backslash i}}} \frac{(1-2\epsilon)\mathcal{N}(z|0,1)}{\epsilon + (1-2\epsilon)\psi(z)} \tag{16}$$
   $$\tilde{h}_i = \tilde{h}_i^{\backslash i} + \tilde{\lambda}_i^{\backslash i}\alpha \tag{17}$$
   $$v_{i,b} = \tilde{\lambda}_i^{\backslash i}(\frac{1}{\alpha_i \tilde{h}_i} - 1) \tag{18}$$
   $$m_{i,b} = \tilde{h}_i + v_{i,b}\alpha \tag{19}$$
   where $z = \tilde{h}_i^{\backslash i}/\sqrt{\tilde{\lambda}_i^{\backslash i}}$ and $\psi(\cdot)$ is the standard normal cdf.
   - Remove r_i from $\tilde{t}_{i,b}$ to obtain $\tilde{t}_i$:
   $$v_i = 1/(1/v_{i,b} + b_i) \tag{20}$$
   $$m_i = v_i(m_{i,b}/v_{i,b} + m_i^{\text{old}} b_i) \tag{21}$$
   (c) Given the new message $\tilde{t}_i$, update $\mathbf{A}$ and h_i:
   $$\mathbf{A} = \mathbf{A} - \frac{\mathbf{a}_i \mathbf{a}_i^{\mathrm{T}}}{\delta + \mathbf{A}_{i,i}} \quad h_i = \sum_j \mathbf{A}_{ij}\frac{m_j}{v_j} \tag{22}$$
   where $\delta = 1/(1/v_i - 1/v_i^{\text{old}})$ and $\mathbf{a}_i$ is the i-th column of $\mathbf{A}$.

4.2. Relaxed belief propagation (RBP)

Just as EP reduces to belief propagation on Bayesian networks or MRFs, REP becomes a relaxed version of belief propagation on these models. In particular, let us consider the joint distribution of N discrete variables $\mathbf{x} = (x_1, \ldots, x_N)$ in a MRF:

$$p(\mathbf{x}) = \frac{1}{Z}\prod_i \psi_i(x_i) \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j)$$

where Z is the normalization constant, $\psi_i(x_i)$ and $\psi_{i,j}(x_i, x_j)$ are unitary and pairwise potential functions, and $\mathcal{E}$ represents the set of edges.

We obtain classical BP updates by adopting a factorized EP approximation (Minka, 2001):

$$q(\mathbf{x}) = \prod_i q(x_i) \propto \prod_i \psi_i(x_i) \prod_{(i,j)\in\mathcal{E}} \tilde{\psi}_{ij}(x_i)\tilde{\psi}_{ij}(x_j)$$

where $\tilde{\psi}_{ij}(x_i)$ and $\tilde{\psi}_{ij}(x_j)$ are factorized approximation to the factor $\psi_{ij}(x_i, x_j)$; they are messages from the factor $t\psi_{ij}$ to the nodes x_i and x_j.

It is well known that if the MRF contains cycles and the variables are strongly coupled, BP can suffer from low approximation quality and divergence. To address this issue, we use the following relaxation factor

$$r_{ij}(x_i, x_j) = r_{ij}(x_i)r_{ij}(x_j) = \tilde{\psi}_{ij}(x_i)^{b_{ij}}\tilde{\psi}_{ij}(x_j)^{b_{ij}}$$

where $b_{ij} \in [0, 1]$. We present the RBP updates below:

1. Initialize $q(x_i) = \psi_i(x_i)$ and $\tilde{\psi}_{ij}(x_i) = 1$.
2. Until all $\tilde{\psi}_{ij}$ converge: Pick an edge $(i, j) \in \mathcal{E}$.
   (a) Remove $\tilde{\psi}_{ij}(x_i)$ from $q(x_i)$:
   $$q^{\backslash ij}(x_i) \propto q(x_i)/\tilde{\psi}_{ij}(x_i).$$
   Similarly, remove $\tilde{\psi}_{ij}(x_j)$ from $q(x_j)$.

