Effective Monte Carlo Variational Inference for Binary-Variable Probabilistic Programs

Geng Ji
Computer Science Department, UC Irvine

Erik B. Sudderth
Computer Science Department, UC Irvine

Abstract

We propose a broadly applicable variational inference algorithm for probabilistic models with binary latent variables, using sampling to approximate expectations required for coordinate ascent updates. Applied to three real-world models for text and image and network data, our approach converges much faster than REINFORCE-style stochastic gradient algorithms, and requires fewer Monte Carlo samples. Compared to hand-crafted variational bounds with model-dependent auxiliary variables, our approach leads to tighter likelihood bounds and greater robustness to local optima. Our method is designed to integrate easily with probabilistic programming languages for effective, scalable, black-box variational inference.

1 Introduction

Variational inference is widely used to estimate the posterior distributions of hidden variables in probabilistic models (Wainwright and Jordan, 2008). Variational bounds are usually optimized via *coordinate ascent variational inference* (CAVI, Jordan et al. (1999)) algorithms which iteratively update single (or small blocks of) variational parameters. Although CAVI updates can be effective for simple models composed from conjugate priors (Blei et al., 2003), for many models the expectations required for exact CAVI updates are intractable: they may require complex integrals for continuous variables, or computation scaling exponentially with the number of dependent discrete variables.

Variational algorithms for models with non-conjugate conditionals have been derived via hand-crafted auxiliary variables that induce looser, but more tractable, bounds on the data log-likelihood (Jordan et al., 1999; Winn and Bishop, 2005). Such bounds typically require complex derivations specialized to the parametric structure of specific distributions (Jaakkola and Jordan, 1999; Gan et al., 2015), and thus do not easily integrate with general-purpose inference systems.

To address these limitations, Paisley et al. (2012), Wingate and Weber (2013) and Ranganath et al. (2014) have explored stochastic gradient algorithms that directly optimize a reparameterized bound involving the log-likelihood gradient or score function, as in the classic REINFORCE policy gradient (Williams, 1992). Due to its simplicity and generality, it has become the "standard" variational inference algorithm for a number of probabilistic programming languages including Edward and TensorFlow Probability (Tran et al., 2016, 2018), WebPPL (Goodman and Stuhlmüller, 2014; Ritchie et al., 2016), Pyro (Bingham et al., 2019), and Gen (Cusumano-Towner et al., 2019). REINFORCE has also been called *black box variational inference* (BBVI, (Ranganath et al., 2014)) because it can be applied to different probabilistic models without specialized derivations. We use these two terms interchangably. Unlike other black-box variational methods (Kucukelbir et al., 2017) that require specific variable reparameterizations (Kingma and Welling, 2014), BBVI provides unbiased gradients for all models including the many practically important models with discrete latent variables. But as we demonstrate in this paper, even for models of moderate size and using control variates, REINFORCE-based variational inference typically requires a large number of iterations (and Monte Carlo samples) for convergence due to its high-variance issue.

In this paper we analyze the poor convergence behavior of previous BBVI methods in more detail, and contrast it with a Monte Carlo variant of the classic CAVI algorithm. Our Monte Carlo CAVI updates have strong asymptotic guarantees (Ye et al., 2019) while showing good convergence behavior even when few samples are used; in experiments, BBVI typically requires about one hundred times more computation to infer posteriors of comparable quality. We demonstrate the potential of Monte Carlo CAVI for black-box inference by applying it to diverse models of text and image and network data. Dramatically, in addition to being easier to derive and implement, Monte Carlo CAVI updates are *superior* to previous hand-crafted variational inference algorithms in predictive accuracy and robustness to initialization.

2 Binary-Variable Models

We consider probabilistic models that generate observations x via binary latent variables z sampled from some joint distribution $p(z,x) = p(z)p(x \mid z)$. Specifically, we take three real-world models as running examples. The first one is deep noisy-OR Bayesian networks that have been used to capture topic interactions within documents (Murphy, 2012; Liu et al., 2016; Ji et al., 2019). Like the logical OR operator, the noisy-OR conditional distribution (Horvitz et al., 1988) assumes the state of a binary variable z_i is independently influenced by the state of each parent:

$$p(z_i = 0 \mid z_{\mathcal{P}(i)}) = \exp\Big(-\sum\nolimits_{k \in \mathcal{P}(i)} w_{k\to i} \cdot z_k\Big). \quad (1)$$

Similarly, observed word tokens x in the bottom layer are generated by their latent topic ancestors z via the noisy-OR relation as well, whose dependence can be determined by any directed acyclic graph.

