3rd Symposium on Advances in Approximate Bayesian Inference, 2020 1–11

Evidence Estimation by Kullback-Leibler Integration for Flow-Based Methods

Anonymous Authors
Anonymous Institution

1. Introduction

Bayesian inference is based on the posterior distribution of a set of latent variables x, given a set of observations y:

$$p(x|y) = \frac{p(y|x)p_0(x)}{p(y)},$$

where $p(y|x)$ is the likelihood of the data, and $p_0(x)$ is the prior. For simplicity we drop the conditioning on y and write only $p(x) := p(x|y)$ for the posterior. The *evidence* $Z = p(y)$ is an important quantity for both model comparison and hyper-parameter optimization, but computing it is typically intractable for interesting problems.

There is a large class of methods to estimate the evidence, including advanced methods like Thermodynamic Integration (TI) (Gelman and Meng, 1998; Lartillot and Philippe, 2006) or Bayesian quadrature (O'Hagan, 1991; Rasmussen and Ghahramani, 2003). Most of these techniques are standalone methods and do not compute nor approximate the posterior.

On the other hand, many methods, such as Variational Inference (VI) aim to find the best posterior approximation. We are particularly interested in a class of approaches, which we refer as *Flow Based Variational Inference*, where the optimization of the variational distribution can be described as an Ordinary Differential Equation (ODE), we give a precise definition in Section 2.1. In this work we show that for flow based VI we can approximate the posterior $p(x)$ and estimate the evidence Z at the same time. More specifically we give a general approach on the computation of the log evidence for flow based methods. We illustrate our approach by using the Stein Variational Gradient Descent (SVGD) (Liu and Wang, 2016) algorithm and show preliminary results on toy problems.

2. Method

2.1. Flow Based Variational Inference

Variational Inference (VI) aims at finding a distribution q that most closely approximates the target distribution p, while constraining q to be a member of some family $\mathcal{Q}$ to make the problem tractable. The quality of the approximation is measured by the Kullback-Leibler (KL) divergence between the distributions q and p. The most common approach (Blei et al., 2017), is to parametrize q with a set of variational parameters χ, thus turning inference into an optimization problem, solving $\arg\min_\chi \mathrm{KL}(q_\chi||p)$.

A different approach is to start with a tractable initial distribution $x(0) \sim q_0$ and apply an ODE flow:

$$\frac{dx(t)}{dt} = \varphi_t(x(t)), \tag{1}$$

for suitable smooth function $\varphi_t \colon \mathbb{R}^D \to \mathbb{R}^D$, such that $\frac{d\mathrm{KL}(q_t||p)}{dt} \leq 0$.

© A. Authors.

Assuming such dynamics for $x(t)$, the evolution of the corresponding density q_t is given by the continuity equation:

$$\frac{dq_t(x)}{dt} = -\nabla \cdot (q_t(x)\varphi(x)), \tag{2}$$

where $\nabla = (\partial/\partial x_1, \ldots, \partial/\partial x_D)$. See Appendix A.1 for a proof in the current context.

In practice, the continuous time variable in (1) must be discretized, leading to iterative algorithms like:

$$x_{t+1} = x_t + \varepsilon_t \varphi_t(x_t). \tag{3}$$

2.2. KL Flow

We will now consider the dynamics of the KL divergence induced by (1). Recall that the KL-divergence is defined by $\mathrm{KL}(q||p) = \int q(x) \log \frac{q(x)}{p(x)} dx$. As we show in Appendix A.2 its time evolution is given by:

$$\frac{d\mathrm{KL}(q_t||p)}{dt} = \int q_t(x)\varphi_t(x) \cdot \nabla(\log q_t(x) - \log p(x))dx \tag{4}$$

$$= -\mathbb{E}_{q_t}\left[\nabla \cdot \varphi_t(x) + \varphi_t(x) \cdot \nabla \log p(x)\right]. \tag{5}$$

Note that Equation (5) is the expectation of the *Stein operator* applied on φ, also used in Liu and Wang (2016).
Equation (4) and Equation (5) can both we used to calculate derivative of the KL but differ in their application. Unlike Equation (5), Equation (4) requires to compute $\nabla \log q_t$. Maoutsa et al. (2020), for example, use an estimator for $\nabla \log q$ based on kernels.