(b) Jointly minimize the penalized divergence over q and b_{ij}. To this end, we first compute $\hat{q}^r(x_i, x_j)$, $q^r(x_i)$, and $q^r(x_j)$:

$$\hat{q}^r(x_i, x_j) = \frac{1}{Z_{ij}} \psi_{ij}(x_i, x_j) \big(\tilde{\psi}_{ij}(x_i)\tilde{\psi}_{ij}(x_j)\big)^{b_{ij}} \cdot q^{\setminus ij}(x_i) q^{\setminus ij}(x_j) \quad (23)$$

$$q^r(x_i) = \sum_{x_j} \hat{q}^r(x_i, x_j) \quad (24)$$

$$q^r(x_j) = \sum_{x_i} \hat{q}^r(x_i, x_j) \quad (25)$$

where Z_{ij} is a normalization constant. Then we conduct line search over b_{ij} to minimize the KL divergence between $\hat{q}^r(x_i, x_j)$ and $q^r(x_i)q^r(x_j)$ with the penalty over b_{ij}. After obtaining b_{ij} and the corresponding $q^r(x_i)$, we calculate the new belief $q(x_i)$:

$$q(x_i) = q^r(x_i)/\tilde{\psi}_{ij}(x_i)^{b_{ij}}. \quad (26)$$

Similarly we compute $q(x_j)$.

(c) Update the message $\tilde{\psi}_{ij}(x_i)$ (and $\tilde{\psi}_{ij}(x_j)$):

$$\tilde{\psi}_{ij}(x_j) \propto q(x_i)/q^{\setminus ij}(x_i).$$

5. Related works

Various message passing algorithms, such as power EP (Minka, 2004), can be interpreted as iterative minimization of a general α-divergence with different α values (Minka, 2005; Zhu & Rohwer, 1995). On MRFs, power EP reduces to fractional BP (Wiegerinck & Heskes, 2003). When $\alpha = 1$, power EP and fractional BP become EP and BP; when $\alpha \neq 1$, minimizing the α-divergence does not require EP's moment matching in message passing updates. However, moment matching may contribute to great empirical performance of BP and EP, and is desirable for many tasks such as classification—where moment matching can help preserve the posterior probability in critical regions. Unlike power EP and fractional BP, REP and RBP not only stabilize message updates but also maintain moment matching whenever possible in an adaptive way.

We can damp the step size for message updates to help convergence (Minka, 2005). The damping method recursively minimizes the KL divergence just as EP with the same same energy function. For cases where EP diverges, the damping method can be very slow with a small step size needed for convergence and provide poor approximations after convergence. By contrast, using the (penalized) divergence different from the KL divergence, REP can improve both algorithmic stability and approximation accuracy over EP.

6. Experiments

6.1. Results on GP classification

For GP classification, we compared REP with EP, power EP (PEP), and damped EP (DEP) on approximation quality, convergence speed, and prediction accuracy. We chose GP classification as a testbed because EP has been shown to be an excellent choice for training GP classification models (Kuss & Rasmussen, 2005). Since there is no previous reported work that uses PEP for GP training, we derived the updates and presented them in Appendix.

Toy example. First, we considered linear classification of five data points shown in Figure 1. The red 'x' and blue 'o' data points belong two classes. The red point on the right is mislabeled. To reflect the true labeling error rate in the data, we set $\epsilon = 0.2$ in (8). To obtain linear classifiers, we used the linear kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$ for the GP training algorithms. After each algorithm converged, we recovered the posterior mean and covariance of the linear classifier $\mathbf{w}$ in the 2-dimensional input space.

