Amortized Monte Carlo Integration

Adam Goliński
University of Oxford

Yee Whye Teh
University of Oxford

Frank Wood
University of British Columbia

Tom Rainforth
University of Oxford

Abstract

Current approaches to amortizing Bayesian inference focus solely on approximating the posterior distribution. Typically, this approximation is in turn used to calculate expectations for one or more target functions. In this paper, we address the inefficiency of this computational pipeline when the target function(s) are known upfront. To this end, we introduce a method for *amortizing Monte Carlo integration*. Our approach operates in a similar manner to amortized inference, but tailors the produced amortization artifacts to maximize the accuracy of the resulting expectation calculation(s). We show that while existing approaches have fundamental limitations in the level of accuracy that can be achieved for a given run time computational budget, our framework can, at least in theory, produce arbitrary small errors for a wide range of target functions with $\mathcal{O}(1)$ computational cost at run time. Furthermore, our framework allows not only for amortizing over possible datasets, but also over possible target functions.

1 Introduction

At its core, Bayesian modeling is rooted in the calculation of expectations: the eventual aim of modeling is typically to make a decision, or to construct predictions for unseen data, both of which take the form of an expectation under the posterior (Robert, 2007). The eventual aim of the vast majority of Bayesian inference problems can thus be summarized in the form of one or more expectations $\mathbb{E}_{p(x|y)}\left[f(x)\right]$, where $f(x)$ is a target function and $p(x|y)$ is the posterior distribution on x for some data y, which we typically only know up to a normalizing constant $p(y)$. Sometimes $f(x)$ is not known up front, or we care about many different $f(x)$, such that it is convenient to just approximate $p(x|y)$ upfront, e.g. in the form of Monte Carlo samples, and then later use this to calculate estimates, rather than address the target expectations directly.

However, it is often the case in practice that a particular target function, or class of target functions, is known a priori. For example, in decision-based settings $f(x)$ takes the form of a loss function. It has been well established in the literature that in such *target-aware* settings the aforementioned pipeline of first approximating $p(x|y)$ and then using this as a basis for calculating $\mathbb{E}_{p(x|y)}\left[f(x)\right]$ is suboptimal as it ignores relevant information in $f(x)$ (Owen, 2013; Lacoste–Julien et al., 2011). As we will later show, the potential gains in such situations can be substantial.

Although it is all too often overlooked, how to adjust for target-aware settings has been well studied in the fixed-dataset context (Hesterberg, 1988; Wolpert, 1991; Oh & Berger, 1992; Evans & Swartz, 1995; Lacoste–Julien et al., 2011). In this paper, we extend these ideas to *amortized* inference settings (Stuhlmüller et al., 2013; Kingma & Welling, 2014; Ritchie et al., 2016; Paige & Wood, 2016; Le et al., 2017, 2018; Maddison et al., 2017; Naesseth et al., 2018), wherein one looks to "compile away" the cost of inference across different possible datasets by learning an artifact that can be used to assist the inference process at run time for a given dataset.

Typically, this amortization artifact takes the form of a parametrized proposal, $q(x;\varphi(y))$, which takes in data y and regresses these to proposal parameters $\varphi(y)$, generally using a deep neural network. Though the exact process varies with context, the inference network is usually trained either by drawing latent-data sample pairs from a fixed joint distribution $p(x,y)$ (Ritchie et al., 2016; Paige & Wood, 2016; Le et al., 2017), or as part of a stochastic variational inference scheme (Hoffman et al., 2013; Kingma & Welling, 2014; Rezende et al., 2014). Once trained, it provides an efficient means of approximately sampling from the posterior of a particular dataset, e.g. using importance sampling.

Our first contribution is highlighting that the shortcomings of existing amortized inference approach for target-aware

problems are analogous to those of the single-dataset setting: even if the inference network fully encapsulates the true posterior, the resulting estimator is still sub-optimal.

