Combine Monte Carlo with Exhaustive Search: Effective Variational Inference and Policy Gradient Reinforcement Learning

Michalis K. Titsias
Department of Informatics
Athens University of Economics and Business
mtitsias@aueb.gr

Abstract

In this paper we discuss very preliminary work on how we can reduce the variance in black box variational inference based on a framework that combines Monte Carlo with exhaustive search. We also discuss how Monte Carlo and exhaustive search can be combined to deal with infinite dimensional discrete spaces. Our method builds upon and extends a recently proposed algorithm that constructs stochastic gradients using local expectations.

1 Introduction

Many problems in approximate variational inference and reinforcement learning involve the maximization of an expectation

$$\mathbb{E}_{q_\theta(\mathbf{x})}[f(\mathbf{x})], \tag{1}$$

where $q_\theta(\mathbf{x})$ is a distribution that depends on parameters θ that we wish to tune. For most interesting applications in machine learning the optimization of the above cost function is very challenging since the gradient cannot be computed analytically. To deal with this, several approaches are based on stochastic optimization where stochastic gradients are used to carry out the optimization.

Specifically, the two most popular approaches are the score function method (Williams, 1992; Glynn, 1990; Paisley et al., 2012; Ranganath et al., 2014; Mnih and Gregor, 2014) and the reparametrization method (Salimans and Knowles, 2013; Kingma and Welling, 2014; Rezende et al., 2014; Titsias and Lázaro-Gredilla, 2014). The reparametrization method is suitable for differential functions $f(\mathbf{x})$ while the score function is completely general and it can be applied to non-smooth functions $f(\mathbf{x})$ as well. Both these methods are heavily based on sampling from the variational distribution $q_\theta(\mathbf{x})$. However, stochastic estimates purely based on Monte Carlo can be inefficient. This is because, while Monte Carlo has the great advantage that gives unbiased estimates it is computationally very intense and even in the simplest cases it requires infinite computational time to give exact answers. For instance, suppose $\mathbf{x}$ takes a finite set of K values. While the exact value of the gradient (based on exhaustive enumeration of all values) can be achieved in $O(K)$ time, Monte Carlo still wastefully requires infinite computational time to provide the exact gradient.

How can we reduce the computational ineffectiveness of Monte Carlo for (at least) discrete spaces? Here, we propose to do this by simply moving computational resources from Monte Carlo to exhaustive enumeration or search so that we construct stochastic estimates by combining Monte Carlo with exhaustive search. A first instance of such an algorithm has been recently proposed in (Titsias and Lázaro-Gredilla, 2015) and it is referred to as local expectation gradients or just local expectations. Here, we extend this approach by proposing i) a new version of the algorithm that can much more efficiently optimize highly correlated distributions $q_\theta(\mathbf{x})$ where $\mathbf{x}$ is a high dimensional

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

discrete vector and ii) by showing how to deal with cases where components of $\mathbf{x}$ take infinite values. In the appendix we also discuss applications to policy gradient optimization in reinforcement learning.

2 A new version of local expectations for correlated distributions

Suppose that the n-dimensional latent vector $\mathbf{x} = (x_1, \ldots, x_n)$ in the cost in (1) is such that each x_i takes M_i values. We consider a variational distribution over $\mathbf{x}$ that is represented as a directed graphical model having the following joint density

$$q_\theta(\mathbf{x}) = \prod_{i=1}^{n} q_\theta(x_i|\text{pa}_i), \tag{2}$$

where $q_\theta(x_i|\text{pa}_i)$ is the conditional factor over x_i given the set of the parents denoted by pa_i and θ is the set of variational parameters. For simplicity we consider θ to be a *global parameter* that influences all factors, but in practice of course each factor could depend only on a subset of dimensions of θ. Next we will make heavily use of a decomposition of $q_\theta(\mathbf{x})$ in the form