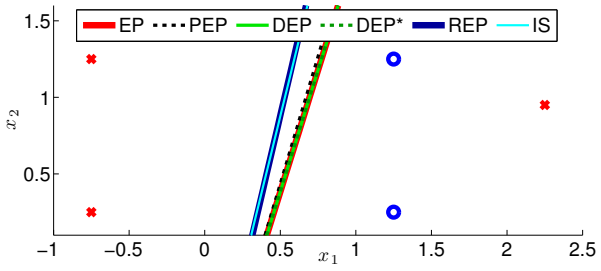
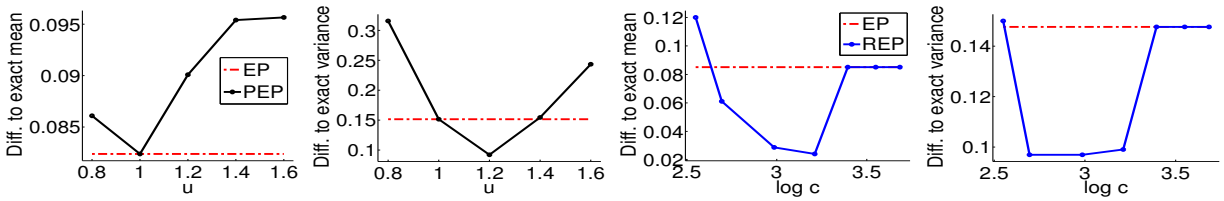


Figure 1. Decision boundaries of EP, power EP, damped EP with step-size 0.5 and with an adaptive step-size, and REP. The red data point on the right is mislabeled.

To measure the approximation quality, we used importance sampling (IS) with 10^8 samples to obtain the exact posterior distribution of the classifier $\mathbf{w}$. We treated the (approximate) posterior means as the estimated classifiers and used them to generate their decision boundaries (see Figure 1). For PEP, we set the power u to 0.8; for REP, we set $c = 20$; for DEP, we used both a fixed step-size 0.5 and an adaptive step-size that is based on a local prediction confidence level (based on (12) and (13)). We denote DEP with the adaptive step-size as DEP* in Figure 1.

The EP decision boundary differs from the exact Bayesian decision boundary significantly. DEP and DEP* give the same wrong decision boundary as EP. The PEP decision boundary is slightly closer to the exact one. The REP decision boundary perfectly overlaps with the exact one. Note that the relaxation pa-



(a) Error in mean est. (b) Error in var. est. (c) Error in mean est. (d) Error in var. est.

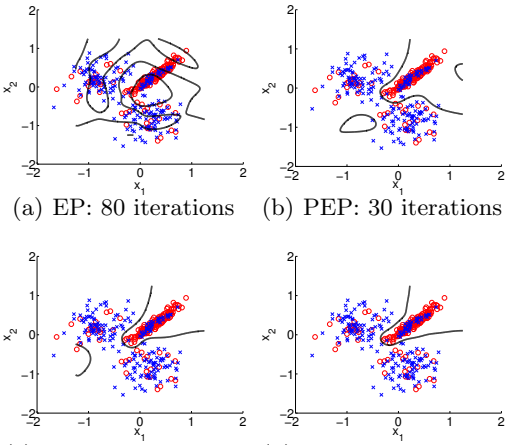
Figure 2. Estimation accuracies of EP, power EP, and REP. The estimation accuracies of damped EP are not visualized above since they are identical to those of EP. (a) and (c): EP vs power EP with different values for the power u; (c) and (d): EP vs REP with different values for the penalty weight c. While REP reduces to EP when c is big, for a wide range of c values, the REP's approximation accuracy is significantly higher than those of EP and power EP.

rameter b_i was automatically pruned to be zero for all the points except the red point on the right (the corresponding $b_i = 0.01$), demonstrating REP's adaptiveness in relaxation.

We also varied the power for PEP, the step size of DEP, and the penalty weight c in (5) for REP to examine their impact on approximate quality. We measured the mean square distance between the estimated and the exact mean vectors, as well as the mean square distances between the estimated and the exact covariance matrices. The results of PEP and REP are summarized in Figure 2. We did not show the estimation accuracies of DEP since they are identical to those of EP. Figures 2.(a)-(b) show that the estimated posterior means of PEP are always worse than what EP achieves. In contrast, when c is big for REP, the l_1 penalty forces the relaxation factor $b_i = 0$. Accordingly, REP reduces to EP and gives the same results. When c is small—for a wide range of values—REP greatly improves the posterior approximation quality.

Finally, for the classification models with various ϵ values (e.g., 0.1 and 0.25) in (8), our further experiments showed that REP consistently provided more accurate posterior estimation than EP and PEP.

