Variational Likelihood-Free Gradient Descent

Jack Simons JACK.SIMONS@BRISTOL.AC.UK
Song Liu SONG.LIU@BRISTOL.AC.UK
Mark Beaumont M.BEAUMONT@BRISTOL.AC.UK
University of Bristol

Abstract

In many scientific applications, we do not have explicit access to the likelihood function. However simulations of the process of interest, using different parameter settings, may give us access to the likelihood function implicitly. The methodology for approximating likelihoods and posterior distributions based on simulated observations can be described as simulation-based inference. In this paper, we propose a simulation-based inference algorithm in which we iteratively update particles to more closely resemble the posterior. Our approach utilises simulations to estimate a density ratio function at each iteration and then uses it to approximate the KL divergence between the particle density and the posterior density. By alternating between gradient descent and density ratio estimation, the approximated KL divergence is minimized. We benchmark the performance of our algorithm on a Gaussian mixture model and the M/G/1 queue process model and report promising results.

1. Introduction

Scientific enquiry requires the construction of testable and falsifiable models that describe phenomena of interest. Historically, this has led to the development of likelihood- and Bayesian-based approaches to inference, focusing on closed-form representations of the likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ and tractable posterior distributions $p(\boldsymbol{\theta}|\boldsymbol{x})$. More recently there has been a trend to focus on the stochastic mechanism whereby the data are generated (Diggle and Gratton, 1984). In this case the likelihood is defined implicitly and a variety of approaches have been proposed to utilise it effectively for statistical inference.

The relationship between our desired posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$ and the likelihood is given in Bayes' formula $p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})}$. Henceforth we shall call $p(\boldsymbol{\theta})$ the prior, and $p(\boldsymbol{x})$ the marginal. Of course exact calculations of the posterior are unfeasible in the absence of the likelihood function. Despite this, much work has been done in literature to give approximate solutions to these problems, e.g. Approximate Bayesian Computation (ABC) (Beaumont et al., 2002; Sisson et al., 2018). For a more general overview see (Cranmer et al., 2020) which covers a variety of simulation-based inference (SBI) methods. In the SBI literature the implicit likelihood is assumed to be computationally expensive, and thus the emphasis of these methods is to provide accurate inference whilst using a minimal number of simulations.

In recent years, it has been noticed that if we can approximate the ratio $\frac{p(\boldsymbol{\theta}|\boldsymbol{x})}{p(\boldsymbol{\theta})} = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x})}$ by a function $r(\boldsymbol{\theta}, \boldsymbol{x})$, then given an observation $\boldsymbol{x}_{\text{obs}}$, we can easily approximate the posterior $p(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})$ by using $r(\boldsymbol{\theta}, \boldsymbol{x}_{\text{obs}})p(\boldsymbol{\theta})$. In Thomas et al. (2021) they leverage logistic regression to estimate $\frac{p(\boldsymbol{x}|\boldsymbol{\theta}_0)}{p(\boldsymbol{x})}$ (where $\boldsymbol{\theta}_0$ is fixed) using samples from a simulator and samples from the marginal $p(\boldsymbol{x})$. Similarly, Hermans et al. first estimates $\frac{p(\boldsymbol{x},\boldsymbol{\theta})}{p(\boldsymbol{x})p(\boldsymbol{\theta})}$ using samples from the joint probability $p(\boldsymbol{x},\boldsymbol{\theta})$ and the marginals $p(\boldsymbol{x})$ and $p(\boldsymbol{\theta})$, then use MCMC to sample from the approximated posterior using $r(\boldsymbol{\theta},\boldsymbol{x}_{\text{obs}})p(\boldsymbol{\theta})$. Instead of estimating the density ratio in one-shot, Hermans et al. also proposes a sequential approach which focuses on estimating the ratio accurately for $\boldsymbol{\theta}$ parameters that likely generated $\boldsymbol{x}_{\text{obs}}$. Durkan et al. points out the links between this sequential approach and conditional density estimators such as Papamakarios and Murray (2016); Lueckmann et al. (2017).

