Black-Box Importance Sampling

Qiang Liu
Dartmouth College

Jason D. Lee
University of Southern California

Abstract

Importance sampling is widely used in machine learning and statistics, but its power is limited by the restriction of using *simple* proposals for which the importance weights can be tractably calculated. We address this problem by studying *black-box importance sampling methods* that calculate importance weights for samples generated from any unknown proposal or black-box mechanism. Our method allows us to use better and richer proposals to solve difficult problems, and (somewhat counter-intuitively) also has the additional benefit of improving the estimation accuracy beyond typical importance sampling. Both theoretical and empirical analyses are provided.

1 Introduction

Efficient Monte Carlo methods are workhorses for modern Bayesian statistics and machine learning. Importance sampling (IS) and Markov chain Monte Carlo (MCMC) are two fundamental tools widely used when it is intractable to draw exact samples from the underlying distribution $p(x)$. IS uses an *simple* proposal distribution $q(x)$ to draw a sample $\{x_i\}$, and attaches it with a set of importance weights that are proportional to the probability ratio $p(x_i)/q(x_i)$. MCMC methods, on the other hand, rely on simulating Markov chains whose equilibrium distribution matches the target distribution.

Unfortunately, both importance sampling (IS) and MCMC have their own critical weaknesses. IS heavily relies on a good proposal $q(x)$ that closely matches the target distribution $p(x)$ to obtain accurate estimates. However, it is critically challenging, or even impossible, to design good proposals for high dimensional complex target distributions, given the restriction of using simple proposals. Therefore, alternative methods that do not require to calculate proposal probabilities would greatly enhance the power of IS, yielding efficient solutions for difficulty problems.

On the other hand, MCMC approximates the target distribution with an (often complex) distribution simulated from a large number of steps of Markov transitions, and has been widely used to solve complex problems. However, MCMC has a long-standing difficulty accessing its convergence, and one may get absurdly wrong results when using non-convergent results [e.g., Morris et al., 1996]. In addition, the computational cost of MCMC becomes critically expensive when the number of data instances is very large (a.k.a. the big data setting). A number of approximate versions of MCMC have been developed recently to deal with the big data issue [e.g., Welling and Teh, 2011, Alquier et al., 2016], but these methods usually no longer converge to the correct stationary distribution.

Motivated by combining the advantages of IS and MCMC, we study *black-box importance sampling* methods that can calculate importance weights for any given sample $\{x_i\}_{i=1}^n$ generated from arbitrary, unknown black-box mechanisms. Such methods allow us to use highly complex proposals that closely match the target distribution, without worrying about the computational tractability of the typical importance weights.

Interestingly, the black-box methods, despite using no information of the proposal distribution, can actually give better estimation accuracy than the typical importance sampling that leverages the proposal information. This appears to be a paradox (using less information yet getting better results), but is consistent with the arguments of O'Hagan [1987] that "Monte Carlo (that uses the proposal information) is fundamentally unsound", and the interesting results of Henmi et al. [2007], Delyon and Portier [2014] that certain types of approximate versions of IS weights reduce the variance over exact IS weights.

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

As an example of application, we apply black-box importance weights to samples simulated by a number of short Markov chains, in which MCMC helps provide a complex proposal that are "crudely" closely to the target distribution, and the black-box weights further refine the result. In this way, we obtain consistent estimators even from un-convergent MCMC results, or approximate MCMC transitions that appear commonly in big data settings.

Beyond MCMC, black-box IS can be used to refine many other approximation methods related to complex generation mechanisms, including variational inference with complex proposals [e.g., Rezende and Mohamed, 2015], bootstrapping [Efron, 2012] and perturb-and-MAP methods [Hazan et al., 2013, Papandreou and Yuille, 2011]. Further, we envision our method can find more applications in many areas where importance sampling or variance reduction plays an importance role, such as probabilistic inference in graphical model [e.g., Liu et al., 2015], variance reduction for variational inference [e.g., Wang et al., 2013, Ranganath et al., 2014] and policy gradient estimation [e.g., Greensmith et al., 2004], covariance shift in transfer learning [e.g., Sugiyama et al., 2008], off-policy evaluation reinforcement learning [e.g., Li et al., 2015], and stochastic optimization [e.g., Zhao and Zhang, 2015].

Our black-box importance weights are calculated by a convex quadratic optimization, based on minimizing a recently proposed kernelized Stein discrepancy that measures the goodness-of-fit of a sample to an unnormalized distribution [Oates et al., 2017, Chwialkowski et al., 2016, Liu et al., 2016]; this makes our method widely applicable for unnormalized distributions that widely appear in machine learning and statistics.

Related Works

Our method is closely related to Briol et al. [2015b,a], Oates et al. [2017], which combine Stein's identity with Bayesian Monte Carlo [O'Hagan, 1991, Ghahramani and Rasmussen, 2002] and control variates, respectively, and can also be interpreted as a form of importance weights similar to our method. The key difference is that the weights in their method can be negative and are not normalized to sum to one, while our approach explicitly optimizes the weights in the probability simplex, which helps provide more stable practical results as we illustrate both theoretically and empirically in our work. We provide a more throughout discussion in Section 3.3.

An alternative approach for black-box weights is to directly approximate the underlying proposal distribution q with an estimator $\hat{q}$ and use the corresponding ratio $p(x)/\hat{q}(x)$ as the importance weight. Henmi et al. [2007], Delyon and Portier [2014] showed that certain types of approximation $\hat{q}$ can improve, rather than deteriorate, the performance compared with the exact importance weight $p(x)/q(x)$. However, the method by Henmi et al. [2007] is not widely applicable since it requires to solve a maximum likelihood estimator in a parametric family that include the proposal distribution; The method in Delyon and Portier [2014] uses a kernel density estimator for q and tends to give unstable empirical results as we show in our experiments. Related to this, there is a literature in semi-supervised learning for covariance shifts [e.g., Nguyen et al., 2010, Sugiyama and Kawanabe, 2012] that estimates the density ratio $p(x)/q(x)$ given two samples $\{x_i\} \sim p$ and $\{y_i\} \sim q$, when both p and q are unknown.

There are also other directions where the advantages of IS and MCMC can be combined, including adaptive importance sampling [e.g., Martino et al., Botev et al., 2013, Beaujean and Caldwell, 2013, Yuan et al., 2013], and sequential Monte Carlo [e.g., Smith et al., 2013, Robert and Casella, 2013, Neal, 2001]. The black-box techniques can be combined with these methods to obtain more powerful, adaptive methods.