Synthetic data. We then compared these algorithms on a nonlinear classification task. We sampled 200 data points for each class: for class 1 the points were sampled from a single Gaussian distribution and, for class 2, the points from a mixture of two Gaussian components. The data points from these two classes are represented by red circles and blue "x", respectively in Figure 3. We randomly flipped the labels of some data points to introduce labeling errors and varied the error rates from 10% to 20%. For each case, we let ϵ match the error rate. We used a Gaussian kernel for all these training algorithms and applied cross-validation on the training data to tune the kernel width. We set the power $u = 0.8$ for power EP, the step-size 0.5 for damped EP, and $c = 10$ for REP.

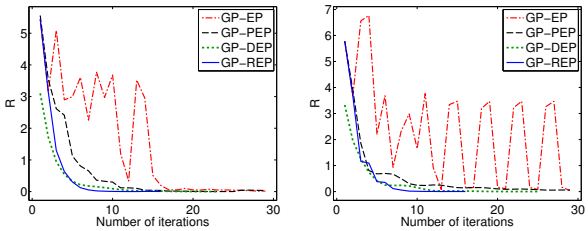


(a) EP: 80 iterations (b) PEP: 30 iterations

(c) DEP: 45 iterations (d) REP: 15 iterations

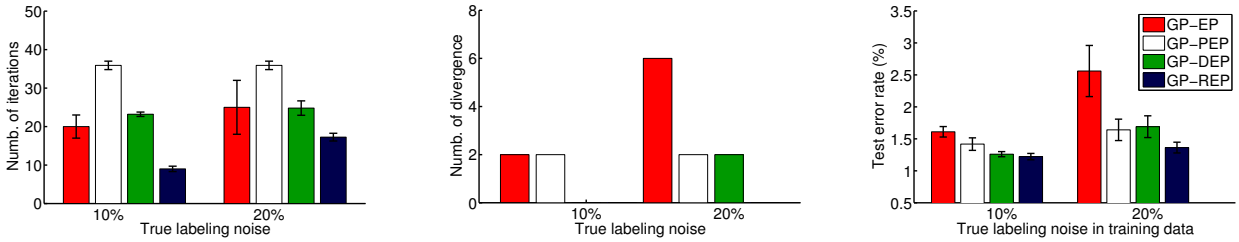
Figure 3. Decision boundaries of EP, power EP, damp EP, and REP. 20% of the data points are mislabeled.

In Figure 3, we visualized the decision boundaries of EP, PEP, DEP, and REP on one of the datasets with 20% labeling errors. Clearly, EP diverged and led to a chaotic decision boundary. PEP and DEP converged in 30 and 45 iterations and their decision boundaries are better than that of EP but still do not match the underlying generative distributions of the data (one Gaussian vs two Gaussians). Using an adaptive step-



(a) 10% labeling error (b) 20% labeling error

Figure 4. Change in GP parameters along iterations.



(a) Number of iterations (b) Number of divergence (c) Test error rate

Figure 5. Comparison on two datasets with different labeling noise levels. REP always converges. Furthermore, with fewer iterations, REP achieves higher prediction accuracies than EP, damped EP, and power EP.

size, DEP* actually diverged all the time. So we did not show its decision boundaries. By contrast, REP converged in only 15 iterations and provides a much more reasonable decision boundary than all the other algorithms.

To illustrate the convergence of PEP, DEP, and REP, we visualized in Figure 4 the change of the GP message parameter $\mathbf{m}$ along iterations: $R(\text{iter}) \equiv \|\mathbf{m}_{\text{iter}} - \mathbf{m}_{\text{iter}-1}\|_2$. Clearly, EP is less stable than the other algorithms.

To conduct a systematic comparison between EP, PEP, DEP, and REP on algorithmic robustness, convergence speed and estimation accuracy, we repeated the experiments 10 times; each time we sampled 400 training and 39,600 test points. Figure 5.(a) shows the number of iterations until convergence. To reach the convergence, we required $R < 10^{-3}$. Clearly, REP converged faster than PEP, DEP, and EP. Figure 5.(b) shows that while EP, DEP, and PEP diverged sometimes, REP *never* did.