Motivated by variational inference algorithms (Blei et al., 2017), we propose to minimize the KL divergence between the approximate posterior and the true posterior. Specifically, we were inspired by a variational approach called Stein Variational Gradient Descent (SVGD) (Liu and Wang, 2016) and optimise a particle distribution to minimize the aforementioned KL divergence. Note that SVGD cannot be readily applied to simulation-based inference due to lack of a tractable likelihood. In our method, we use density ratio estimation to approximate the log ratio term in the KL divergence and perform gradient updates on particles to reduce the approximated KL divergence. We obtain promising results on toy and real-world problems using the proposed method.

In the rest of the paper: we formally introduce our method in Section 2, show toy and real-world experiment results in Section 3, and discuss and conclude in Section 4.

2. The Proposed Method

Suppose we have a set of particles $\Theta_q = \{\boldsymbol{\theta}_i\}_{i=1}^{n_q}$ with an empirical probability density function q. We hope to iteratively update Θ_q so that q converges to a good approximation of the target posterior. To achieve this, we minimize the KL divergence between q and the target posterior $p(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})$. Specifically, our algorithm repeats the following two steps:

- Approximate $\text{KL}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})]$ using particles Θ_q and simulations.
- Perform gradient descent update on particles in Θ_q to reduce the approximated KL divergence.

The similarities of our method, Variational Likelihood-Free Gradient Descent (VLFGD) to the classic SVGD algorithm (Liu and Wang, 2016; Liu, 2017) are seen in this second point: both perform gradient descent on a set of particles to minimize the KL divergence between them and the target posterior. However, beyond this the differences in the two methods are considerable - they have separate motivations, derivations and applications - and thus they should be considered as distinct methods.

2.1. Approximating the KL Divergence

Like other variational inference methods, we hope to find a variational distribution q that minimizes the KL divergence between q and the target posterior $p(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})$:

$$\text{KL}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})] := \int q(\boldsymbol{\theta}) \log\left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})}\right) \mathrm{d}\boldsymbol{\theta} = \int q(\boldsymbol{\theta}) \log\left(\frac{q(\boldsymbol{\theta})p(\boldsymbol{x}_{\text{obs}})}{p(\boldsymbol{x}_{\text{obs}},\boldsymbol{\theta})}\right) \mathrm{d}\boldsymbol{\theta}. \quad (1)$$

Although in Bayesian inference, $p(\boldsymbol{x}_{\text{obs}})$ is usually seen as a constant, here we treat it as the density $p(\boldsymbol{x})$ evaluated at $\boldsymbol{x}_{\text{obs}}$. This treatment is crucial for developing our methodology.

Without access to density functions $q(\boldsymbol{\theta}), p(\boldsymbol{x})$ and $p(\boldsymbol{x}, \boldsymbol{\theta})$, it is impossible to evaluate $\log\left[\frac{q(\boldsymbol{\theta})p(\boldsymbol{x})}{p(\boldsymbol{x},\boldsymbol{\theta})}\right]$ in the KL divergence. However, it is possible to estimate the log density ratio function using samples from the constituent densities (Sugiyama et al., 2012). One of the simplest log density ratio estimators is the logistic regression classifier (see Chapter 4 in Sugiyama et al. (2012)). Suppose density functions $q(\boldsymbol{z})$ and $p(\boldsymbol{z})$ are densities from which positive and negative samples are drawn in a binary classification task. Logistic regression (assuming perfectly balanced class priors) solves the following optimization:

$$\min_g \int q(\boldsymbol{z}) \log\left[1 + \exp(-g(\boldsymbol{z}))\right] \mathrm{d}\boldsymbol{z} + \int p(\boldsymbol{z}) \log[1 + \exp(g(\boldsymbol{z}))] \mathrm{d}\boldsymbol{z}. \tag{2}$$

Some algebra shows that the minimum is attained when $g = \log \frac{q}{p}$. When solving (2) in practice, we approximate the objective using empirical samples from q and p and restrict g to belong to some parametric family. Therefore, to estimate the ratio $\frac{q(\boldsymbol{\theta})p(\boldsymbol{x})}{p(\boldsymbol{x},\boldsymbol{\theta})}$, we minimize

$$\min_g \int p(\boldsymbol{x}) q(\boldsymbol{\theta}) \log\left[1 + \exp(-g(\boldsymbol{\theta}, \boldsymbol{x}))\right] \mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{x} + \int p(\boldsymbol{\theta}, \boldsymbol{x}) \log[1 + \exp(g(\boldsymbol{\theta}, \boldsymbol{x}))] \mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{x}. \tag{3}$$

