Variational Inference on Deep Exponential Family by using Variational Inferences on Conjugate Models

Mohammad Emtiyaz Khan
Center for Advanced Intelligence Project
RIKEN, Tokyo, Japan
emtiyaz@gmail.com

Wu Lin*
wu.lin@uwaterloo.ca

Abstract

In this paper, we propose a new variational inference method for deep exponential-family (DEF) models. Our method converts non-conjugate factors in a DEF model to easy-to-compute conjugate exponential-family messages. This enables local and modular updates similar to variational message passing, as well as stochastic natural-gradient updates similar to stochastic variational inference. Such updates make our algorithm highly suitable for large-scale learning. Our method exploits the structure of the deep network and can be useful to reduce the variance of stochastic methods for variational inference.

1 Introduction

In this paper, we propose a new variational inference method for deep exponential-family (DEF) models. DEFs were recently proposed by Ranganath et al. (2015) and they contain many existing deep models as special cases, e.g. Sigmoid Belief Network and Deep Latent Gaussian models. Inference in these models is intractable due to non-conjugate factors. To solve this problem, variational inference based on stochastic approximation is applied, e.g. Ranganath et al. (2015) use the black-box variational inference method.

Such stochastic gradient-descent (SGD) methods extend the applicability of variational inference to many intractable deep models (Ranganath et al., 2013; Salimans et al., 2013; Paisley et al., 2012; Titsias and Lázaro-Gredilla, 2014), but they lack the modularity and efficiency of classical variational inference methods such as variational message passing (VMP) (Winn and Bishop, 2005). For example, the gradient estimation in SGD might have high-variance when computed naively without taking the structure of the problem into account (Kingma and Welling, 2013). In contrast, message passing algorithms can exploit the structure and can do efficient local computations. Unfortunately, VMP does not apply to DEFs since DEFs contain non-conjugate factors.

In this paper, we derive a modular variational inference algorithm that combines stochastic methods with message passing algorithms. Our method converts non-conjugate factors to easy-to-compute conjugate exponential-family messages, thereby enabling updates similar to variational message passing. The conversion is computationally efficient since it only requires local computations of gradients at nodes. Moreover, our method enables the use of doubly-stochastic gradients, similar to stochastic variational inference. Such local and stochastic updates make our algorithm suitable for large-scale learning. Even with such updates, our method is guaranteed to converge.

*The author was a freelance researcher during this work and had no affiliations.

Workshop on Bayesian Deep Learning, NIPS 2016, Barcelona, Spain.

2 Conjugate-Computation Variational Inference (CVI)

Our algorithm for DEF is based on a variational inference method called conjugate-computation variational inference[2]. Given a probabilistic model with a factor graph containing both conjugate and non-conjugate terms, CVI computes a conjugate approximation of the non-conjugate terms, thereby converting a non-conjugate problem into a conjugate problem. A stochastic variational inference on the resulting conjugate problem is equivalent to a proximal-gradient step, which is guaranteed to converge to a local maximum of the variational lower bound. The proximal-gradient step is also a natural-gradient step making it suitable for variational inference Hoffman et al. (2013). We now give a few details of the method.

Given a model $p(\mathbf{y}, \mathbf{z})$, where $\mathbf{y}$ is the data and $\mathbf{z}$ is the latent vector, we wish to estimate an approximation $q(\mathbf{z})$ to the posterior distribution $p(\mathbf{z}|\mathbf{y})$ by maximizing the following lower bound to the log-marginal likelihood:

$$\mathcal{L}(q) := \mathbb{E}\left[\log p(\mathbf{y}, \mathbf{z}) - \log q(\mathbf{z})\right] \tag{1}$$