Figure 5.(c) shows that, with 10% labeling errors in the training set, REP gave significantly higher prediction accuracies than EP and PEP. The test errors of EP, DEP, and PEP were averaged only over the converged cases out of the 10 runs; their average accuracies would degrade if we include their divergent cases here. Note that we did not introduce labeling errors in the test data and the prediction error rates can be lower than the labeling error rates in the training sets. With 10% labeling errors, the prediction accuracy of REP is slightly higher than that of DEP with no significance, but REP converges three time faster (see Figure 5.(a)). With 20% labeling errors in the training set, with much fewer number of iterations, REP significantly outperforms all the alternative algorithms in terms of prediction accuracy.

Real data. Finally we tested these algorithms on five UCI benchmark datasets: Heart, Pima, Diabetes, Haberman, and Spam. For PEP and REP, we tuned the power and penalty weight based on a small validation set. For DEP, we used a step-size 0.5; a smaller step size would make DEP really slow for convergence.

For the Heart dataset, the task is to detect heart diseases with 13 features per sample. We randomly split the dataset into 81 training and 189 test samples 100 times. For the Pima dataset, we randomly split it into 319 training and 213 test samples, again 100 times. For the Diabetes dataset, medical measurements and personal history are used to predict whether a patient is diabetic. Rätsch et al. (2001) split the UCI Diabetes dataset into two groups (468 training and 300 test samples) for 100 times. We used the same partitions in our experiments. For the Haberman's survival dataset, the task is to estimate whether a patient survives more than five years (including 5 years) after a surgery for breast cancer. The whole dataset contains information from 306 patient samples and 3 attributes per sample. We randomly split the dataset into 183 training and 123 test samples 100 times. Note that we did *not* add any labeling errors to these four datasets. Figure 6 summarizes the averaged results. REP outperforms the competing algorithms significantly.

For the Spam dataset, the task is to detect spam emails. We partitioned the dataset to have 276 training and 4325 test samples, and flipped the labels of multiple data points randomly from both the training and test sets. The experiment was repeated for 100 times. Figure 7 demonstrated that, with various additional labeling errors, REP achieves significantly higher prediction accuracies than EP, DEP, and PEP.

6.2. Results on MRF inference

Now we test RBP, BP, damped BP, and fractional BP for inference on a binary MRF model $x_i \in \{-1, 1\}$ with weak or strong suppressive interactions:

$$p(\mathbf{x}) = \frac{1}{Z} \exp\Big(\sum_i J_i x_i - \sum_{ij} J_{ij} x_i x_j\Big).$$

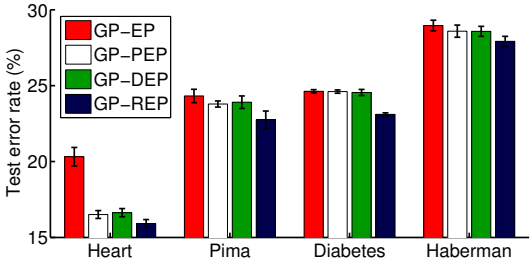


Figure 6. Test error rates of EP, DEP, PEP and REP on four UCI benchmark datasets without additional labeling noise.

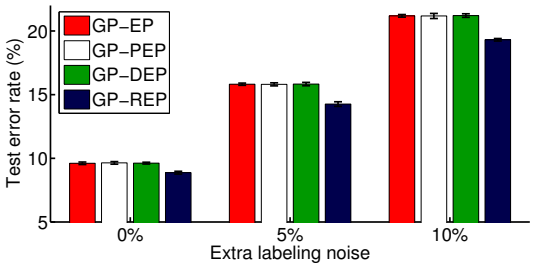


Figure 7. Test error rates on the Spam dataset with additional labeling noise.

