Conditional Expectation Propagation

Zheng Wang, Shandian Zhe
School of Computing, University of Utah
Salt Lake City, UT 84112
wzhut@cs.utah.edu, zhe@cs.utah.edu

Abstract

Expectation propagation (EP) is a powerful approximate inference algorithm. However, a critical barrier in applying EP is that the moment matching in message updates can be intractable. Handcrafting approximations is usually tricky, and lacks generalizability. Importance sampling is very expensive. While Laplace propagation provides a good solution, it has to run numerical optimizations to find Laplace approximations in every update, which is still quite inefficient. To overcome these practical barriers, we propose conditional expectation propagation (CEP) that performs conditional moment matching given the variables outside each message, and then takes expectation w.r.t the approximate posterior of these variables. The conditional moments are often analytical and much easier to derive. In the most general case, we can use (fully) factorized messages to represent the conditional moments by quadrature formulas. We then compute the expectation of the conditional moments via Taylor approximations when necessary. In this way, our algorithm can always conduct efficient, analytical fixed point iterations. Experiments on several popular models for which standard EP is available or unavailable demonstrate the advantages of CEP in both inference quality and computational efficiency.

1 INTRODUCTION

Expectation propagation (EP) (Minka, 2001) is a popular posterior inference algorithm. It approximates the factors of the joint probability with exponential-family terms, also called messages (in graphical models), and iteratively updates each message via moment matching. EP often produces fast and accurate posterior estimations, and have been applied in many Bayesian learning tasks (Herbrich et al., 2007; Graepel et al., 2010; Hernández-Lobato and Adams, 2015). EP is the cornerstone of the influential machine learning library, Infer.NET (Minka et al., 2014).

Nonetheless, a critical barrier of using EP is that the moment matching can be intractable during the message updates. When a factor (*e.g.,* the likelihood of a data point) is complex and includes many latent variables, the normalizer of the tilted distribution (that is proportional to the factor multiplying with the messages from the other factors) is likely to be intractable and so are the moments. Handcrafting an approximation to the moments is usually tricky, and limited to particular types of factors. One can consider importance sampling, which, however, requires a large number of samples to obtain reliable estimations and hence is very expensive.

To mitigate this issue, Eskin et al. (2004) proposed Laplace propagation (LP) that uses Gaussian messages and calculates the Laplace approximation to the tilted distribution. This is fulfilled by finding the mode and computing the inverse Hessian at the mode. The approximation, which is a Gaussian distribution, is then used for moment matching. Despite its effectiveness, LP has to repeatedly conduct numerical optimizations to find the modes in message updates, and therefore is still quite inefficient.

To overcome these practical barriers, we propose conditional expectation propagation (CEP) algorithm, which not only bypasses the (possibly) tricky moment matching and so the nontrivial approximation designs, but also enjoys efficient and analytical fixed point iterations. Specifically, we observe that although the moments of the tilted distributions can be intractable, the conditional moments given a set of variables fixed are often tractable and analytical. In the most general case, we can consider the conditional moments for a single variable (or two) given all the others. These moments (if unavailable) can be explicitly represented by quadrature formulas. Therefore, we first introduce factorized approximations (i.e., messages) and match the conditional moments for the variable(s) in each message. Next, we update the message by computing the expectation of the conditional moments w.r.t the (current) approximate posterior of the remaining variables. We show the connection of this method to the standard EP. Finally, if necessary, to enable tractable expectation computation for the conditional moments, we use their first or second order Taylor approximations at the moments of the remaining variables. In this way, CEP always performs efficient, analytical updates and is general to all kinds of factors, without nontrivial approximation designs, numerical optimizations or importance sampling.

For evaluation, we first examined CEP in Bayesian probit and logistic regression where the standard moment matching is analytical or has accurate approximations. For logistic regression, we represented the conditional moments with Gauss-Hermite quadrature. On both simulation and real-world datasets, CEP obtains inference quality and running efficiency close to EP, and is much faster than LP and importance sampling based approaches. We then applied CEP in Bayesian tensor decomposition where the moment matching is intractable. CEP largely improves upon LP and a classical tensor decomposition algorithm, and is close to or significantly better than the variational message passing (VMP) in prediction accuracy. Meanwhile, CEP achieves over 40x speedup against LP. Finally, we adapted CEP to the assumed density filtering (ADF) framework to perform streaming Bayesian decomposition. Our method outperforms the state-of-the-art approach (Du et al., 2018) by a large margin.

2 EXPECTATION PROPAGATION

We first review the expectation propagation (EP) algorithm Minka (2001). Given the observations $\mathcal{D}$, the posterior distribution of a Bayesian model has a general form,

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{1}{Z}\prod\nolimits_i f_i(\boldsymbol{\theta}_i) \tag{1}$$

where $\boldsymbol{\theta}$ are the latent random variables, Z the normalization constant, and $\{f_i(\boldsymbol{\theta}_i)\}_i$ the factors that link to the prior or data likelihoods. For example, f_0 may correspond to the prior, and f_n the likelihood of the n-th data point ($n \geq 1$). Each $\boldsymbol{\theta}_i$ is a subset of or identical to $\boldsymbol{\theta}$. Bayesian inference aims to compute the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$. However, the exact computation is usually infeasible, due to the high dimensional and intractable integral in calculating the normalizer Z.

To address this issue, EP approximates each factor f_i by an exponential-family term,

$$\tilde{f}_i(\boldsymbol{\theta}_i) \propto \exp\left(\boldsymbol{\lambda}_i^\top \phi(\boldsymbol{\theta}_i)\right), \tag{2}$$

where $\boldsymbol{\lambda}_i$ and $\phi(\boldsymbol{\theta}_i)$ are the natural parameters and sufficient statistics, respectively. Then the approximate posterior distribution is given by

$$q(\boldsymbol{\theta}) \propto \prod_i \tilde{f}_i(\boldsymbol{\theta}_i), \tag{3}$$

and trivial to compute through the summation of the natural parameters. In the factor graph representation (Kschischang et al., 2001), the approximation factor $\tilde{f}_i(\boldsymbol{\theta}_i)$ is also defined as the message from factor f_i to the variables $\boldsymbol{\theta}_i$. Commonly used messages are Gaussian factors.

EP repeatedly refines each approximation factor $\tilde{f}_i$ through three steps, message deletion, projection and update. In the message deletion step, we compute a calibrating distribution $q^{\backslash i}(\boldsymbol{\theta})$ by removing $\tilde{f}_i$ from the current posterior $q(\boldsymbol{\theta})$, $q^{\backslash i}(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta})/\tilde{f}_i(\boldsymbol{\theta}_i)$. This is equivalent to multiplying the messages from all the other factors, and hence a surrogate for these context factors. In the projection step, we first construct a tilted distribution $\hat{p}_i(\boldsymbol{\theta})$ from multiplying the true factor f_i with the calibrating distribution,

$$\hat{p}_i(\boldsymbol{\theta}) = \frac{1}{Z_i} f_i(\boldsymbol{\theta}_i) q^{\backslash i}(\boldsymbol{\theta}), \tag{4}$$

where Z_i is the normalizer. We then project the tilted distribution to the exponential family by minimizing the KL divergence between $\hat{p}_i(\boldsymbol{\theta})$ and a new approximate posterior $q^*(\boldsymbol{\theta})$. It is well known that the minimum can be obtained by moment matching. That is, we compute the expectation of $\phi(\boldsymbol{\theta})$ w.r.t $\hat{p}_i$ and set it to the moment of $q^*(\boldsymbol{\theta})$,

$$\mathbb{E}_{q^*}\big(\phi(\boldsymbol{\theta})\big) = \mathbb{E}_{\hat{p}_i}\big(\phi(\boldsymbol{\theta})\big) = \nabla_{\boldsymbol{\lambda}^{\backslash i}} \log(Z_i) \tag{5}$$

where $\boldsymbol{\lambda}^{\backslash i}$ are the natural parameters of the calibrating distribution $q^{\backslash i}$. Finally, we update the message $\tilde{f}_i$ via $\tilde{f}_i(\boldsymbol{\theta}_i) \propto q^*(\boldsymbol{\theta})/q^{\backslash i}(\boldsymbol{\theta})$.