Preliminary and Notation Let $k(x, x') \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel; we denote by $k(x, \cdot)$ the one-variable function for each fixed x. The reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ of $k(x, x')$ is the closure of linear span $\{f \colon f = \sum_{i=1}^{m} a_i k(x, x_i), \quad a_i \in \mathbb{R}, \quad m \in \mathbb{N}, \ x_i \in \mathcal{X}\}$, equipped with an inner product $\langle f, \ g \rangle_{\mathcal{H}} = \sum_{ij} a_i b_j k(x_i, x_j)$ for $f = \sum_{i=1}^{m} a_i k(x, x_i)$ and $g = \sum_i b_i k(x, x_i)$. One can verify that such $\mathcal{H}$ has a *reproducing* property in that $f = \langle f, \ k(x, \cdot) \rangle_{\mathcal{H}}$. We use $\mathcal{O}(\cdot)$ for the Big O in probability notation.

2 Background: Kernelized Stein Discrepancy

We give a brief introduction to Stein's identity and kernelized Stein discrepancy (KSD) [Liu et al., 2016, Oates et al., 2017, Chwialkowski et al., 2016] which forms the foundation of our method.

Let $p(x)$ be a continuously differentiable (also called smooth) density supported on $\mathcal{X} \subseteq \mathbb{R}^d$. We say that a smooth function $f(x)$ is in the Stein class of $p(x)$ if

$$\int_{\mathcal{X}} \nabla_x (p(x) f(x)) dx = 0, \tag{1}$$

which can be implied by a zero boundary condition $\oint_{\partial X} p(x) f(x) dS = 0$ when $\mathcal{X}$ is bounded, or $\lim_{r \to \infty} \oint_{B_r} f(x) p(x) dS = 0$ when $\mathcal{X} = \mathbb{R}^d$, where B_r denotes the sphere with radius r. For $f(x)$ in the Stein

class of $p(x)$, Stein's identity shows

$$\begin{aligned} &\mathbb{E}_{x\sim p}[\boldsymbol{s}_p(x)f(x) + \nabla_x f(x)] = 0, \\ &\text{where} \qquad \boldsymbol{s}_p(x) = \nabla_x \log p(x), \end{aligned} \tag{2}$$

which is in fact a direct rewrite of (1) using the product rule of derivatives. We call $\boldsymbol{s}_p(x) \coloneqq \nabla_x \log p(x)$ the score function of $p(x)$. Note that calculating $\boldsymbol{s}_p(x)$ does not depend on the normalization constant in $p(x)$, that is, when $p(x) = f(x)/Z$ where Z is the normalization constant and is often critically difficult to calculate, we have $\boldsymbol{s}_p(x) = \nabla_x \log f(x)$, independent of Z. This property makes Stein's identity a powerful practical tool for handling unnormalized distributions that widely appear in machine learning and statistics.

We can "kernelize" Stein's identity with a smooth positive definite kernel $k(x, x')$ for which $k(x, x')$ is in the Stein class of $p(x)$ for each fixed $x' \in \mathcal{X}$ (we say such $k(x, x')$ is in the Stein class of p in this case). By first applying (2) on $k(x, x')$ with fixed x' and subsequently with fixed x, we can get the following kernelized version of Stein's identity:

$$\begin{aligned} &\mathbb{E}_{x\sim p}\big[k_p(x, x')\big] = 0, \quad \forall x' \in \mathcal{X}, \\ &\mathbb{E}_{x,x'\sim p}\big[k_p(x, x')\big] = 0, \end{aligned} \tag{3}$$

where x, x' are i.i.d. drawn from p and $k_p(x, x')$ is a new kernel function defined via

$$\begin{aligned} k_p(x, x') = {} & \boldsymbol{s}_p(x)^\top k(x, x')\boldsymbol{s}_p(x') + \boldsymbol{s}_p(x)^\top \nabla_{x'} k(x, x') \\ & + \boldsymbol{s}_p(x')^\top \nabla_x k(x, x') + \operatorname{trace}(\nabla_{x,x'} k(x, x')). \end{aligned} \tag{4}$$

See Theorem 3.5 of Liu et al. [2016]. We remark that $k_p(x, x')$ can be easily calculated with given $k(x, x')$ and $\boldsymbol{s}_p(x)$, even when $p(x)$ is unnormalized.

If we now replace the expectation $\mathbb{E}_p[\cdot]$ in (3) with $\mathbb{E}_q[\cdot]$ of a different smooth density $q(x)$ supported on $\mathcal{X}$, (3) would not equal zero; instead, it gives a non-negative discrepancy measure of p and q:

$$\mathbb{S}(q,\ p) = \mathbb{D}(q,\ p)^2 = \mathbb{E}_{x,x'\sim q}\big[k_p(x, x')\big] \geq 0, \tag{5}$$

where $\mathbb{D}(q,\ p)$ is called the kerenelized Stein discrepancy (KSD), and $\mathbb{S}(q,\ p)$ is the square of KSD, introduced for notation convenience. Here $\mathbb{E}_{x,x'\sim q}\big[k_p(x, x')\big]$ is always nonnegative because $k_p(x, x')$ can be shown to be positive definite if $k(x, x')$ is positive definite [e.g., Liu et al., 2016, Theorem 3.6].

In addition, one can further show that $\mathbb{S}(q,\ p)$ equals zero if and only if $p = q$ if $k(x, x')$ is strictly positive definite in certain sense: strictly integrally positive definite in Liu et al. [2016], and *cc*-universal in Chwialkowski et al. [2016] and Oates et al. [2016]. Meanwhile, $k_p(x, x')$ is obviously not strictly positive definite, since $p(x)$ is an eigenfunction with zero eigenvalue as suggested by (3). In fact, let $\mathcal{H}_p$ be the RKHS related to $k_p(x, x')$, then all the functions $h(x)$ in $\mathcal{H}_p$ are orthogonal to $p(x)$ in that $\mathbb{E}_p[h(x)] = 0$. Such $\mathcal{H}_p$ were first studied in Oates et al. [2017], in which it was used to define an infinite dimensional control variate for variance reduction.