Our second is in introducing AMCI, a framework for performing *amortized Monte Carlo integration*. Though still based around learning amortized proposals distributions, AMCI varies from standard amortized inference approaches in three respects. Firstly it operates in a target-aware fashion, incorporating information about $f(x)$ into the amortization artifacts, increasing the efficiency at run time. Secondly, rather than relying purely on self-normalization, AMCI employs two separate proposals for estimating the unnormalized target integral $\mathbb{E}_{p(x)}\left[f(x)p(y|x)\right]$ and the marginal likelihood $\mathbb{E}_{p(x)}\left[p(y|x)\right]$. This means that it can, at least in principle, return single sample estimates with arbitrarily low mean squared error, unlike standard approaches whose attainable MSE is lower bounded for a given $f(x)$ and number of samples (Owen, 2013). Finally, to account for cases in which multiple possible target functions may be of interest, AMCI also allows for amortization of parametrized functions $f(x;\theta)$ through the use of pseudo prior $p(\theta)$.

2 Background

2.1 Importance sampling

Importance Sampling (IS) is a common sampling method that forms the key building block for many more advanced inference schemes. The standard Importance Sampling (IS) approach requires the target distribution $p(x)$ to be a normalized probability distribution.

$$\mu := \mathbb{E}_{p(x)}\left[f(x)\right] = \int f(x)\frac{p(x)}{q(x)}q(x)dx$$
$$\approx \hat{\mu} := \frac{1}{N}\sum_{n=1}^{N} f(x_n)w_n \text{ where } x_n \sim q(x) \quad (1)$$

and $w_n := p(x_n)/q(x_n)$ is known as an importance weight.

When the target distribution is not normalized we can self-normalise the estimate by dividing by the unbiased normalising constant estimate $\hat{Z} := \frac{1}{N}\sum_{n=1}^{N} w_n$. This approach is called self-normalized importance sampling (SNIS). For example, to estimate an expectation over a posterior $p(x|y)$, one typically uses SNIS as follows

$$\mathbb{E}_{p(x|y)}[f(x)] = \frac{\int \frac{f(x)p(x,y)}{q(x)}q(x)\mathrm{d}x}{\int \frac{p(x,y)}{q(x)}q(x)\mathrm{d}x} \approx \frac{\sum_n^N f(x_n)w_n}{\sum_n^N w_n} \quad (2)$$

where $x_n \sim q(x)$, and $w_n := p(x_n, y)/q(x_n)$.

2.2 Optimal importance sampling proposal

For a general unknown target, the optimal proposal is the posterior $q(x|y) = p(x|y)$ (see e.g. (Rainforth, 2017, 5.3.2.2)). However, this no longer holds if we have some information about $f(x)$. In this target-aware scenario, the optimal behavior turns out to depend on whether one is performing SNIS or standard IS.

For the non self-normalized case the optimal proposal can be shown to be $q(x) \propto p(x|y)|f(x)|$ (Owen, 2013). In the particular case where $f(x) \geq 0\,\forall x$, this actually leads to a zero-variance estimator as we have $q(x) = p(x|y)f(x)/\mu$ and hence according to Eq. (1) $\hat{\mu} := (1/N)\Sigma_{n=1}^{N} f(x_n)w_n = \mu$ for any value of N, even a single sample. This result can be achieved whenever $f(x)$ is upper or lower bounded for a known bound b, by for example, setting $g(x) = b - f(x)$ such that $g(x) \geq 0\,\forall x$ and noting that $\mathbb{E}_{p(x|y)}\left[f(x)\right] = b - \mathbb{E}_{p(x|y)}\left[g(x)\right]$.

In the self-normalized case, the optimal proposal instead transpires to be $q(x) = |f(x) - \mu|p(x)$ (Hesterberg, 1988). In this case one can no longer achieve a zero variance estimator for finite N and nonconstant $f(x)$: the variance of the estimator is lower bounded by $\mathbb{E}_p(|f(x) - \mu|)^2/N$ (Owen, 2013).

