Proceedings of Machine Learning Research 101:1033–1048, 2019 ACML 2019

Trust Region Sequential Variational Inference

Geon-Hyeong Kim[2] GHKIM@AI.KAIST.AC.KR
Youngsoo Jang[2] YSJANG@AI.KAIST.AC.KR
Jongmin Lee[2] JMLEE@AI.KAIST.AC.KR
Wonseok Jeon[3] JEONWONS@MILA.QUEBEC
Hongseok Yang[2] HONGSEOK.YANG@KAIST.AC.KR
Kee-Eung Kim[1,2] KEKIM@KAIST.AC.KR
[1] *Graduate School of AI, KAIST, Daejeon, Republic of Korea*
[2] *School of Computing, KAIST, Republic of Korea*
[3] *MILA, McGill University, Canada*

Editors: Wee Sun Lee and Taiji Suzuki

Abstract

Stochastic variational inference has emerged as an effective method for performing inference on or learning complex models for data. Yet, one of the challenges in stochastic variational inference is handling high-dimensional data, such as sequential data, and models with non-differentiable densities caused by, for instance, the use of discrete latent variables. In such cases, it is challenging to control the variance of the gradient estimator used in stochastic variational inference, while low variance is often one of the key properties needed for successful inference. In this work, we present a new algorithm for stochastic variational inference of sequential models which trades off bias for variance to tackle this challenge effectively. Our algorithm is inspired by variance reduction techniques in reinforcement learning, yet it uniquely adopts their key ideas in the context of stochastic variational inference. We demonstrate the effectiveness of our approach through formal analysis and experiments on synthetic and real-world datasets.

Keywords: sequential variational inference, black-box variational inference

1. Introduction

One of the recent trends in machine learning is to learn a sophisticated probabilistic model from complex data. Such a model is typically built on top of deep neural network, and the learning commonly involves approximating the model's intractable posterior. Computing a good approximate posterior efficiently has been the aim of many research projects, which led to a wide variety of approaches, such as stochastic variational inference when the model yields explicit likelihood densities (Kingma and Welling, 2014; Hoffman et al., 2013; Burda et al., 2016) and game-theoretic framework when likelihoods are only implicit (Goodfellow et al., 2014).

Our goal is to advance the state of the art of generic stochastic variational inference for sequential data. In particular, we want to have an inference algorithm that can analyze efficiently models with non-differentiable densities, such as those using discrete latent variabmles. In fact, tackling such non-differentiable models or the ones with discrete latent variables has been an active research topic in the variational-inference community. In the

past few years, different types of techniques have been developed. Some focused on reducing the variance of existing gradient estimator with clever control variate (Grathwohl et al., 2018; Tucker et al., 2017), while others focused on relaxing discrete random variables to differentiable continuous ones (Maddison et al., 2017b; Jang et al., 2017) or removing the smoothness assumption from existing techniques (Lee et al., 2018). Although significant progress has been made, experts agree that discrete latent variables and non-differentible densities are still stumbling blocks for stochastic variational inference. This sentiment is consistent with the experiments reported later in this paper.

In the paper, we present a new algorithm for black-box variational inference for sequential data. Our algorithm is designed to handle both of continuous latent variables and discrete latent variables effectively. It is based on three ideas, all coming from the reinforcement-learning literature.

The first idea is to separate out a problematic part of the standard ELBO objective for state space models, and to approximate this part using a separate optimization process. Intuitively, the part computes a version of the ELBO from a state at time step $t > 1$, instead of the usual $t = 1$. We call it *future ELBO*. The future ELBOs at different time steps satisfy recursive equations, which our algorithm uses to compute their approximations. These approximate future ELBOs are used during the estimation of the gradient of the ELBO, and help reduce the variance of the estimator. The future ELBOs correspond to critics in the actor-critic method of reinforcement learning. Briefly, we may consider that actor-critic is combining REINFORCE algorithm with approximating future information, especially value function in reinforcement learning and in our case, it corresponds to future ELBO.

The second idea is to use so called trust region. Our variational-inference algorithm computes approximate future ELBOs for a variational distribution q_{ϕ_0}, but uses them to estimate the ELBO and its gradient at a different variational distribution q_{ϕ_1}. To manage the potential harm of this seemingly incorrect use of these future ELBOs, our algorithm uses trust region and limits the application of the future ELBOs to variational distributions sufficiently close to q_{ϕ_0}.

The third is natural gradient, which is derived from a second-order approximation of trust region constraint. This means that when the algorithm computes the direction of changing the variational distribution q_ϕ, it uses a notion of distance or divergence inherent to distributions themselves, not the choice of their parameterization. Natural gradients have been used in the context of variational inference previously (Hoffman et al., 2013; Regier et al., 2017).

We point out that while all of these ideas are inspired by policy gradient methods in reinforcement learning, in particular, by TRPO (Schulman et al., 2015), their concrete implementation in our algorithm is unique. While most of reinforcement learning algorithms strive for sample efficiency due to the high cost of generating samples, our algorithm does not do so. This is an acceptable option because generating samples is significantly cheaper in stochastic variational inference than in reinforcement learning. This in turn provides simplicity over TRPO in terms of formal analysis and implementation.

Our algorithm has the theoretical property that in an ideal setting, each update to the variational distribution improves its ELBO. Also, our experiments with time-series models

and discrete latent variables show that the algorithm outperforms the relaxation based on concrete distribution, one of the best techniques for this kind of problem.

2. Background

We briefly review the basics of stochastic variational inference that needed to follow our paper.