The second type of models are sigmoid belief networks (Neal, 1992), which are layered binary generative models where the activation probability of each node is determined by the sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$. The activation $z_{i,j}$ of node j in layer i depends on the states of nodes in the preceding layer z_{i+1}:

$$p(z_{i,j} = 1 \mid z_{i+1}) = \sigma(w_{i,j}^T z_{i+1} + c_j). \quad (2)$$

Here, the possibly sparse weight vector $w_{i,j}$ determines which parents directly influence the activation of z_i. In our experiments, two layers of binary latent variables are used to generate pixel values x at the finest scale.

Finally, we consider a simplified version of the non-parametric relational model of Miller et al. (2009), where each entity i is described by a set of D hidden binary features $z_{id} \sim \text{Bernoulli}(\rho)$. The probability that undirected link x_{ij} between entities i and j is present depends on the number of shared features:

$$p(x_{ij} = 1 \mid z) = \Phi\Big(w_0 + \sum\nolimits_{d=1}^{D} w_d z_{id} z_{jd}\Big). \quad (3)$$

Here, Φ is the CDF of the standard normal distribution, or probit function. Real-valued weight w_d controls the change in link probability when entities share feature d, and $\Phi(w_0)$ is the (small) probability of link occurrence for entities that share no features.

```
 1 import torch
 2 from pyro import plate, sample
 3 from pyro.distributions import Bernoulli
 4
 5 class BN(torch.nn.Module):
 6   def __init__(self, params):
 7     super(BN, self).__init__()
 8     self.b, self.W1, self.c1, self.W2, self.c2 = params
 9     self.D_H2, self.D_H1 = self.W2.shape
10
11   @abstractmethod
12   def squash_fun(self, x):
13     raise NotImplementedError
14
15   def model(self, data):
16     dat_axis = plate('dat_axis', data.shape[0], dim=-2)
17     top_axis = plate('top_axis', self.D_H2, dim=-1)
18     mid_axis = plate('mid_axis', self.D_H1, dim=-1)
19     bot_axis = plate('bot_axis', data.shape[1], dim=-1)
20     with dat_axis, top_axis:
21       z_top = sample('z_top', Bernoulli(
22                      probs=self.squash_fun(self.b)))
23       wz_top = torch.matmul(z_top, self.W2) + self.c2
24     with dat_axis, mid_axis:
25       z_bot = sample('z_bot', Bernoulli(
26                      probs=self.squash_fun(wz_top)))
27       wz_bot = torch.matmul(z_bot, self.W1) + self.c1
28     with dat_axis, bot_axis:
29       sample('x', Bernoulli(
30              probs=self.squash_fun(wz_bot)), obs=data)
31
32 class NoisyOrBN(BN):
33   def squash_fun(self, x):
34     return torch.ones([]) - torch.exp(-x)
35
36 class SigmoidBN(BN):
37   def squash_fun(self, x):
38     return torch.sigmoid(x)
```

Figure 1: Pyro specification of three-layer Bayesian networks. By defining different squashing functions (line 33 and 37), the noisy-OR topic model and sigmoid belief network may be easily created from the abstract base class.

3 Variational Inference for PPLs

Probabilistic programming languages (PPLs) provide flexible but precise frameworks for defining probabilistic models, and performing inference queries given observed data. Fig. 1 shows the power of PPLs by defining noisy-OR topic networks and sigmoid belief networks with compact, integrated Pyro code. The grand promise of PPLs is that given a generative model specification, appropriate inference code can be automatically generated, enabling rapid model exploration even for non-expert users. In this section, we develop a Monte Carlo VI algorithm and show its effectiveness against the REINFORCE variational gradients, which is the standard variational algorithm for many PPLs.

3.1 Monte Carlo Coordinate Ascent VI

Exact posterior inference is intractable for models like those in Sec. 2 due to the combinatorial number of latent feature combinations. Mean field variational inference algorithms seek an approximate posterior $q(z)$ from a tractable family with simpler dependencies by maximizing the *evidence lower bound* (ELBO):

$$\mathcal{L}(q) = \mathbb{E}_{q(z)}\left[\log p(z,x) - \log q(z)\right] \leq p(x). \quad (4)$$

In this work, we make a "naive" mean-field approximation so that $q(z) = \prod_i q(z_i)$ is fully factorized. Coordinate ascent variational inference (CAVI, Jordan et al. (1999); Winn and Bishop (2005); Blei et al. (2017)) iteratively optimizes each factor of the variational density while holding all others fixed, producing iterations that monotonically increase the ELBO and converge to a (local) maximum.