2.3. Log-Evidence by Flow Integration

Now that we have expressions for the rate of change in the KL-divergence under a flow, we can construct an equation for the log of the evidence.

Theorem 1 *Given a target distribution $p(x) = \frac{1}{Z}\exp(-V(x))$ and a variational distribution $q_t(x)$ with a flow defined by $\frac{dq_t(x)}{dt} = -\nabla \cdot (q_t(x)\varphi_t(x))$ and an initial density q_0, then*

$$\log Z \geq \mathbb{E}_{q_0}\left[-\log q_0(x) - V(x)\right] - \int_0^\infty \frac{d}{dt} KL(q_t||p)dt, \tag{6}$$

with strict equality if $KL(q_\infty||p) = 0$.

Proof: The KL-divergence between q_t and p can be expressed as:

$$\mathrm{KL}(q_t\|p) = \int_X q_t(x) \log \frac{q_t(x)}{p(x)} dx = \int_X q_t(x) \log q_t(x) dx + \int_X q_t(x)V(x)dx + \log Z$$
$$= -\mathbb{H}\left[q_t\right] + \mathbb{E}_{q_t}\left[V(x)\right] + \log Z, \tag{7}$$

where $\mathbb{H}[q]$ denotes the entropy of q. We can connect Equation (7) to (5) by writing the KL as an integrated path:

$$\mathrm{KL}(q_\infty\|p) = \mathrm{KL}(q_0\|p) + \int_0^\infty \frac{d}{dt}\mathrm{KL}(q_t\|p)dt.$$

(Note the resemblance with TI where we integrate the path from the prior to the posterior.)

If we assume that the final distribution q_∞ perfectly approximates the target, i.e. $\mathrm{KL}(q_\infty \| p) \approx 0$ and replace $\mathrm{KL}(q_0 || p)$ by Equation (7) at $t = 0$ we obtain the result:

$$\log Z = \mathbb{H}[q_0] - \mathbb{E}_{q_0}[V(x)] - \int_0^\infty \frac{d}{dt}\mathrm{KL}(q_t \| p) dt \quad \blacksquare$$

2.4. Training and Integrating

In Equation (6), the expectations with respect to q_0 can be computed analytically or approximated easily, provided the sampling from q_0 is inexpensive. We propose to estimate the integral as a part of the training procedure. We apply the same time-discretization used for the inference and approximate it with a simple numerical quadrature.

The term $\frac{d}{dt}\mathrm{KL}(q_t || p)$ can be replaced by Equation (4) or (5) and reuses existing computation from the inference as we will show in Section 2.5. Additionally, if the inference reaches a fixed-point, i.e. $\frac{dx(t)}{dt} = 0 \equiv \varphi(x) = 0$, Equation (4) guarantees that $\frac{d\mathrm{KL}}{dt} = 0$ and the integral is finite.

Let T be the number of steps until convergence and take ε_t to be the same as in Equation (3), then the integral can be approximated by:

$$\int_0^\infty \frac{d\mathrm{KL}(q_t || p)}{dt} dt \approx \sum_{i=0}^{T} \varepsilon_i \left. \frac{d\mathrm{KL}(q_t || p)}{dt} \right|_{t=i}.$$

2.5. Stein Variational Gradient Descent

We present a brief overview of SVGD and show how Theorem 1 can be applied to estimate $\log Z$. We refer the reader to Liu and Wang (2016) for a full treatment of the theory of SVGD.