Suppose $Z_p := \left\{\left(\boldsymbol{\theta}_i^{(0)}, \boldsymbol{x}_i^{(0)}\right)\right\}_{i=1}^{n_p}$ are unbiased samples from the joint probability $p(\boldsymbol{x}, \boldsymbol{\theta})$, we can approximate the objective function in (3) using sample averages:

$$\min_{\boldsymbol{w}} \frac{1}{n_p n_q} \sum_{i=1}^{n_p} \sum_{j=1}^{n_q} \log\left[1 + \exp\left(-g_{\boldsymbol{w}}\left(\boldsymbol{\theta}_j, \boldsymbol{x}_i^{(0)}\right)\right)\right] + \frac{1}{n_p} \sum_{i=1}^{n_p} \log\left[1 + \exp\left(g_{\boldsymbol{w}}\left(\boldsymbol{\theta}_i^{(0)}, \boldsymbol{x}_i^{(0)}\right)\right)\right], \tag{4}$$

where we have restricted g to belong to a function family parameterized by $\boldsymbol{w}$. In this work, we parameterize g as a neural network where $\boldsymbol{w}$ are the weights of the neural network. After obtaining the minimizer $\widehat{\boldsymbol{w}}$, we use $g_{\widehat{\boldsymbol{w}}}(\cdot, \boldsymbol{x}_{\text{obs}})$ as an estimator of $\log\left[\frac{q(\cdot)p(\boldsymbol{x}_{\text{obs}})}{p(\boldsymbol{x}_{\text{obs}},\cdot)}\right]$.

2.2. Gradient Descent on KL Divergence

In the proposed method, we want to slightly perturb our particles so that the updated particles will more closely resemble the true target posterior. Consider a perturbed particle set $\Theta'_q = \{\boldsymbol{\theta}'_i\}_{i=1}^{n_q}$ where $\boldsymbol{\theta}'_i = \boldsymbol{\theta}_i + \boldsymbol{\epsilon}_i$. Suppose particles in Θ'_q have an empirical probability density function $q_{\boldsymbol{\epsilon}}$, then the "optimal update" is obtained by the following optimization:

$$\min_{\boldsymbol{\epsilon}_i, i=1\ldots n_q} \mathrm{KL}[q_\epsilon(\boldsymbol{\theta}) | p(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})] = \int q_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}) \log \frac{q_{\boldsymbol{\epsilon}}(\boldsymbol{\theta}) p(\boldsymbol{x}_{\text{obs}})}{p(\boldsymbol{\theta}, \boldsymbol{x}_{\text{obs}})} \mathrm{d}\boldsymbol{\theta}, \text{ subject to } \|\boldsymbol{\epsilon}_i\| \le \epsilon_0, \tag{5}$$

where ϵ_0 is a small constant. Since $\boldsymbol{\epsilon}_i$ are only small perturbations ($\|\boldsymbol{\epsilon}_i\| \le \epsilon_0$), we can assume $\log \frac{q_{\boldsymbol{\epsilon}}(\boldsymbol{\theta})p(\boldsymbol{x}_{\text{obs}})}{p(\boldsymbol{\theta},\boldsymbol{x}_{\text{obs}})} \approx \log \frac{q(\boldsymbol{\theta})p(\boldsymbol{x}_{\text{obs}})}{p(\boldsymbol{\theta},\boldsymbol{x}_{\text{obs}})}$ for all $\boldsymbol{\theta}$, so that $g_{\widehat{\boldsymbol{w}}}(\cdot, \boldsymbol{x}_{\text{obs}})$ is still a reasonable estimator of $\log \frac{q_{\boldsymbol{\epsilon}}(\cdot)p(\boldsymbol{x}_{\text{obs}})}{p(\cdot,\boldsymbol{x}_{\text{obs}})}$. The minimization in (5) can be approximated by

$$\min_{\boldsymbol{\epsilon}_i, i=1\ldots n_q} \frac{1}{n_q} \sum_{i=1}^{n_q} g_{\widehat{w}}(\boldsymbol{\theta}_i + \boldsymbol{\epsilon}_i, \boldsymbol{x}_{\text{obs}}), \text{ subject to } \|\boldsymbol{\epsilon}_i\| \le \epsilon_0. \tag{6}$$