The refinement of each message can be sequential or parallel. The key step in the refinement is the moment matching (5). Through repeated moment matching, the approximate posterior is improved by assimilating the critical statistics in the true posterior. Due to the fixed point iteration nature, EP often converges fast (although convergence is not always guaranteed) and presents prominent computational advantages over alternative approaches, such as sampling.

3 CONDITIONAL EXPECTATION PROPAGATION

Despite the impressive success of EP, applying EP can be troublesome when the moment matching is intractable. Given a complex factor $f_i(\boldsymbol{\theta}_i)$, it is very likely that the log normalizer of the tilted distribution, $\log(Z_i)$, is intractable and hence the moment calculation (see (5)). In this case, we might need to handcraft an approximation of the moments. However, this is often tricky and the approximation is hard to generalize to other types of complex factors. A simple and general approach is to use importance sampling, but it requires a large number of samples to obtain a reliable estimation, hence is very costly and will deprive the computational advantage of EP. Although this problem can be alleviated by training a machine learning model to predict the moments (Heess et al., 2013; Jitkrittum et al., 2015), the training data collection and replenishment still require a lot of importance sampling. In addition, learning predictors for distinct kinds of factors or even the same kind with new observations may also require us to supplement training samples and/or retrain from scratch (Jitkrittum et al., 2015). While Laplace propagation (Eskin et al., 2004) can completely sidestep importance sampling and provide a general solution by using Gaussian messages and calculating Laplace approximation to the titled distribution, it has to run numerical optimizations, say, L-BFGS, to find the mode and to construct the approximation in every message update, which is still quite inefficient.

To overcome these barriers, avoiding nontrivial approximation designs, costly importance sampling and numerical optimizations, we propose the conditional expectation propagation (CEP) algorithm, presented as follows.

3.1 CONDITIONAL MOMENT MATCHING

We observe that while the moments of the tilted distribution (4) can be intractable, the conditional moments of a subset of variables given the others can be analytical and easy to derive. For example, when a tilted distribution is proportional to a production factor multiplying with a Gaussian distribution, i.e., $\hat{p}_i(\boldsymbol{\theta}) \propto \mathcal{N}(x_n|\boldsymbol{\theta}_1^\top\boldsymbol{\theta}_2, \beta)\mathcal{N}(\boldsymbol{\theta}|\mu, \boldsymbol{\Sigma})$ where $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^\top, \boldsymbol{\theta}_2^\top]^\top$, computing the moments of $\boldsymbol{\theta}$ is tricky, but $\boldsymbol{\theta}_1$ given $\boldsymbol{\theta}_2$ fixed and $\boldsymbol{\theta}_2$ given $\boldsymbol{\theta}_1$ fixed is trivial — both $\hat{p}(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)$ and $\hat{p}(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1)$ are Gaussian distributions. This resembles Gibbs sampling — while the posterior is intractable, the conditional distribution of each variable given all the others (and observations) is tractable and we can repeatedly sample from the conditional distributions.

Therefore, we introduce factorized messages and derive analytical conditional moments in the first step. W.l.o.g, we partition $\boldsymbol{\theta}$ into $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ and approximate each factor f_i with $\prod_m \tilde{f}_{im}(\boldsymbol{\theta}_m)$ where each $\tilde{f}_{im}(\boldsymbol{\theta}_m) \propto \exp\left(\boldsymbol{\lambda}_{im}^\top \phi(\boldsymbol{\theta}_m)\right)$. Hence, the approximate posterior $q(\boldsymbol{\theta})$ and the calibrating distribution $q^{\backslash i}(\boldsymbol{\theta})$ are both factorized over $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$. Note that factorized messages are widely adopted by EP and other message passing algorithms, especially in large-scale applications (Graepel et al., 2010; Zhe et al., 2016a). To update each $\tilde{f}_{im}$ in the standard EP, we need to compute the moments of $\boldsymbol{\theta}_m$ with the tilted distribution $\hat{p}_i(\boldsymbol{\theta}) \propto q^{\backslash i}(\boldsymbol{\theta}_m)q^{\backslash i}(\boldsymbol{\theta}_{\backslash m})f_i(\boldsymbol{\theta}_m, \boldsymbol{\theta}_{\backslash m})$ where $\boldsymbol{\theta}_{\backslash m}$ are all the latent variables excluding $\boldsymbol{\theta}_m$. It can be seen that

$$\mathbb{E}_{\hat{p}_i(\boldsymbol{\theta})}\big(\phi(\boldsymbol{\theta}_m)\big) = \mathbb{E}_{\hat{p}_i(\boldsymbol{\theta}_{\backslash m})}\big[\mathbb{E}_{\hat{p}_i(\boldsymbol{\theta}_m|\boldsymbol{\theta}_{\backslash m})}\big(\phi(\boldsymbol{\theta}_m)\big)\big] \quad (6)$$

where $\hat{p}_i(\boldsymbol{\theta}_{\backslash m})$ is $\hat{p}_i(\boldsymbol{\theta})$ marginalizing out $\boldsymbol{\theta}_m$, and

$$\hat{p}_i(\boldsymbol{\theta}_m|\boldsymbol{\theta}_{\backslash m}) \propto q^{\backslash i}(\boldsymbol{\theta}_m)f_i(\boldsymbol{\theta}_m, \boldsymbol{\theta}_{\backslash m}). \quad (7)$$

We first obtain an analytical form of the conditional moment $\mathbb{E}_{\hat{p}_i(\boldsymbol{\theta}_m|\boldsymbol{\theta}_{\backslash m})}\big(\phi(\boldsymbol{\theta}_m)\big)$. This is straightforward for many factors. But uniformly, this can always be achieved by using fully factorized messages, and representing the conditional moment (which is now for a single variable) with a quadrature formula, $\mathbb{E}_{\hat{p}_i(\theta_m|\boldsymbol{\theta}_{\backslash m})}\big(\phi(\theta_m)\big) \approx \sum_j \alpha_j g(\gamma_j, \boldsymbol{\theta}_{\backslash m})$ where $\{\gamma_j, \alpha_j\}$ are quadrature nodes and weights. In addition, we can also use bi-variable or triple-variable messages with two-dimensional or three-dimensional quadrature formulas that are more complex. However, higher-dimensional quadratures are not recommended due to the degradation of accuracy and explosion of the computational cost.

3.2 EXPECTED CONDITIONAL MOMENT

To update the message $\tilde{f}_{im}$, EP requires us to further compute the expectation of the conditional moment w.r.t the marginal tilted distribution $\hat{p}_i(\boldsymbol{\theta}_{\backslash m})$ (see (6)). However, $\hat{p}_i(\boldsymbol{\theta}_{\backslash m})$ can be intractable for tricky f_i as well. To overcome this problem, we observe that (due to the factorized posterior form) EP also maintains the moment matching between $\hat{p}_i(\boldsymbol{\theta}_{\backslash m})$ and $q(\boldsymbol{\theta}_{\backslash m})$ — the (marginal) approximate posterior for $\boldsymbol{\theta}_{\backslash m}$. Therefore, we can assume they are close in high density regions and use $q(\boldsymbol{\theta}_{\backslash m})$ as a surrogate for $\hat{p}_i(\boldsymbol{\theta}_{\backslash m})$. The conditional moment is a function of $\boldsymbol{\theta}_{\backslash m}$, and w.l.o.g can be further represented as a function of their sufficient statistics, $\mathbb{E}_{\hat{p}_i(\boldsymbol{\theta}_m|\boldsymbol{\theta}_{\backslash m})}(\phi(\boldsymbol{\theta}_m)) = h(\Phi_m)$ where $\Phi_m = \{\phi(\boldsymbol{\theta}_1), \ldots, \phi(\boldsymbol{\theta}_{m-1}), \phi(\boldsymbol{\theta}_{m+1}), \ldots, \phi(\boldsymbol{\theta}_M)\}$. Therefore, we next compute the expected conditional moment $\mathbb{E}_{q(\boldsymbol{\theta}_{\backslash m})}\big(h(\Phi_m)\big)$.