SVGD belongs to the class of algorithms described in Section 2.1. Liu and Wang (2016) start with Equation (1) and derive an optimal φ_t, in the sense that it maximizes the expectation in Equation (5). When φ_t is constrained to lie in the unit ball in a Reproducing Kernel Hilbert Space (RKHS) the value of this expectation is called the *kernelized Stein discrepancy*. This goodness of fit measure was first introduced by Liu et al. (2016), they furthermore showed that its value is attained for:

$$\varphi_t^* = \frac{\varphi_{q_t,p}}{\|\varphi_{q_t,p}\|_k}, \text{ where } \varphi_{q_t,p}(x) = \mathbb{E}_{y\sim q_t}\left[k(x,y)\nabla \log p(y) + \nabla k(x,y)\right]. \tag{8}$$

In practice the variational distribution q_t is approximated by an *empirical distribution* based on a set of samples $\{x_i^t\} \sim q_t$, i.e. $q_t(x) = \frac{1}{N}\sum_{i=1}^N \delta(x_i^t - x)$. Therefore, we start by drawing N samples $\{x_i^0\}_{i=0,\dots,N}$ from the initial distribution q_0, and repeatedly update their positions using φ_t^* and Equation (3). Note that, since q_t is itself discretized, we only have an empirical estimate of φ_t. We explore the implications of this approximation in the experiments of Section 3.2. The inference steps are summarized in Algorithm 1. We believe that due to the non-parametric nature of SVGD the odds of satisfying the condition $\mathrm{KL}(q_\infty || p) \approx 0$ are good, making it well suited for flow integration.

2.6. SVGD KL Flow Estimators

Since SVGD calculates φ_t at each iteration, we can also calculate $\frac{d\text{KL}}{dt}$. In this sections we describe three estimators that can be used.

Stein discrepancy: The most obvious approach is to insert Equation (8) into Equation (5) and use the empirical estimation of the result, i.e. the negative kernelized Stein discrepancy:

$$\frac{d}{dt}\,\text{KL}(q^t\|p) \approx -\frac{1}{N}\sum_{i,j}^{N}\Bigg[k(x_i,x_j)\nabla\log p(x_i)\cdot\nabla\log p(x_j) + \nabla\log p(x_i)\cdot\nabla_{x_j}k(x_i,x_j) + \nabla\log p(x_j)\cdot\nabla_{x_i}k(x_i,x_j) + \sum_{k=1}^{D}\frac{\partial^2 k(x_i,x_j)}{\partial(x_i)_k\partial(x_j)_k}\Bigg]. \tag{9}$$

Considering Equation (5), we see that this quantity depends on both φ and its gradient. Since we do not know what happens to the gradient as φ approaches 0, there is no guarantee that (13) will go to 0.

Unbiased Stein discrepancy: An alternative is the unbiased estimator of the Stein discrepancy as described in Liu et al. (2016). For details, see Appendix B.2.

RKHS-Norm: We propose a third estimator which is guaranteed to converge to 0 if the SVGD algorithm converges to a fixed solution. Following from Liu et al. (2016), the Stein discrepancy can also be written as $\frac{d\text{KL}}{dt} = -\|\varphi\|_k^2$, where $\|\cdot\|_k$ is the *RKHS-norm* for a given kernel k.

By definition[1] we have $\varphi(x) = \sum_{i=1}^{N}\alpha_i k(x,x_i)$ and $\|\varphi\|_k^2 = \sum_{i,j}^{N}\alpha_i k(x_i,x_j)\alpha_j$ where $\{x_i\}_{i=1}^{N}$ are the set of particles representing q_t. Let φ be the vector $\varphi_i = \varphi(x_i)$, then we can invert this equation for all particles to obtain $\alpha = K^{-1}\varphi$, where K is the kernel matrix. This leads to the final estimator for $\frac{d\text{KL}}{dt}$:

$$\frac{d\text{KL}(q_t||p)}{dt} \approx -\|\varphi\|_k^2 = -\varphi^\top K^{-1}\varphi. \tag{10}$$

This estimator is not only the guaranteed to go to 0 with φ but also avoids second order derivatives. Due to numerical instabilities of the kernel matrix it may need to be regularized in order to be inverted, i.e. $K^{-1} \approx (K+\nu I)^{-1}$ for $0 < \nu << 1$. We compare the different estimators experimentally in Section 3.1.