Concretely, to update a variational factor $q(z_i)$, CAVI requires the *complete conditional* $p(z_i \mid z_{-i}, x)$ of z_i given all other latent variables z_{-i} and the observations x. For binary latent-variable models, each $q(z_i)$ is a Bernoulli distribution. The optimal value for its logit $\tau_i \triangleq \log \frac{q(z_i=1)}{q(z_i=0)}$ equals

$$\tau_i = \mathbb{E}_{q(z_{-i})}\left[\log \frac{p(z_i = 1 \mid z_{-i}, x)}{p(z_i = 0 \mid z_{-i}, x)}\right], \tag{5}$$

where the expectation is with respect to the current iteration's variational distributions $q(z_{-i}) = \prod_{j \neq i} q(z_j)$.

While CAVI provides a uniform way to optimize the ELBO, it is not computationally tractable because for non-conjugate conditionals like those in Eqs. (1,2,3), computing the expectations in Eq. (5) requires enumerating the exponentially many joint configurations of variables in the Markov blanket of z_i. We propose a Monte Carlo version of the classical CAVI algorithm, which uses sampling to approximate the expectations needed for optimal variational parameter updates:

$$\begin{aligned}\tau_i &\approx \frac{1}{M}\sum_{m=1}^{M} \log \frac{p(z_i = 1 \mid z_{-i}^{(m)}, x)}{p(z_i = 0 \mid z_{-i}^{(m)}, x)} \qquad (6)\\ &= \frac{1}{M}\sum_{m=1}^{M} \log \frac{p(z_i = 1 | u_{\mathcal{P}(i)}^{(m)}) \prod_{j \in \mathcal{C}(i)} p(u_j^{(m)} | z_i = 1, u_{\mathcal{P}(j)}^{(m)})}{p(z_i = 0 | u_{\mathcal{P}(i)}^{(m)}) \prod_{j \in \mathcal{C}(i)} p(u_j^{(m)} | z_i = 0, u_{\mathcal{P}(j)}^{(m)})},\end{aligned}$$

where $z_{-i}^{(m)}$ are samples drawn from $q(z_{-i})$, $u^{(m)} \triangleq \{z_{-i}^{(m)} \cup x\}$, and $\mathcal{C}(i)$ is the set of children of variable i. The second line of Eq. (6) shows more explicitly how the model structure is leveraged to avoid unnecessary computations with variables outside the Markov blanket. Importantly, the complexity of Monte Carlo CAVI is linear in the number of samples even for models with high-order dependencies.

3.2 Comparison with REINFORCE

The REINFORCE method optimizes Eq. (4) via stochastic gradient ascent, computing unbiased ELBO gradients via Monte Carlo samples $z^{(m)}$ drawn from $q(z)$. To avoid constraints, for binary latent variables, noisy gradient updates are parameterized via variational logits τ_i:

$$\frac{\partial \mathcal{L}}{\partial \tau_i} \approx \frac{1}{M}\sum_{m=1}^{M} \left.\frac{\partial \log q(z_i)}{\partial \tau_i}\right|_{z_i^{(m)}} \cdot \Big(\log p(z_i^{(m)} \mid z_{-i}^{(m)}, x) - \log q(z_i^{(m)})\Big). \tag{7}$$

Note that in Eq. (7), Rao Blackwellization is used to analytically marginalize variables outside the Markov blanket of the variable being updated, provably reducing variance (Ranganath et al., 2014).

Monte Carlo CAVI (CAVI for short) and REINFORCE (BBVI for short) are both general-purpose tools for inference. Both methods only require the ability to sample from the (binary) variational distribution and to evaluate (components of) the model log-probability. As for the speed comparison with BBVI, CAVI is twice as slow per iteration when the number of samples is the same. That's because CAVI evaluates the log densities in Eq. (6) twice, assigning z_i to bothg 0 and 1, while BBVI only needs one of them in Eq. (7), depending on the value of sample $z_i^{(m)}$.