Even with $q(\boldsymbol{\theta}_{\backslash m})$, the expectation might still be intractable. However, with the nice and easy form of $q(\boldsymbol{\theta}_{\backslash m})$, we can use the Taylor expansion at the moments of $\boldsymbol{\theta}_{\backslash m}$, i.e., $\mathbb{E}_q(\Phi_m)$, to derive the first-order or second-order approximation to $h(\Phi_m)$,

$$\hat{h}_1(\Phi_m) = h(\mathbb{E}_q(\Phi_m)) + \nabla h(\mathbb{E}_q(\Phi_m))^\top \mathbf{u} \quad (8)$$

$$\hat{h}_2(\Phi_m) = \hat{h}_1(\Phi_m) + \frac{1}{2}\mathrm{tr}\big(\mathbf{S} \cdot \nabla\nabla h(\mathbb{E}_q(\Phi_m))\big) \quad (9)$$

where $\mathbf{u} = \Phi_m - \mathbb{E}_q(\Phi_m)$ and $\mathbf{S} = \mathbf{u}\mathbf{u}^\top$. Taking expectation over the Taylor approximations, we have

$$\mathbb{E}_{q(\boldsymbol{\theta}_{\backslash m})}\big(h(\Phi_m)\big) \approx h(\mathbb{E}_q(\Phi_m)) \quad \text{or} \tag{10}$$

$$h(\mathbb{E}_q(\Phi_m)) + \frac{1}{2}\mathrm{tr}\big(\mathrm{cov}(\Phi_m) \cdot \nabla\nabla h(\mathbb{E}_q(\Phi_m))\big), \tag{11}$$

Both (10) and (11) are analytical and straightforward to compute. Using (11) can be more accurate but requires extra calculations. To reduce the cost, we can use the diagonal Hessian and covariance matrix.

The other steps are the same as standard EP. As we can see, through deriving analytical conditional moments and their expectations, our method bypasses the tricky moment matching, and still conducts efficient and analytical message refinements. We do not need to run expensive importance sampling, numerical optimizations or design nontrivial moment approximations. The proposed CEP is summarized in Algorithm 1.

3.3 CONNECTION TO EP

A key difference between CEP and EP is that CEP uses the approximate posterior $q(\boldsymbol{\theta}_{\backslash m})$ to replace the marginal tilted distribution $\hat{p}_i(\boldsymbol{\theta}_{\backslash m})$ in computing the expected conditional moments as in (6). This can lead to a different fixed point. But under certain conditions, EP's fixed point can still coincide with CEP's.

Lemma 3.1. *When the conditional moment h is part of the sufficient statistics of $\boldsymbol{\theta}_{\backslash m}$, i.e., each element of h belongs to Φ_m, the fixed points of EP are also that of CEP without Taylor approximations.*

The proof is given in the supplementary material. While in most cases, CEP may reach a fixed point different from EP, we found that the quality of the estimated posteriors is close to that of EP in our experiments. In the difficult cases for standard EP, CEP can easily and rapidly converge to good posterior estimations.

3.4 ALGORITHM COMPLEXITY

Given N factors and M dimensional sufficient statistics for $q(\boldsymbol{\theta})$, the time and space complexity of CEP using the first-order Taylor approximation and the second-order with diagonal Hessian or covariance matrix are both $\mathcal{O}(NM)$. If we use the full Hessian and full covariance matrix, the time and space complexity are $\mathcal{O}(NM^2)$.

4 CEP FOR BAYESIAN TENSOR DECOMPOSITION AND LOGISTIC REGRESSION

As a case study, we apply CEP in two popular models, Bayesian tensor decomposition and logistic regression.

Algorithm 1 Conditional Expectation Propagation (CEP)

```
Initialize $q(\boldsymbol{\theta}) = 1$ and all the messages $\tilde{f}_{im}(\boldsymbol{\theta}_m) = 1$.
repeat
    Pick a factor $f_i$ and message $\tilde{f}_{im}$,
      • Message deletion: Calculate the calibrating distribution, $q^{\backslash i}(\boldsymbol{\theta}_m) \propto q(\boldsymbol{\theta}_m)/\tilde{f}_{im}(\boldsymbol{\theta}_m)$.
      • Projection: Derive the conditional moment of $\boldsymbol{\theta}_m$ w.r.t $\hat{p}_i(\boldsymbol{\theta}_m|\boldsymbol{\theta}_{\backslash m})$ in (7), and then compute its expectation w.r.t $q(\boldsymbol{\theta}_{\backslash m})$. If the expectation is intractable, use (10) or (11) for approximations. The expected conditional moments are used to construct a new posterior $q^*(\boldsymbol{\theta}_m)$.
      • Update: Update the message based on the new posterior: $\tilde{f}_{im} \propto q^*(\boldsymbol{\theta}_m)/q^{\backslash i}(\boldsymbol{\theta}_m)$.
until all the $\tilde{f}_{im}(\boldsymbol{\theta}_m)$ converge
```

4.1 BAYESIAN TENSOR DECOMPOSITION

Denote a K-mode tensor by $\mathcal{Y} \in \mathbb{R}^{d_1 \times \ldots \times d_K}$, where d_k is the dimension of the k-th mode, corresponding to d_k objects (*e.g.,* users or items). The entry value at location $\mathbf{i} = (i_1, \ldots, i_K)$ is denoted by $y_\mathbf{i}$. We introduce an R dimensional embedding vector $\mathbf{u}_j^k$ to represent each object j in mode k. A $d_k \times R$ embedding matrix is formed by stacking all the embedding vectors in mode k, $\mathbf{U}^k = [\mathbf{u}_1^k, \ldots, \mathbf{u}_{d_k}^k]^\top$. Tensor decomposition aims to use these embeddings $\mathcal{U} = \{\mathbf{U}^k\}_k$ to reconstruct the observed entries.

We use a Bayesian model based on the classical CANDECOMP/PARAFAC (CP) decomposition (Harshman, 1970) to sample the observed tensor entries. We consider continuous and binary observations. Each embedding vector $\mathbf{u}_j^k$ is first sampled from a Gaussian prior distribution. Given the embeddings, a continuous entry value $y_\mathbf{i}$ is sampled from

$$p(y_\mathbf{i}|\mathcal{U}, \tau) = \mathcal{N}(y_\mathbf{i}|\mathbf{1}^\top(\mathbf{u}_{i_1}^1 \circ \ldots \circ \mathbf{u}_{i_K}^K), \tau^{-1}) \tag{12}$$

where $\mathbf{1}$ is the vector full of ones, $\circ$ the Hadamard (or element-wise) product, and τ the inverse variance. We assign τ a Gamma prior, $p(\tau) = \mathrm{Gam}(\tau|a_0, b_0)$. If $y_\mathbf{i}$ is binary, it is sampled from

$$p(y_\mathbf{i}|\mathcal{U}) = \psi\big((2y_\mathbf{i} - 1)\mathbf{1}^\top(\mathbf{u}_{i_1}^1 \circ \ldots \circ \mathbf{u}_{i_K}^K)\big) \tag{13}$$

where $\psi(\cdot)$ is the cumulative density function (CDF) of the standard Gaussian distribution.