2.1. Variational Inference

Consider a probabilistic model defined by a density function p over random variables $z \in \mathcal{Z} \subseteq \mathbb{R}^M$ and $x \in \mathcal{X} \subseteq \mathbb{R}^L$. Assume that x is observed. We would like to compute the posterior distribution of latent z under observed x. Unfortunately, in most cases, it is not possible to compute the exact posterior. This difficulty led to the development of multiple approaches for approximating the posterior, in particular, variational inference which phrases approximate posterior inference as optimization. A variational inference algorithm assumes a family of approximating distributions $\{q_\phi(z;x)\}_\phi$, and sets up an optimization problem for q_ϕ by picking an objective function that intuitively measures the approximation quality of $q_\phi(z;x)$ to the posterior $p(z|x)$ for a given x. A recent popular choice is stochastic gradient ascent, which optimizes ϕ by following a sample-based estimate of the gradient of the objective.

A common optimization objective is evidence lower bound (ELBO), given by

$$\mathcal{L}(\phi;x) = \mathbb{E}_{z\sim q_\phi(\_;x)}\left[\log \frac{p(x,z)}{q_\phi(z;x)}\right]. \tag{1}$$

By Jensen's inequality, $\log p(x) \geq \mathcal{L}(\phi;x)$, from which the name ELBO came. The inequality becomes equality precisely when $q_\phi(z;x)$ is $p(z|x)$, the posterior itself.

To perform stochastic gradient ascent, two well-known algorithms, score estimator (referred to as REINFORCE (Williams, 1992) and likelihood ratio estimator (Glynn, 1990)) and reparameterization estimator (also known as pathwise estimator) (Kingma and Welling, 2014), are widely used. The reparameterization estimator generally shows better performance than the score estimator in terms of variance. However, it has a limited scope of applicability. In particular, it has the difficulty in handling models with discrete latent variables. Overcoming this limitation is still an active research topic, with multiple proposals such as the handling of discrete latent variables via the Gumbel-softmax trick (Maddison et al., 2017b; Jang et al., 2017) and the addition of a correction term for non-differentiable boundaries to the estimator (Lee et al., 2018).

2.2. State Space Model

The state space model is a standard representation for dynamics behind sequential data. It assumes a latent variable z formed by a random sequence $z_1, \ldots, z_T$ in $\mathbb{R}^M$, and an observed variable x of a similar sequence form: $x = (x_1, \ldots, x_T)$ with each x_t having a value in $\mathbb{R}^L$. Then, the joint density of x and z is represented by initial joint density $p(x_1, z_1)$ and joint

transition density $p(x_{t+1}, z_{t+1}|x_1, \ldots, x_t, z_1, \ldots, z_t)$ as follows:

$$p(x, z) = p(x_1, z_1) \prod_{t=2}^{T} p(x_t, z_t|x_{1:t-1}, z_{1:t-1}) \tag{2}$$

Here we use the subscript notation $x_{i:j}$ to denote the subsequence $(x_i, x_{i+1}, \ldots, x_j)$. Note that this setup permits time-dependent transition densities on x_t and z_t.

Variational inference is frequently employed for approximating the posterior $p(z|x)$ of the state space model. A popular choice of an approximating distribution $q_\phi(z; x)$ in this case is the one with the following factozation:

$$q_\phi(z; x) = q_\phi(z_1; x) \prod_{t=2}^{T} q_\phi(z_t|z_{1:t-1}; x). \tag{3}$$

3. Trust Region Variational Inference with Approximating Future ELBO

We now present our algorithm for performing stochastic gradient ascent. Our algorithm uses following key ingredients: (i) surrogate objective approximating the ELBO with a variant of the actor-critic method; (ii) handling the potential incorrectness caused from future ELBO approximation through trust region; (iii) better gradient direction using natural gradient. We explain these ingredients one at a time, and show how the original ELBO optimization gets gradually transformed to our algorithm. We also show that when this problem is solved exactly, the iteration of our algorithm improves the ELBO, even though the algorithm is derived from our surrogate objective.

3.1. Variance Reduction via Function Approximation

The first ingredient of our algorithm is a particular type of approximation of the ELBO inspired by the algorithms in reinforcement learning. In order to achieve this, we first reformulate the ELBO by identifying recurrent terms that arise in state space models.

Consider the following ELBO for the state space model, defined in (2) and (3):

$$\mathcal{L}(\phi; x) = \mathbb{E}_{z \sim q_\phi(\_; x)} \left[\sum_{t=1}^{T} \log \frac{p(x_t, z_t|x_{1:t-1}, z_{1:t-1})}{q_\phi(z_t|z_{1:t-1}; x)} \right]. \tag{4}$$

When the target density p is not smooth, the update of the parameter ϕ is often done using the score estimator of the gradient of ELBO.

For the state space model, the gradient of ELBO has the following form:

$$\nabla_\phi \mathcal{L}(\phi; x) = \mathbb{E}_{z \sim q_\phi(\_; x)} \left[\sum_{t=1}^{T} \nabla \log q_\phi(z_t|z_{1:t-1}; x) \sum_{t'=t}^{T} \log \frac{p(x_{t'}, z_{t'}|x_{1:t'-1}, z_{1:t'-1})}{q_\phi(z_{t'}|z_{1:t'-1}; x)} \right] \tag{5}$$

We can further reformulate the gradient by introducing the *future ELBO* $\Gamma_\phi(z_{1:t})$ given the prefix of latent variables $z_{1:t}$, $0 \leq t \leq T$:

$$\begin{aligned}
\Gamma_\phi(z_{1:0}) &= \mathcal{L}(\phi), \\
\Gamma_\phi(z_{1:t}) &= \mathbb{E}_{z_{t+1:T} \sim q_\phi(_;x)}\left[\sum_{t'=t}^{T-1} \log \frac{p(x_{t'+1}, z_{t'+1}|x_{1:t'}, z_{1:t'})}{q_\phi(z_{t'+1}|z_{1:t'}; x)}\right], \\
\Gamma_\phi(z_{1:T}) &= 0.
\end{aligned}$$