One can further consider kernel $k_p^+(x, x') = k_p(x, x') + 1$, whose corresponding RKHS $\mathcal{H}_p^+$ consists of functions of form $h(x) + c$ with $h \in \mathcal{H}_p$ and c is a constant in $\mathbb{R}$. Therefore, $\mathcal{H}_p^+$ includes functions with arbitrary values of mean $\mathbb{E}_p h$. Oates et al. [2016] showed that $\mathcal{H}_p^+$ is dense in $L^2(\mathcal{X}, p)$ when $k(x, x')$ in *c*-universal and $\mathcal{X}$ is a compact subset of $\mathbb{R}^d$; see Oates et al. [2016] for more discussion.

3 Stein Importance Weights

Let $\{x_i\}_{i=1}^n$ be a set of points in $\mathbb{R}^d$ and we want to find a set of weights $\{w_i\}_{i=1}^n$, $w_i \in \mathbb{R}$, such that the weighted sample $\{x_i, w_i\}_{i=1}^n$ closely approximates the target distribution $p(x)$ in the sense that

$$\sum_{i=1}^n w_i h(x_i) \approx \mathbb{E}_p[h(x)],$$

for general test function $h(x)$. For this purpose, we define an empirical version of the KSD in (5) to measure the discrepancy between $\{x_i, w_i\}_{i=1}^n$ and $p(x)$,

$$\mathbb{S}(\{x_i, w_i\},\ p) = \sum_{i,j=1}^n w_i w_j k_p(x_i, x_j) = \boldsymbol{w}^\top \boldsymbol{K}_p \boldsymbol{w},$$

where $\boldsymbol{K}_p = \{k_p(x_i, x_j)\}_{i,j=1}^n$ and $\boldsymbol{w} = \{w_i\}_{i=1}^n$, and we assume the weights are self normalized, that is, $\sum_i w_i = 1$. We then select the optimal weights by minimizing the discrepancy $\mathbb{S}(\{x_i, w_i\},\ p)$,

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \left\{ \boldsymbol{w}^\top \boldsymbol{K}_p \boldsymbol{w}, \quad s.t. \quad \sum_{i=1}^n w_i = 1, \quad w_i \geq 0 \right\}, \tag{6}$$

where in addition to the normalization constraint $\sum_i w_i = 1$, we also restrict the weights to be non-negative; these two simple constraints have important practical implications as we discuss in the sequel. Note that the optimization in (6) is a convex quadratic programming that can be efficiently solved by off-the-shelf optimization tools. For example, both mirror descent and Frank Wolfe take $O(n^2/\epsilon)$ to find the optimum with ϵ-accuracy. Solving (6) does not require to know how the points $\{x_i\}_{i=1}^n$ are generated, and hence gives a *black-box* importance sampling.

Theoretically, minimizing the empirical KSD can be justified by the following bound.

Proposition 3.1. *Let $h(x)$ be a test function and $h - \mathbb{E}_p h \in \mathcal{H}_p$. Assume $\sum_{i=1}^n w_i = 1$, we have*

$$|\sum_{i=1}^n w_i h(x_i) \; - \; \mathbb{E}_p h| \; \leq \; C_h \sqrt{\mathbb{S}(\{x_i, w_i\}, \; p)}, \quad (7)$$

where $C_h = ||h - \mathbb{E}_p h||_{\mathcal{H}_p}$, which depends on h and p, but not on $\{x_i, w_i\}_{i=1}^n$.

Remark 1. Oates et al. [2017, Theorem 3] has a similar result which does not require $\sum_i w_i = 1$, but has a constant term larger than C_h when $\sum_i w_i = 1$ does hold. We propose to enforce $\sum_i w_i = 1$ because it gives exact estimation for constant functions $h(x) = c$, and is a common practice for importance sampling (which is referred to as self-normalized importance sampling). In our empirical results, we find that the normalized weights can significantly stabilize the algorithm, especially for high dimensional models.

2. One can show that the $\mathbb{S}(q, \; p)$ as defined in (5) can be treated as a square maximum mean discrepancy (MMD) between p and q, equipped with the (p-specific) kernel $k_p(x, x')$. In the light of this, bound (7) is a form of the worse case bounds of the kernel-based quadrature rules [e.g., Chen et al., 2010, Bach, 2015, Huszár and Duvenaud, 2012, Niederreiter, 2010]. The use of the special kernel $k_p(x, x')$ allows us to calculate the discrepancy tractably for general unnormalized distributions, in contrast with MMD with typical kernels that are intractable to calculate due to the need for evaluating the a term related to the expectation of the kernel under distribution p.

3.1 Practical Applications

Our method as summarized in Algorithm 1 can be used to refine any sample $\{x_i\}_{i=1}^n$ generated with arbitrary black-box mechanisms, and allows us to apply importance sampling in cases that are otherwise difficult. As an example, we can generate $\{x_i\}_{i=1}^n$ by simulating n parallel MCMC chains for m steps, where the length m of the chains can be smaller than what is typically used in MCMC, because it just needs to be large enough to bring the distribution of $\{x_i\}_{i=1}^n$ "roughly" close to the target distribution. This also makes it easy to parallelize the algorithm compared with running a single long chain. In practice, one may heuristically decide if m is large enough by checking the variance of the estimated weights $\{w_i\}_{i=1}^n$ (or the effective sample size). One can also simulate $\{x_i\}_{i=1}^n$ using MCMC with approximate translation kernels as these required for massive datasets [e.g., Welling and Teh, 2011, Alquier et al., 2016], so our method provides a new solution for big data problems.

We should remark that when $\{x_i\}_{i=1}^n$ is simulated from n independent MCMC initialized from a distribution $q_0(x)$, the weight $w_0(x) = n^{-1}p(x)/q_0(x)$ does provide a valid importance sampling weights in that $\sum_i w_0(x_i)h(x_i)$ gives an unbiased estimator [MacEachern et al., 1999, Theorem 6.1]. However, this weight does not update as we run more MCMC steps, and performs poorly in practice.

Algorithm 1 Stein Importance Sampling

1. Generate $\{x_i\}_{i=1}^n$ using any mechanism that is believed to resemble $p(x)$ (e.g., by running n independent MCMC chains for a small number of steps, or using parametric bootstrap).
2. Calculate importance weights $\{\hat{w}_i\}_{i=1}^n$ by (6).
3. Calculate $\sum_i \hat{w}_i h(x_i)$ to approximate $\mathbb{E}_p[h]$for test function h.