To estimate the posterior of the embeddings $\mathcal{U}$ with EP, the major hurdle is to approximate the likelihood of each observation, (12) or (13). Due to the Hadamard product, the log normalizer of the titled distribution is intractable and the moments are tricky to compute. Hence, we turn to CEP. Note that the prior factors belong to the exponential family and do not need approximations. For the continuous entry, we approximate (12) by $\tilde{f}_{\mathbf{i}}(\tau)\prod_k \tilde{f}_{\mathbf{i}}^k(\mathbf{u}_{i_k}^k)$ where $\tilde{f}_{\mathbf{i}}(\tau)$ is a Gamma term and each $\tilde{f}_{\mathbf{i}}^k(\mathbf{u}_{i_k}^k)$ a Gaussian term. The approximate posterior $q(\mathcal{U}, \tau)$ and the calibrating distribution $q^{\setminus \mathbf{i}}(\mathcal{U}, \tau)$ are hence factorized over τ and all $\{\mathbf{u}_j^k\}_{j,k}$. To update each message $\tilde{f}_{\mathbf{i}}^k(\mathbf{u}_{i_k}^k)$, we first derive the conditional moment of $\mathbf{u}_{i_k}^k$ given the other embeddings and τ fixed. Note that the conditional tilted distribution,

$$\hat{p}_{\mathbf{i}}(\mathbf{u}_{i_k}^k|\mathbf{u}_{\mathbf{i}}^{\setminus k}, \tau) \propto q^{\setminus \mathbf{i}}(\mathbf{u}_{i_k}^k)\mathcal{N}(y_{\mathbf{i}}|\mathbf{z}_{\mathbf{i}}^{\setminus k^\top}\mathbf{u}_{i_k}^k, \tau^{-1})$$

is Gaussian. Here $\mathbf{u}_{\mathbf{i}}^{\setminus k}$ are all the embedding vectors associated with entry $\mathbf{i}$ but excluding $\mathbf{u}_{i_k}^k$, $\mathbf{z}_{\mathbf{i}}^{\setminus k}$ the Hadmard product of the vectors in $\mathbf{u}_{\mathbf{i}}^{\setminus k}$, and $q^{\setminus \mathbf{i}}(\mathbf{u}_{i_k}^k) = \mathcal{N}(\mathbf{u}_{i_k}^k|\mathbf{m}_{i_k}^k, \mathbf{S}_{i_k}^k)$, the calibrating distribution of $\mathbf{u}_{i_k}^k$. We can easily obtain the conditional mean and covariance,

$$\mathrm{cov}(\mathbf{u}_{i_k}^k|\mathbf{u}_{\mathbf{i}}^{\setminus k}, \tau) = \big[\mathbf{S}_{i_k}^{k\ -1} + \tau(\mathbf{z}_{\mathbf{i}}^{\setminus k}\mathbf{z}_{\mathbf{i}}^{\setminus k^\top})\big]^{-1}, \tag{14}$$

$$\mathbb{E}(\mathbf{u}_{i_k}^k|\mathbf{u}_{\mathbf{i}}^{\setminus k}, \tau) = \mathrm{cov}(\mathbf{u}_{i_k}^k|\mathbf{u}_{\mathbf{i}}^{\setminus k}, \tau)\big[\mathbf{S}_{i_k}^{k\ -1}\mathbf{m}_{i_k}^k + \tau y_{\mathbf{i}}\mathbf{z}_{\mathbf{i}}^{\setminus k}\big]. \tag{15}$$

To further obtain their expectations w.r.t $q(\mathbf{u}_{\mathbf{i}}^{\setminus k}, \tau)$ so as to update $\tilde{f}_{\mathbf{i}}^k(\mathbf{u}_{\setminus i_k}^k)$, we can follow (10) to take the expectation over their first-order Taylor expansion at the moments of $\mathbf{u}_{\mathbf{i}}^{\setminus k}$ and τ. This is trivial — we replace the terms $\mathbf{z}_{\mathbf{i}}^{\setminus k}$, $\mathbf{z}_{\mathbf{i}}^{\setminus k}\mathbf{z}_{\mathbf{i}}^{\setminus k^\top}$ and τ by their expectations in (14)(15). Due to the factorized posterior, we have

$$\begin{aligned}\mathbb{E}_q(\mathbf{z}_{\mathbf{i}}^{\setminus k}) &= \mathbb{E}_q(\mathbf{u}_{i_1}^1) \circ \ldots \circ \mathbb{E}_q(\mathbf{u}_{i_{k-1}}^{k-1}) \\ &\circ \mathbb{E}_q(\mathbf{u}_{i_{k+1}}^{k+1}) \circ \ldots \circ \mathbb{E}_q(\mathbf{u}_{i_K}^K), \qquad (16) \\ \mathbb{E}_q(\mathbf{z}_{\mathbf{i}}^{\setminus k}\mathbf{z}_{\mathbf{i}}^{\setminus k^\top}) &= \mathbb{E}_q(\mathbf{u}_{i_1}^1\mathbf{u}_{i_1}^{1\ \top}) \circ \ldots \circ \mathbb{E}_q(\mathbf{u}_{i_{k-1}}^{k-1}\mathbf{u}_{i_{k-1}}^{k-1\ \top}) \\ &\circ \mathbb{E}_q(\mathbf{u}_{i_{k+1}}^{k+1}\mathbf{u}_{i_{k+1}}^{k+1\ \top}) \circ \ldots \circ \mathbb{E}_q(\mathbf{u}_{i_K}^K\mathbf{u}_{i_K}^{K\ \top}). \qquad (17)\end{aligned}$$

We can use the second-order Taylor approximation as well (see (11)), but our investigation shows that it does not outperform the first-order approach and is slower.

We now look into how to update message $\tilde{f}_{\mathbf{i}}(\tau)$. Note that the calibrating distribution is a Gamma distribution, $q^{\setminus \mathbf{i}}(\tau) = \mathrm{Gam}(\tau|a^{\setminus \mathbf{i}}, b^{\setminus \mathbf{i}})$. We first find that the conditional tilted distribution is also a Gamma distribution,

$$\begin{aligned}&\hat{p}_{\mathbf{i}}(\tau|\mathbf{u}_{i_1}^1, \ldots, \mathbf{u}_{i_K}^K) = \mathrm{Gam}(\tau|\hat{a}, \hat{b}) \\ &\propto q^{\setminus \mathbf{i}}(\tau)\mathcal{N}(y_{\mathbf{i}}|\mathbf{1}^\top(\mathbf{u}_{i_1}^1 \circ \ldots \circ \mathbf{u}_{i_K}^K), \tau^{-1})\end{aligned}$$

where $\hat{a} = a^{\setminus \mathbf{i}} + \frac{1}{2}$ and $\hat{b} = b^{\setminus \mathbf{i}} + \frac{1}{2}(y_{\mathbf{i}} - \mathbf{1}^\top(\mathbf{u}_{i_1}^1 \circ \ldots \circ \mathbf{u}_{i_K}^K))^2$. Next, we take the expectation over $\hat{b}$ w.r.t to the posterior $q(\mathbf{u}_{i_1}^1, \ldots, \mathbf{u}_{i_K}^K)$ ($\hat{a}$ is constant and so $\mathbb{E}_q(\hat{a}) = \hat{a}$). This is analytical and straightforward:

$$\begin{aligned}\mathbb{E}_q(\hat{b}) &= b^{\setminus \mathbf{i}} + \frac{1}{2}y_{\mathbf{i}}^2 - y_{\mathbf{i}}\mathbf{1}^\top\big[\mathbb{E}_q(\mathbf{u}_{i_1}^1) \circ \ldots \circ \mathbb{E}_q(\mathbf{u}_{i_K}^K)\big] \\ &+ \frac{1}{2}\mathrm{tr}\big[\mathbb{E}_q(\mathbf{u}_{i_1}^1\mathbf{u}_{i_1}^{1\ \top}) \circ \ldots \circ \mathbb{E}_q(\mathbf{u}_{i_K}^K\mathbf{u}_{i_K}^{K\ \top})\big]. \qquad (18)\end{aligned}$$

As in Algorithm 1, $\mathbb{E}_q(\hat{a})$ and $\mathbb{E}_q(\hat{b})$ are then used to build a new Gamma posterior for τ to update $\tilde{f}_{\mathbf{i}}(\tau)$.