2.3 Inference amortization

As explained in the introduction, inference amortization involves learning an inference artifact that regresses from datasets to proposal parameters. Out of several variants, we focus on the inference amortization method introduced by Paige & Wood (2016) as this is the one AMCI builds upon. The inference network is defined as a neural network φ with parameters η and the approximate proposal is $q(x;\varphi(y;\eta))$. We would like a choice of η to minimize $D_{KL}\left[p(x|y) \;||\; q(x;\varphi(y;\eta))\right]$ across possible instantiations of dataset y, hence the objective is

$$\mathcal{J}(\eta) = \mathbb{E}_{p(y)}\left[D_{KL}\left[p(x|y) \;||\; q(x;\varphi(y;\eta))\right]\right]$$
$$= \mathbb{E}_{p(x,y)}\left[-\log q(x;\varphi(y;\eta))\right] + \text{const wrt } \eta \quad (3)$$

This objective requires us to be able to sample from the joint distribution $p(x, y)$. The entire objective can be optimized using gradient methods since the reparameterized gradient can be easily evaluated:

$$\nabla_\eta \mathcal{J}(\eta) = \mathbb{E}_{p(x,y)}\left[-\nabla_\eta \log q(x;\varphi(y;\eta))\right] \quad (4)$$

3 Amortized Monte Carlo Integration

Amortized Monte Carlo integration (AMCI) is a framework for amortizing the cost of calculating expectations. Though strongly motivated by Bayesian settings, AMCI can be applied in any Monte Carlo integration setting wherein we wish to calculate $\mathbb{E}_{\pi(x)}\left[f(x)\right]$ for some reference distribution $\pi(x)$, known only up to a normalizing constant. Moreover, because a generic integration $\int_{x\in\mathcal{X}} f(x)dx$ can always be expressed as an expecta-

tion $\mathbb{E}_{q(x)}[f(x)\pi(x)/q(x)]$ through importance sampling, AMCI allows amortizing integration more generally.

3.1 Estimator

Existing amortized inference methods often evaluate the expectations using SNIS with the approximate posterior $q(x|y)$ as the proposal. As described in the section 2.2, the variance of this estimator, and hence its error, is lower bounded. AMCI overcomes this limitation by using a new estimator consisting of two separate non self-normalized importance sampling estimators, each using a separate proposal. This means each can be tailored to their respective needs, which can give substantial improvements over SNIS. Moreover, when $f(x)$ is lower or upper bounded, the AMCI construction allows us to achieve a zero-variance estimate if optimal proposals are used for each of the estimators. Thus AMCI can, in principle, provide arbitrarily large improvements over previous approaches, since there is not limit on the level of accuracy it can achieve with only a single sample.

AMCI is based around the estimator

$$\mathbb{E}_{p(x|y)}[f(x)] \approx \frac{\frac{1}{N}\sum_n^N \frac{f(x_n)p(x_n,y)}{q_1(x_n|y)}}{\frac{1}{M}\sum_m^M \frac{p(x_m,y)}{q_2(x_m^*|y)}} \tag{5}$$

where a separate set of samples are drawn for the numerator and denominator by sampling $x_n \sim q_1(x|y)$ and $x_m^* \sim q_2(x|y)$ respectively. The optimal sampling proposal for the expectation in the numerator is $q_1(x|y) \propto |f(x)|p(x|y)$ while for the denominator it is $q_2(x|y) \propto p(x|y)$. When using these proposals, the estimator gives zero error, even when $N = M = 1$.

The above estimator requires taking $T = N + M$ samples, but only N or M are used to evaluate each of the individual estimators. Given that in practice we do not have access to the perfectly optimal proposals, it is more efficient to make use of all T samples in each of the estimators. The motivates the definition of full AMCI estimator for $\mathbb{E}_{p(x|y)}[f(x)]$

$$\frac{\frac{\alpha}{N}\sum_n^N \frac{f(x_n)p(x_n,y)}{q_1(x_n|y)} + \frac{1-\alpha}{M}\sum_m^M \frac{f(x_m^*)p(x_m^*,y)}{q_2(x_m^*|y)}}{\frac{\beta}{N}\sum_n^N \frac{p(x_n,y)}{q_1(x_n|y)} + \frac{1-\beta}{M}\sum_m^M \frac{p(x_m^*,y)}{q_2(x_m^*|y)}}, \tag{6}$$

where both the numerator and denominator are in a form of a convex combination of estimators with respect to samples from both proposals. The level of interpolation is set by parameters α, β which vary between 0 and 1. Setting $\alpha = 1$ and $\beta = 0$ corresponds to the estimator in Eq. (5) and is optimal if our proposals are perfect. Setting $\alpha = 1, \beta = 1$ or $\alpha = 0, \beta = 0$ corresponds to the SNIS estimator with proposals q_1, q_2, respectively.