But as shown in the toy example in Fig. 2, BBVI requires much more iterations to converge to the same optima than CAVI, even under the best learning rate

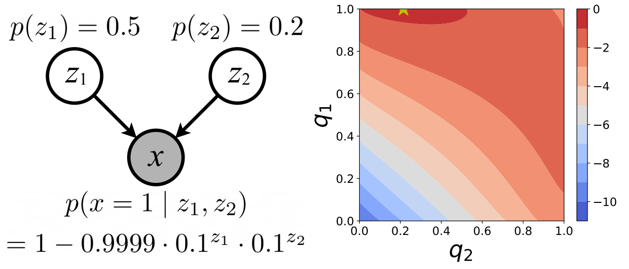


Figure 2: *Left:* Graphical illustration of a toy noisy-OR model with two latent nodes. *Right:* Contour plot of the toy model's ELBO as a function of the variational parameters $q_1 \triangleq q(z_1 = 1)$ and $q_2 \triangleq q(z_2 = 2)$, when the observation $x = 1$. The yellow star indicates the global optimum.

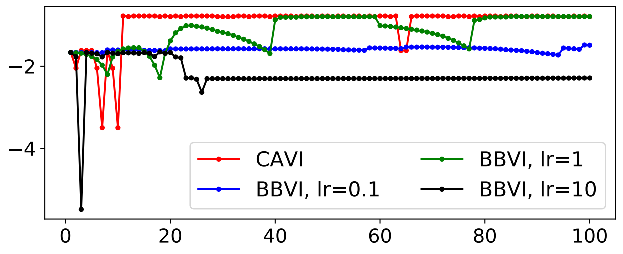


Figure 3: ELBO (y axis) over iterations (x axis) on the toy model. The initial values $q_1 = 0.5, q_2 = 0.9$ are selected intentionally from the low-probability region in Fig. 5. As expected, the ELBO of CAVI fluctuates in the first few iterations, but then quickly moves to the optimum (red). In contrast, the ELBO of BBVI updates drops frequently across iterations even under the best learning rate (green). The number of samples used for each method is 2.

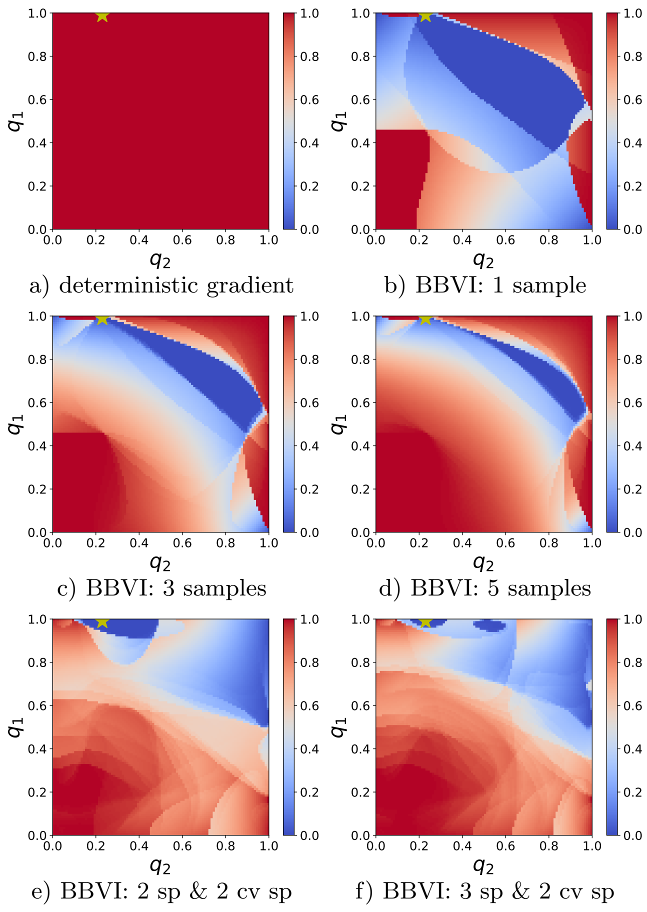


Figure 4: The probability that a gradient update of q would increase the ELBO. Each point of the plots indicates the current value of q_1 and q_2. a) the true gradient; b-d) niosy gradients estimated with various numbers of samples; e-f) noisy gradients with extra samples used to estimate the baseline control variate (cv). The learning rate is 1.

tuned (Fig. 3). That is because comparing to the analytic gradient that always increases the ELBO, the randomness caused by Monte Carlo sampling in BBVI may change the variational parameters in wrong directions, especially in areas around the global optimum (Fig. 4). Note that adding the control variate is not able to entirely solve this problem.