Using the future ELBOs and their temporal difference Δ_ϕ given by

$$\Delta_\phi^\gamma(z_{1:t}) = \log \frac{p(x_t, z_t|x_{1:t-1}, z_{1:t-1})}{q_\phi(z_t|z_{1:t-1}; x)} + \gamma\Gamma_\phi(z_{1:t}) - \Gamma_\phi(z_{1:t-1})$$

and $\Delta_\phi(z_{1:t})$ denotes $\Delta_\phi^1(z_{1:t})$. The gradient of ELBO in (5) can be simplified as follows (the derivation is provided in Lemma 1 in the Appendix A.):

$$\nabla_\phi \mathcal{L}(\phi; x) = \mathbb{E}_{z \sim q_\phi(\_;x)}\left[\sum_{t=1}^{T} \nabla_\phi \log q_\phi(z_t|z_{1:t-1}; x)\Delta_\phi(z_{1:t})\right].$$

If we use a naive Monte-Carlo estimation of the future ELBOs $\Gamma_\phi(z_{1:t})$, the algorithm readily reduces to using REINFORCE (Williams, 1992) for the variational inference task. However, this approach would exhibit too much variance mostly due to its sequential nature, and thus impractical for all but very simple models. Ideally, we would like to develop a low-variance unbiased estimator for the future ELBOs so that we can reduce the variance of the ELBO gradient while incurring no bias.

The approach we propose in this paper is to train an expressive function approximator with parameter w (e.g. neural networks) to approximate the future ELBOs

$$\hat{\Gamma}^w(z_{1:t}) \approx \Gamma_\phi(z_{1:t})$$

without incurring too much bias, so that we can improve the overall perfomance of stochastic variational inference. In the later sections, we will experimentally show that it is advantageous to introduce even a significant amount of bias to reduce the variance of the ELBO gradient. Careful readers would notice that this approach is reminiscent of actor-critic methods in reinforcement learning.

3.2. Monotonic Improvement

In order to further improve stochastic gradient ascent, we can perform line search along the gradient. To be specific, at iteration k, given the ELBO gradient $\nabla_\phi \mathcal{L}(\phi; x)|_{\phi_k}$ computed at ϕ_k, we seek to find optimal step size α^* that maximizes the ELBO at each iteration,

$$\alpha^* = \operatorname*{argmax}_\alpha \mathcal{L}(\phi'_\alpha; x)$$

where $\phi'_\alpha = \phi_k + \alpha\nabla_\phi \mathcal{L}(\phi; x)|_{\phi_k}$, and set $\phi_{k+1} \leftarrow \phi'_{\alpha^*}$. One of the major challenges in this approach is that it would require multiple evaluations of ELBO during line search. If

we use the naive Monte-Carlo estimator for (4), the variance will be too large to obtain α^* reliably. We would like to reduce the variance of ELBO estimation, ideally reusing the function approximator $\hat{\Gamma}^w$ introduced in the previous section. Note that we cannot simply take the prediction from $\hat{\Gamma}^w$ as ELBO estimation since it is trained for ϕ_k, not ϕ'_α

In order to achieve this, we introduce the following surrogate objective:

$$\widetilde{\mathcal{L}}(\phi', \phi_k; x) = \mathcal{L}(\phi_k; x) + \mathbb{E}_{z \sim q_{\phi'}(\_;x)} \left[\sum_{t=1}^{T} \Delta_{\phi_k}(z_{1:t}) \right].$$

The above formula suggests that we can just focus on evaluating the second term for each ϕ' during line search, using the prediction from the function approximator $\hat{\Gamma}^w$ trained for ϕ_k.

The surrogate objective is a first-order approximation to the ELBO at ϕ_k, in the sense that

$$\widetilde{\mathcal{L}}(\phi', \phi_k; x)|_{\phi'=\phi_k} = \mathcal{L}(\phi_k; x)$$
$$\nabla_{\phi'} \widetilde{\mathcal{L}}(\phi', \phi_k; x)|_{\phi'=\phi_k} = \nabla_{\phi} \mathcal{L}(\phi; x)|_{\phi=\phi_k}$$

Nonetheless, the surrogate objective is destined to be inaccurate as we increase the distance between ϕ' and ϕ_k. Hence, we perform search only in the neighborhood of ϕ_k defined in terms of KL divergence:

$$\begin{aligned} &\text{maximize } \widetilde{\mathcal{L}}(\phi', \phi_k; x) \\ &\text{subject to } D_{\mathrm{KL}}(q_{\phi'}(z; x) \| q_{\phi_k}(z; x)) \leq \delta, \end{aligned} \tag{6}$$

with the constant δ specifying the radius of the neighborhood region. This constrained optimization also suggests that we should follow the direction of the natural gradient (Amari, 1998; Hoffman et al., 2013; Schulman et al., 2015; Regier et al., 2017), which can be obtained by multiplying the inverse of Fisher information matrix to the plain gradient. The overall algorithm is shown in Algorithm 1.

We can provide a formal guarantee on the monotonic improvement of policies found in each iteration.