Intuitively, the optimal coefficients, (α^*, β^*), should be close to those with perfect proposals, i.e. $\alpha \approx 1,\ \beta \approx 0$, provided we use sufficiently accurate inference networks. More formally, under certain conservative assumptions, we can show that $\alpha^* = N/((T-N)\frac{\mathrm{Var}[f(x_1)w_1]}{\mathrm{Var}[f(x_1^*)w_1^*]} + N)$ and $\beta^* = N/((T-N)\frac{\mathrm{Var}[w_1]}{\mathrm{Var}[w_1^*]} + N)$, the derivations for which are given in Appendix A. Similar analysis should also be possible to guide the selection of N and M, but we leave this to future work.

3.2 Amortization

To be able to evaluate this estimator AMCI needs to learn to amortize the approximations q_1 and q_2. The standard objective allowing to amortize the posterior distribution $p(x|y)$ is described in section 2.3. To learn the approximation to the optimal sampling proposal $|f(x)|\,p(x|y)$ AMCI seeks to minimize $D_{KL}\left[|f(x)|\,p(x|y) \;||\; q(x;\varphi(y;\eta))\right]$ across possible instantiations of dataset y:

$$\begin{aligned}\mathcal{J}_1(\eta) &= \mathbb{E}_{p(y)}\left[D_{KL}\left[|f(x)|\,p(x|y) \;||\; q(x;\varphi(y;\eta))\right]\right] \\ &= -\,\mathbb{E}_{p(x,y)}\left[|f(x)|\log q(x;\varphi(y;\eta))\right] + \text{const wrt } \eta\end{aligned}$$

AMCI learns to amortize $q_1(x|y)$ using $\mathcal{J}_1$ above, as well as $q_2(x|y)$ using $\mathcal{J}$ in Eq. (3). Both objectives can be jointly optimized using the same samples from $p(x, y)$.

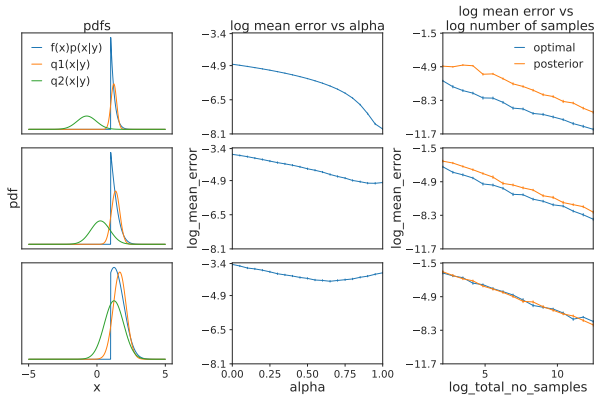
The distributions $q_1(x|y)$ and $q_2(x|y)$ are related and it is reasonable to assume that part of the computation required to determine those proposals is shared. To take advantage of this, we use a single neural network with shared weights to determine the parameters of both distributions.

To account for cases in which multiple possible target functions may be of interest, AMCI also allows for amortization over parametrized functions. If the target function can be parameterized as $f(x;\theta)$, then by extending our target distribution with a pseudo prior $p(\theta)$ we are able to amortize over possible target functions as well. The choice of $p(\theta)$ determines how much importance we assign to different possible functions that we would like to amortize over. Since in practice perfect performance is unattainable over the entire space of θ the choice of $p(\theta)$ is important and it will have important effect on the performance of the system.