For binary tensor entries, we use $\prod_k \tilde{f}_{\mathbf{i}}^k(\mathbf{u}_{i_k}^k)$ to approximate the likelihood in (13). The conditional tilted distribution for each message $\tilde{f}_{\mathbf{i}}^k(\mathbf{u}_{i_k}^k)$ is

$$\hat{p}_{\mathbf{i}}(\mathbf{u}_{i_k}^k|\mathbf{u}_{\mathbf{i}}^{\setminus k}) \propto q^{\setminus \mathbf{i}}(\mathbf{u}_{i_k}^k)\psi\big((2y_{\mathbf{i}} - 1)\mathbf{z}_{\mathbf{i}}^{\setminus k^\top}\mathbf{u}_{i_k}^k\big).$$

The log normalizer is tractable and so we can derive the conditional moments analytically. They have the same form as the moments required in EP for Bayesian probit regression (Dušek, 2013). We can then use Taylor approximations (10)(11) to compute the expected conditional moments and to update each $\tilde{f}_{\mathbf{i}}^k(\mathbf{u}_{i_k}^k)$. The details for both continuous and binary tensors are given in the supplementary material.

4.2 BAYSESIAN LOGISTIC REGRESSION

Given the observed classification instances $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$ and binary labels $\mathbf{y} = [y_1, \ldots, y_n]$, the joint probability of the Bayesian logistic model is

$$p(\mathbf{y}, \mathbf{w}|\mathbf{X}) = p(\mathbf{w})\prod_{i=1}^n 1/\big(1 + \exp(-(2y_i - 1)\mathbf{w}^\top\mathbf{x}_i)\big),$$

where the prior $p(\mathbf{w})$ is usually chosen as a Gaussian distribution. The moment matching in regard to the logistic likelihood is intractable, and there is a smart approximation (Gelman et al., 2013). Here, we consider CEP instead. We choose fully factorized messages $\prod_m \tilde{f}_{im}(w_m)$ to approximate each logistic likelihood, where each message $\tilde{f}_{im}(w_m)$ is Gaussian and so are $q(w_m)$ and $q^{\setminus i}(w_m)$. The conditional tilted distribution is

$$\hat{p}_i(w_m|\mathbf{w}_{\setminus m}) \propto q^{\setminus i}(w_m)g_{im}(w_m|\mathbf{w}_{\setminus m}),$$

where $\mathbf{w}_{\setminus m}$ are $\mathbf{w}$ excluding w_m, and $g_{im}(w_m|\mathbf{w}_{\setminus m}) = 1/\big(1 + \exp(-(2y_i - 1)w_m x_{im} - (2y_i - 1)\mathbf{w}_{\setminus m}^\top\mathbf{x}_{i\setminus m})\big)$. Here x_{im} is the m-th element of $\mathbf{x}$ and $\mathbf{x}_{i\setminus m}$ the remaining elements. The moments of w_m are still intractable. Nevertheless, because the conditional distribution is for a single variable, we can approximate the moments by a quadrature formula. Since $q^{\setminus i}(w_m)$ is Gaussian, we can use Gauss-Hermite quadrature. Given quadrature nodes and weights $\{(\gamma_j, \alpha_j)\}_j$, the conditional moments can be represented by

$$\mathbb{E}(w_m|\mathbf{w}_{\backslash m}) \approx \frac{\sum_j \alpha_j \gamma_j g_{im}(\gamma_j|\mathbf{w}_{\backslash m})}{\sum_j \alpha_j g_{im}(\gamma_j|\mathbf{w}_{\backslash m})}, \quad (19)$$

$$\mathbb{E}(w_m^2|\mathbf{w}_{\backslash m}) \approx \frac{\sum_j \alpha_j \gamma_j^2 g_{im}(\gamma_j|\mathbf{w}_{\backslash m})}{\sum_j \alpha_j g_{im}(\gamma_j|\mathbf{w}_{\backslash m})}. \quad (20)$$

Note that the quadrature nodes are determined by $q^{\backslash i}(w_m)$; both the nodes and weights are constant to $\mathbf{w}_{\backslash m}$.

To update $\tilde{f}_{im}$, we compute the expectation of the conditional moments w.r.t $q(\mathbf{w}_{\backslash m})$. To this end, we can use their first-order or second-order Taylor approximations and then take expectation as shown in (10) and (11). Both are straightforward to derive and efficient to calculate. The details are provided in the supplementary material.

5 RELATED WORK

Expectation propagation (EP) (Minka, 2001) is a deterministic approximate inference algorithm that unifies assumed-density-filtering (ADF) (Maybeck, 1982; Lauritzen, 1992; Boyen and Koller, 1998) and loopy belief propagation (Murphy et al., 1999). It iteratively minimizes (local) KL divergence via moment matching to update factor approximations, which can be viewed as message passing and updates in graphical model representations. EP updates are essentially fixed point iterations. Although the convergence is not guaranteed, EP often converges fast in practice and produces accurate posterior estimations. EP can be further extended to power EP (Minka, 2004) that minimizes the α-divergence during the message update. Another important variant is stochastic EP (SEP) (Li et al., 2015), which only stores and updates a single or a few approximate factors and hence can largely reduce the memory cost.

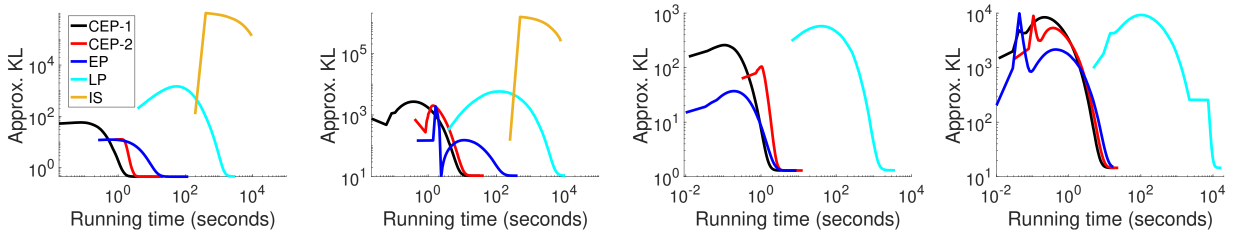
Using EP can be troublesome when the moment matching is intractable. While importance sampling can solve this problem in principle, it needs massive samples to obtain a reliable estimation and is very expensive. To address this issue, Eskin et al. (2004) developed Laplace propagation (LP) to find the Laplace approximation to the tilted distribution. While getting rid of sampling, LP requires repeated numerical optimizations during the message updates and can still be costly. Recently, several excellent works propose to use machine learning models to predict the moments (Heess et al., 2013; Eslami et al., 2014; Jitkrittum et al., 2015).The training examples are collected by running importance sampling in the first a few EP iterations. Heess et al. (2013) used neural-networks, Eslami et al. (2014) random forests and Jitkrittum et al. (2015) Gaussian processes with random features. In (Jitkrittum et al., 2015; Eslami et al., 2014), the predictive variance or an uncertainty score is used to monitor the quality of the predicted moments – if it exceeds a threshold, importance sampling is used again to improve the estimation and to replenish the training data. Despite the promising performance, these methods have to run a lot of importance sampling to collect sufficient training examples, which might still be expensive. Moreover, extra efforts need to be made to design models, extract features and choose hyper-parameters. There are also concerns in the generalizability to new data points; the learned model only specializes on one type of factors, and usually cannot generalize to other types. To address these practical barriers, we propose CEP that bypasses the tricky moment matching, instead seeks for easy and analytical conditional moments and then takes the expectation over the conditional moments. The expectation can be efficiently calculated through Taylor approximations. In this way, CEP avoids to manually design approximations or to run costly importance sampling and numerical optimizations. It can still conduct efficient and analytical message refinements. We can immediately adapt CEP to a stochastic or streaming version, like SEP and ADF.