Theorem 1. *Given the parameter ϕ_k at iteration k, assume that Δ_{ϕ_k} is bounded. Then, for any ϕ with $\widetilde{\mathcal{L}}(\phi, \phi_k; x) > \mathcal{L}(\phi_k; x)$, there exists $\delta > 0$ such that*

$$\mathcal{L}(\phi; x) \geq \mathcal{L}(\phi_k; x).$$

Careful readers would again notice that Algorithm 1 and the above theoretical result are reminiscent of Trust-Region Policy Optimization (TRPO) (Schulman et al., 2015) in reinforcement learning. However, our algorithm takes advantage of the fact that we can freely sample latent variable z during line search. In reinforcement learning, this amounts to executing every policy encountered during line search, which would be out of the question. Thus, TRPO relies on importance sampling to evaluate policies on the same batch of samples, while our algorithm uses the direct Monte-Carlo method. This leads us to simpler theoretical analysis and more robust experimental results compared to TRPO in reinforcement learning. In Appendix C, we analyze this difference from a theoretical point of view.

Algorithm 1 Trust Region Sequential Variational Inference

Input: $x,\ p(x,z),\ \delta$.

Initialize ϕ_0, α, and $\hat{\Delta}$.
for $k = 0, 1, \ldots$ **do**
  Compute natural gradient $H^{-1}g$, where H is Fisher information matrix and g is gradient $\nabla_\phi \widetilde{\mathcal{L}}(\hat{\phi}, \phi_k; x)$.
  Compute parameter $\hat{\phi} = \phi_k + \alpha H^{-1} g$.
  if $\widetilde{\mathcal{L}}(\hat{\phi}, \phi_k; x) > \mathcal{L}(\phi_k; x)$ and $D_{\mathrm{KL}}(q_{\hat{\phi}}(\_; x) \| q_{\phi_k}(\_; x)) \leq \delta$ **then**
    Set $\phi_{k+1} \leftarrow \hat{\phi}$.
  else
    Adjust α and break.
  end if
end for

4. Related Work

The reparameterization estimator (Kingma and Welling, 2014) is a technique of choice for controlling high variance during the estimation of the gradient in stochastic variational inference. It is known to perform better than the score estimator, also known as REINFORCE (Williams, 1992). However, the reparameterization estimator has a limited scope of applicability, because it works only for models with differentiable densities. In recent years, there are studies to overcome this limitation and make the trick apply for models with discrete latent variables or more generally models with non-differentiable densities. Some tried to relax the discrete variables and apply this trick to the relaxed models (Maddison et al., 2017b; Jang et al., 2017), while others developed clever control variate to REINFORCE so as to reduce its variance (Mnih and Gregor, 2014; Grathwohl et al., 2018). There is also a work that combines both of these approaches (Tucker et al., 2017). Finally, there has been an attempt to remove the smoothness assumption in the reparameterization estimator (Lee et al., 2018). Note that all of these work focus on changing two key algorithms, REINFORCE and the reparameterization estimator, so that the new algorithm has a wider applicability or has less variance. In contrast, our work attempts to adopt a different kind of policy-search algorithms from reinforcement learning for variational inference.

There have been multiple attempts to replace the ELBO by new better objectives for variational inference, such as the so called IWAE and FIVO objectives inspired by importance sampling (Burda et al., 2016) and sequential Monte-Carlo (Maddison et al., 2017a; Naesseth et al., 2018; Le et al., 2018). These objectives give tighter lower bounds to the marginal likelihood of observed data than the ELBO. Intuitively, this means that the objectives are capable of improving the approximation of a given variational distribution. Our work is orthogonal to this line of research, and combining them is an interesting future topic.

The connection between probabilistic inference and reinforcement learning has been explored in the past. The recent review article of Levine (Levine, 2018) provides a good

overview of past and recent studies on this topic with an emphasis on using probabilistic inference techniques, such as stochastic variational inference, for finding good robust policies in variational inference. In particular, the article shows how entropy reinforcement learning is related to stochastic variational inference (Levine, 2018), which is also used to improve the performance of an algorithm by Igl et al. (2018). The use of reinforcement learning techniques for probabilistic inference is explored by Weber et al. (2015), who also suggested the use of value function and critic to reduce variance as in our work. However, they stopped at deriving formulas and suggesting ideas, without showing how they should be converted to a practical algorithm. Our work carries out this maturing step by using further ideas, such as trust region and natural gradient, from reinforcement learning, adjusting them to the setting of variational inference, and showing the theoretical and practical stability of the final algorithm for variational inference.

There have been several attempts to apply natural gradient and trust region to variational inference. Hoffman et al. (2013) applied natural gradient to stochastic variational inference, and Theis and Hoffman (2015) improved it using trust region. However, both of these works use the setup where approximation distributions have global parameters and local parameters. It is not clear how to apply their results to the setting of amortized variational inference considered in this paper. Also, Arenz et al. (2018) applied the trust-region-based policy search algorithm for variational inference, but they did not consider state space models and did not exploit the recursive relationship on future ELBOs as we do in this paper.

5. Experiments

We implemented our algorithm and compared it experimentally with existing approaches. We used two types of models in our experiments. The first is a model for a linear dynamical system. This model does not use any discrete variables, and has a differentiable density. Thus, it is amenable to algorithms known to perform well, such as reparameterization estimator. We chose this model to see how our algorithm compares with existing algorithms on such relatively easy models. The second is sequential deep generative models for two real-world music datasets (Boulanger-lewandowski et al., 2012), which again use both discrete and continuous latent variables. In this last case, models are not fully specified, but include neural networks with unknown parameters. We follow the recipe of variational auto-encoder and extend our algorithm such that it not just learns an approximate posterior but also (the parameters of) a model itself.

In all of these experiments, our algorithm performs as good as or better than existing approaches. Explaining these findings form the rest of this section.