On the other hand, Fig. 5 shows CAVI behaves better than BBVI under the same sampling budgets, in the sense that most areas have higher probability of ELBO improvement. The low-probability regions are much farther away from the global optimum comparing to the ones in BBVI. More importantly, unlike the gradient-based BBVI method that has to follow the gradient directions step by step, CAVI does not need to tune the learning rate, and changes the variational factor of each variable directly to the optimal point. Because of this, even if q is initialized in the blue region unluckily, CAVI is able to escape from it in just a

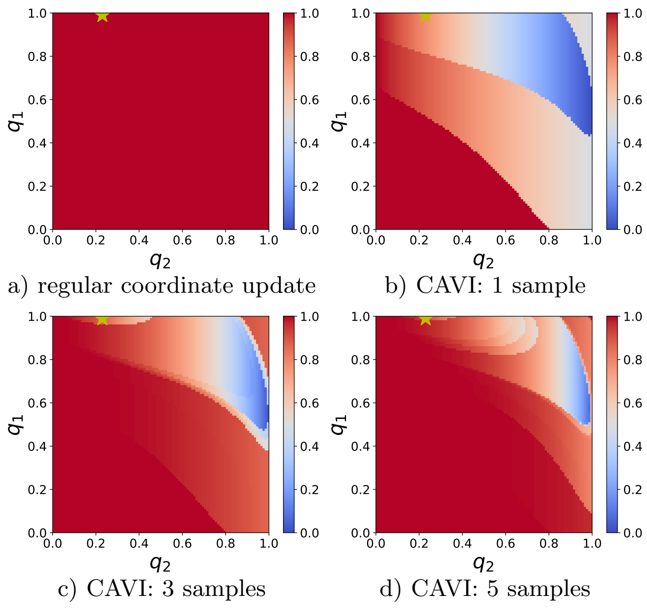


Figure 5: The probability that a coordinate update of q would increase the ELBO. a) the regular coordinate ascent update where expectations are computed numerically; b-d) Monte Carlo CAVI using various numbers of samples.

couple of iterations and rapidly converge to the global optimum, as illustrated in the red curve in Fig. 3.

4 Experiments

We compare the proposed algorithm with BBVI and auxiliary-variable coordinate methods on the three models described in Sec. 2. We use the same datasets as in the original papers, and evaluate their average test ELBOs using Monte Carlo sampling. We find that our Monte Carlo CAVI method has multiple appealing advantages over the baseline approaches.

4.1 Text Data and Noisy-OR Relations

As in Ji et al. (2019), we use the tiny 20 Newsgroups dataset to test the inference performance, and follow the same model structure that has 44 latent topic nodes spanned in two layers, and 100 observed token nodes. The edge weights are fixed to the values learned through the variational training algorithm of the paper, without the local model pruning.

4.1.1 Faster Convergence than BBVI

We compare the performance of different variational methods on the test set, which contains 4,872 documents in total. The variational distribution q is all initialized to 0.5. The trace plot of average ELBO is shown in Fig. 6(a). Similar to the auxiliary-variable coordinate algorithm in Ji et al. (2019), CAVI con-