6 EXPERIMENT

6.1 BAYESIAN PROBIT AND LOGISTIC REGRESSION

We first examined our approach on two classical models, Bayesian logistic regression (BLR) and probit regression (BPR). EP has analytical forms for moment matching in BPR. While in BLR the moments are intractable due to the logistic function, we can develop accurate approximations based on quadrature rules (Gelman et al., 2013).

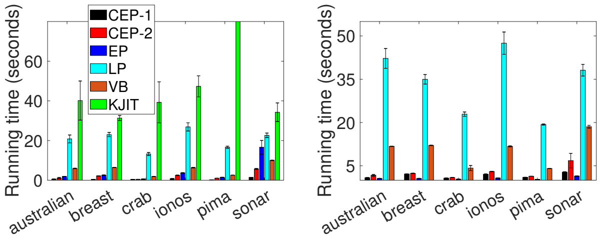
Methods. As in (Graepel et al., 2010), we used fully factorized messages. This is important for large-scale applications. We implemented two versions of the proposed algorithm: CEP-1 that uses the first-order Taylor approximation (see (10)) and CEP-2 the second-order (see (11)). Note that due to the fully factorized approximation, CEP-2 is equivalent to using the diagonal Hessian and covariance in the Taylor approximation. In BLR, we used the Gauss-Hermite quadrature to represent the conditional moments with 9 nodes (see Section 4.2). We compared with EP, Laplace propagation (LP), EP with importance sampling (IS) for moment matching, and variational Bayes(VB). For a fair comparison, these methods employed fully factorized posteriors as well. In addition, we compared with the kernel based just-in-time (KJIT) learning to predict the moments (Jitkrittum et al., 2015) for EP updates. KJIT uses IS to dynamically generate training examples and update the predictor. For LP, we ran L-BFGS to find the mode and constructed the Laplace approximation in each message update. The maximum number of iterations was set to 100. For factorized EP in BLR, we followed (Gelman et al., 2013) to develop a two dimen-



(a) Simu1 in BLR (b) Simu2 in BLR (c) Simu1 in BPR (d) Simu2 in BPR

Figure 1: Running time *vs.* approximate KL divergence. BPR and BLR are short for Bayesian probit regression and Bayesian logistic regression, respectively.

sional Gauss-Hermite quadrature to compute the moments (see Section 2.2.1 of the supplementary material for details). Note that the original approach in (Gelman et al., 2013) is inappropriate for factorized EP, resulting in much worse performance. For IS and KJIT, we generated $500K$ samples to estimate each moment. Note that we only tested IS and KJIT for BLR, where the exact moments are intractable. We used a flat Gaussian, $\mathcal{N}(\cdot|0, 10^6)$, as the initialization for all the messages in EP, CEP-1, CEP-2, LP and IS. For VB, the initial posterior of each variable was set to the prior. We used the original implementation of KJIT with C# under Infer.Net framework (`https://github.com/wittawatj/kernel-ep`), which employs a joint Gaussian term to approximate each likelihood (rather than a factorized one). All the other methods were implemented with MATLAB 2017. We ran all the algorithms on a single Linux server with Intel(R) i7 CPUs and 24GB memory.



(a) Bayesian logistic regression (b) Bayesian probit regression

Figure 2: Average running time on six real datasets.

Synthetic datasets. To examine the inference quality of our approach, we followed (Li et al., 2015) to simulate two datasets for BLR and BPR, respectively. Each dataset include $10,000$ samples and 4 classification weights. Each weight was sampled from the standard normal prior $\mathcal{N}(\cdot|0, 1)$. In the first dataset for each model, the feature values of each instance $\mathbf{x}_n$ were independently sampled from a single Gaussian distribution $\mathcal{N}(\cdot|0, 1)$ while in the second dataset from a mixture of Gaussian distributions with 5 components, $\frac{1}{5}\big(N(\cdot|-2, \frac{1}{2}) + N(\cdot|-1, \frac{1}{2}) + N(\cdot|0, \frac{1}{2}) + N(\cdot|1, \frac{1}{2}) + N(\cdot|2, \frac{1}{2})\big)$. The label y_n was sampled from the generative model. Hence, the second dataset for each model is more heterogeneous. To evaluate the inference quality, we ran No-U-Turn (NUTS) (Hoffman and Gelman, 2014) sampler for $100K$ iterations as the burn-in time and then drew $50K$ posterior samples, from which we estimated a Gaussian distribution as the golden standard. We then ran CEP-1, CEP-2, EP, LP, IS and KJIT to obtain the posterior estimations of the classification weights. We computed the KL divergence between the golden standard and the posteriors estimated by each method. Note that we did not compare with VB because VB minimizes the KL divergence in a reverse direction and this metric may not be fair to VB. Although taking much longer running time than CEP-$\{1,2\}$, EP and LP, KJIT still obtains much larger approximate KL divergence, implying much worse inference quality or even a failure. The running time and the KL divergence of KJIT are $\{9595.4,\ 10150.3\}$ seconds and $\{14.7,\ 1.4e13\}$ on the two simulation datasets for BLR. We show for all the other methods how the KL divergence varies along with running time in Fig. 1. As we can see, similar to KJIT, IS spent the most running time but resulted in the worst performance (Fig. 1a and b). This might be due to the unstable accuracy of the estimated moments, even with $500K$ samples. In both BPR and BLR, our algorithms, CEP-1 and CEP-2, obtain almost the same approximate KL divergence with EP, implying the same inference quality. Although LP ends up with a similar KL divergence as well, it is far slower than CEP-1, CEP-2 and EP. CEP-1 is slightly faster than CEP-2, because CEP-2 uses the second order Taylor approximation and needs extra computation. While the running time of EP, CEP-1 and CEP-2 are close in BPR, EP is slower than both CEP-1 and CEP-2 in BLR. The reason might be that EP has to use a two-dimensional Gauss-Hermite quadrature for moment matching, while CEP-1 and CEP-2 just one dimensional quadrature.

Real datasets. To further examine the predictive performance and computational efficiency, we tested all the methods on six real-world datasets from UCI machine learning repository(`https://archive.ics.uci.edu/ml/index.php`), *australian*, *breast*, *crab*,*ionos*, *pima* and *sonar*. We randomly split each dataset into a half for training and the other half for testing. We split the dataset for five times and reported the average test log likelihood, area under ROC curve (AUC), and running time of each method. The test log likelihoods are summarized in Table 1. Due to the space limit, the AUC results are provided in the supplementary material. The running time are shown in Fig. 2.

From Table 1a and b, we can see that CEP-1 and CEP-2 nearly always obtain average test log-likelihoods (a bit) larger than or close to that of EP. It demonstrates that our approach can have the same predictive performance as EP. In most cases, CEP-2 is slightly better than CEP-1, and this might be due to the usage of the second-order Taylor approximations. LP achieves a close prediction accuracy to EP as well, but it is much slower (see Fig. 2). VB performs best in *crab* dataset, but is suboptimal in the other datasets. KJIT, however, performs the worst in most cases, especially on *pima* and *sonar*. In fact, KJIT's performance is quite unstable. It often results in an exceedingly small log-likelihood, implying a complete failure. We have to run several times to guarantee reasonable accuracy.