Instead of actually solving (6), we use an approximate solution

$$\widehat{\boldsymbol{\epsilon}}_i := -\eta \cdot \nabla_{\boldsymbol{\theta}_i} \frac{1}{n_q} \sum_{j=1}^{n_q} g_{\widehat{w}}(\boldsymbol{\theta}_j, \boldsymbol{x}_{\text{obs}}) = -\frac{\eta}{n_q} \cdot \nabla_{\boldsymbol{\theta}_i} g_{\widehat{w}}(\boldsymbol{\theta}_i, \boldsymbol{x}_{\text{obs}}), \tag{7}$$

where η is a small constant (e.g. 0.0001) chosen post hoc.

Simply speaking, in order to move our particles closer to the true posterior, we push them along the negative gradient of the approximate KL divergence by a small amount.

2.3. Alternating Between KL Estimation and Gradient Update

Our estimation in Section 2.1 is tied to a specific $q(\boldsymbol{\theta})$. In other words, once we update particles in Θ_q, our log ratio estimator is no longer accurate. Therefore, we need to re-estimate the log density ratio every time after our particles are perturbed. This naturally leads to a procedure alternating between the KL divergence estimation in Section 2.1 and the gradient update in Section 2.2. This idea is summarised in Algorithm 1.

```
Algorithm 1: VLFGD
Inputs: Observation $\boldsymbol{x}_{\text{obs}}$; initial particles $\Theta_q^{(1)}$; sampler of prior $p(\boldsymbol{\theta})$; observation
        simulator $\text{sim}(\boldsymbol{\theta})$; step size $\eta$; maximum iteration $T$.
Outputs: Particles at iteration $T$: $\Theta_q^{(T)}$.
Draw unbiased samples $Z_p := \left\{\left(\boldsymbol{\theta}_i^{(0)}, \boldsymbol{x}_i^{(0)}\right)\right\}_{i=1}^{n_p}$ where $\boldsymbol{\theta}_i^{(0)} \sim p(\boldsymbol{\theta}), \boldsymbol{x}_i^{(0)} = \text{sim}\left(\boldsymbol{\theta}_i^{(0)}\right)$.
for $t$ from 1 to $T$ do
    Obtain $\widehat{\boldsymbol{w}}$ by solving (4)
    $\boldsymbol{\theta}_i^{(t+1)} := \boldsymbol{\theta}_i^{(t)} + \widehat{\boldsymbol{\epsilon}}_i$, where $\widehat{\boldsymbol{\epsilon}}_i$ is defined in (7).
end
```

3. Experiments

3.1. Mixture of two Gaussians

We apply our method to a simple model with a tractable posterior. The likelihood and prior are defined as

$$p(\theta) \sim U(\theta_{\text{low}}, \theta_{\text{high}}), \quad p(x|\theta) \sim \alpha\mathcal{N}(\theta, \sigma_1^2) + (1-\alpha)\mathcal{N}(\theta, \sigma_2^2),$$

respectively. As in (Papamakarios et al., 2019), we set

$$\theta_{\text{low}} = -10, \ \theta_{\text{high}} = 10, \ \alpha = 0.5, \ \sigma_1^2 = 1, \ \sigma_2^2 = 0.01, \text{ and } x_{\text{obs}} = 0.$$

The tractable posterior $p(\theta|x_{\text{obs}}) \propto 0.5\mathcal{N}(0, \sigma_1^2) + 0.5\mathcal{N}(0, \sigma_2^2)$. Because the parameter we wish to infer is one dimensional, it allows us to visualize our update steps. We run Algorithm 1 with 1000 particles ($n_q = 1000$). The histograms of particles $\Theta_q^{(t)}$ at iterations $t = 1, 15, 200$ are plotted at the top row of Figure 1. It can be seen that $\Theta_q^{(t)}$ converges