There are many other cases where black-box IS can be found useful. For example, we can simulate $\{x_i\}_{i=1}^n$ from bootstrapping or perturbed maximum *a posteriori* (MAP) [Papandreou and Yuille, 2011, Hazan et al., 2013], that is, $x_i = \arg\max_x \tilde{p}(x)$ where $\tilde{p}(x)$ is a perturbed version of $p(x)$, or the bootstrapping likelihood. The idea of using importance weighted bootstrapping to carry out Bayesian calculation has been discussed before [e.g., Efron, 2012], but was limited to simple cases when the bootstrap distribution is computable. Black-box IS can also be used to refine the results of variational inference [e.g., Wainwright and Jordan, 2008], especially for the cases with complex variational proposal distributions [e.g., Salimans et al., 2015, Rezende and Mohamed, 2015].

3.2 Convergence Rate

Our procedure does not assume the generation mechanism of $\{x_i\}_{i=1}^n$, but if $\{x_i\}_{i=1}^n$ is indeed generated "nicely", error bounds can be easily established using Proposition 3.1: if there exists a set of "reference" positive normalized weights $\{w_i^*\}_{i=1}^n$ such that $\mathbb{S}(\{x_i, \; w_i^*\}, \; p) = \mathcal{O}(n^{-\delta})$, then the mean square error of our estimator with weight $\{\hat{w}_i\}_{i=1}^n$ returned by (6) should also be $\mathcal{O}(n^{-\delta})$ by following (6) and (7).

To gain more intuition, assume $k_p(x, x')$ has a set of eigenfunctions $\{\phi_\ell(x)\}$ and eigenvalues $\{\lambda_\ell\}$ such that $k_p(x, x') = \sum_\ell \lambda_\ell \phi_\ell(x)\phi_\ell(x')$, then we have

$$\begin{aligned} |\sum_i \hat{w}_i h(x_i) - \mathbb{E}_p h|^2 &\leq C_h^2 \; \mathbb{S}(\{x_i, \hat{w}_i\}, \; p) \\ &= C_h^2 \; \sum_\ell \lambda_\ell \; (\sum_i w_i \phi_\ell(x_i) - \mathbb{E}_p \phi_\ell)^2, \end{aligned}$$

where $C_h = ||h - \mathbb{E}_p h||_{\mathcal{H}_p}$ and we used the fact that $\mathbb{E}_p \phi_\ell = 0$ since $\phi_\ell \in \mathcal{H}_p$. Therefore, it is enough to find a set of positive and normalized reference weights

whose error on estimating $\mathbb{E}_p\phi_\ell$ is low. Note that such reference weight does not necessarily need to be practically computable to establish the bound.

As an obvious example, when $\{x_i\}_{i=1}^n$ is i.i.d. drawn from an (unknown) proposal distribution $q(x)$, the typical importance sampling weight $w_i^* \propto p(x_i)/q(x_i)$ (up to the normalization) can be used as a reference weight to establish an $\mathcal{O}(n^{-1/2})$ error rate as the typical Monte Carlo methods have.

Theorem 3.2. *Assume $h - \mathbb{E}_p h \in \mathcal{H}_p$ and $\{x_i\}_{i=1}^n$ is i.i.d. drawn from $q(x)$ with the same support as $p(x)$. Define $w_*(x) = p(x)/q(x)$ and assume $\mathbb{E}_{x\sim p}(|w_*(x)k_p(x,x)|) < \infty$, and $\mathbb{E}_{\{x,x'\}\sim p}[w_*(x)w_*(x')k_p(x,x')^2] < \infty$. For $\{\hat{w}_i\}_{i=1}^n$ defined in (6), we have*

$$|\sum_{i=1}^n \hat{w}_i h(x_i) - \mathbb{E}_p h| = \mathcal{O}(n^{-1/2}).$$

Interestingly, it turns out the typical importance weight $w_*(x) \propto p(x)/q(x)$ is not the best possible reference weight; better options can be constructed using various variance reduction techniques to give convergence rates better than the typical $\mathcal{O}(n^{-1/2})$ rate.

Theorem 3.3. *Assume $\{x_i\}_{i=1}^n$ is i.i.d. drawn from $q(x)$ and $w_*(x) = p(x)/q(x)$. Let $\{\phi_\ell\}_{\ell=1}^\infty$ be the set of orthogonal eigenfunctions w.r.t. $p(x)$ with eigenvalues $\{\lambda_\ell\}_{\ell=1}^\infty$. Assume all the following quantities are upper bounded by M uniformly for $\forall x \in \mathcal{X}$: $\sum_{\ell=1}^\infty \lambda_\ell$, $w_*(x)$, $|\phi(x)|$, $\max_{\ell,\ell'} \mathrm{var}_{x\sim q}[w_*(x)^2\phi_\ell(x)\phi'_\ell(x)]$, we have*

$$\mathbb{E}_{\boldsymbol{x}\sim q}\big[|\sum_i \hat{w}_i h(x_i) - \mathbb{E}_p h|^2\big]^{1/2} = \mathcal{O}\Big(n^{-(1+\alpha)/2}\Big),$$

where α is a number that satisfies $0 < \alpha \le 1$ and is decided by the bound M and the decay of the eigenvalues $\mathcal{R}(n) = \sum_{\ell>n}\lambda_\ell$ of kernel $k_p(x,x')$. See Theorem B.5 in Appendix for more details.

The proof of Theorem 3.3 (see Section 2.2 in Appendix) is based on constructing a reference weight using a control variates method based on the orthogonal basis functions $\{\phi_\ell\}$. Our constructed reference weights can be treated as a perturbed version of the typical importance weights $w_*(x) \propto p(x)/q(x)$ as used in Theorem (3.2), where the perturbation is introduced to cancel the estimation error and increase the accuracy. Since this reference weights concentrate around $w_* \propto p(x)/q(x)$ which is positive, we can zero out its negative values without much impact on the error bound. This provides a justification on the non-negative garrote constraint, and allows us to construct a set of non-negative reference weights for our proof.

Similar theoretical analysis can be found in Briol et al. [2015b], Bach [2015]. In particular, Briol et al. [2015b] used a similar "reference weight" idea to establish a convergence rate for Bayesian Monte Carlo. The main technical challenge in our proof is to make sure that the reference weight satisfies the non-negative and self-normalization constraints. Section 2.2 in Appendix provides more detailed discussions.

3.3 Other "Super-Efficient" Weights

We review several other types of "super-efficient" weights that also give better convergence rates than the typical $\mathcal{O}(n^{-1/2})$ rate; this includes Bayesian Monte Carlo and the related (linear) control variates method, as well as methods based on density approximation of the proposal distributions, which can be interpreted as *multiplicative* control variates [Nelson, 1987] that reduce the variance.