We consider the case of a mean-field approximation $q(\mathbf{z}) = \prod_k q(z_k)$ which factorizes over all entries of $\mathbf{z}$ (the method itself applies to more general distributions, e.g. structured mean-field). Given a posterior estimate $q_t(\mathbf{z})$ at the t'th iteration and a non-conjugate[3] factor $f_a(\mathbf{z})$ of $p(\mathbf{y}, \mathbf{z})$, CVI computes an exponential-family approximation of the factor f_a in the $t+1$'th iteration as shown below:

$$f_a(\mathbf{z}) \approx \prod_k \text{ExpFam}(z_k, \tilde{\lambda}_{f_a \to z_k}), \text{ where } \tilde{\lambda}_{f_a \to z_k} \leftarrow w_i \tilde{\lambda}_{f_a \to z_k} + (1 - w_i) \widehat{\nabla}_{\mu_k} \mathbb{E}_{q_t}[\log f_a] \tag{2}$$

where $\text{ExpFam}(z_k, \tilde{\lambda})$ denotes an (unnormalized) exponential-family distribution over z_k with natural parameter $\tilde{\lambda}$, the scalar $w_i \in (0, 1)$ is the step-size, and $\widehat{\nabla}_{\mu_k} \mathbb{E}_{q_t}(\log f_a)$ is the *stochastic* gradient w.r.t. the mean parameter μ_k of $q(z_k)$ at $q_t(z_k)$. Given the above conjugate approximation at the $t+1$ iteration, CVI updates the posterior $q(\mathbf{z})$ using a *stochastic* variational inference on the conjugate model. The algorithm repeats these two steps until convergence, i.e. update $\tilde{\lambda}_{f_a \to z_k}$ for all non-conjugate nodes and factors, and compute $q(\mathbf{z})$ on the conjugate approximation. An illustrative example is shown in Figure 1.

The above approximation is very similar to Expectation Propagation (EP), but instead of doing moment-matching as in EP, the CVI approximation maximizes the variational lower bound of (1). The advantage of this approximation is that, unlike EP, it can handle stochastic updates, i.e. the gradient estimates can be stochastic (need to be unbiased and bounded variance). In addition, the non-conjugate factor can also be randomly selected. Once a factor's approximation is updated, $q(z_k)$ for all the variables z_k in its neighborhood are updated. This enables stochastic updates where not only the data can be selected at random but also nodes can be updated at random.

The above update corresponds to a stochastic proximal-gradient method for which convergence is guaranteed under very mild continuity assumption of the lower bound. Connection to proximal-gradient methods shows that our method is a natural-gradient method that uses a KL-divergence instead of a Euclidean distance (see Khan et al. (2016) for details). Therefore, our method is an extension of stochastic variational inference Hoffman et al. (2013) to non-conjugate models. Because of this property, we expect our method to perform better than naive SGD based methods, e.g. black-box variational inference Ranganath et al. (2013), that ignore the geometry of the variational parameter space.

Another important difference with existing methods such as Ranganath et al. (2013) and Salimans et al. (2013) is that, in our method, the gradient are averaged in $\tilde{\lambda}$ before updating the variational parameter $\boldsymbol{\lambda}$. Therefore our method is expected to be more stable during learning than existing methods.

[2]This work is currently under submission.

[3]A factor is non-conjugate w.r.t. a variable z_k when it does not take the same functional form as $q(z_k)$ with respect to z_k.