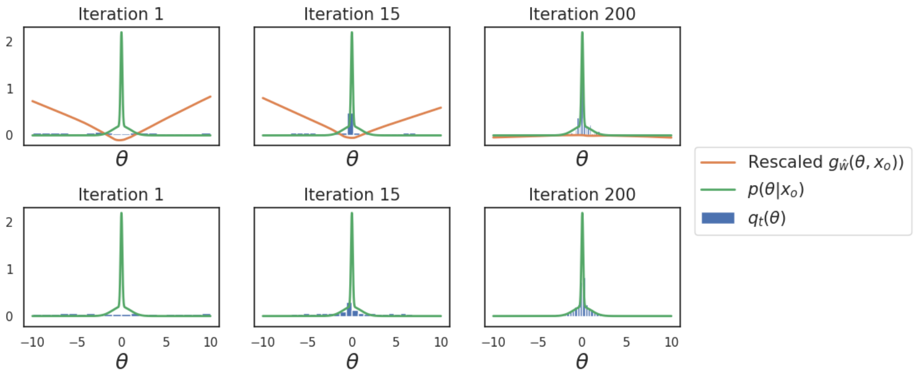
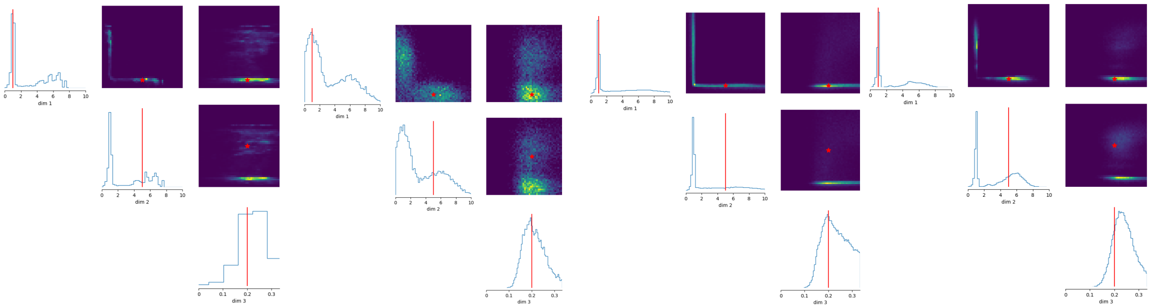


Figure 1: Top row: VLFGD particles. Bottom row: SVGD particles.



(a) VLFGD (b) ABC-Rej (c) NRE (d) SRE

Figure 2: Resulting aggregated posteriors from 10 runs of VLFGD, ABC-Rejection, NRE and SRE with 100k simulation budget applied to the M/G/1 Queueing model.

to the target posterior $p(\theta|x_{\text{obs}})$ as t increases. For comparison, we plot the histograms of SVGD particles at the bottom row. We can see that our VLFGD updates mimic the updates observed in SVGD without accessing the true posterior. Moreover, we plot the estimated log ratio $g_{\hat{\boldsymbol{w}}}(\cdot, x_{\text{obs}})$ for VLFGD updates. It can be seen that at iteration 1 and 15, the negative gradient of $g_{\hat{\boldsymbol{w}}}(\cdot, x_{\text{obs}})$ points to 0 where the true posterior is high. This confirms that $g_{\hat{\boldsymbol{w}}}(\cdot, x_{\text{obs}})$ can indeed guide the Θ_q toward a good posterior approximate.

3.2. M/G/1 Queue

The M/G/1 process aims to model a simple queue with customers arriving at random times v_i and being served for random amounts of time s_i by a single server. The output of the process is the inter-departure times $d_i - d_{i-1}$ of the customers. The distribution of these objects is dictated by 3 parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$. Precise details of the model are found in Appendix C, and we employ the same set up as (Papamakarios et al., 2019).

It is easy to simulate observations using this model. However, its likelihood is intractable and consequently its posterior is intractable. Inference is considerably more challenging than that in Section 3.1 not only because the dimension of both $\boldsymbol{x}$ and $\boldsymbol{\theta}$ have increased, but also due to the highly non-linear and noisy nature of the simulator. In Figure 3.2, we compare our method, VLFGD, to ABC-Rejection (Pritchard et al., 1999), Neural Ratio Estimation (NRE) and its sequential version Sequential Ratio Estimation (SRE) described in (Hermans et al., 2020) with the latter two implemented in Python with the toolbox sbi (Tejero-Cantero et al., 2020).

The marginal and joint posteriors resulting from VLFGD, Rejection ABC, and SRE have reasonable amount of mass near the ground-truth parameters $\boldsymbol{\theta}^\star$. This is in contrast to NRE which has almost no mass assigned to the ground-truth parameter $\theta_2^\star$. Both NRE, SRE and our method give strongly peaked marginal posteriors. The posteriors achieved from our method and SRE are very similar, with peaks in almost identical locations. All methods result in the main mode of $p(\theta_2|\boldsymbol{x}_{\text{obs}})$ in a different location to the ground truth $\theta_2^\star$.