3. Experiments

In this section we discuss some experiments studying the simplest example: approximating a 2D Gaussian distribution. We empirically explore the different parameters of the algorithm, namely the estimator for $d\text{KL}/dt$, the step size ε, the number of particles and the problem setup. We show an unsuccessful attempt to use this method to compute the log evidence in a Bayesian generalized linear regression in Appendix C and provide a hypothesis to explain

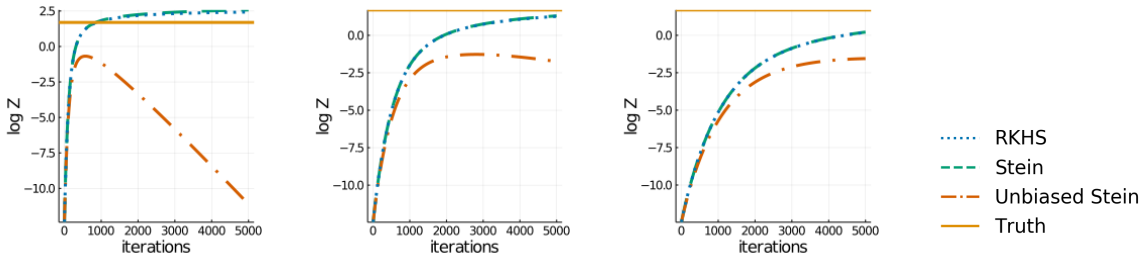
1. Here we only show the solution in 1 dimension, but it can easily be extended to higher dimensions for common kernels.

this issue.
Following Liu and Wang (2016), we used a RBF kernel $k(x, y) = \exp(-0.5\|x - y\|^2/l^2)$. We set l to $\frac{\text{median}(A)}{\log N}$ at each iteration, where A is the matrix of the pairwise Euclidean distances between the particles. We use a constant step size $\varepsilon_t = \varepsilon$ for all experiments.

3.1. Evaluating Estimators and Step Size

We first compare the effect of the step size and the different flow estimators. For these experiments we draw 200 particles from $q_0 = \mathcal{N}(0, I_2)$ and set $p = \mathcal{N}(\mu = [4, 5], \Sigma = [1, 0.5; 0.5, 1])$. Figure 1 shows the results obtained by the estimators: the *RKHS*-norm as described in Equation (10), the *Stein* discrepancy as in Equation (13) and the *unbiased Stein* estimator as described in Equation (16). We compare it to the true value $\frac{1}{2}\log|2\pi\Sigma|$.

The RKHS-norm estimator turns out to be the most reliable and accurate of the three available flow estimators. As expected the naïve Stein discrepancy tends to overestimate more than the RKHS-norm, as it naturally comes with a stronger bias from the estimation of Equation (5). A general issue with all estimators is the steepness of the function at the origin. This magnifies all approximation errors introduced by the discretization. A smaller step size improves the result but requires more computational resources to converge.