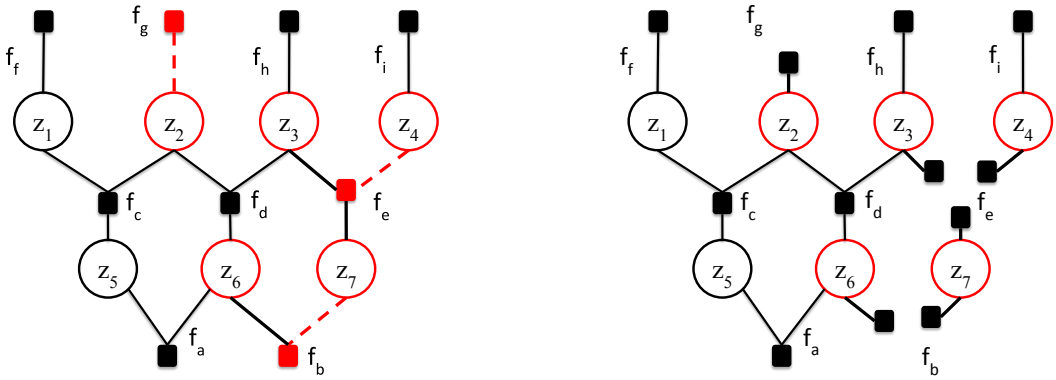


Figure 1: Figure (a) shows an example of non-conjugate graphical model. A dashed red-line indicates a non-conjugate relationship between the variable and factor, e.g. factor f_e is non-conjugate with respect to variable z_4. We call such factors the non-conjugate factors (shown in red f_b, f_e, and f_g) and the neighboring nodes of these factors the non-conjugate nodes (shown in red z_2, z_3, z_4, z_6, and z_7). Figure (b) shows the approximate conjugate model obtained by using approximation shown in (2). The algorithm uses stochastic variational inference on this conjugate model to update $q(\mathbf{z})$. We repeat these two steps until convergence.

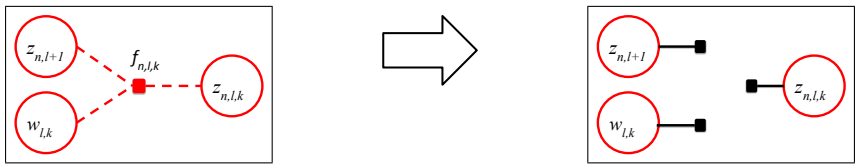


Figure 2: The left figure shows a non-conjugate factor of DEFs and the right figure shows the conjugate approximation obtained using (2).

3 CVI for Deep Exponential-Family Models

We now show how CVI simplifies variational inference on DEFs. Given data points $\mathbf{y}_n$, a DEF model has L layers of latent variables $\mathbf{z}_n := \{\mathbf{z}_{n,1}, \mathbf{z}_{n,2}, \ldots, \mathbf{z}_{n,L}\}$, where $\mathbf{z}_{n,l}$ for l'th layer is of length K_l. Each layer has a parameter $\mathbf{W}_l$ whose column $\mathbf{w}_{l,k}$ is assumed to follow an exponential-family distribution with natural parameter $\boldsymbol{\eta}_{\mathbf{w}_{l,k}}$. Each entry in the top latent vector $\mathbf{z}_{n,L}$is assumed to follow an exponential-family distribution as well (denote its natural parameter by $\eta_{z_{n,L,k}}$). The subsequent latent vectors are then sampled as shown below:

$$p(\mathbf{z}_{n,l}|\mathbf{z}_{n,l+1}, \mathbf{w}_{l,k}) = \prod_{k=1}^{K_l} \text{ExpFam}_l\left[z_{n,l,k}, g_l(\mathbf{z}_{n,l+1}^T \mathbf{w}_{l,k})\right], \tag{3}$$

where g_l are link functions. Finally, the latent vector $\mathbf{z}_{n,1}$ are combined with $\mathbf{w}_{0,d}$ to model the d'th dimension in $\mathbf{y}_n$.

Following Ranganath et al. (2015), we wish to find the following approximation (K_0 is equal to the length of $\mathbf{y}_n$):

$$q(\mathbf{z}, \mathbf{w}) = \left[\prod_{l=0}^{L-1}\prod_{k=1}^{K_l} \text{ExpFam}_l(\mathbf{w}_{l,k}, \boldsymbol{\lambda}_{w_{l,k}})\right]\left[\prod_{n=1}^{N}\prod_{l=1}^{L}\prod_{k=1}^{K_l} \text{ExpFam}_l(z_{n,l,k}, \lambda_{z_{n,l,k}})\right] \tag{4}$$

The problem is non-conjugate due to factors $f_{n,l,k} := \text{ExpFam}[z_{n,l,k}, g_l(\mathbf{z}_{n,l+1}^T, \mathbf{w}_{l,k})]$ and we can convert the problem to a conjugate one by projecting these factor into exponential family. The expectation of the corresponding term in the lower bound is $\mathbb{E}_q[\log f_{n,l,k}]$ and it depends on the parameters of the distributions $q(z_{n,l,k})$, $q(\mathbf{z}_{n,l+1})$, and $q(\mathbf{w}_{l,k})$. Therefore, using (2) we can compute

the following site-parameters by computing the gradient of the factor $f_{n,l,k}$ with respect to the mean parameter of the above three distributions:

$$\tilde{\lambda}_{f_{n,l,k}\to z_{n,l,k}}, \quad \tilde{\lambda}_{f_{n,l,k}\to \mathbf{z}_{n,l+1}}, \quad \text{and } \widetilde{\boldsymbol{\lambda}}_{f_{n,l,k}\to \mathbf{w}_{l,k}}. \tag{5}$$