Bayesian Monte Carlo and Control Variates Bayesian Monte Carlo [O'Hagan, 1991, Ghahramani and Rasmussen, 2002] was originally developed to evaluate integrals using Bayesian inference procedure with Gaussian prior, which turns out to be equivalent to a weighted form $\sum_i w_i h(x_i)$ with w_i being a set of weights independent of the test function h; unlike our method, these weights are not normalized to sum to one and can take negative values.

From a RKHS perspective, one can interpret Bayes MC as approximating $\mathbb{E}_p[h(x)]$ with $\mathbb{E}_p[\hat{h}(x)]$ where $\hat{h}(x)$ is an approximation of $h(x)$ constructed by kernel linear regression based on the data-value pair $\{x_i, h(x_i)\}_{i=1}^n$. Let $k_0(x,x')$ be the kernel used in Bayes MC, then one can show that Bayes MC estimate equals $\sum_i \hat{w}_i h(x_i)$ with $\hat{\boldsymbol{w}} = [\hat{w}_i]_{i=1}^n = (\boldsymbol{K}_0 + \lambda I)^{-1}\boldsymbol{b}$, where $\boldsymbol{K}_0 = [k_0(x_i,x_j)]_{ij}$ and $\boldsymbol{b} = [\mathbb{E}_{x\sim p}(k_0(x,x_i))]_{i=1}^n$, and λ a regularization coefficient. Equivalently, Bayes MC can be treated as minimizing the maximum mean discrepancy (MMD) between $\{x_i, w_i\}$ and p, with a form of

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \big\{\boldsymbol{w}^\top \boldsymbol{K}_0 \boldsymbol{w} - 2\boldsymbol{b}^\top \boldsymbol{w} + \lambda||\boldsymbol{w}||_2^2\big\}.$$

One of the main difficulty of Bayesian MC, however, is that it depends on $\boldsymbol{b} = \mathbb{E}_p[k_0(x,x')]$, which can be intractable to calculate for complex $p(x)$.

The control variates method [e.g., Liu, 2008] also relies on a (kernel) linear regressor $\hat{h}(x)$, but estimates $\mathbb{E}_p h$ with a bias-correction term $\frac{1}{n}\sum_{i=1}^n (h(x_i) - \hat{h}(x_i)) + \mathbb{E}_p[\hat{h}(x_i)]$, which can also be rewritten into a weighted form. Note that when $\lambda = 0$ and $\boldsymbol{K}_0$ is strictly positive definite, the $\hat{h}(x_i)$ becomes an interpolation of $h(x)$ (i.e., $h(x_i) = \hat{h}(x_i)$), and control variates and Bayes

MC becomes equivalent. In control variates, one can also use only a subset of the data to estimate $\hat{h}(x)$ and use the remaining data to estimate the expectation of the difference $h(x) - \hat{h}(x)$; this ensures the resulting estimator is unbiased.

Closely related to our work, Oates et al. [2017] and Briol et al. [2015b] proposed to use the Steinalized kernel $k_p^+(x, x') = k_p(x, x') + 1$ in control variates and Bayesian MC, respectively,[1] for which $\boldsymbol{b} = \mathbb{E}_{x\sim p}[k_p^+(x, x')] = 1$. We can show that their method is equivalent to using the following weight

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \{\boldsymbol{w}^\top \boldsymbol{K}_p \boldsymbol{w} + (\sum_i w_i - 1)^2 + \lambda ||w||_2^2\}.$$

This form is similar to our (6), but does not enforce the non-negative garrote constraint [Breiman, 1995] and replacing the normalization constraint $\sum_i w_i = 1$ with a quadratic regularization with regularization coefficient of one. Here the L2 penalty $\lambda||w||_2^2$ is necessary for ensuring numerical stability in practice. In our case, it is the non-negative constraint that helps stabilize the optimization problem, without needing to specify a regularization parameter.

Approximating the Proposal Distribution Another (perhaps less well known) set of methods are based on replacing the importance weight $w_*(x) = p(x)/q(x)$ with an approximate version $\tilde{w}(x) = p(x)/\hat{q}(x)$, where $\hat{q}(x)$ is an estimator of proposal density $q(x)$ from $\{x_i\}_{i=1}^n$. While we may naturally expect that such approximation would decrease the accuracy compared with the typical IS that uses the exact $q(x)$, surprising results [Henmi et al., 2007, Delyon and Portier, 2014] show that in certain cases the approximate weights $\tilde{w}(x)$ actually improve the accuracy. To gain an intuition why this can be the case, observe that we have $\tilde{w}(x) = [p(x)/q(x)] \cdot [q(x)/\hat{q}(x)]$, where the second term $q(x)/\hat{q}(x)$ may acts as a (multiplicative) control variate [Nelson, 1987] which can decrease the variance if it is negatively correlated with the other parts of the estimator. For asymptotic analysis, it is common to expand multiplicative control variates using Taylor expansion, which reduces it to linear control variates.

In particular, Henmi et al. [2007] showed that when $q(x)$ is embedded in a parametric family $\mathcal{Q} = \{q(x \mid \theta), \theta \in \Theta\}$, replacing $w_*(x)$ with the approximate weight $\tilde{w}(x) = p(x)/\hat{q}(x)$, where $\hat{q}$ is the maximum likelihood estimator of $q(x)$ within $\mathcal{Q}$, would guarantee to decrease the asymptotic variance compared with the standard IS. The result in Delyon and Portier [2014] forms a nonparametric counterpart of Henmi et al. [2007], in which it is shown that taking $\hat{q}(x)$ to be a leave-one-out kernel density estimator of $q(x)$ would give super-efficient error rate $\mathcal{O}(n^{-(1+\alpha)/2})$ where α is a positive number that depends on the smoothness of $q(x)$ and $p(x)h(x)$.

[1] $k_p(x, x')$ can be not used directly in Bayesian Monte Carlo since it only includes functions with zero mean.