$$q_\theta(\mathbf{x}) = q_\theta(x_{1:i-1}) q_\theta(x_i|\text{pa}_i) q_\theta(x_{i+1:n}|x_i, x_{1:i-1}),$$

where we intuitively think of $q_\theta(x_{1:i-1}) = \prod_{j=1}^{i-1} q_\theta(x_j|\text{pa}_j)$ as the factor associated with the *past*, the single conditional $q_\theta(x_i|\text{pa}_i)$ as the factor representing the *present* and the remaining term $q_\theta(x_{i+1:n}|x_i, x_{1:i-1}) = \prod_{j=i+1}^{n} q_\theta(x_j|\text{pa}_j)$ as the factor representing the *future*.

To maximize the cost in (1) we take gradients with respect to θ so that the exact gradient can be written as

$$\nabla_\theta \mathbb{E}_{q_\theta(\mathbf{x})}[f(\mathbf{x})] = \sum_{i=1}^{n} \sum_{x_{1:i-1}} q_\theta(x_{1:i-1}) \left[\sum_{x_i} \nabla_\theta q_\theta(x_i|\text{pa}_i) \left[\sum_{x_{i+1:n}} q_\theta(x_{i+1:n}|x_i, x_{1:i-1}) f(\mathbf{x}) \right] \right]$$

To get an unbiased stochastic estimate we can firstly observe that the only problematic summation that is not already an expectation is the summation over x_i. Therefore, our idea is to deal with the problematic summation over x_i by performing exhaustive search, while for the remaining variables we can use Monte Carlo. We need two Monte Carlo operations that require sampling from the past factor $q_\theta(x_{1:i-1})$ and sampling also from the future factor $q_\theta(x_{i+1:n}|x_i, x_{1:i-1})$. These two operations need to be treated slightly differently since the past factor is independent from x_i, while the future factor does depend on x_i. We can approximate the expectation under the past by drawing a sample $x_{1:i-1}^{(s)}$ from $q_\theta(x_{1:i-1})$ yielding

$$\nabla_\theta \mathbb{E}_{q_\theta(\mathbf{x})}[f(\mathbf{x})] \approx \sum_{i=1}^{n} \sum_{m=1}^{M_i} \nabla_\theta q_\theta(x_i = m|\text{pa}_i^{(s)}) \left[\sum_{x_{i+1:n}} q_\theta(x_{i+1:n}|x_i = m, x_{1:i-1}^{(s)}) f(x_{1:i-1}^{(s)}, x_{i:n}) \right]$$

where we have explicitly written the sum over all possible values of x_i. To get now an unbiased estimate of this we will need to draw M_i samples from all possible future factors $q_\theta(x_{i+1:n}|x_i = m, x_{1:i-1}^{(s)}), m = 1, \ldots, M_i$ which gives

$$\nabla_\theta \mathbb{E}_{q_\theta(\mathbf{x})}[f(\mathbf{x})] \approx \sum_{i=1}^{n} \sum_{m=1}^{M_i} \nabla_\theta q_\theta(x_i = m|\text{pa}_i^{(s)}) f(x_{1:i-1}^{(s)}, x_i = m, x_{i+1:n}^{(s,m)}), \tag{3}$$

where $x_{i+1:n}^{(s,m)} \sim q_\theta(x_{i+1:n}|x_i = m, x_{1:i-1}^{(s)})$. All these samples can be intuitively thought of as several possible imaginations of the future produced by exhaustively enumerating all values of x_i at present time. Also the computations inside the sum $\sum_{i=1}^{n}$ can be done in parallel. This is because we can a draw single sample $\mathbf{x}^{(s)}$ from $q_\theta(\mathbf{x})$ beforehand and then move along the path $\mathbf{x}^{(s)}$ and compute all n terms involving the sums $\sum_{m=1}^{M_i}$ in parallel.