5.1. Experimental Setup

When implementing our algorithm for these experiments, we made certain choices on several parts of the algorithm. First, adopting function approximator $\hat{\Gamma}^w(z_{1:t})$ gives rise to the requirement of learning it. There are previous studies about effective ways to learn function approximator in reinforcement learning. Among them, we chose generalized advantage estimator (GAE) (Schulman et al., 2016) with parameter $\lambda = \gamma = 0.9$ to compute the

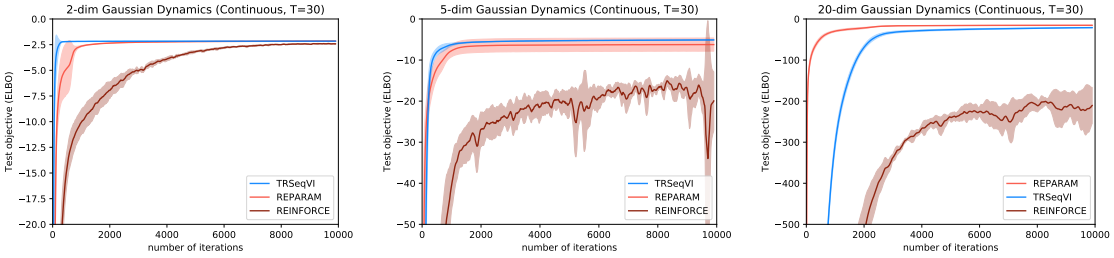

Figure 1: Learning curves for three linear dynamical systems with dimensions 2, 5, and 20. Note that dimensions are 2, 5, 20 from left.

Algorithm	Dimension	Test ELBO	CPU Time(s)
TRSeqVI	2	$-2.160(\pm 0.001)$	100.291
	5	$-4.977(\pm 0.006)$	133.835
	20	$-19.477(\pm 0.466)$	466.726
REPARAM	2	$-2.150(\pm 0.001)$	61.513
	5	$-6.215(\pm 1.601)$	99.212
	20	$-15.236(\pm 0.003)$	422.635
SCORE	2	$-2.377(\pm 0.007)$	38.621
	5	$-11.139(\pm 0.634)$	93.297
	20	$-230.324(\pm 26.694)$	428.335

Table 1: Consumed CPU time for the $1,000$ iteration

approximation, which is defined as follows:

$$\hat{A}^{\phi} = \sum_{t=1}^{T} (\gamma\lambda)^{t-1} \Delta_{\phi}^{\gamma}(z_{1:t}).$$

The GAE controls trade-off between variance and bias through γ and λ, which are generally near 1. Next, to compute natural gradient efficiently, we applied conjugate gradient algorithm with trust region constrained by 10^{-3}. Finally, we used Adam optimizer (Kingma and Ba, 2015) with learning rate 10^{-3} to perform stochastic gradient ascent.

5.2. Linear Dynamical Systems

We first conducted an experiment on linear dynamical systems with three different dimensions $d \in \{2, 5, 20\}$. The underlying dynamics in each of these cases is fixed, and has the following form:

$$z_t = Az_{t-1} + v_t,$$
$$x_t = Cz_t + w_t,$$

where z_t and x_t are d-dimensional latent and observed variables at time step t, the next A and C are $d \times d$ randomly-chosen matrices with determinant 1, and v_t and w_t denote the d-dimension Gaussian noises with variance $0.01I_d$.

Algorithms	JSB	Nottingham
TRSeqVI (C)	$-8.831(\pm 0.096)$	$-2.438(\pm 0.064)$
REPARAM (C)	$-8.586(\pm 0.014)$	$-3.122(\pm 0.051)$
TRSeqVI (D)	$\mathbf{-7.687(\pm 0.096)}$	$\mathbf{-2.135(\pm 0.053)}$
REPARAM (D)	$-8.579(\pm 0.025)$	$-3.195(\pm 0.288)$
SCORE (D)	$-8.666(\pm 0.088)$	$-3.491(\pm 0.495)$

Table 2: Results for test ELBO of JSB and Nottingham.

Our variational distributions use information only at the current and previous steps and have the form:

$$q_\phi(z_t|z_{1:t-1}; x) = q_\phi(z_t|z_{t-1}; x_t)$$

We compared three algorithms: reparameterization estimator (denoted as REPARAM), score estimator (denoted as SCORE), and our algorithm (denoted as TRSeqVI).

The result is shown in the Figure 1 for dimension 2, 5, and 20. For the 20-dimension case, we also report a zoomed-in version of the graph to highlight the difference between REPARAM and TRSeqVI. In all three cases, our algorithm converges faster and has less standard error than the other two algorithms, and achieves as good an ELBO as REPARAM, which outperforms SCORE in terms of ELBO. Although the inner loop of TRSeqVI performs more computation than those of REPARAM and SCORE (due to line search), it does not incur too much overhead. Table 1 shows the CPU time of the three algorithms during the first $1,000$ iteration. Since the initial setup time for running these algorithms is usually needed, the differences shown in the table have little impact on the overall time of these algorithms.

5.3. Polyphonic Music Datasets

The last experiment is on two polyphonic music datasets, JSB chorales and Nottingham folk tunes (Boulanger-lewandowski et al., 2012). On these datasets, we need to learn both models and variational distributions. The model learning is done by the stochastic gradient ascent with respect to the ELBO objective, while the inference of variational distributions is carried out by three different algorithms, SCORE, REPARAM and TRSeqVI. The generative models for these datasets can be formulated with or without using discrete latent variables, although it is more intuitive to use discrete ones. For TRSeqVI and REPARAM, we try both alternatives; for REPARAM, using models without discrete latent variables means that we do not need to use relaxation and introduce additional approximation. Quite naturally, using models with discrete latent variables gives better results in both cases than using the ones without those variables.