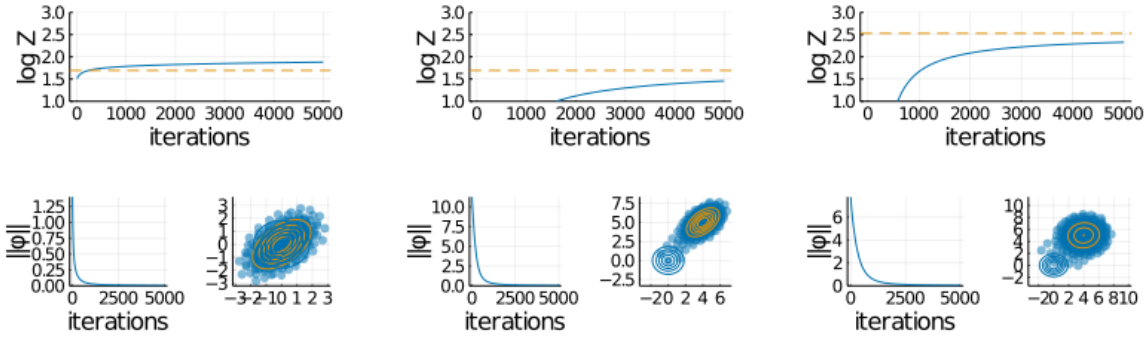


a) $\varepsilon = 0.05$ b) $\varepsilon = 0.01$ c) $\varepsilon = 0.005$

Figure 1: $\log Z$ estimation as a function of the number of iterations for different estimators of the kernelized Stein discrepancy and different step sizes.

3.2. Varying the Target

Our preliminary experiments show that one of the most critical factors for a good approximation of the log-evidence is the choice of target and initial distributions. We evaluate the approximation quality for different targets with varying covariances and/or means. For these experiments we used 500 particles and a step size of 0.05. The convergence of $\log Z$ as well as $\|\varphi\|$ and an illustration of the problem are shown in Figure 2. As can be seen from the figures the method typically converges to a slight overestimation of $\log Z$. Large differences in covariance and/or mean lead to worst results, an observation confirmed in the linear regression experiment in Appendix C. A simple solution would be to take a Laplace approximation as a starting point, which might also avoid the strong steepness at the beginning.



a) $p = \mathcal{N}([0,0],[1,0.5;0.5,1])$ b) $p = \mathcal{N}([4,5],[1,0.5;0.5,1])$ c) $p = \mathcal{N}([4,5],[2,0.1;0.1,2])$

Figure 2: Approximating $\log Z$ of different Gaussian targets p starting from a standard normal.

3.3. Effect of the Number of Particles

One of the approximations involved is the empirical estimation of Equation (5) with a finite number of samples. In Figure 3, we reuse the problem of Section 3.1 but only examine the convergence of the *RKHS*-norm approach. We fix the step size $\varepsilon = 0.05$ and run the problem with 50, 100 and 200 particles. The figure shows that increasing the number of particles makes the estimate more accurate. Unfortunately, SVGD scales poorly with the number of particles, thus putting practical limits on the accuracy of the computation.

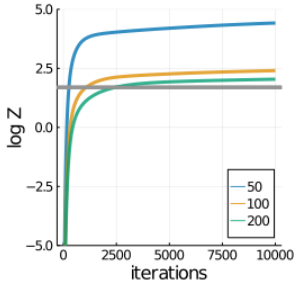


Figure 3: Estimation of $\log Z$ as function of the number of iterations for a varying number of particles.

4. Conclusion

We presented *KL flow integration*, a way to extend flow based variational inference algorithms to also estimate the log evidence, or at least a lower bound thereof. As an example, we added log evidence estimation to SVGD and provided some first experimental results. We intend to continue to study, analytically and empirically, the issues arising when working on more difficult problems such as Bayesian generalized linear regression. Since our framework is not limited to a single algorithm, we also plan on exploring applications to more flows, such as the work of Maoutsa et al. (2020) and other expressive parametric models.

References

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.

Andrew Gelman and Xiao-Li Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical science*, pages 163–185, 1998.

Nicolas Lartillot and Hervé Philippe. Computing bayes factors using thermodynamic integration. *Systematic biology*, 55(2):195–207, 2006.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in Neural Information Processing Systems 29*, pages 2378–2386. Curran Associates, Inc., 2016.

Qiang Liu, Jason Lee, and Michael Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *Proceedings of Machine Learning Research*, volume 48, pages 276–284. PMLR, New York, New York, USA, 20–22 Jun 2016.

Dimitra Maoutsa, Sebastian Reich, and Manfred Opper. Interacting particles solutions of fokker-planck eqations through gradient-log-density estimation. *Entropy*, 22, 2020.