The above algorithm is simpler and much more effective than the initial algorithm in (Titsias and Lázaro-Gredilla, 2015) which is based on the Markov blanket around x_i, and it operates similarly to Gibbs sampling where a single future sample is used. Notice, however, that for fully factorised distributions where the past, present and future are independent the above procedure essentially becomes equivalent to local expectations in (Titsias and Lázaro-Gredilla, 2015). So it is only for the correlated distributions that the above stochastic gradients differ than the one proposed in (Titsias and Lázaro-Gredilla, 2015). In appendix we describe how the above algorithm can be used in reinforcement learning.

3 Dealing with infinite dimensional spaces

For several applications, as those involving Bayesian non-parametric models or Poisson latent variable models, some random variables x_i can take countably infinite values. How can we extend the previous algorithm and the algorithms proposed in (Titsias and Lázaro-Gredilla, 2015) to deal with that? The solution is rather simple: we need to transfer computations from exhaustive search back to Monte Carlo.

Suppose that some random variable x_i takes infinite values so in order to evaluate the stochastic gradient in (3) we need to perform the following infinite sum $\sum_{m=1}^{\infty} \nabla_\theta q_\theta(x_i = m|\text{pa}_i^{(s)}) f(x_{1:i-1}^{(s)}, x_i = m, x_{i+1:n}^{(s,m)})$, which next in order to simplify notation we shall write as

$$\sum_{x_i=1}^{\infty} \nabla_\theta q_\theta(x_i) f(x_i).$$

We introduce a possibly adaptive integer $T \geq 1$ and write the above as

$$\sum_{x_i=1}^{T} \nabla q_\theta(x_i) f(x_i) + \sum_{x_i=T+1}^{\infty} \nabla q_\theta(x_i) f(x_i).$$

The first finite sum can be computed exactly through exhaustive enumeration. Thus, to get an overall unbiased estimate we need to get an unbiased estimate for the second term. For that we will use Monte Carlo. More precisely, to use Monte Carlo we need to apply the score function method so that the second term is written as

$$\begin{aligned}
\sum_{x_i=T+1}^{\infty} \nabla q_\theta(x_i) f(x_i) &= \sum_{x_i=T+1}^{\infty} q_\theta(x_i) \nabla \log q_\theta(x_i) f(x_i) \\
&= (1 - Q(T)) \sum_{x_i=T+1}^{\infty} \frac{q_\theta(x_i)}{1 - Q(T)} \nabla \log q_\theta(x_i) f(x_i) \\
&= (1 - Q(T)) \sum_{x_i=T+1}^{\infty} q_{v_i}(x_i | x_i > T) \nabla \log q_\theta(x_i) f(x_i)
\end{aligned}$$

where $Q(T) = \sum_{x_i=1}^{T} q_\theta(x_i)$ is the cumulative distribution function and $q_\theta(x_i|x_i > T)$ is the truncated variational distribution over the space $x_i > T$. To get now an unbiased estimate we simply need to draw independent samples from $q_\theta(x_i|x_i > T)$. So overall the stochastic gradient is

$$\sum_{x_i=1}^{T} \nabla q_\theta(x_i) f(x_i) + \frac{1 - Q(T)}{S} \sum_{s=1}^{S} \nabla \log q_\theta(x_i^{(s)}) f(x_i^{(s)}). \tag{4}$$

This gradient is unbiased for any value of T. However, in practice to efficiently reduce variance we will need to choose/adapt T so that the probability $Q(T)$ becomes large and the contribution of the second term is small. In practice, to implement this in a black box manner we can adaptively choose T so that at each optimization iteration $Q(T)$ is above a certain threshold such as 0.95.

4 Discussion

We have presented methods to combine Monte Carlo and exhaustive search in order to reduce variance in stochastic optimization of variational objectives and also to deal with infinite dimensional discrete spaces. The appendix describes a related algorithm for policy gradient reinforcement learning.

References

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

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

Mnih, A. and Gregor, K. (2014). Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*.

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

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

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*.