Fig. 2 shows the average running time of each method. As we can see, CEP-1, CEP-2 and EP are much faster than LP, which is consistent with the results from the simulation (see Fig. 1). Note that KJIT is much slower than LP, especially for *pima*, where KJIT's average running time is 904.626 seconds. CEP-1 and CEP-2 are also significantly faster than VB. In BPR, the speeds of CEP-1 and CEP-2 are close to that of EP, while in BLR, EP is slower, especially on *sonar*, which might be again due to the two dimensional quadrature for moment matching. The results confirm the computational advantage of our approach.

6.2 BAYESIAN TENSOR DECOMPOSITION

We next examined our approach in Bayesian tensor decomposition (as in Section 4.1). Due to the production term in the likelihood (see (12)(13)), the moment matching is intractable and hence standard EP is infeasible. To the best of our knowledge, efficient and accurate approximations remain absent.

Datasets. We used two real-world datasets (Zhe et al., 2016b): (1) *Alog*, a $200 \times 100 \times 200$ tensor with continuous entry values, and (2) *Enron*, a binary tensor of size $203 \times 203 \times 200$. *Alog* represents the three-way (user, action, resource) interactions in a file access log, and includes 0.33% nonzero entries. *Enron* describes the three-way relationship (sender, receiver, time) in emails, with 0.01% nonzero entries.

Methods. We compared with LP, variational message passing (VMP), and a state-of-the-art tensor decomposition algorithm, CP-WOPT (Acar et al., 2011). VMP has been proven successful in matrix decomposition when the EP-based messages are unavailable (Stern et al., 2009). We implemented VMP for tensor decomposition. As in (Stern et al., 2009), to ensure tractable updates, VMP uses an augmented model representation for binary tensors, where for each entry $\mathbf{i}$, a latent continuous entry value $z_{\mathbf{i}}$ is first sampled from $\mathcal{N}\big(z_{\mathbf{i}}|\mathbf{1}^\top(\mathbf{u}^1_{i_1} \circ \ldots \circ \mathbf{u}^K_{i_K}), 1\big)$, and then the observation $y_{\mathbf{i}}$ determined from a step function, $\mathbb{1}(y_{\mathbf{i}} = 1)\mathbb{1}(z_{\mathbf{i}} > 0) + \mathbb{1}(y_{\mathbf{i}} = 0)\mathbb{1}(z_{\mathbf{i}} \leq 0)$ where $\mathbb{1}(\cdot)$ is the indicator function. We implemented our approach, CEP, based on the first-order Taylor approximations in (10) (the second-order based one turns out be to worse although with more computation).

The test setting is the same as in (Zhe et al., 2016b). We evaluated all the methods via a 5-fold cross validation. Specifically, we randomly split the nonzero entries into 5 folds where 4 folds and the same number of zero entries were sampled for training. We used the rest of nonzero entries and sampled 0.1% of the remaining zero entries for test. In so doing, the nonzero and zero entries were considered as equally important, and we avoided the evaluation being dominated by the large portion of zeros. We chose the rank, i.e., the dimension of the embedding vectors, from $\{3, 5, 8, 10\}$.

We report the average root-mean-square-error (RMSE) for *Alog* and AUC for *Enron* as well as their standard deviations in Fig. 3 a and c. We can see that CEP outperforms LP and CP-WOPT by a large margin in all the cases. The prediction accuracy of CEP is close to (when the rank is 3 or 5 for *Alog*) or significantly better than VMP (in all the other cases). Note that the inferior performance of VMP in binary data might be due to the extra variational approximations for the latent continuous variables in the augmented model representation. The reason why LP is far worse than CEP might be that the numerical optimizations were saturated in poor local maxima, resulting in inferior Laplace approximations. We also report the per-iteration running time of CEP, LP and VMP in Fig. 3 b and d. As we can see, CEP has a close speed to VMP, and is much faster than LP — on average with over 40 times speedup. The results consistently demonstrate the advantage of CEP in both predictive performance and computational efficiency. Finally, we show in Fig. 4 how the predictive performance of CEP, LP and VMP varied along with the running iterations when the rank is 10 (more results are given in the supplementary material). As we can see, the prediction accuracy of all the methods converges quickly and keeps stable with more iterations.

6.3 STREAMING TENSOR DECOMPOSITION

Finally, we adapted CEP to the assumed density filtering (ADF) framework, denoted by ADF-CEP, for streaming tensor decomposition. We compared with the state-of-the-art Bayesian streaming tensor decomposition algorithm, POST (Du et al., 2018), which is essentially VMP adjusted to the streaming variational Bayes (Broderick et al., 2013). We also compared with CP-WOPT that performs

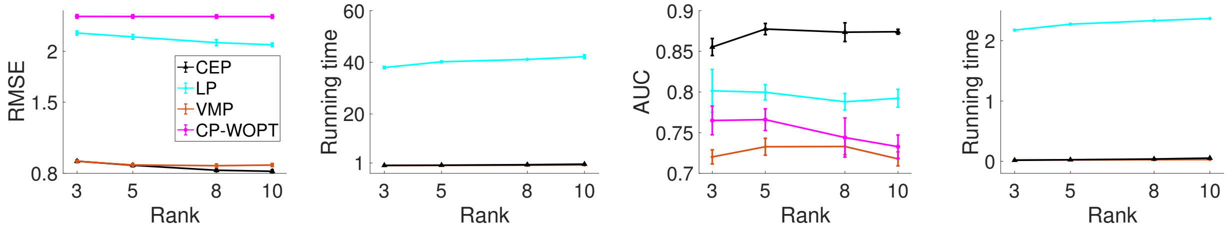
Dataset	CEP-1	CEP-2	LP	EP	VB	KJIT
australian	$\mathbf{-0.449 \pm 0.012}$	-0.450 ± 0.015	-0.451 ± 0.012	-0.450 ± 0.015	-0.449 ± 0.015	-0.451 ± 0.013
breast	-0.579 ± 0.020	$\mathbf{-0.565 \pm 0.020}$	-0.576 ± 0.018	-0.585 ± 0.024	-0.587 ± 0.023	-0.576 ± 0.017
crab	-0.316 ± 0.004	-0.315 ± 0.003	-0.349 ± 0.012	-0.313 ± 0.003	$\mathbf{-0.277 \pm 0.004}$	-0.327 ± 0.004
ionos	-0.316 ± 0.022	$\mathbf{-0.302 \pm 0.023}$	-0.316 ± 0.022	-0.309 ± 0.025	-0.332 ± 0.032	-0.339 ± 0.018
pima	-0.541 ± 0.007	$\mathbf{-0.540 \pm 0.006}$	-0.541 ± 0.006	-0.541 ± 0.007	-0.542 ± 0.007	-0.608 ± 0.014
sonar	-0.522 ± 0.017	$\mathbf{-0.513 \pm 0.027}$	-0.519 ± 0.016	-0.531 ± 0.026	-0.578 ± 0.026	-1.085 ± 0.053

(a) Bayesian logistic regression

Dataset	CEP-1	CEP-2	LP	EP	VB
australian	$\mathbf{-0.428 \pm 0.012}$	-0.431 ± 0.015	$\mathbf{-0.428 \pm 0.012}$	-0.435 ± 0.015	-0.441 ± 0.017
breast	$\mathbf{-0.583 \pm 0.018}$	-0.592 ± 0.023	-0.594 ± 0.023	-0.615 ± 0.032	-0.621 ± 0.035
crab	-0.228 ± 0.009	-0.226 ± 0.009	-0.231 ± 0.009	-0.250 ± 0.008	$\mathbf{-0.197 \pm 0.010}$
ionos	-0.286 ± 0.012	$\mathbf{-0.277 \pm 0.015}$	-0.319 ± 0.009	-0.307 ± 0.010	-0.458 ± 0.036
pima	$\mathbf{-0.553 \pm 0.009}$	-0.554 ± 0.009	$\mathbf{-0.553 \pm 0.009}$	-0.554 ± 0.009	-0.557 ± 0.010
sonar	-0.533 ± 0.032	$\mathbf{-0.528 \pm 0.032}$	-0.579 ± 0.030	-0.553 ± 0.039	-0.891 ± 0.070

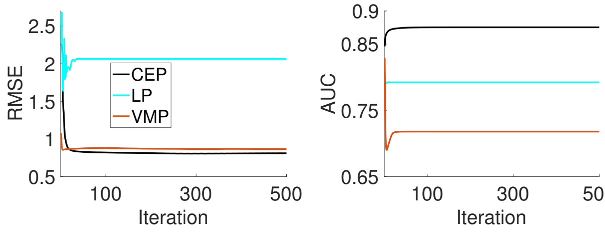
(b) Bayesian probit regression.