This step converts the non-conjugate factor $f_{n,l,k}$ into three conjugate factors as illustrated in Figure 2. The natural parameter of variables $z_{n,l+1,k}$ and $\mathbf{w}_{l,k}$ can then be updated by summing the contributions from all neighboring factors:

$$\boldsymbol{\lambda}_{\mathbf{w}_{l,k}} \leftarrow \boldsymbol{\eta}_{\mathbf{w}_{l,k}} + \sum_{n=1}^{N} \widetilde{\boldsymbol{\lambda}}_{f_{n,l,k}\to \mathbf{w}_{l,k}}, \qquad \lambda_{z_{n,l+1,k}} \leftarrow \tilde{\lambda}_{f_{n,l+1,k}\to z_{n,l+1,k}} + \sum_{j=1}^{K_l} \tilde{\lambda}_{f_{n,l,j}\to z_{n,l+1,k}} \tag{6}$$

The above inference step is equivalent to inference in a conjugate exponential-family model. These computations can be carried out locally at a node. The algorithm works as follows: we first initialize $\tilde{\lambda}_{f_a\to z_k}$ for all non-conjugate factors f_a, and repeat the following two steps until convergence:

1. Randomly select a factor $f_{n,l,k}$ and update $\tilde{\lambda}_{f_{n,l,k}\to u}$ for all neighboring nodes u.
2. At all the neighboring nodes u of f_a, update the natural parameter using (6).

4 Results

We present preliminary results on a one-layer Sigmoid Belief Network (SBN) using a very similar setup as Titsias and Lázaro-Gredilla (2015). In SBN, each hidden variable $z_{n,l,k}$ is modeled by using a Bernoulli distribution as shown below:

$$\log p(z_{n,l,k}|\mathbf{z}_{n,l+1}, \mathbf{w}_{l,k}) = z_{n,l,k}(\mathbf{z}_{n,l+1}^T \mathbf{w}_{l,k}) - \log[1 + \exp(\mathbf{z}_{n,l+1}^T \mathbf{w}_{l,k})] \tag{7}$$

Additionally, we assume a Gaussian prior on $\mathbf{w}_{l,k}$. The second term above is known as the logistic-log partition (LLP) function. This function makes the inference intractable.

As a baseline method, we use the Black-Box Variational Inference (BBVI) method of Ranganath et al. (2013) to approximation the LLP function. We call this method the 'Partial-BBVI' method since we only partially approximate the gradient using the black-box method. For variance reduction, we use the method suggested in Salimans (2014) (see Eq. 8). The fully black-box method was too slow to converge on the medium-size dataset we considered in our experiments, which is why we used the Partial-BBVI method.

We also compare to the local expectation gradient method of Titsias and Lázaro-Gredilla (2015) called 'LeGrad'.

For CVI, we use the local expectation-gradient method to compute the gradient-approximation for the LLP function. Note that computing gradients with respect to the mean parameter is trivial for both Gaussian and Bernoulli distributions.

For all methods, we use Adam (Kingma and Ba, 2014) to take stochastic gradient steps. We set the learning rate to 0.05, 0.6, and 0.1 respectively for Partial-BBVI, LeGrad, and CVI respectively. We used a decay factor of 0.9 and 0.999 for the mean and variance respectively, and ϵ was set to 1e-8. For Partial-BBVI, we used 20 samples to approximate the gradient, while for the other two methods we used 1 sample from $q(\mathbf{z})$ and 5 samples from $q(\mathbf{w})$.

We compare on two datasets. The first dataset is the Voting dataset which contains Yes/No votes from 435 senators on 17 issues. The second dataset is a smaller version of the MNIST dataset containing a total of 3000 binary images of size 28×28. For the Voting dataset, we use half of the votes for 20% of the senators as the test set. We use the batch method during training since the data size is small. For MNIST, we use 1000 images for testing, treating half of each image as test entries. For this data, we use a mini-batch of size 500 during training and a quarter of the pixels in the image randomly selected at each iteration.