The latent variables of the model for JSB chorals is 32 dimensional, and the one for Nottingham is 64 dimensional, regardless of whether they are formed in terms of discrete or continuous ones. Variational distributions have the form

$$q_\phi(z_t|z_{t:1-1}; x) = q_\phi(z_t|z_{1:t-1}; x_{1:t})$$

which mean that they depend on information on observations up to time step t. The $(z_{1:t-1}; x_{1:t-1})$ parts are summarized by recurrent neural networks, and we use diagonal

covariance matrices when models are built in terms of continuous latent variables. Also, we use a data encoder and latent encoder to encode data and latent variable. In detail, we use a single-layer LSTM for recurrent neural network. Also, all of probability densities, data encoder, and latent encoder are fully connected neural networks with one hidden layer. The hidden layers here have the same size as the dimension of the latent variable.

In Table 2, (C) means that an algorithm uses continuous latent variables with the Gaussian distribution, and (D) means that an algorithm uses discrete latent variables with the Bernoulli distribution. We run the algorithms in the table, for $300,000$ iterations in both of JSB and Nottingham domains. As seen in this table, TRSeqVI outperforms the other alternatives in both datasets, regardless of whether we use a model with countinuous latent variables or the one with discrete variables. It is worth noting that the best performer is TRSeqVI(D), which beats its continuous counterpart TRSeqVI(C).

6. Conclusion

In this work, we introduced and analyzed a new algorithm for variational inference which successfully tackles the issue for non-differentiable models, such as the one with discrete latent variables. To achieve this goal, we applied three key ideas, approximation function of future ELBO, trust region update, and natural gradient, all of which are inspired from reinforcement learning. By taking advantage of the difference between variational inference and reinforcement learning, we improved these ideas from reinforcement learning by designing a method for efficient line search and obtaining a simple proof for theoretical guarantee. Our experiments show the promise of the performance of our algorithm on state space models with discrete latent variables.

A natural follow-up is to design a specialized algorithm for learning generative models that goes well with the inference algorithm presented in this paper. In our experiment, we just used the stochastic gradient on the standard ELBO objective, but given that alternative objectives for model learning are actively explored and pursued recently, there may be a different objective for model learning that works better with the algorithm in the paper. Another possible follow-up is to combine our algorithm with existing techniques. When the density of a model is differentiable with respect to certain latent variables, we would like to make our algorithm use this smoothness property by, for instance, applyinging reparameterization selectively only on those variables.

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-2016-0-00464) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation). We also acknowledge the support by MSIT (IITP No. 2019-0-00075-001). The work was conducted at High-Speed Vehicle Research Center of KAIST with the support of Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD). Yang was supported by the Engineering Research Center Program through the National Research Foundation of Korea (NRF) funded by the Korean Government MSIT (NRF-2018R1A5A1059921), and also by Next-Generation Information

Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (2017M3C4A7068177).

References

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2): 251–276, 1998.

Oleg Arenz, Mingjun Zhong, and Gerhard Neumann. Efficient gradient-free variational inference using policy search. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 234–243, 2018.

Nicolas Boulanger-lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in highdimensional sequences: Application to polyphonic music generation and transcription. In *In ICML 29*. Citeseer, 2012.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

Peter W Glynn. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84, 1990.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Will Grathwohl, Dami Choi, Yuhuai Wu, Geoff Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *ICLR*, 2018.

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

Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational reinforcement learning for POMDPs. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2117–2126, 2018.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Second International Conference on Learning Representations, ICLR*, 2014.

Tuan Anh Le, Maximilian Igl, Tom Rainforth, Tom Jin, and Frank Wood. Auto-encoding sequential monte carlo. In *International Conference on Learning Representations*, 2018.

Wonyeol Lee, Hangyeol Yu, and Hongseok Yang. Reparameterization gradient for non-differentiable models. In *Advances in neural information processing systems (NeurIPS'18)*, pages 5558–5568, 2018.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Chris J Maddison, John Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Teh. Filtering variational objectives. In *Advances in Neural Information Processing Systems*, pages 6576–6586, 2017a.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017b.

Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1791–1799, 2014.

Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, pages 968–977, 2018.

Jeffrey Regier, Michael I Jordan, and Jon McAuliffe. Fast black-box variational inference through stochastic trust-region optimization. In *Advances in Neural Information Processing Systems*, pages 2399–2408, 2017.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

Lucas Theis and Matt Hoffman. A trust-region method for stochastic variational inference with applications to streaming data. In *International Conference on Machine Learning*, pages 2503–2511, 2015.

George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pages 2627–2636, 2017.

Theophane Weber, Nicolas Heess, Ali Eslami, John Schulman, David Wingate, and David Silver. Reinforced variational inference. In *Advances in Neural Information Processing Systems (NIPS) Workshops*, 2015.

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

Appendix A. Details of Gradient Derivation

Lemma 1. *For any ϕ, using future ELBO does not cause any bias. It means,*

$$\nabla_\phi \mathcal{L}(\phi; x) = \mathbb{E}_{z\sim q_\phi(\_;x)}\left[\sum_{t=1}^{T} \nabla_\phi \log q_\phi(z_t|z_{1:t-1};x)\Delta_\phi(z_{1:t})\right]$$

holds.