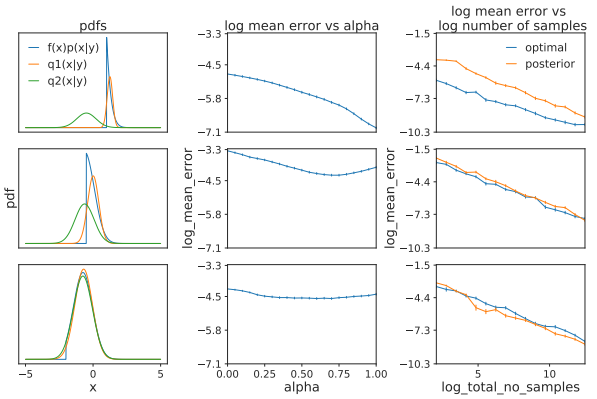
Amortization over the space of targets requires another modification to the objective $\mathcal{J}$ and to our inference network φ – it now also needs to take θ as input. AMCI seeks a choice of η which performs well across possible instantiations of dataset y and possible values of θ. Hence we need to take an expectation over $p(y)p(\theta)$ yielding

$$\begin{aligned}&\mathcal{J}_1^{'}(\eta) \\ &= \mathbb{E}_{p(y)\,p(\theta)}\left[D_{KL}\left[|f(x;\theta)|\,p(x|y) \;||\; q(x;\varphi(y,\theta;\eta))\right]\right] \\ &= -\mathbb{E}_{p(x,y)\,p(\theta)}\left[|f(x;\theta)|\log q(x;\varphi(y,\theta;\eta))\right] + \text{const}\end{aligned}$$

Note that $p(x, y)p(\theta)$ is tractable and so the entire objective can be optimized using gradient methods.



(a) Fixed integrand $f(x)$.
Rows represent examples for different values of y,
respectively: $-1.5, 0.5, 2.5$.



(b) Parameterized integrand $f(x;\theta)$.
Rows represent examples for different values of (y,θ),
respectively: $(-1,1)$, $(-1.25,-0.5)$, $(-1.5,-2)$.

Figure 1: Results of the amortization experiments. Uncertainty bars in columns 2, 3 are estimated over a 1000 runs. Middle column errors are reported for 64 samples. For AMCI the number of samples from both proposals was equal, i.e. $M = N$. Columns represent, respectively: shapes of probability density functions, log mean error vs α (Eq. (6)), log mean error vs log number of samples for optimal proposal ($\alpha = 1$) and for posterior proposal ($\alpha = 0$).

4 Experiments

We demonstrate the performance of the AMCI estimator on two illustrative examples, and compare it with the SNIS estimator with posterior proposal in Eq. (2). In experiment one we fix the target function $f(x)$ whilst in experiment two, we allow a parameterized target function $f(x;\theta)$. We also investigate the effects of varying the parameters α and β in Eq. (6).

4.1 Fixed target function

For a fixed target function the model is as follows

$$p(x) = \mathcal{N}(x;0,1) \qquad p(y|x) = \mathcal{N}(y;x,1)$$
$$q(x|y) = \mathcal{N}\left(x;\varphi(y;\eta)\right) \qquad f(x) = \mathbb{1}_{x>1}$$

The posterior $p(x|y) = \mathcal{N}(x;\frac{y}{2},\frac{1}{2})$ can be determined analytically, and so can the true value of the expectation $\mathbb{E}_{p(x|y)}\left[f(x)\right] = 1 - \Phi(1)$, where Φ is a standard normal distribution cumulative density function.

4.2 Parameterized target function

In this case $p(x)$ and $p(y|x)$ remain the same, but the definition of $q(x|y)$ and $f(x;\theta)$ changes

$$q(x|y) = \mathcal{N}\left(x;\varphi(\eta,y,\theta)\right) \qquad f(x;\theta) = \mathbb{1}_{x>\theta}$$
$$\theta \sim \text{Uniform}[-5,5]$$

We constrain the range of θ to be between -5 and 5 as the region of interest. The posterior is still $p(x|y) = \mathcal{N}(x;\frac{y}{2},\frac{1}{2})$ and the true value of the expectation is now dependent on θ as follows $\mathbb{E}_{p(x|y)}\left[f(x;\theta)\right] = 1 - \Phi(\theta)$.

4.3 Implementation details

The inference networks and the posterior distributions are combined into one neural network. The network used 2 fully connected layers with 10 hidden nodes per layer and it was trained using Adam (Kingma & Ba, 2015).