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

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

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

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

A Combine Monte Carlo and exhaustive search for policy optimization

Consider a finite horizon Markov decision process (MDP) with joint density

$$p(\alpha_{0:h-1}, s_{1:h}|s_0) = \prod_{t=0}^{h-1} \pi_\theta(\alpha_t|s_t)p(s_{t+1}|s_t, \alpha_t). \tag{5}$$

Let $R_{t+1} = R(s_t, \alpha_t, s_{t+1})$ denote the reward that the agent receives when starting at state s_t, performing action α_t and ending up at state s_{t+1}. The agent wishes to tune the policy parameters θ so that to maximize the expected total reward

$$v_\theta(s_0) = \mathbb{E}\left[\sum_{t=0}^{h-1} R_{t+1}\right]$$

where the expectation is taken under the distribution $p(\alpha_{0:h-1}, s_{1:h}|s_0)$ given by eq. (5). The gradient is explicitly written as

$$\begin{aligned}\nabla_\theta v_\theta(s_0) &= \sum_{k=0}^{h-1} \sum_{\alpha_{0:h-1}, s_{1:h}} p(s_{1:k}, \alpha_{0:k-1}|s_0)\nabla_\theta \pi_\theta(\alpha_k|s_k)p(s_{k+1:h}, \alpha_{k+1:h-1}|s_k, \alpha_k)\left[\sum_{t=0}^{h-1} R_{t+1}\right] \\ &= \sum_{k=0}^{h-1} f_\theta^k(s_0)\end{aligned} \tag{6}$$

where

$$p(s_{1:k}, \alpha_{0:k-1}|s_0) = \prod_{t=0}^{k-1} \pi_\theta(\alpha_t|s_t)p(s_{t+1}|s_t, \alpha_t)$$

$$p(s_{k+1:h}, \alpha_{k+1:h-1}|s_k, \alpha_k) = p(s_{k+1}|s_k, \alpha_k) \prod_{t=k+1}^{h-1} \pi_\theta(\alpha_t|s_t)p(s_{t+1}|s_t, \alpha_t).$$

The function $f_\theta^k(s_0)$ can be thought as the gradient information collected by the agent at time k, i.e. when the agent takes action α_k. The probability distribution $p(s_{1:k}, \alpha_{0:k-1}|s_0)$ describes the sequence of states and actions that precede action α_k, while $p(s_{k+1:h}, \alpha_{k+1:h-1}|s_k, \alpha_k)$ describes those generated after α_k is taken. Observe that while $p(s_{1:k}, \alpha_{0:k-1}|s_0)$ is independent from the current action α_k, $p(s_{k+1:h}, \alpha_{k+1:h-1}|s_k, \alpha_k)$ depends on it since clearly the future states and actions are influenced by the current action.

The exact value of $f_\theta^k(s_0)$ is computationally intractable and we are interested in expressing low variance unbiased estimates that can be realistically acquired through actual experience. The key idea of our approach is

to explore all possible actions at time k, and then combine this with Monte Carlo sampling. By taking advantage of the Markov structure of $f_\theta^k(s_0)$ we re-arrange the summations as

$$f_\theta^k(s_0) = \sum_{\substack{s_{1:k} \\ \alpha_{0:k-1}}} p(s_{1:k}, \alpha_{0:k-1}|s_0) \sum_{\alpha_k} \nabla_\theta \pi_\theta(\alpha_k|s_k) \sum_{\substack{s_{k+1:h} \\ \alpha_{k+1:h-1}}} p(s_{k+1:h}, \alpha_{k+1:h-1}|s_k, \alpha_k) \left[\sum_{t=0}^{h-1} R_{t+1}\right]$$

$$= \sum_{\substack{s_{1:k} \\ \alpha_{0:k-1}}} p(s_{1:k}, \alpha_{0:k-1}|s_0) \sum_{m=1}^{\mathcal{A}(s_k)} \nabla_\theta \pi_\theta(\alpha_k = m|s_k) \sum_{\substack{s_{k+1:h} \\ \alpha_{k+1:h-1}}} p(s_{k+1:h}, \alpha_{k+1:h-1}|s_k, \alpha_k = m) \left[\sum_{t=0}^{h-1} R_{t+1}\right]$$