Proof. We start the proof from (5):

$$\begin{aligned}
\nabla_\phi \mathcal{L}(\phi; x) &= \sum_{t=1}^{T}\int q_\phi(z;x)\nabla_\phi \log q_\phi(z_t|z_{1:t-1};x)\sum_{t'=t}^{T}\log\frac{p(x_{t'},z_{t'}|x_{1:t'-1},z_{1:t'-1})}{q_\phi(z_{t'}|z_{1:t'-1};x)}dz \\
&= \sum_{t=1}^{T}\int q_\phi(z_{1:t};x)\nabla_\phi \log q_\phi(z_t|z_{1:t-1};x)\left\{\log\frac{p(x_t,z_t|x_{1:t-1},z_{1:t-1})}{q_\phi(z_t|z_{1:t-1};x)}+\Gamma_\phi(z_{1:t})\right\}dz_{1:t} \\
&= \sum_{t=1}^{T}\int q_\phi(z_{1:t};x)\nabla_\phi \log q_\phi(z_t|z_{1:t-1};x)\Delta_\phi(z_{1:t})dz_{1:t} \\
&= \mathbb{E}_{z\sim q_\phi(_;x)}\left[\sum_{t=1}^{T}\nabla_\phi \log q_\phi(z_t|z_{1:t-1};x)\Delta_\phi(z_{1:t})\right].
\end{aligned}$$

Note that

$$\begin{aligned}
&\int q_\phi(z_{1:t};x)\nabla_\phi \log q_\phi(z_t|z_{1:t-1};x)\Gamma_\phi(z_{1:t-1})dz_{1:t} \\
&= \int q_\phi(z_{1:t-1};x)\Gamma_\phi(z_{1:t-1})\int \nabla_\phi q_\phi(z_t|z_{1:t-1};x)dz_t dz_{1:t-1} = 0
\end{aligned}$$

□

Appendix B. Proofs of Theorem 1

In this section, we clarify the condition of Theorem 1 and provide proof. Before starting proof, we introduce a lemma needed to prove theorem.

Proof of Theorem 1.

$$\begin{aligned}
&\mathcal{L}(\phi;x) - \mathcal{L}(\phi_k;x) \\
&= \mathbb{E}_{z\sim q_\phi(_;x)}\left[\sum_{t=1}^{T}\{\log p(x_t,z_t|x_{1:t-1},z_{1:t-1}) - \log q_\phi(z_t|z_{1:t-1};x)\} - \mathcal{L}(\phi_k;x)\right] \\
&= \mathbb{E}_{z\sim q_\phi(_;x)}\left[\sum_{t=1}^{T}\left\{\Delta_{\phi_k}(z_{1:t};x) - \log\frac{q_\phi(z_t|z_{1:t-1};x)}{q_{\phi_k}(z_t|z_{1:t-1};x)}\right\}\right] \\
&= \mathbb{E}_{z\sim q_\phi(_;x)}\left[\sum_{t=1}^{T}\Delta_{\phi_k}(z_{1:t};x)\right] - D_{\mathrm{KL}}(q_\phi(z_{1:T};x)\|q_{\phi_k}(z_{1:T};x)) \\
&= d(x) - D_{\mathrm{KL}}(q_\phi(z_{1:T};x)\|q_{\phi_k}(z_{1:T};x)) \geq d(x) - \delta \geq 0
\end{aligned}$$

if $\delta \leq d(x)$ holds for

$$d(x) = \widetilde{\mathcal{L}}(\phi, \phi_k; x) - \mathcal{L}(\phi_k; x) = \mathbb{E}_{z \sim q_\phi(\_;x)} \left[\sum_{t=1}^{T} \Delta_{\phi_k}(z_{1:t}; x) \right].$$

□

Appendix C. Theoretical Analyze for Alternative Line Search

In this section, we analyze for the condition of δ if we rely on importance sampling like TRPO. For this analysis, define another surrogate objective $\mathcal{L}'(\phi, \phi_k; x)$ as

$$\mathcal{L}'(\phi, \phi_k; x) = \mathcal{L}(\phi_k; x) + \mathbb{E}_{z \sim q_{\phi_k}(\_;x)} \left[\sum_{t=1}^{T} \frac{q_\phi(z_t | z_{1:t-1}; x)}{q_{\phi_k}(z_t | z_{1:t-1}; x)} \Delta_{\phi_k}(z_{1:t}; x) \right].$$

Before introducing a theorem, we need following lemma:

Lemma 2. *Assume that for any parameter ϕ and ϕ_k, there are finite constants $\Delta_{\max}(x) > 0$ and $\delta > 0$ which satisfy*

$$\Delta_{\max}(x) = \max_{z_{1:t}} |\Delta_{\phi_k}(z_{1:t}; x)|,$$
$$\delta \geq D_{\mathrm{KL}}(q_\phi(z_{1:T}; x) \| q_{\phi_k}(z_{1:T}; x)).$$

Then,

$$\left| \mathbb{E}_{z \sim q_\phi(\_;x)} \left[\sum_{t=1}^{T} \Delta_{\phi_k}(z_{1:t}; x) \right] - \mathbb{E}_{z \sim q_{\phi_k}(\_;x)} \left[\sum_{t=1}^{T} \frac{q_\phi(z_t | z_{1:t-1}; x)}{q_{\phi_k}(z_t | z_{1:t-1}; x)} \Delta_{\phi_k}(z_{1:t}; x) \right] \right| \leq T \Delta_{\max}(x) \sqrt{2\delta}$$

holds.