4 Experiments

We empirically evaluate our method and compare it with the methods mentioned above, first on an illustrative toy example based on Gaussian mixture, and then on Bayesian probit regression. The methods we tested all have a form of $\sum_i w_i h(x_i)$, where the weights w_i are decided by one of the following algorithms:

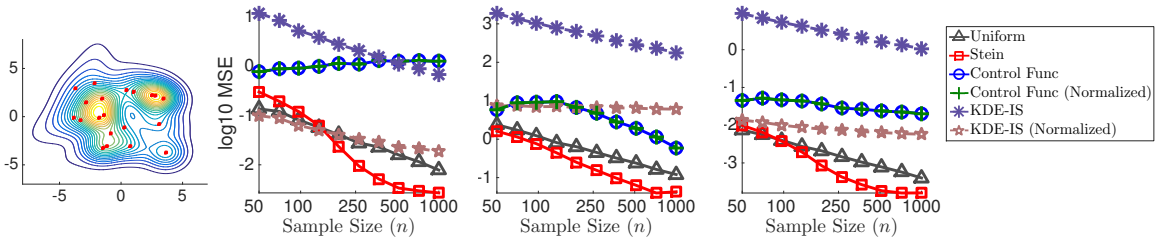
1. Uniform weights $w_i = 1/n$ (`Uniform`).

2. Our method that solves (6) (referred as `Stein`), for which we use RBF kernel $k(x, x') = \exp(-\frac{1}{h}||x - x'||_2^2)$; the bandwidth h is heuristically chosen to be the median of the pairwise square distance of data $\{x_i\}_{i=1}^n$ as suggested by Gretton et al. [2012].

3. The control functional method `Control Func` following the empirical guidance in Oates et al. [2017], which is also equivalent to Bayesian MC with kernel $k_p^+(x, x') = k_p(x, x') + 1$. Note that the weights $\{w_i\}$ in this method may be negative and do not necessarily sum to one. We also test a modified version of it $\sum_i w_i h(x_i) / \sum_i w_i$ that normalizes the weights and refer it as `Control Func (Normalized)`. The kernel $k(x, x')$ and the bandwidth are taken to be the same as our method. We follow Oates et al. [2017]'s guidance to select that an L2 regularization coefficient to stabilize the algorithm.

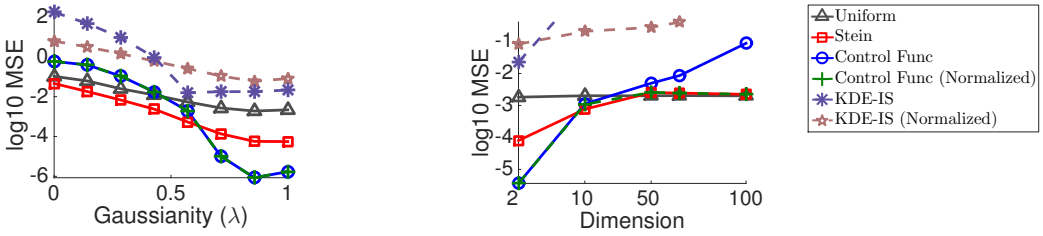
4. The kernel density estimator (KDE) based method by Delyon and Portier [2014] (`KDE`), which uses weights $w_i = n^{-1} p(x_i)/\hat{q}_i(x_i)$, where $\hat{q}_i(x)$ is a leave-one-out KDE of form $\hat{q}_i(x) = \sum_{j\neq i} k(x, x_j)/n$. We report the result when using RBF kernel with bandwidth decided by the rule of thumb $h = \hat{\sigma}\big(\frac{d2^{d+5}\Gamma(d/2+3)}{(2d+1)n}\big)^{1/(4+d)}$, where $\hat{\sigma}$ is the standard deviation of $\{x_i\}_{i=1}^n$ and d is the dimension of x. We also tested the choice of kernel and bandwidth suggested in Delyon and Portier [2014] but did not find consistent improvement. Similar to the case of the control functional method, we also test a self-normalized version of `KDE` and denote it by `KDE (Normalized)`.

We evaluate these methods by comparing their mean square errors (MSE) for estimating $\mathbb{E}_p h_j$, with $h_j(x)$ taken to be x^j, $(x^j)^2$ or $\cos(\omega x^j + b)$, where x^j is the j-th component of vector x; we calculate the MSE for each h_j and report the average MSE over $j = 1, \ldots, d$. For $h_j(x) = \cos(\omega x^j + b)$, we draw $\omega \sim \mathcal{N}(0, 1)$ and $b \sim \text{Uniform}([0, 2\pi])$ and average the MSE over 20 random trials.



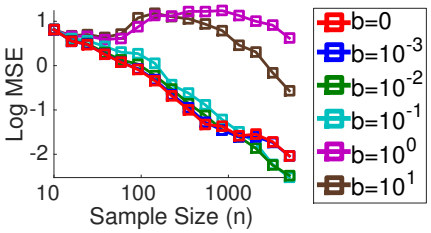
(a) Contour of $p(x)$ (b) $\mathbb{E}(x^j)$ (c) $\mathbb{E}((x^j)^2)$ (d) $\mathbb{E}(\cos(\omega x^j + b))$

Figure 1: Gaussian Mixture Example. (a) The contour of the distribution $p(x)$ that we use; the red dots represent the centers of the mixture components. The sample $\{x_i\}$ is i.i.d. drawn from $p(x)$ itself. (b) - (c) The MSE of the different weighting schemes for estimating $\mathbb{E}_p h_j$, when $h_j(x)$ equals x^j, $(x^j)^2$, and $\cos(\omega x^j + b)$, respectively, where x^j is the j-th component of vector x; the MSE is calculated for each $j = 1, \ldots, d$ and the average MSE is reported. For $h = \cos(\omega x^j + b)$ in (d), we draw $\omega \sim \mathcal{N}(0, 1)$ and $b \sim \text{Uniform}([0, 2\pi])$, averaged over 20 trials.



(a) Varying Gaussianity (b) Varying Dimension

Figure 2: (a) Results on $p(x|\lambda)$ where λ indexes the Gaussianity: $p(x|\lambda)$ equals $\mathcal{N}(0, 1)$ when $\lambda = 1$ and it reduces to the $p(x)$ in Figure 1(a) when $\lambda = 0$. (b) Results on standard Gaussian distribution with increasing dimensions. The sample size is fixed to be $n = 100$ in both (a) and (b). The MSE is for estimating $\mathbb{E}((x^j)^2)$, averaged on different coordinates $j = 1, \ldots, d$.