Table 1: Average test log-likelihoods on six real datasets.



(a) *Alog* (b) *Alog* running time (c) *Enron* (d) *Enron* running time

Figure 3: Prediction accuracy and per-iteration running time of all the methods for tensor decomposition. Note that the running time of VMP and CEP are close.



(a) *Alog* (b) *Enron*

Figure 4: Average prediction accuracy *v.s.* running iteration. The rank of the embeddings is 10.

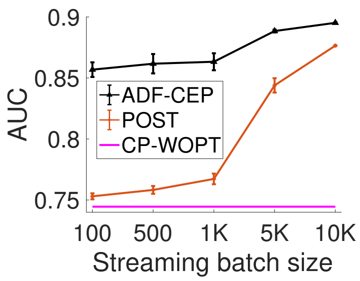


Figure 5: The predictive performance for streaming decomposition. The rank of the embeddings is 8.

static decomposition. We tested on a real-world binary tensor, *DBLP* (Du et al., 2018), which depicts the bibliography relationships (author, conference, keyword). The tensor is $10K \times 200 \times 10K$, including $\%0.001$ nonzero entries. We followed (Du et al., 2018) to sample 80% nonzero entries and randomly sampled the same number of zero entries to obtain a balanced training set. We then sampled 50 test sets from the remaining entries — each comprises 200 nonzero and $1,800$ zero elements. We randomly shuffled the training entries, and partitioned them into many small batches. These batches were streamed to ADF-CEP and POST. After all the batches were processed, we evaluated the predictive performance of the learned models on the 50 test sets. We varied the batch sizes from $\{100, 500, 1K, 5K, 10K\}$, and set the rank of the embeddings to 8. We conducted 5 runs for each method, and report the average and standard deviation of AUC in Fig. 5. As we can see, ADF-CEP consistently outperforms POST and CP-WOPT by a large margin.

7 CONCLUSION

We have developed conditional expectation propagation (CEP) to overcome the practical barriers of EP when the moment matching is intractable. CEP conducts efficient and analytical updates, without the need for hand-crafted approximations, expensive sampling or numerical optimizations. The performance of CEP in the inference tasks of several important Bayesian models is encouraging. In the future work, we will continue to explore CEP in theory, *e.g.,* energy functions, and apply CEP in more complex models, *e.g.,* Bayesian neural networks.

References

Acar, E., Dunlavy, D. M., Kolda, T. G., and Morup, M. (2011). Scalable tensor factorizations for incomplete data. Chemometrics and Intelligent Laboratory Systems, 106(1):41–56.

Boyen, X. and Koller, D. (1998). Tractable inference for complex stochastic processes. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, pages 33–42. Morgan Kaufmann Publishers Inc.

Broderick, T., Boyd, N., Wibisono, A., Wilson, A. C., and Jordan, M. I. (2013). Streaming variational bayes. In Advances in Neural Information Processing Systems, pages 1727–1735.

Du, Y., Zheng, Y., Lee, K.-c., and Zhe, S. (2018). Probabilistic streaming tensor decomposition. In 2018 IEEE International Conference on Data Mining (ICDM), pages 99–108. IEEE.

Dušek, O. (2013). Expectation propagation for a probit regression model. Technical report, Institute of Formal and Applied Linguistics.

Eskin, E., Smola, A. J., and Vishwanathan, S. (2004). Laplace propagation. In Advances in neural information processing systems, pages 441–448.

Eslami, S. A., Tarlow, D., Kohli, P., and Winn, J. (2014). Just-in-time learning for fast and flexible inference. In Advances in Neural Information Processing Systems, pages 154–162.

Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). Bayesian data analysis. chapter 10. Chapman and Hall/CRC.

Graepel, T., Candela, J. Q., Borchert, T., and Herbrich, R. (2010). Web-scale Bayesian click-through rate prediction for sponsored search advertising in Microsoft's Bing search engine. Omnipress.

Harshman, R. A. (1970). Foundations of the PARAFAC procedure: Model and conditions for an"explanatory"multi-mode factor analysis. UCLA Working Papers in Phonetics, 16:1–84.

Heess, N., Tarlow, D., and Winn, J. (2013). Learning to pass expectation propagation messages. In Advances in Neural Information Processing Systems, pages 3219–3227.

Herbrich, R., Minka, T., and Graepel, T. (2007). Trueskill$^{\mathrm{TM}}$: A bayesian skill rating system. In Schölkopf, B., Platt, J. C., and Hoffman, T., editors, Advances in Neural Information Processing Systems 19, pages 569–576. MIT Press.

Hernández-Lobato, J. M. and Adams, R. (2015). Probabilistic backpropagation for scalable learning of bayesian neural networks. In International Conference on Machine Learning, pages 1861–1869.

Hoffman, M. D. and Gelman, A. (2014). The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. Journal of Machine Learning Research, 15(1):1593–1623.

Jitkrittum, W., Gretton, A., Heess, N., Eslami, S., Lakshminarayanan, B., Sejdinovic, D., and Szabó, Z. (2015). Kernel-based just-in-time learning for passing expectation propagation messages. In Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, pages 405–414. AUAI Press.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. IEEE Transactions on information theory, 47(2):498–519.

Lauritzen, S. L. (1992). Propagation of probabilities, means, and variances in mixed graphical association models. Journal of the American Statistical Association, 87(420):1098–1108.

Li, Y., Hernández-Lobato, J. M., and Turner, R. E. (2015). Stochastic expectation propagation. In Advances in Neural Information Processing Systems, pages 2323–2331.

Maybeck, P. S. (1982). Stochastic models, estimation, and control, volume 3. Academic press.

Minka, T. (2004). Power EP. Technical report, Technical report, Microsoft Research, Cambridge.

Minka, T., Winn, J., Guiver, J., and Knowles, D. (2014). Infer .net 2.4, 2010. microsoft research cambridge.

Minka, T. P. (2001). Expectation propagation for approximate bayesian inference. In Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence, pages 362–369. Morgan Kaufmann Publishers Inc.

Murphy, K. P., Weiss, Y., and Jordan, M. I. (1999). Loopy belief propagation for approximate inference: An empirical study. In Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, pages 467–475. Morgan Kaufmann Publishers Inc.

Stern, D. H., Herbrich, R., and Graepel, T. (2009). Matchbox: large scale online Bayesian recommendations. In Proceedings of the 18th international conference on World wide web, pages 111–120. ACM.

Zhe, S., Lee, K.-c., Zhang, K., and Neville, J. (2016a). Online spike-and-slab inference with stochastic expectation propagation. NIPS WS.

Zhe, S., Zhang, K., Wang, P., Lee, K.-c., Xu, Z., Qi, Y., and Ghahramani, Z. (2016b). Distributed flexible nonlinear tensor factorization. In Advances in Neural Information Processing Systems, pages 928–936.