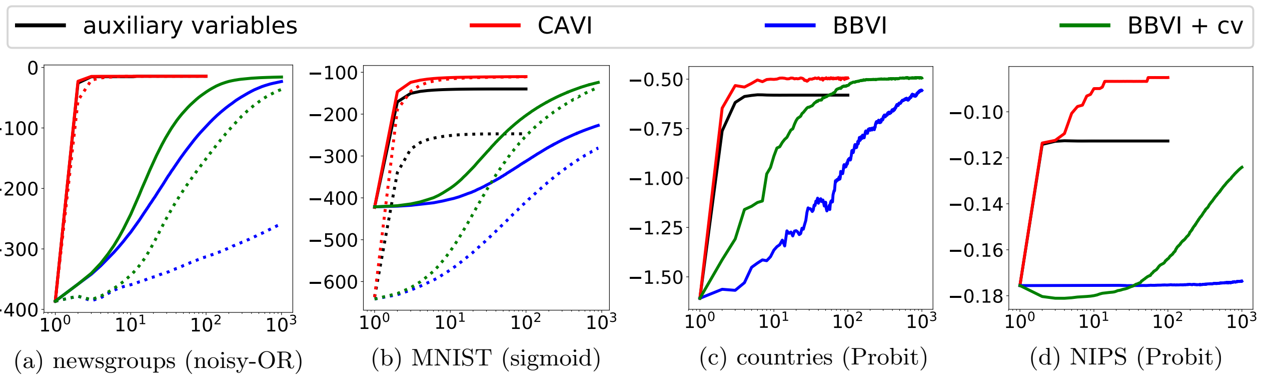


Figure 6: Improvement of average test ELBO (y axis) over iterations (x axis) on four different datasets. Unless specified otherwise, we use 10 samples for both CAVI and BBVI. Our CAVI algorithm always converges to values higher than or similar to those of the model-dependent auxiliary-variable methods. It also converges in a speed orders of magnitude faster than BBVI, whose learning rate has been tuned for the best performance on each dataset. (a): CAVI behaves more robustly than BBVI when the sampling budget drops from 10 (solid) to 2 (dotted). (b): CAVI behaves more robustly than the auxiliary-variable method when the initialization changes from the marginal prior for each node (solid) to 0.5 (dotted). (c): Only on this very small dataset, BBVI with control variate (decaying average baseline) converges within 1,000 iterations. (d): CAVI is clearly better than the other methods on this larger network dataset.

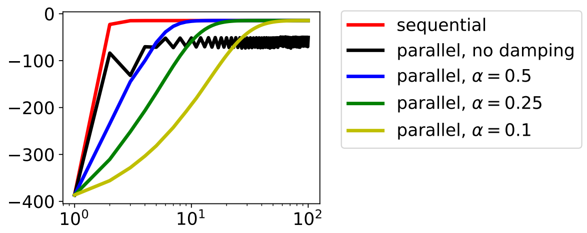


Figure 7: While the direct parallelization of CAVI (black) fails to converge, trails with damping all end up in good local optima similar to the sequential CAVI update (red). Larger damping rate α helps to converge faster.

verges in about 10 iterations. The ELBO of the two methods at convergence are also very close (CAVI -14.51 v.s auxiliary variables -14.53). When the sample size drops from 10 to 2, the convergence speed of CAVI only slows down slightly.

On the contrary, BBVI is still far from convergence even after 1,000 iterations, just getting an average test ELBO of -15.76 with the control variate, and -21.22 without. We believe the slow ELBO improvement of BBVI is largely due to the low probability area around the optimum point shown in the toy example of Fig. 4, especially for the last few hundreds of iterations. In addition, the dotted lines show that BBVI is much more vulnerable to the change in sample size.

4.1.2 Parallel CAVI Updates via Damping

Coordinate algorithms update the parameters one at a time, while holding all the others fixed. Comparing to gradient-based methods that change all the parameters together, this sequential setup naturally prohibits CAVI from being embarrassingly parallelized. Similar to Sun et al. (2013), we find reliable parallelization can be achieved through damping. As shown in Eq. (8), the damping update sets the vector of logits τ at iteration $t+1$ as a linear combination of its value in the previous iteration t, and the parallel coordinate update for all variables:

$$\tau_{t+1} = (1-\alpha)\cdot\tau_t + \alpha\cdot\tau_{\text{parallel}}. \tag{8}$$

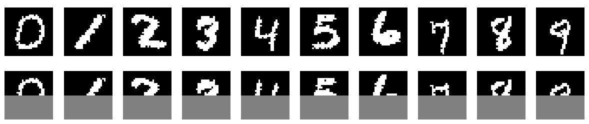
The parallel updates across all dimensions share the same set of Monte Carlo samples. Fig. 7 illustrates that without damping, the ELBO of the parallel update (black line) oscillates and never converges. Damping helps avoid this problem, as shown by the blue, green and yellow curves. We find that the damping rate α generally does not affect the ELBO at convergence, but just slows down the convergence speed slightly, as shown in Fig. 7.