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

Carl Edward Rasmussen and Zoubin Ghahramani. Bayesian monte carlo. *Advances in neural information processing systems*, pages 505–512, 2003.

Dilin Wang, Ziyang Tang, Chandrajit Bajaj, and Qiang Liu. Stein variational gradient descent with matrix-valued kernels. In *Advances in Neural Information Processing Systems 32*, pages 7836–7846. Curran Associates, Inc., 2019.

Appendix A. Detailed derivations

A.1. Conservation Law for Density

To prove the conservation law for a probability density under an ODE flow, consider the time derivative of the expectation of an arbitrary smooth function $g(x) : \mathbb{R}^D \to \mathbb{R}$:

$$\begin{aligned}\frac{d}{dt}\mathbb{E}\left[g(X(t))\right] =& \mathbb{E}\left[\nabla g(X(t)) \cdot \frac{dX(t)}{dt}\right] = \mathbb{E}\left[\nabla g(X(t)) \cdot \varphi(X(t))\right] \\ =& \int \nabla g(x) \cdot q_t(x)\varphi(x)dx = -\int g(x)\nabla \cdot (q_t(x)\varphi(x))\, dx,\end{aligned}$$

where $\cdot$ is the dot product and $\nabla = \left(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_D}\right)$ is the gradient operator. We used the ODE flow from Equation (1), the chain rule and Green's identity. Naturally we can also express the derivative by the change of q_t :

$$\frac{d}{dt}\mathbb{E}\left[g(X(t))\right] = \int \frac{dq_t(x)}{dt} g(x)dx. \tag{11}$$

Since this is true for all functions g, we obtain the following gradient flow for q_t:

$$\frac{dq_t(x)}{dt} = -\nabla \cdot (q_t(x)\varphi(x)) \tag{12}$$

Note that this is classic result for in physics of conserved quantities, known as a continuity equation.

A.2. KL Derivatives

Given $\mathrm{KL}(q||p) = \int q(x) \log q(x) - \log p(x) dx$ the time derivative is given by:

$$\begin{aligned}
\frac{d\mathrm{KL}(q||p)}{dt} &= \frac{d}{dt} \int q_t(x) \left(\log q_t(x) - \log p(x)\right) \\
&= - \int \nabla \cdot (q_t(x)\varphi_t(x))(\log q_t(x) - \log p(x)) dx \\
&= \int q_t(x)\varphi_t(x) \cdot \nabla(\log q_t(x) - \log p(x)) dx \\
&= \int \nabla q_t(x) \cdot \varphi_t(x)) dx - \int q_t(x)\varphi_t(x) \cdot \nabla \log p(x) dx \\
&= - \mathbb{E}_{q_t} \left[\nabla \cdot (\varphi_t(x)) - \varphi_t(x) \cdot \nabla \log p(x)\right],
\end{aligned}$$

where we used Equation (2), Green's identity and the fact that $\lim_{x\to\infty} q_t(x) = 0$.

Appendix B. SVGD Algorithm and estimators

B.1. SVGD Algorithm with Flow Integration

We summarize the different steps presented in Section 2 with Algorithm 1. Note that SVGD does not actually calculate the full time derivative at every step, but the information required is present at every step, namely the updated φ (Liu et al., 2016).

Algorithm 1: Stein Variational Gradient Descent with Flow Integration
Input: Initial distribution q_0, target distribution $p(x) = \frac{1}{Z}\exp(-V(x))$, step size ε_t, # iterations T, kernel function $k(x, y)$, KL derivative estimator $F(\varphi, x) \approx \frac{d\mathrm{KL}}{dt}$ (see Equation (4), (5) or (10)).
Output: Set of particles $\{x_i\}_{i=1}^N$, log-evidence approximation $\log \tilde{Z}$
Init: Sample initial particles $\{x_i\}_{i=1}^N$ from q_0, compute $\mathrm{KL} = \mathbb{H}[q_0] - \mathbb{E}_{q_0}[V(x)]$
for $t = 1 : T$ **do**
  $\varphi_i = \frac{1}{N}\sum_{j=1}^N -k(x_j^k, x)\nabla V(x_i^k) + \nabla k(x_i^k, x),\ \forall i = 1 : N$ # SVGD step
  $\mathrm{KL} \leftarrow \mathrm{KL} + \varepsilon_t F(\varphi, x)$
  $x_i \leftarrow x_i + \varepsilon_t \varphi_i,\ \forall i = 1 : N$
end