4.4 Results

We find that fixing $\beta = 0$ universally results in the smallest error of the estimates and hence we do not report results on varying β in the plots to improve clarity.

Results of the experiments are presented in Fig. 1a and Fig. 1b respectively. The values of y and θ plotted were chosen to be illustrative of the phenomena observed.

We find that the AMCI estimator performs better than the SNIS estimator with posterior proposal when the optimal proposal differs from the posterior. In some cases, when the optimal sampling proposal differs greatly from the posterior the mean error using the AMCI estimate are over an order of magnitude smaller than for the SNIS estimate for the same total sample budget, see the first row in the Figure 1. However, when the optimal proposal is similar to the posterior, AMCI and SNIS give more similar performance, see the second and third rows. We also find in those cases setting $\alpha < 1$ results in lower mean error.

4.5 Discussion

Further work will be focused on scaling the AMCI framework to larger problems, automating the process of online hyperparameter tuning, correlating the samples for the individual estimators to further reduce the variance of the estimator, and using more flexible neural density estimators, e.g. normalizing flows, rather than parametric families as the amortized proposal distributions.

Acknowledgments

We would like to thank Tuan Anh Le for his input, enthusiasm and sharing parts of his previous work, as well as Bradley Gram-Hansen for both the enthusiasm and taking time to proof-read the manuscript.

AG is supported by the UK EPSRC CDT in Autonomous Intelligent Machines and Systems. TR and YWT are supported in part by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement no. 617071. FW is supported under DARPA PPAML through the U.S. AFRL under Cooperative Agreement FA8750-14-2-0006, Sub Award number 61160290-111668.

References

Evans, Michael and Swartz, Tim. Methods for approximating integrals in statistics with special emphasis on bayesian integration problems. *Statistical science*, pp. 254–272, 1995.

Hesterberg, Timothy Classen. *Advances in importance sampling*. PhD thesis, Stanford University, 1988.

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

Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

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

Lacoste–Julien, Simon, Huszar, Ferenc, and Ghahramani, Zoubin. Approximate inference for the loss-calibrated bayesian. In Gordon, Geoffrey, Dunson, David, and Dudík, Miroslav (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 416–424, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

Le, Tuan Anh, Baydin, Atılım Güneş, and Wood, Frank. Inference compilation and universal probabilistic programming. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1338–1348, Fort Lauderdale, FL, USA, 2017. PMLR.

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

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

Naesseth, Christian A, Linderman, Scott W, Ranganath, Rajesh, and Blei, David M. Variational sequential monte carlo. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

Oh, Man-Suk and Berger, James O. Adaptive importance sampling in monte carlo integration. *Journal of Statistical Computation and Simulation*, 41(3-4):143–168, 1992.

Owen, Art B. *Monte Carlo theory, methods and examples*. 2013.

Paige, Brooks and Wood, Frank. Inference networks for sequential monte carlo in graphical models. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, 2016.

Rainforth, Tom. *Automating Inference, Learning, and Design using Probabilistic Programming*. PhD thesis, 2017.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of The 31st International Conference on Machine Learning*, 2014.

Ritchie, Daniel, Horsfall, Paul, and Goodman, Noah D. Deep amortized inference for probabilistic programs. *arXiv preprint arXiv:1610.05735*, 2016.

Robert, Christian. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.

Stuhlmüller, Andreas, Taylor, Jacob, and Goodman, Noah. Learning stochastic inverses. In *Advances in neural information processing systems*, pp. 3048–3056, 2013.

Wolpert, Robert L. Monte carlo integration in bayesian statistical analysis. *Contemporary Mathematics*, 115: 101–116, 1991.

A Derivation of the optimal parameter values for the AMCI estimator

In this section, we derive the optimal values of α and β in terms of minimizing the mean squared error (MSE) of the estimator in Equation 6. We assume that we are allocated a total sample budget of T samples, such that $M = T - N$.