4. Discussions and Conclusion

In this paper, we proposed an iterative algorithm to produce a particle distribution that resembles the target posterior. To achieve this, we move the particles according to the direction that minimizes a KL divergence between the particle distribution and the true posterior. The KL divergence is approximated using logistic regression. Experiments on a Gaussian mixture model and the M/G/1 queuing model show promising results.

Methodologically, our method is similar to the sequential approach used in Hermans et al. (2020) in the sense that both approaches estimate a ratio function repeatedly while the particles (or the sampler) are being updated. We estimate $\frac{q(\boldsymbol{\theta})p(\boldsymbol{x})}{p(\boldsymbol{x},\boldsymbol{\theta})}$ but Hermans et al. estimates $\frac{p(\boldsymbol{x}|\boldsymbol{\theta})\tilde{p}(\boldsymbol{\theta})}{\tilde{p}(\boldsymbol{x})\tilde{p}(\boldsymbol{\theta})}$ where $\tilde{p}(\boldsymbol{\theta})$ is a proposal prior and $\tilde{p}(\boldsymbol{x})$ is the marginal of $p(\boldsymbol{x}|\boldsymbol{\theta})\tilde{p}(\boldsymbol{\theta})$. At convergence, $q(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{x})$ and thus our ratio converges to 1, while the ratio used by Hermans et al. converges to the likelihood to evidence ratio. Instead of modelling the posterior directly, Tran et al. considers modelling the likelihood $q_{\boldsymbol{w}}(\boldsymbol{x}|\boldsymbol{\theta})$ and minimizing the KL divergence between $q(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})$ and $p(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})$. The KL divergence, up to a constant, can be written as (see (4) in Tran et al. (2017) for details) $\int q_{\boldsymbol{w}}(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}}) \log \frac{q_{\boldsymbol{w}}(\boldsymbol{x}_{\text{obs}}|\boldsymbol{\theta})}{p(\boldsymbol{x}_{\text{obs}}|\boldsymbol{\theta})} \mathrm{d}\boldsymbol{\theta} + F_{\boldsymbol{w}}(\boldsymbol{\theta})$, where $F_{\boldsymbol{w}}(\boldsymbol{\theta})$ is a term that can be easily approximated using Monte-Carlo via samples of $q(\boldsymbol{\theta})$. If $q_{\boldsymbol{w}}(\boldsymbol{x}|\boldsymbol{\theta})$ is an implicit generative model (i.e., a neural sampler that produces $\boldsymbol{x}$ given an input $\boldsymbol{\theta}$), one can use logistic regression to estimate the likelihood ratio $\frac{q_{\boldsymbol{w}}(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{\theta})}$ given a $\boldsymbol{\theta}$ and then use it to approximate the KL. The optimal $\boldsymbol{w}$ can be found by minimizing this approximated KL. In comparison, we directly model the posterior using a particle distribution and estimate the ratio between the approximated posterior and the true posterior. Put simply, Tran et al. models the likelihood function and estimates the *likelihood ratio*, while we model the posterior and estimate the *posterior ratio*.

Although our method is almost completely free from tuning, in the future we can consider a more automated algorithm where hyper-parameters (such as η) in our algorithm can be tuned according to some criterion, since low-maintenance methods are always preferred by scientific users. Further, we could also consider a sequential version of our method by gradually zeroing in on estimating a log-ratio which is more relevant to our observation.

References

Mark A Beaumont, Wenyang Zhang, and David J Balding. Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

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

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

Peter J Diggle and Richard J Gratton. Monte carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46 (2):193–212, 1984.

Conor Durkan, Iain Murray, and George Papamakarios. On contrastive learning for likelihood-free inference. In *International Conference on Machine Learning*, pages 2771–2781. PMLR, 2020.

Joeri Hermans, Volodimir Begy, and Gilles Louppe. Likelihood-free mcmc with amortized approximate ratio estimators. In *International Conference on Machine Learning*, pages 4239–4248. PMLR, 2020.

Qiang Liu. Stein variational gradient descent as gradient flow. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3118–3126, 2017.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in Neural Information Processing Systems*, 29, 2016.