4.2 Image Data and Sigmoid Relations

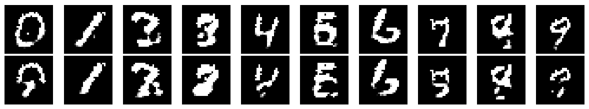
Following Gan et al. (2015), we build a fully-connected network with three layers. The two layers at the top have 100 nodes each, and the observed layer at bottom corresponds to the binarized images. We use the test set of MNIST, which contains 10,000 images each with 28×28 pixels. Edge weights of the network are learned from the training set through the public code of Gan et al. (2015) for coordinate-ascent variational training using the Pólya-Gamma trick.

4.2.1 Robustness to Bad Initializations

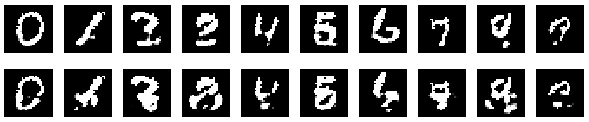
As presented in Fig. 6(b), similar conclusions can be drawn for this model. Moreover, the auxiliary variable method (Gan et al., 2015) performs very badly when q



Input: bottom parts of the images (top row) are missing.



Prior marginal init: CAVI (top) v.s BBVI (bottom).



0.5 init: CAVI (top) v.s BBVI (bottom).

Figure 8: Examples of MNIST digit completion. CAVI works better than BBVI under both initializations.

is uniformly initialized to 0.5 (dotted black line). The result improves if we initialize q to be the marginal prior of each node (solid black line), obtained via a Monte Carlo estimate over one million samples.

In contrast, CAVI is not as sensitive to initialization. As shown by the red lines in Fig. 6(b), CAVI performs better than the auxiliary-variable method under both initialization strategies. Examples of digit completion in Fig. 8 also illustrate this difference clearly.

We believe the reasons for the performance difference between CAVI and the auxiliary-variable method are two-fold. First, the auxiliary-variable objective is a lower bound of the ELBO, so it is expected that the result will be worse than CAVI, which optimizes the ELBO directly. Second, with more latent variables added in, the optimization surface becomes more complicated, so the data-augmented algorithm gets stuck in bad local optima more easily. We find under repeated trials with random variable update orders, the variance of the ELBOs at convergence is much larger for the auxiliary-variable method than CAVI.

4.3 Link Data and Probit Relations

We test the performance on two datasets from the original paper of Miller et al. (2009). The first one is the country dataset, which describes various relations (such as “accusation” and “economic aid”) between 14 countries during 1950 to 1965 (Rummel, 1976). In particular, we use the “conference” relation, which consists of symmetric connections indicating if two countries co-participate in any international conference. We set $D = 4$, and model parameters $w_d = 2, w_0 = -2, \rho = 0.5$ are selected through grid search. The features are initialized as the prior value ρ. As shown in Fig. 6(c), on this very small model, BBVI with control variate finally reaches the same performance of CAVI after using over 10 times more iterations. For a baseline comparison, we derive an auxiliary-variable variational method based on the trick of Probits from thresholded Gaussians (Albert and Chib, 1993). See the appendix for full details.

The second relational dataset is the NIPS coauthorship data by Globerson et al. (2007), where a link indicates two individuals being coauthors of a paper in one of the first 17 NIPS conferences. Following Miller et al. (2009) and Palla et al. (2012), we pick the 234 most connected authors, and set $D = 10, w_d = 2, w_0 = -2, \rho = 0.1$. On this larger dataset, the advantage of CAVI over the auxiliary variable method and BBVI is very obvious, as shown in Fig. 6(d).

5 Discussion

We have developed a Monte Carlo variational inference framework applicable to any probabilistic model with binary latent variables. The proposed method converges much faster than BBVI, and is less sensitive to the sample size and initialization. Relative to model-specific auxiliary bounds, our Monte Carlo CAVI algorithm directly optimizes a tighter likelihood bound, and is more robust to initialization in spite of being simpler to derive and implement.

While this submission focused on models with binary variables for simplicity, it is straightforward to generalize the Monte Carlo update of Eq. (6) to general discrete variables with more domain values v:

$$\begin{aligned} q(z_i = v) &\propto \exp\Big\{\frac{1}{M}\sum_{m=1}^{M} \log p\Big(z_i = v \mid z_{-i}^{(m)}, x\Big)\Big\} \quad (9) \\ &= \exp\Big\{\frac{1}{M}\sum_{m=1}^{M} \log p\Big(z_i = v \mid u_{\mathcal{P}(i)}^{(m)}\Big) \\ &\qquad + \sum_{j\in\mathcal{C}(i)} \log p\Big(u_j^{(m)} \mid z_i = v, u_{\mathcal{P}(j)}^{(m)}\Big)\Big\}. \end{aligned}$$

Note that computational complexity grows linearly with the number of discrete states.