(c) Stein with relaxed $w_i \geq -b$ constraint

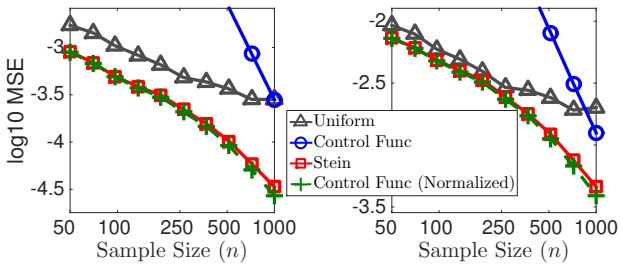
Figure 3: The result of our method on the $p(x)$ in Figure 1(a) when the non-negative constraint $w_i \geq 0$ replaced by a general lower bound $w_i \geq -b$ with different values of b. The MSE is for estimating $\mathbb{E}((x^j)^2)$, averaged on different coordinates j.

Gaussian Mixture We start with a 2-D Gaussian mixture distribution $p(x) = \sum_j \beta_j \mathcal{N}(x;\ \mu_j, \sigma_j^2)$ with 20 randomly located mixture components shown in Figure 1(a), and draw $\{x_i\}_{i=1}^n$ from $p(x)$ itself. The MSEs for estimating $\mathbb{E}_p h$ with different $h(x)$ as the sample size n increases are shown in Figure 1(b)-(d), where we generally find that our method tends to perform among the best.

In Figure 2(a), we study the performance of the algorithms on distributions with different Gaussianity, where we replace $p(x)$ with a series of distributions $p(x \mid \lambda)$ whose random variable is $(1 - \lambda)x + \lambda\xi$ where $x \sim p$, $\xi \sim \mathcal{N}(0, 1)$ and $\lambda \in [0, 1]$ controls the Gaussianity of $p(x \mid \lambda)$: it reduces to $p(x)$ when $\lambda = 0$ and equals $\mathcal{N}(0, 1)$ when $\lambda = 1$. We observe that `Stein` tends to perform the best when the distribution has high non-Gaussianity, but is suboptimal compared with `Control Func` when the distribution is close to Gaussian.

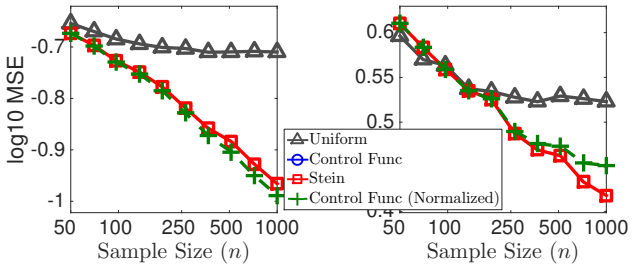
In Figure 2(b), we consider how the different algorithms scale to high dimensions by setting $p(x)$ to be the standard Gaussian distribution with increasing dimensions. We generally find that our `Stein` tends to perform among the best under the different settings, except for low dimensional standard Gaussian under which `Control Func` performs the best. The self-normalized versions of `KDE` and `Control Func` can help to stabilize the algorithm in various cases, for example, `KDE (Normalized)` significantly improves over `KDE` in all the cases, and `Control Func (Normalized)` is significantly better than `Control Func` in high dimensional cases as shown in Figure 2(b).

Figure 3 shows the performance of our method with the non-negativity constraint ($w_i \geq 0$) replaced by ($w_i \geq -b$) where b is a positive number that takes different



(a) Estimating $\mathbb{E}(x^j)$ (b) Estimating $\mathbb{E}((x^j)^2)$

Figure 4: Results of Bayesian probit model with simulated data. We generate $\{x_i\}_{i=1}^n$ by simulating n parallel chains of stochastic gradient Langevin Dynamics with a mini-batch size of 100 for 100-steps. `KDE` and `KDE (Normalized)` perform significantly worse in this case, and are not show in the figure.



(a) Estimating $\mathbb{E}(x^j)$ (b) Estimating $\mathbb{E}((x^j)^2)$

Figure 5: Results of Bayesian probit model on the Covtype dataset. We generate $\{x_i\}_{i=1}^n$ by simulating n parallel chains of stochastic gradient Langevin Dynamics with a mini-batch size of 100 for 1000-steps. The unnormalized `Control Func`, as well as `KDE` and `KDE (Normalized)` perform significantly worse in this case, and are not show in the figure.

values. We find that the result of $w_i \geq 0$ generally performs the best when n is small (e.g., $n < 1000$), but is slightly suboptimal when n is large. Because the stability in the small n case is more practically important than the large n case, given that the absolute difference on MSE would be negligible in the large n region, we think enforcing $w_i \geq 0$ is a simple and good practical procedure.

Bayesian Probit Model We consider the Bayesian probit regression model for binary classification. Let $D = \{\chi_\ell, \zeta_\ell\}_{\ell=1}^N$ be a set of observed data with feature vector χ_ℓ and binary label $\zeta_\ell \in \{0, 1\}$. The distribution of interest is $p(x) := p(D|x)p_0(x)$ with

$$p(D|x) = \prod_{\ell=1}^{N} \left[\zeta_\ell \Phi(x^\top \chi_\ell) + (1 - \zeta_\ell)(1 - \Phi(x^\top \chi_\ell))\right],$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution, and $p_0(x) = \mathcal{N}(x; 0,\ 0.1 \times I)$ is the prior, where I is the identity matrix.

To test our method, we simulate $\{x_i\}_{i=1}^n$ by running n parallel chains of stochastic Langevin dynamics [Welling and Teh, 2011]. Since this method is an inexact MCMC, its stationary distribution is different from the target distribution $p(x)$, and as a result, directly averaging $\{x_i\}_{i=1}^n$ with uniform weights (`Unif`) can give relatively poor results (although it is still possible to get consistent estimation for $\mathbb{E}_p h$ by averaging the temporal trajectory with a properly decreasing step-size and weighting scheme as shown by Teh et al. [2016]).

We apply the black-box weights to refine the result of stochastic Langevin dynamics. Figure 4 shows the result on a small simulated dataset with 100 data instances and 10 features. We can find that `Stein` and `Control Func (Normalized)` significantly improve the performance over `Unif`. Interestingly, we find that the unnormalized `Control Func`, as well as `KDE` and `KDE (normalized)` (not show in the figure) perform significantly worse in this case.

Figure 5 shows the result on the Forest Covtype dataset from the UCI machine learning repository [Bache and Lichman, 2013]; it has 54 features, and is reprocessed to get binary labels following Collobert et al. [2002]. For our experiment, we take the first 10,000 data points, so that it is feasible to evaluate the ground truth with No-U-Turn Sampler (NUTS) [Hoffman and Gelman, 2014]. We again find that `Stein` and `Control Func (Normalized)` improves over the uniform weights, and the unnormalized `Control Func` and `KDE` and `KDE (normalized)` again perform significantly worse and are not shown in the figure.