Jan-Matthis Lueckmann, Pedro J Goncalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H Macke. Flexible statistical inference for mechanistic models of neural dynamics. *arXiv preprint arXiv:1711.01861*, 2017.

George Papamakarios and Iain Murray. Fast ε-free inference of simulation models with bayesian conditional density estimation. In *Advances in neural information processing systems*, pages 1028–1036, 2016.

George Papamakarios, David Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 837–848. PMLR, 2019.

Jonathan K Pritchard, Mark T Seielstad, Anna Perez-Lezaun, and Marcus W Feldman. Population growth of human y chromosomes: a study of y chromosome microsatellites. *Molecular biology and evolution*, 16(12):1791–1798, 1999.

Scott A Sisson, Yanan Fan, and Mark Beaumont. *Handbook of approximate Bayesian computation*. CRC Press, 2018.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

Alvaro Tejero-Cantero, Jan Boelts, Michael Deistler, Jan-Matthis Lueckmann, Conor Durkan, Pedro J. Gonçalves, David S. Greenberg, and Jakob H. Macke. sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, 5(52):2505, 2020. doi: 10.21105/joss.02505. URL https://doi.org/10.21105/joss.02505.

Owen Thomas, Ritabrata Dutta, Jukka Corander, Samuel Kaski, Michael U Gutmann, et al. Likelihood-free inference by ratio estimation. *Bayesian Analysis*, 2021.

Dustin Tran, Rajesh Ranganath, and David M Blei. Hierarchical implicit models and likelihood-free variational inference. *arXiv preprint arXiv:1702.08896*, 2017.

Appendix A. Particle Collapse

Observe that the minimum of Equation (1) for an empirical distribution q is achieved when all particles take values equal to the mode(s) of the target distribution. However, we desire approximate samples of the posterior rather than approximations of the mode(s). Fortunately, this potential issue is circumvented as a density ratio estimator is used to estimate the KL divergence instead of approximating the KL using the empirical density functions (delta functions) represented by particles.

Appendix B. NN Details

Specifications of our NN:

- Mixed Gaussian, Section 3.1: MLP with 50 hidden units, 3 layers, batch-size 64, learning rate 1e-4, max epochs 1000, optimizer Adam
- M/G/1, Section 3.2: MLP with 50 hidden units, 3 layers, batch-size 256, learning rate 1e-4, max epochs 1000, optimizer Adam

Appendix C. M/G/1 Queue Experiment Set-up

We follow a very similar implementation to that of (Papamakarios et al., 2019). The model is defined by three equations

$$\forall i \quad s_i \sim U(\theta_1, \theta_2), \quad v_i - v_{i-1} \sim \mathrm{Exp}(\theta_3), \quad d_i - d_{i-1} = s_i + \max(0, v_i - d_{i-1}). \tag{8}$$

These can be interpreted as follows: the server takes $s_i \sim U(\theta_1, \theta_2)$ to serve customer i. The time between customer i's arrival v_i and customer $(i-1)$'s arrival v_{i-1}, is such that $v_i - v_{i-1} \sim \mathrm{Exp}(\theta_3)$. Parameters $(\theta_1, \theta_2, \theta_3)$ are the parameters we wish to do inference on. The priors we place on these parameters are given by

$$\theta_1 \sim U(0, 10), \qquad \theta_2 \sim U(0, 10), \qquad \theta_3 \sim U(0, 1/3).$$

The output from the process is the inter-departure times $d_i - d_{i-1}$ where d_i denotes the departure time of customer i. We have $d_i - d_{i-1} = s_i + \max(0, v_i - d_{i-1})$.

For our experiments we take the total number of customers $I = 50$. To reduce the dimensionality and make inference feasible we take a summary statistic of the output: we record the (0, 25, 50, 75, 100)th quantiles of the inter-departure times and output that. In Papamakarios and Murray (2016) they introduce a pilot run of 100K simulations to allow for the normalisation of $\boldsymbol{x}$ (to have zero mean and unit covariance) - we do not employ this as we feel this would be unfeasible in many other scientific applications.

We take the ground truth parameters $\boldsymbol{\theta}^{\star} = (1, 5, 0.2)$ and simulate $\boldsymbol{x}_{\text{obs}}$ from these parameters.