We report the following mean-absolute error in reconstructing the test set:

$$MAE := \frac{1}{N_{te}} \sum_{n,d} |y_{n,d} - \hat{y}_{n,d}| \tag{8}$$

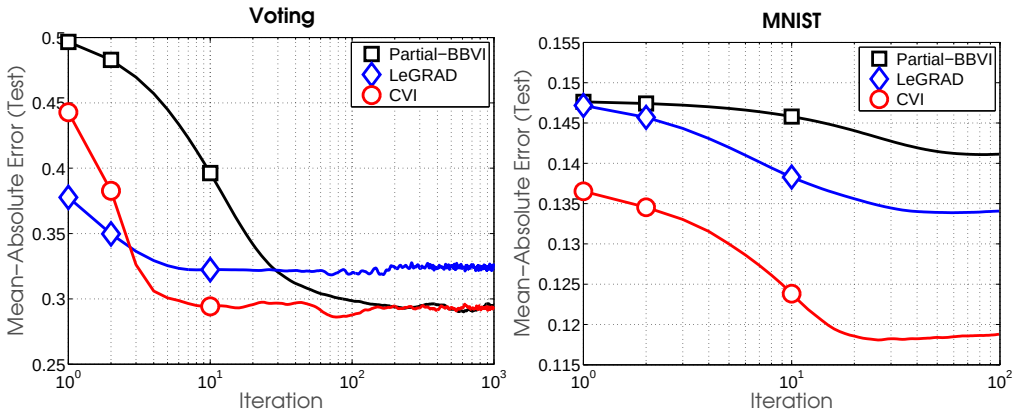


Figure 3: MAE on the test set vs iterations on Voting and MNIST datasets.

where $y_{n,d}$ is a test entry, $\hat{y}_{n,d}$ is its reconstruction by a method, and N_{te} is the total number of test entries $y_{n,d}$. We set $\hat{y}_{n,d} = p(y_{n,d}|\mathbb{E}_q(\mathbf{z}), \mathbb{E}_q(\mathbf{w}))$. This prediction is an approximation that ignores the variance, but we use it since it is cheap to compute in each iteration. A random coin flip gives an error of about 0.5 and a lower error is an improvement over it.

Figure 3 shows the evolution of MAE for the two datasets. All algorithms start with the same initial values (the first iteration is not shown in the figure). We observe that CVI gives a slightly lower test error and is as fast as LeGrad to converge. Partial-BBVI is slowest to converge which is expected due to high variance problem. A better performance of our method might be because of the fact that our method is a natural-gradient method, although this observation needs further investigation and multiple runs of our experiments are necessary to establish significance. However, looking at the reconstructions of a few test images shown in Figure 4 we observe that CVI gives more crisp reconstructions than the other methods. It is plausible that CVI is able to learn a better model, but more experiments are necessary to confirm such claims.

5 Conclusions

In this paper, we propose a new variational inference method for deep exponential-family (DEF) models. Our method enables local and modular updates similar to variational message passing, and stochastic natural-gradient updates similar to stochastic variational inference. These two features make our method highly suitable for large-scale variational inference.

Although we have considered DEFs in this paper, our method can potentially be applied to other deep models. The current trend in deep generative models is to construct a deep model by adding deep neural networks to existing probabilistic graphical model (e.g. Krishnan et al. (2015); Johnson et al. (2016)). Our method is ideal for such scenarios where non-conjugacy is introduced due to the inclusion of a deep network. We are also exploring application to variational auto-encoders where a similar algorithm can be derived.

References

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

Johnson, M. J., Duvenaud, D., Wiltschko, A. B., Datta, S. R., and Adams, R. P. (2016). Composing graphical models with neural networks for structured representations and fast inference. *arXiv preprint arXiv:1603.06277*.

Khan, M. E., Babanezhad, R., Lin, W., Schmidt, M., and Sugiyama, M. (2016). Faster Stochastic Variational Inference using Proximal-Gradient Methods with General Divergence Functions. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Krishnan, R. G., Shalit, U., and Sontag, D. (2015). Deep kalman filters. *arXiv preprint arXiv:1511.05121*.