The computation of the derivative of the KL divergence is written only as F_t. This is because there are two ways to estimate F_t: it can either be estimated by computing the Stein discrepancy or from the RKHS norm of φ.

B.2. Estimation Based on the Stein Discrepancy

The derivative of the KL divergence is equal to the negative of the Stein discrepancy, which is defined by

$$\mathbb{D}(q, p) = \max_{\varphi \in \mathcal{H}} \left\{ \mathbb{E}_{x\sim q}\left[\mathrm{trace}(A_p\varphi(x))\right]^2 \mid \|\varphi\| \leq 1 \right\} \tag{13}$$

where $\mathcal{A}_p\varphi = \varphi\nabla \log p + \nabla\varphi$ is the Stein operator and the maximum is attained by

$$\varphi^* = \frac{\varphi_{q,p}}{\|\varphi_{q,p}\|} \text{ where } \varphi_{q,p}(x) = \mathbb{E}_{y\sim q}\left[\mathcal{A}_p k(x,y)\right] \tag{14}$$

Thus the derivative of the KL divergence can be obtained by applying the Stein operator twice to the kernel of the RKHS:

$$\begin{aligned}\frac{d}{dt}\,\mathrm{KL}(q_t\|p) \approx -\frac{1}{N}\sum_{i,j}^{N}\Bigg[&k(x_i,x_j)\nabla\log p(x_i)\cdot\nabla\log p(x_j) + \nabla\log p(x_i)\cdot\nabla_{x_j}k(x_i,x_j)\\ &+\nabla\log p(x_j)\cdot\nabla_{x_i}k(x_i,x_j) + \sum_{k=1}^{D}\frac{\partial^2 k(x_i,x_j)}{\partial(x_i)_k\partial(x_j)_k}\Bigg].\end{aligned} \tag{15}$$

However, according to Liu et al. (2016) this is a biased estimator for the Stein discrepancy, an unbiased one can be constructed by leaving out the terms where $i = j$.

$$\begin{aligned}\frac{d}{dt}\,\mathrm{KL}(q_t\|p) \approx -\frac{1}{N}\sum_{i\neq j}^{N}\Bigg[&k(x_i,x_j)\nabla\log p(x_i)\cdot\nabla\log p(x_j) + \nabla\log p(x_i)\cdot\nabla_{x_j}k(x_i,x_j)\\ &+\nabla\log p(x_j)\cdot\nabla_{x_i}k(x_i,x_j) + \sum_{k=1}^{D}\frac{\partial^2 k(x_i,x_j)}{\partial(x_i)_k\partial(x_j)_k}\Bigg].\end{aligned} \tag{16}$$

Unfortunately neither of these converge to 0 when $\varphi \to 0$, making them unsuitable to approximate an improper integral.

B.3. Estimating the Gradient Using RKHS Norm

Liu et al. (2016) showed that $\mathbb{D}(q,p) = \|\varphi\|_k^2$, so that

$$\frac{d}{dt}\,\mathrm{KL}(q_t\|p) = -\mathbb{E}\left[\mathrm{trace}\left(\mathcal{A}_p\varphi(x)\right)\right] = -\mathbb{D}(q_t,p) = -\|\varphi_{q_t,p}\|^2. \tag{17}$$