Let the true values of the expectations in the numerator and denominator be denoted as Z_N and Z_D, respectively. We also define the following shorthands for the unbiased importance sampling estimators with respect to proposals q_1 and q_2 in Equation 6 $a_1 = \frac{1}{N}\sum_n^N \frac{f(x_n)p(x_n,y)}{q_1(x_n|y)}$, $b_1 = \frac{1}{M}\sum_m^M \frac{f(x_m^*)p(x_m^*,y)}{q_2(x_m^*|y)}$, $a_2 = \frac{1}{N}\sum_n^N \frac{p(x_n,y)}{q_1(x_n|y)}$, $b_2 = \frac{1}{M}\sum_m^M \frac{p(x_m^*,y)}{q_2(x_m^*|y)}$, where $x_n \sim q_1(x|y)$ and $x_m^* \sim q_2(x|y)$.

We start by considering the estimator according to Equation 6

$$\frac{Z_N}{Z_D} \approx I := \frac{\alpha a_1 + (1-\alpha)b_1}{\beta a_2 + (1-\beta)b_2}. \tag{7}$$

Using the central limit theorem, then as $N, M \to \infty$, we have

$$\to \frac{Z_N + \sigma_N \xi_N}{Z_D + \sigma_D \xi_D}, \quad \text{where} \quad \xi_N, \xi_D \sim \mathcal{N}(0,1) \tag{8}$$

are correlated standard normal random variables and σ_N and σ_D are the standard deviation of the estimators for numerator and denominator respectively. Specifically we have

$$\begin{aligned}\sigma_N^2 =& \mathrm{Var}[\alpha a_1 + (1-\alpha)b_1] \\ =& \alpha^2 \mathrm{Var}_{q_1}[a_1] + (1-\alpha)^2 \mathrm{Var}_{q_2}[b_1],\end{aligned}$$

which by the weak law of large numbers

$$= \frac{\alpha^2}{N}\mathrm{Var}_{q_1}[f(x_1)w_1] + \frac{(1-\alpha)^2}{M}\mathrm{Var}_{q_2}[f(x_1^*)w_1^*]$$

where $w_1 = p(x_1, y)/q_1(x_1|y)$, $w_1^* = p(x_1^*, y)/q_2(x_1^*|y)$, $x_1 \sim q_1(x|y)$, and $x_1^* \sim q_2(x|y)$. Analogously,

$$\sigma_D^2 = \frac{\beta^2}{N}\mathrm{Var}_{q_1}[w_1] + \frac{(1-\beta)^2}{M}\mathrm{Var}_{q_2}[w_1^*].$$

Now going back to Equation 8 and using Taylor's Theorem on $1/\left(Z_D + \sigma_D \xi_D\right)$ about $1/Z_D$ gives

$$\begin{aligned}I =& \frac{Z_N + \sigma_N \xi_N}{Z_D}\left(1 - \frac{\sigma_D \xi_D}{Z_D}\right) + O(\epsilon) \\ =& \frac{Z_N}{Z_D} + \frac{\sigma_N \xi_N}{Z_D} - \frac{Z_N \sigma_D \xi_D}{Z_D^2} - \frac{\sigma_N \sigma_D \xi_N \xi_D}{Z_D^2} + O(\epsilon)\end{aligned}$$

where $O(\epsilon)$ represents asympotitically dominated terms. Note here the importance of using Taylor's theorem, rather just a Taylor expansion, to to confirm that these terms are indeed asympotitically dominated. We can further drop the $\frac{\sigma_N \sigma_D \xi_N \xi_D}{Z_D^2}$ term as this will be order $O(1/\sqrt{MN})$ and will thus be asymptotically dominated, giving

$$= \frac{Z_N}{Z_D} + \frac{\sigma_N \xi_N}{Z_D} - \frac{Z_N \sigma_D \xi_D}{Z_D^2} + O(\epsilon).$$

To calculate the MSE of I, we start with the standard bias variance decomposition

$$\mathbb{E}\left[\left(I - \frac{Z_N}{Z_D}\right)^2\right] = \mathrm{Var}\left[I\right] + \left(\mathbb{E}\left[I - \frac{Z_N}{Z_D}\right]\right)^2.$$

Considering first the bias squared term, we see that this depends only on the higher order terms $O(\epsilon)$, while the variance does not. It straightforwardly follows that the variance term will be asymptotically dominant, so we see that optimizing for the variance is asymptotically equivalent to optimizing for the MSE.