We are exploring applications to other models with non-binary discrete variables, and integrating Monte Carlo CAVI updates into PPLs like Gen. We believe it will provide a compelling alternative to previous black-box variational methods as a scalable inference engine for probabilistic programs.

Acknowledgments. This research supported in part by NSF RI Award No. IIS-1816365, and a Facebook Probability and Programming research award.

References

Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679.

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., and Goodman, N. D. (2019). Pyro: Deep universal probabilistic programming. *Journal of Machine Learning Research*, 20(1):973–978.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Cusumano-Towner, M. F., Saad, F. A., Lew, A. K., and Mansinghka, V. K. (2019). Gen: A general-purpose probabilistic programming system with programmable inference. In *Conference on Programming Language Design and Implementation*, pages 221–236.

Gan, Z., Henao, R., Carlson, D., and Carin, L. (2015). Learning deep sigmoid belief networks with data augmentation. In *Artificial Intelligence and Statistics*, pages 268–276.

Globerson, A., Chechik, G., Pereira, F., and Tishby, N. (2007). Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8:2265–2295.

Goodman, N. D. and Stuhlmüller, A. (2014). The Design and Implementation of Probabilistic Programming Languages. http://dippl.org.

Horvitz, E. J., Breese, J. S., and Henrion, M. (1988). Decision theory in expert systems and artificial intelligence. *International Journal of Approximate Reasoning*, 2(3):247–302.

Jaakkola, T. S. and Jordan, M. I. (1999). Variational probabilistic inference and the QMR-DT network. *Journal of Artificial Intelligence Research*, 10:291–322.

Ji, G., Cheng, D., Ning, H., Yuan, C., Zhou, H., Xiong, L., and Sudderth, E. B. (2019). Variational training for large-scale noisy-OR Bayesian networks. In *Conference on Uncertainty in Artificial Intelligence*.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In *International Conference on Learning Representations*.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2017). Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18(1):430–474.

Liu, J., Ren, X., Shang, J., Cassidy, T., Voss, C. R., and Han, J. (2016). Representing documents via latent keyphrase inference. In *International Conference on World Wide Web*, pages 1057–1067.

Miller, K., Jordan, M. I., and Griffiths, T. L. (2009). Nonparametric latent feature models for link prediction. In *Advances in Neural Information Processing Systems*, pages 1276–1284.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Neal, R. M. (1992). Connectionist learning of belief networks. *Artificial Intelligence*, 56(1):71–113.

Paisley, J. W., Blei, D. M., and Jordan, M. I. (2012). Variational Bayesian inference with stochastic search. In *International Conference on Machine Learning*.

Palla, K., Knowles, D. A., and Ghahramani, Z. (2012). An infinite latent attribute model for network data. In *International Conference on Machine Learning*.

Ranganath, R., Gerrish, S., and Blei, D. M. (2014). Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822.

Ritchie, D., Horsfall, P., and Goodman, N. D. (2016). Deep amortized inference for probabilistic programs. *arXiv preprint arXiv:1610.05735*.

Rummel, R. J. (1976). *Attributes of nations and behavior of nation dyads, 1950-1965*. Inter-university Consortium for Political Research.

Sun, D., Wulff, J., Sudderth, E. B., Pfister, H., and Black, M. J. (2013). A fully-connected layered model of foreground and background flow. In *Conference on Computer Vision and Pattern Recognition*, pages 2451–2458.

Tran, D., Hoffman, M. W., Moore, D., Suter, C., Vasudevan, S., and Radul, A. (2018). Simple, distributed, and accelerated probabilistic programming. In *Advances in Neural Information Processing Systems*, pages 7598–7609.

Tran, D., Kucukelbir, A., Dieng, A. B., Rudolph, M., Liang, D., and Blei, D. M. (2016). Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256.

Wingate, D. and Weber, T. (2013). Automated variational inference in probabilistic programming. *arXiv preprint arXiv:1301.1299*.

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

Ye, L., Beskos, A., De Iorio, M., and Hao, J. (2019). On the efficacy of monte carlo implementation of cavi. *arXiv preprint arXiv:1905.03760*.