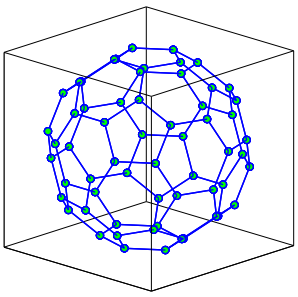
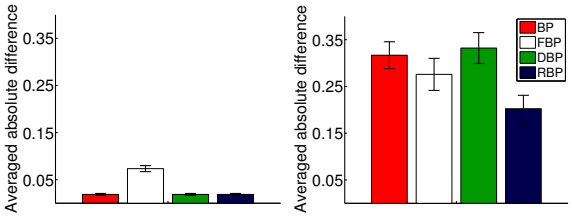


Figure 8. 3-D MRF.

The MRF has 60 nodes and its structure is a Buckminster Fuller geodesic dome as shown in Figure 8. In all cases, we chose the single node parameter randomly as $J_i \sim \mathcal{N}(0,1)$. For the weak interaction condition, we sampled each edge weight independently $J_{ij} \sim |(\mathcal{N}(1,1)|$. For the strong interaction condition, we chose edge weights $J_{ij} \sim |(\mathcal{N}(5,5)|$ independently for each edge. We repeated experiments 50 times for each condition. To obtain the exact single-node marginal distributions, $p(x_i)$, we used the junction tree algorithm. We set the power 0.8 for fractional BP, the step-size 0.8 for damped BP, and $c = 0.1$ for REP. To measure the estimation accuracy, we calculated the averaged absolute difference between the exact and approximate single-node marginal distributions. We did not report the number of divergent cases for each algorithm, but the divergence is reflected in the averaged absolute difference.

The results are summarized in Figure 9.(a)-(b). On the weak interaction case, BP, damped BP, and RBP all



(a) Weak interactions (b) Strong interactions

Figure 9. Approximate inference on discrete MRF.

achieved comparable results, while factional BP performed worse than all the others. This is not surprising because BP works well on MRFs with weak interactions. RBP shrinked its relaxation and almost all the estimated b_{ij} were exactly zeros. Thus RBP behaved like BP. Fractional BP did worse because it did not use the KL minimization to capture important moment statistics. For the strong interaction case, RBP relaxed the moment matching constraint (more b_{ij} are estimated to be nonzero) and achieved significantly higher accuracy than all the other methods.

7. Conclusions

In the paper we have presented the new REP inference method based on the l_1-penalized divergence. Unlike the α-divergence minimization in power EP, the l_1-penalized divergence minimization adaptively relaxes the moment matching constraint in EP and BP. Experimental results demonstrate that the new inference algorithm avoids divergence and improves approximation quality performance over the previous message passing methods.

Acknowledgments

This work was supported by NSF IIS-0916443, NSF ECCS-0941533, NSF CAREER Award IIS-1054903, and the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370.

References

Heskes, T., Opper, M., Wiegerinck, W., Winther, O., and Zoeter, O. Approximate inference techniques with expectation constraints. *Journal of Statistical Mechanics: Theory and Experiment*, 11:11015, 2005.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2): 498–519, 1998.

Kuss, M. and Rasmussen, C. E. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6(10):1679–1704, 2005.

Minka, T. P. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

Minka, T. P. Power EP. Technical Report MSR-TR-2004-149, Microsoft Research, Cambridge, January 2004.

Minka, T. P. Divergence measures and message passing. Technical Report MSR-TR-2005-173, Microsoft Research, Cambridge, 2005.

Pearl, J. Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the American Association of Artificial Intelligence National Conference on AI*, pp. 133–136, Pittsburgh, PA, 1982.

Rätsch, G., Onoda, T., and Müller, K.-R. Soft margins for adaboost. *Mach. Learn.*, 42:287–320, March 2001.

Wiegerinck, W. and Heskes, T. Fractional belief propagation. *Advances in Neural Information Processing Systems*, 12:438–445, 2003.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. Understanding belief propagation and its generalizations. In Lakemeyer, Gerhard and Nebel, Bernhard (eds.), *Exploring artificial intelligence in the new millennium*, pp. 239–269. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

Yuille, A. L. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14:1691 – 1722, 2002.

Zhu, H. and Rohwer, R. Bayesian invariant measurements of generalisation for continuous distributions. Technical Report NCRG/4352, Aston University, Aston Triangle, 1995.