Now using the standard relationship $\mathrm{Var}(X+Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\,\mathrm{Cov}(X,Y)$, we have

$$\begin{aligned}\mathrm{Var}[I] =& \mathrm{Var}\left[\frac{\sigma_N \xi_N}{Z_D}\right] + \mathrm{Var}\left[\frac{Z_N \sigma_D \xi_D}{Z_D^2}\right] - 2\mathrm{Cov}\left[\frac{\sigma_N \xi_N}{Z_N}, \frac{Z_N \sigma_D \xi_D}{Z_D^2}\right] + O(\epsilon) \\ =& \frac{\sigma_N^2}{Z_D^2} + \frac{Z_N^2 \sigma_N^2}{Z_D^4} - 2\frac{\sigma_N Z_N \sigma_D}{Z_D^3}\mathrm{Cov}[\xi_N, \xi_D] + O(\epsilon)\end{aligned}$$

$$= \frac{\alpha^2}{NZ_D^2}\mathrm{Var}_{q_1}[f(x_1)w_1] + \frac{(1-\alpha)^2}{MZ_D^2}\mathrm{Var}_{q_2}[f(x_1^*)w_1^*] + \frac{Z_N^2\beta^2}{NZ_D^4}\mathrm{Var}_{q_1}[w_1] + \frac{Z_N^2(1-\beta)^2}{MZ_D^4}\mathrm{Var}_{q_2}[w_1^*]$$
$$- 2\frac{Z_N}{Z_D^3}\mathrm{Corr}[\xi_N, \xi_D]\left(\frac{\alpha^2}{N}\mathrm{Var}_{q_1}[f(x_1)w_1] + \frac{(1-\alpha)^2}{M}\mathrm{Var}_{q_2}[f(x_1^*)w_1^*]\right)\left(\frac{\beta^2}{N}\mathrm{Var}_{q_1}[w_1] + \frac{(1-\beta)^2}{M}\mathrm{Var}_{q_2}[w_1^*]\right)$$

To assist in the subsequent analysis, we assume that there is no correlation, $\mathrm{Corr}[\xi_N, \xi_D] = 0$. Though this assumption is unlikely to be exactly true, there are two reasons we believe it is reasonable. Firstly, because we expect to set $\alpha \approx 1$ and $\beta \approx 0$, the correlation should generally be small in practice as the two estimators rely predominantly on independent sets of samples. Secondly, we believe this is generally a relatively conservative assumption: if one were to presume a particular correlation, there are adversarial cases with the opposite correlation where this assumption is damaging. Nonetheless, catering for non-zero correlations is something one may wish to look into in future work.

Given this assumption is now straightforward to optimize for α and β by finding where the gradient is zero as follows

$$\nabla_\alpha(\mathrm{Var}[I]Z_D^2) = \frac{2\alpha\mathrm{Var}_{q_1}[f(x_1)w_1]}{N} - \frac{2(1-\alpha)\mathrm{Var}_{q_2}[f(x_1^*)w_1^*]}{T-N} = 0$$
$$\Rightarrow \alpha^* = \frac{N}{(T-N)\frac{\mathrm{Var}_{q_1}[f(x_1)w_1]}{\mathrm{Var}_{q_2}[f(x_1^*)w_1^*]} + N}$$

noting that

$$\nabla_\alpha^2(\mathrm{Var}[I]Z_D^2) = \frac{\mathrm{Var}_{q_1}[f(x_1)w_1]}{N} + \frac{\mathrm{Var}_{q_2}[f(x_1^*)w_1^*]}{T-N} > 0$$

and hence it's a local minimum. Analogously

$$\beta^* = \frac{N}{(T-N)\frac{\mathrm{Var}_{q_1}[w_1]}{\mathrm{Var}_{q_2}[w_1^*]} + N}.$$

We note that it is possible to estimate all the required variances here using previous samples. It should therefore be possible to adaptively set α and β by using these equations along with empirical estimates for these variances.