Proof.

$$\begin{aligned}
&\left| \mathbb{E}_{z \sim q_\phi(_;x)} \left[\sum_{t=1}^{T} \Delta_{\phi_k}(z_{1:t}; x) \right] - \mathbb{E}_{z \sim q_{\phi_k}(_;x)} \left[\sum_{t=1}^{T} \frac{q_\phi(z_t | z_{1:t-1}; x)}{q_{\phi_k}(z_t | z_{1:t-1}; x)} \Delta_{\phi_k}(z_{1:t}; x) \right] \right| \\
&\leq \sum_{t=1}^{T} \left| \mathbb{E}_{z \sim q_\phi(_;x)} \left[\Delta_{\phi_k}(z_{1:t}; x) \right] - \mathbb{E}_{z \sim q_{\phi_k}(_;x)} \left[\frac{q_\phi(z_t | z_{1:t-1}; x)}{q_{\phi_k}(z_t | z_{1:t-1}; x)} \Delta_{\phi_k}(z_{1:t}; x) \right] \right| \\
&= \sum_{t=1}^{T} \left| \int \left\{ q_\phi(z_{1:t-1}; x) - q_{\phi_k}(z_{1:t-1}; x) \right\} q_\phi(z_t | z_{1:t-1}; x) \Delta_{\phi_k}(z_{1:t}; x) dz_{1:t} \right| \\
&\leq \sum_{t=1}^{T} \Delta_{\max}(x) \int \left| q_\phi(z_{1:t-1}; x) - q_{\phi_k}(z_{1:t-1}; x) \right| q(z_t | z_{1:t-1}; x) dz_{1:t} \\
&\leq \sum_{t=1}^{T} \Delta_{\max}(x) \sqrt{2 D_{\mathrm{KL}} \left(q_\phi(z_{1:t-1}; x) \| q_{\phi_k}(z_{1:t-1}; x) \right)} \\
&\leq \sum_{t=1}^{T} \Delta_{\max}(x) \sqrt{2 D_{\mathrm{KL}} \left(q_\phi(z_{1:T}; x) \| q_{\phi_k}(z_{1:T}; x) \right)} \\
&= T \Delta_{\max}(x) \sqrt{2\delta}.
\end{aligned}$$

Note that the inequality $D_{\mathrm{TV}}(q_\phi(z_{1:t};x)\|q_{\phi_k}(z_{1:t};x)) \le \sqrt{2D_{\mathrm{KL}}(q_\phi(z_{1:t};x)\|q_{\phi_k}(z_{1:t};x))}$ for any t comes from Pinsker's inequality. □

Now, we construct a theorem, which is similar to Theorem 1:

Theorem 2. *Given the parameter ϕ_k at iteration k, assume that Δ_{ϕ_k} is bounded. Then, for any ϕ with $\mathcal{L}'(\phi,\phi_k;x) > \mathcal{L}(\phi_k;x)$, there exists $\delta > 0$ such that*

$$\mathcal{L}(\phi;x) \ge \mathcal{L}(\phi_k;x)$$

Proof. Fix ϕ_k and suppose ϕ satisfies $\mathcal{L}'(\phi,\phi_k;x) > \mathcal{L}(\phi_i;x)$. Now, set notations $\Delta_{\max}(x)$ and $d'(x)$ as

$$\Delta_{\max}(x) = \max_{z_{1:t}} |\Delta_{\phi_k}(z_{1:t};x)|,$$
$$d'(x) = \mathcal{L}'(\phi,\phi_k;x) - \mathcal{L}(\phi_k;x).$$

Then, $\Delta_{\max}(x)$ is finite and $d(x) > 0$ from the condition of theorem. Using these notations with Lemma 2, we obtain following

$$\begin{aligned}
&\mathcal{L}(\phi;x) - \mathcal{L}(\phi_k;x)\\
&= \mathbb{E}_{z\sim q_\phi(_;x)}\left[\sum_{t=1}^{T}\left\{\Delta_{\phi_k}(z_{1:t};x) - \log\frac{q_\phi(z_t|z_{1:t-1};x)}{q_{\phi_k}(z_t|z_{1:t-1};x)}\right\}\right]\\
&= \mathbb{E}_{z\sim q_\phi(_;x)}\left[\sum_{t=1}^{T}\Delta_{\phi_k}(z_{1:t};x)\right] - D_{\mathrm{KL}}(q_\phi(z_{1:T};x)\|q_{\phi_k}(z_{1:T};x))\\
&\ge \mathbb{E}_{z\sim q_{\phi_k}(_;x)}\left[\sum_{t=1}^{T}\frac{q_\phi(z_t|z_{1:t-1};x)}{q_{\phi_k}(z_t|z_{1:t-1};x)}\Delta_{\phi_k}(z_{1:t})\right] - T\Delta_{\max}(x)\sqrt{2\delta} - D_{\mathrm{KL}}(q_\phi(z_{1:T};x)\|q_{\phi_k}(z_{1:T};x))\\
&\ge \mathbb{E}_{z\sim q_{\phi_k}(_;x)}\left[\sum_{t=1}^{T}\frac{q_\phi(z_t|z_{1:t-1};x)}{q_{\phi_k}(z_t|z_{1:t-1};x)}\Delta_{\phi_k}(z_{1:t})\right] - T\Delta_{\max}(x)\sqrt{2\delta} - \delta\\
&\ge d'(x) - (\delta + T\Delta_{\max}(x)\sqrt{2\delta})
\end{aligned}$$

Assume that $\delta < 1$, then $\delta < \sqrt{\delta}$ and therefore,

$$\mathcal{L}(\phi;x) - \mathcal{L}(\phi_k;x) \ge d'(x) - \sqrt{\delta}(1 + \sqrt{2}T\Delta_{\max}(x)).$$

It means, for any $\delta > 0$ with

$$\delta \le \min\left\{1, \left(\frac{d'(x)}{1+\sqrt{2}T\Delta_{\max}(x)}\right)^2\right\}$$

satisfies Theorem 2. □

It means, when we use $\widetilde{\mathcal{L}}$, it is enough to $\delta < \min\{1, d(x)\}$. But, when we use $\mathcal{L}'$, then δ is more strongly constrained, since in general $d(x) \approx d'(x) \ll 1$ and therefore, square greatly reduces constraint.