5 Conclusion

We propose a *black-box importance sampling* method that calculates importance weights without knowing the proposal distribution, which also has the additional benefit of providing variance reduction. We expect our method provides a powerful tool for solving many difficult problems were previously intractable via importance sampling.

Acknowledgment This work is supported in part by NSF CRII 1565796. We thank Lester Mackey, Chris Oates and Francois-Xavier Briol for their valuable comments.

References

Pierre Alquier, Nial Friel, Richard Everitt, and Aidan Boland. Noisy Monte Carlo: Convergence of markov chains with approximate transition kernels. *Statistics and Computing*, 26(1-2):29–47, 2016.

Francis Bach. On the equivalence between quadrature rules and random features. *arXiv preprint arXiv:1502.06800*, 2015.

K. Bache and M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Frederik Beaujean and Allen Caldwell. Initializing adaptive importance sampling with markov chains. *arXiv preprint arXiv:1304.7808*, 2013.

Zdravko I Botev, Pierre L'Ecuyer, and Bruno Tuffin. Markov chain importance sampling with applications to rare event probability estimation. *Statistics and Computing*, 23(2):271–285, 2013.

Leo Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995.

François-Xavier Briol, Chris Oates, Mark Girolami, and Michael A Osborne. Frank-Wolfe Bayesian Quadrature: Probabilistic integration with theoretical guarantees. In *NIPS*, 2015a.

François-Xavier Briol, Chris Oates, Mark Girolami, Michael A Osborne, Dino Sejdinovic, et al. Probabilistic integration: A role for statisticians in numerical analysis? *arXiv preprint http://arxiv.org/abs/1512.00933*, 2015b.

Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. In *UAI*, 2010.

Kacper Chwialkowski, Heiko Strathmann, and Arthur Gretton. A kernel test of goodness of fit. In *ICML*, 2016.

Ronan Collobert, Samy Bengio, and Yoshua Bengio. A parallel mixture of SVMs for very large scale problems. *Neural computation*, 14(5):1105–1114, 2002.

Bernard Delyon and François Portier. Integral approximation by kernel smoothing. *arXiv preprint arXiv:1409.0733*, 2014.

Bradley Efron. Bayesian inference and the parametric bootstrap. *The annals of applied statistics*, 6(4):1971, 2012.

Zoubin Ghahramani and Carl E Rasmussen. Bayesian Monte Carlo. In *NIPS*, 2002.

Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Tamir Hazan, Subhransu Maji, and Tommi Jaakkola. On sampling from the gibbs distribution with random maximum a-posteriori perturbations. In *NIPS*, pages 1268–1276, 2013.

Masayuki Henmi, Ryo Yoshida, and Shinto Eguchi. Importance sampling via the estimated sampler. *Biometrika*, 94(4):985–991, 2007.

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

Ferenc Huszár and David Duvenaud. Optimally-weighted Herding is Bayesian quadrature. In *UAI*, 2012.

Lihong Li, Rémi Munos, and Csaba Szepesvári. Toward minimax off-policy value estimation. In *AISTATS*, 2015.

Jun S Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

Qiang Liu, John W Fisher III, and Alexander T Ihler. Probabilistic variational bounds for graphical models. In *NIPS*, pages 1432–1440, 2015.

Qiang Liu, Jason D Lee, and Michael I Jordan. A kernelized stein discrepancy for goodness-of-fit tests and model evaluation. In *ICML*, 2016.

Steven N MacEachern, Merlise Clyde, and Jun S Liu. Sequential importance sampling for nonparametric Bayes models: The next generation. *Canadian Journal of Statistics*, 27(2):251–267, 1999.

Luca Martino, Victor Elvira, David Luengo, and Jukka Corander. Layered adaptive importance sampling. *Statistics and Computing*, pages 1–25.

RD Morris, X Descombes, and J Zerubia. The Ising/Potts model is not well suited to segmentation tasks. In *Digital Signal Processing Workshop Proceedings, 1996., IEEE*, pages 263–266. IEEE, 1996.

Radford M Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

Barry L Nelson. On control variate estimators. *Computers & Operations Research*, 14(3):219–225, 1987.

XuanLong Nguyen, Martin J Wainwright, Michael Jordan, et al. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *Information Theory, IEEE Transactions on*, 56(11):5847–5861, 2010.

Harald Niederreiter. *Quasi-Monte Carlo Methods*. Wiley Online Library, 2010.

Chris J Oates, Jon Cockayne, François-Xavier Briol, and Mark Girolami. Convergence rates for a class of estimators based on stein's identity. *arXiv preprint arXiv:1603.03220*, 2016.

Chris J Oates, Mark Girolami, and Nicolas Chopin. Control functionals for Monte Carlo integration. *Journal of the Royal Statistical Society, Series B*, 2017.

Anthony O'Hagan. Monte carlo is fundamentally unsound. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 36(2/3):247–249, 1987.

Anthony O'Hagan. Bayes–Hermite quadrature. *Journal of statistical planning and inference*, 29(3):245–260, 1991.

George Papandreou and Alan L Yuille. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *ICCV*, pages 193–200. IEEE, 2011.

Rajesh Ranganath, Sean Gerrish, and David M Blei. Black box variational inference. In *AISTATS*, pages 814–822, 2014.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, 2015.

Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

Tim Salimans, Diederik P Kingma, and Max Welling. Markov chain Monte Carlo and variational inference: Bridging the gap. In *ICML*, 2015.

Adrian Smith, Arnaud Doucet, Nando de Freitas, and Neil Gordon. *Sequential Monte Carlo Methods in Practice*. Springer Science & Business Media, 2013.

Masashi Sugiyama and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT Press, 2012.

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*, 2008.

Yee Whye Teh, Alexandre H Thiery, and Sebastian J Vollmer. Consistency and fluctuations for stochastic gradient langevin dynamics. *Journal of Machine Learning Research*, 17, 2016.

Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2): 1–305, 2008.

Chong Wang, Xi Chen, Alex J Smola, and Eric P Xing. Variance reduction for stochastic gradient optimization. In *Advances in Neural Information Processing Systems*, pages 181–189, 2013.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *ICML*, 2011.

Xiukai Yuan, Zhenzhou Lu, Changcong Zhou, and Zhufeng Yue. A novel adaptive importance sampling algorithm based on markov chain and low-discrepancy sequence. *Aerospace Science and Technology*, 29(1):253–261, 2013.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling. *ICML*, 2015.