Using the fact that φ lies in the RKHS spanned by k and the representer theorem we know that:

$$\varphi(x) = \sum_{i=1}^{n}\alpha_i k(x,x_i) \text{ and } \|\varphi\|_k^2 = \sum_{i,j=1}^{n}\alpha_i k(x_i,x_j)\alpha_j \tag{18}$$

where $\{x_i\}$ are the SVGD particles. Let α denote the vector (α_i), these coefficients are unknown, but letting φ denote the vector $(\varphi(x_i))$ and K the matrix $(k(x_i,x_j))$, they can be found by inverting K, i.e.

$$\alpha = K^{-1}\varphi \tag{19}$$

The RKHS norm of φ can then be found as

$$\|\varphi\|_k^2 = \varphi^\top K^{-1}\varphi = \sum_{i,j=1}^{n}\varphi(x_i)K_{ij}^{-1}\varphi(x_j) \tag{20}$$

When the problem is more than 1 dimensional (i.e. the SVGD particles are vectors rather than scalars) the general form of Equation (20) is a tensor equation. This is because in higher dimensions the functions φ are vector-valued functions and the kernel matrix-valued. Details of this more general formulation of SVGD can be found in Wang et al. (2019). However, as is the case in vanilla SVGD, while such generality can be interesting it is not always needed in practice. When using SVGD with scalar-valued kernels the norm of φ can be found using the simple relation $\|\varphi\|^2 = \sum_i^D \|\varphi_i\|^2$, where φ_i are the component functions, whose norm is calculated as in (20).

Appendix C. Bayesian Generalized Linear Regression

We applied our methods in the problem of Bayesian generalized linear regression. Given some inputs $x_i \in \mathbb{R}^D$ and output $y_i \in \mathbb{R}$ we define the model:

$$y_i = \Phi(x_i) \cdot w + \chi, \tag{21}$$

where $\Phi : \mathbb{R}^D \to \mathbb{R}^M$, $\chi \sim \mathcal{N}(0, \frac{1}{\sqrt{\beta}})$, $w \sim \mathcal{N}(0, \Sigma_0)$. Since both the likelihood and the prior are Gaussians, the posterior and therefore the evidence can be found analytically,

$$\log Z = 2 \log \left| 2\pi \left(\Sigma_0^{-1} + \beta \Phi(\boldsymbol{x})^\top \Phi(\boldsymbol{x}) \right)^{-1} \right|,$$

where $\Phi(\boldsymbol{x})$ is the (row)-vector of observations.

We create a toy dataset by sampling $x_i \in \mathcal{U}[-3, 3]$, we set $w = [2, -1, 0.2]$ and $\Phi(x) = [1, x, x^2]$, finally we obtain y_i via Equation (21). We sample N particles from an initial Gaussian distribution centered around the MAP and with covariance $0.1 I_M$. We set the prior covariance Σ_0 to be $0.1 I_M$ as well and the likelihood precision to be $\beta = 2$

Figure 4 show the convergence result for 50, 100 and 200 particles and a step size of 0.001 over 500 iterations. Our algorithm completely overestimate the true value of $\log Z$.

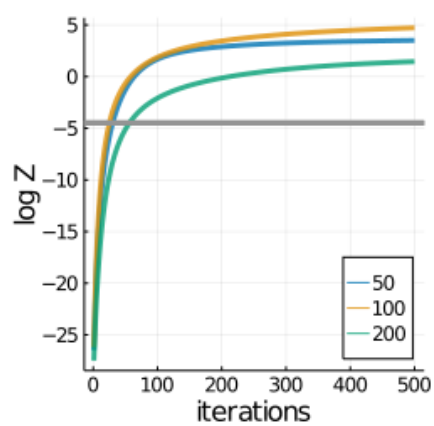


Figure 4: Log evidence computation for a linear regression problem

By inspecting the initial values of φ ($\sim 10^5$) we realized that they were the cause of the

issue. Since the posterior is particularly peaked, the gradients at the initial distribution grow quadratically with the distance to the true weight. For optimization it is not a large issue, but for integration it causes additional problem as we need to integrate over an extremely steep curve. This is still an issue we need to figure out how to solve and maybe some methods for integrating discontinuous functions could come as a solution.