where $\mathcal{A}(s_k^{(i)})$ denotes the number of possible actions when we are at state $s_k^{(i)}$. To get now an unbiased estimate we can sample from $p(s_{1:k}, \alpha_{0:k-1}|s_0)$ and from each conditional $p(s_{k+1:h}, \alpha_{k+1:h-1}|s_k, \alpha_k = m)$. More precisely, by generating a single path $(s_{1:k}^{(i)}, \alpha_{0:k-1}^{(i)})$ from the past factor $p(s_{1:k}, \alpha_{0:k-1}|s_0)$ we obtain

$$\sum_{m=1}^{\mathcal{A}(s_k^{(i)})} \nabla_\theta \pi_\theta(\alpha_k = m|s_k^{(i)}) \sum_{\substack{s_{k+1:h} \\ \alpha_{k+1:h-1}}} p(s_{k+1:h}, \alpha_{k+1:h-1}|s_k^{(i)}, \alpha_k = m) \left[\sum_{t=0}^{h-1} R_{t+1}^{(i)}\right]$$

Notice that the reward values $R_{t+1}^{(i)}$ are now indexed by i to emphasize their dependence on the drawn trajectory. However, the above quantity still remains intractable as the summation over all future trajectories, i.e. over the set $(s_{k+1:h}, \alpha_{k+1:h-1})$, is not feasible. To deal with that we can use again Monte Carlo by sampling from each conditional MDP with distribution $p(s_{k+1:h}, \alpha_{k+1:h-1}|s_k^{(i)}, \alpha_k = m)$ so that we start at state $s_k^{(i)}$, we take action $\alpha_k = m$ and subsequently we follow the current policy. Thus, overall we generate $\mathcal{A}(s_k^{(i)})$ total future trajectories, denoted by $(s_{k+1:h}^{(i,m)}, \alpha_{k+1:h-1}^{(i,m)})_{m=1}^{\mathcal{A}(s_k^{(i)})}$, and obtain the estimate

$$\sum_{m=1}^{\mathcal{A}(s_k^{(i)})} \nabla_\theta \pi_\theta(\alpha_k = m|s_k^{(i)}) \left[\sum_{t=0}^{k-1} R_{t+1}^{(i)} + \sum_{t=k}^{h-1} R_{t+1}^{(i,m)}\right]$$

Here, is rather crucial to see that the sum of rewards is split into two different terms: $\sum_{t=0}^{k-1} R_{t+1}^{(i)}$ which is the sum of rewards obtained before the action α_k is taken, and $\sum_{t=k}^{h-1} R_{t+1}^{(i,m)}$ which is the sum of rewards obtained by performing action $\alpha_k = m$ and then following the policy. Unlike the rewards in second term that depend on the current action (and therefore are indexed by both i and m), the first term is just a constant with respect to the action α_k. By taking advantage of that, the above unbiased estimate simplifies to

$$\sum_{m=1}^{\mathcal{A}(s_k^{(i)})} \nabla_\theta \pi_\theta(\alpha_k = m|s_k^{(i)}) \left[\sum_{t=k}^{h-1} R_{t+1}^{(i,m)}\right] \tag{7}$$

where we used that $\sum_{m=1}^{\mathcal{A}(s_k^{(i)})} \nabla_\theta \pi_\theta(\alpha_k = m|s_k^{(i)}) \left[\sum_{t=0}^{k-1} R_{t+1}^{(i)}\right] = \sum_{t=0}^{k-1} R_{t+1}^{(i)} \sum_{m=1}^{\mathcal{A}(s_k^{(i)})} \nabla_\theta \pi_\theta(\alpha_k = m|s_k^{(i)}) = 0$.