Paisley, J., Blei, D., and Jordan, M. (2012). Variational bayesian inference with stochastic search. *arXiv preprint arXiv:1206.6430*.

Ranganath, R., Gerrish, S., and Blei, D. M. (2013). Black box variational inference. *arXiv preprint arXiv:1401.0118*.

Ranganath, R., Tang, L., Charlin, L., and Blei, D. M. (2015). Deep exponential families. In *AISTATS*.

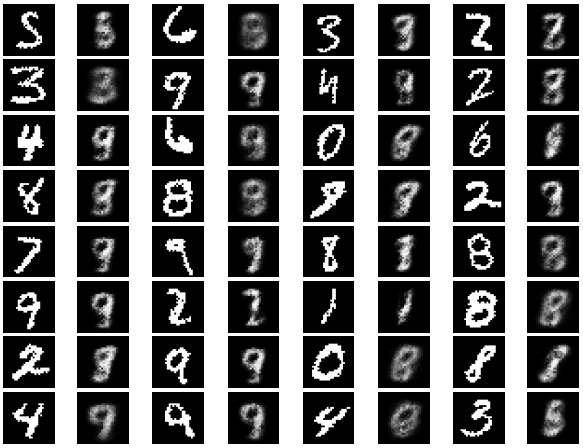
Salimans, T. (2014). Implementing and automating fixed-form variational posterior approximation through stochastic linear regression. *arXiv preprint arXiv:1401.2135*.

Salimans, T., Knowles, D. A., et al. (2013). Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882.

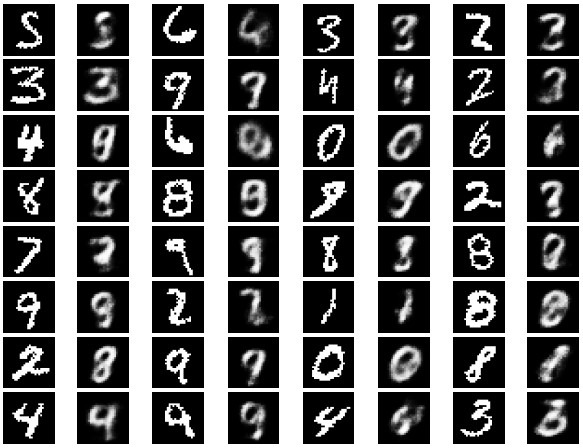
Titsias, M. and Lázaro-Gredilla, M. (2014). Doubly stochastic variational Bayes for non-conjugate inference. In *International Conference on Machine Learning*.

Titsias, M. and Lázaro-Gredilla, M. (2015). Local expectation gradients for black box variational inference. In *Advances in Neural Information Processing Systems*, pages 2638–2646.

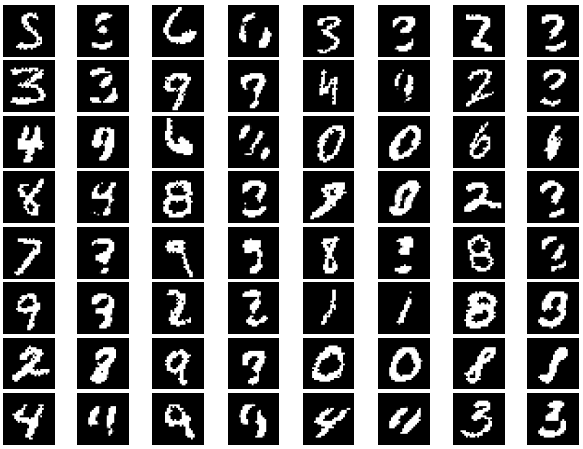
Winn, J. and Bishop, C. M. (2005). Variational message passing. *Journal of Machine Learning Research*, 6(Apr):661–694.



(a) Partial-BBVI



(b) LeGrad



(c) CVI

Figure 4: Reconstructions of a few test images. The odd numbered columns show the actual images and the even numbered columns show their reconstructions. We observe that CVI gives less blurry reconstructions.