Annealed Stein variational gradient descent with localized Gaussian kernel

Zuheng(David) Xu ZUHENG.XU@STAT.UBC.CA
Student number: 24617185

1. Introduction

Stein variational gradient descent (SVGD) (Liu and Wang, 2016) is a particle-based variational inference method, which iteratively transforms a set of samples from a reference distribution to the target distribution. In general, SVGD only requires the gradient of the log density function of the target distribution. This fits well into the framework of Bayesian inference, which usually has intractable normalizing constant in target distribution. Comparing to traditional sampling methods such as Markov Chain Monte Carlo (MCMC), SVGD uses deterministic updates and does not experience slow convergence due to autocorrelation between samples. A big advantage is that it leverage the gradient information of the target distribution and leads to a more efficient sampling procedure.

A common issue of SVGD is the mode degeneracy—particles collapse to a small number of modes Zhuo et al. (2018). To mitigate this problem, D'Angelo and Fortuin (2021) propose *annealed Stein variational gradient descent* (A-SVGD) by introducing an annealing function in the update rule of SVGD, encouraging particles to be more spread. Although this annealing strategy is able to improves the performance of SVGD when the target distribution has multiple modes, we find that the performance is quite sensitive to the value of kernel bandwidth. Inspired by a few synthetic example, we propose to use a localized Gaussian kernel by using the curvature of $\log p$ at the position of each particle. The second order information of $\log p$ leads to an effective choice of the kernel, which is localized on each particle and will be updated adaptively as algorithm runs. multiple empirical experiments show computational gains of our kernel learning scheme over the standard RBF kernel with universal bandwidth.

2. Stein variational gradient descent

In this section, we briefly summarize the development of SVGD and its underlying theory. Similar to normalizing flow (Kobyzev et al., 2020), SVGD aims to learn an invertible transformation from a reference distribution to the target distribution p. Instead of composing a series of parametric functions and optimizing over all the parameters, SVGD iteratively performs a non-parametric transformation to the current distribution and decreases the KL divergence to the target.

Let X be a random variable follows a distribution with density function $q(x)$, which dominates p. Let $T : \mathbb{R}^d \to \mathbb{R}^d$ be an invertible and continuously differentiable map. In particular, we focus on the perturbed identity map:

$$T(x) := x + \epsilon\phi(x), \quad \phi \in \mathcal{H}_k^d \text{ and } \phi \text{ is smooth.}$$

Here $\mathcal{H}_k^d$ is a *Reproducing kernel Hilbert space*(RKHS), a Hilbert space induced by a positive definite kernel $\kappa(x, x') : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. We denote q_T as the push forward measure of $Y := T(X)$, of which

the density function can be written as

$$q_T(y) = q\left(T^{-1}(y)\right) \cdot \left|\det \nabla T^{-1}(y)\right|.$$

The development of SVGD arise from a critical connection between the KL divergence and the *Stein's operator*. Considering the directional derivative of $\mathrm{D}_{\mathrm{KL}}\left(q_T||p\right)$ along $\phi \in \mathcal{H}_k^d$,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathrm{D}_{\mathrm{KL}}\left(q_T||p\right)\Big|_{\epsilon=0} &= -\int q(x)\left(\nabla \log p(x)^T\phi(x) - \operatorname{tr}\nabla\phi(x)\right)\\ &= -\mathbb{E}_q\left[\nabla \log p(X)^T\phi(X) + \operatorname{tr}\nabla\phi(X)\right].\end{aligned}$$

A key observation here is that $\nabla \log p(X)^T\phi(X) + \operatorname{tr}\nabla\phi(X)$ is the *Stein operator* applied to $\phi(x)$. Then since $\phi \in \mathcal{H}_k^d$, the reproducing property of the kernel $\kappa(\cdot,\cdot)$ yields that

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathrm{D}_{\mathrm{KL}}\left(q||p\right)\Big|_{\epsilon=0} = -\left\langle\phi, \mathbb{E}_q\left[\kappa(x,\cdot)\nabla \log p(X)^T\phi(X) + \nabla_x\kappa(x,\cdot)\right]\right\rangle_{\mathcal{H}_k^d}.$$

Note that the gradient is the directional derivative along the steepest descent direction. By maximizing the right hand side over $\phi \in \mathcal{H}_k^d$, we obtain that

$$\phi^\star(\cdot) = \mathbb{E}_q\left[\kappa(x,\cdot)\nabla \log p(X)^T\phi(X) + \nabla\kappa(x,\cdot)\right], \quad \text{and} \tag{1}$$

$$\nabla_\epsilon \mathrm{D}_{\mathrm{KL}}\left(q||p\right)|_{\epsilon=0} = \|\phi^\star\|^2_{\mathcal{H}_k^d} = \mathrm{d}_\kappa(q,p). \tag{2}$$

Here $\mathrm{d}_\kappa(q,p)$ is the *kernelized Stein discrepancy* (KSD) Liu et al. (2016) between q and p. Thus, by iteratively applying the following transformation to X_0,

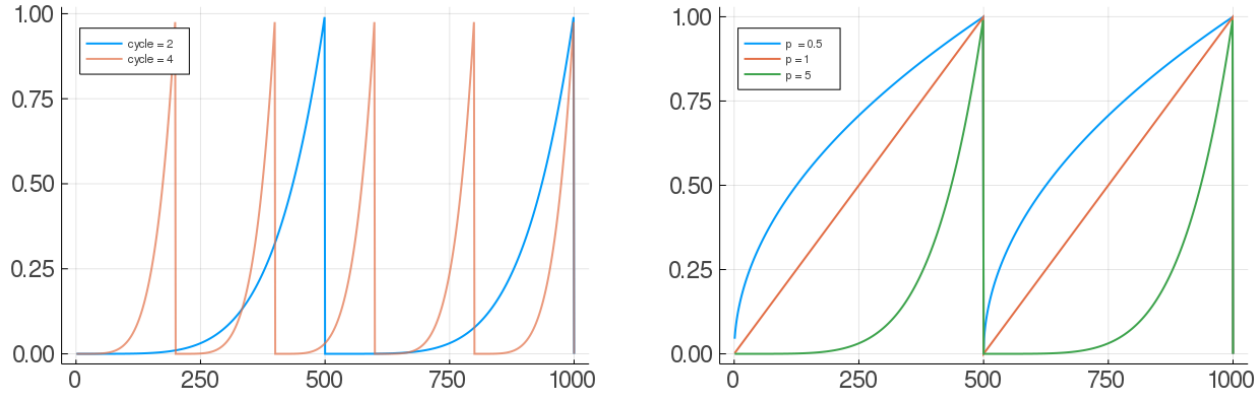
$$X_0 \sim q_0, \quad X_{k+1} \leftarrow T(X_k), \quad T(X_k) := X_k + \epsilon\phi^\star(X_k), \quad k = 0, 1, \ldots. \tag{3}$$

We are able to construct a non-parametric normalizing flow that decreases the KL-divergence sequentially. As revealed in Eq. (2), the fixed point of this transformation is achieved if and only if $\mathrm{d}_\kappa(q,p) = 0$. It is worth pointing out that in general $\mathrm{d}_\kappa(q,p) = 0$ does not imply that $q = p$. In fact, this requires a proper choice of the kernel function $\kappa(\cdot,\cdot)$. A typical choice is the standard RBF kernel, $\kappa(x,x') = \exp(-\frac{1}{h}\|x-x'\|^2$ and the general Gaussian kernel, $\kappa(x,x') = \exp(-(x-x')^T\Sigma^{-1}(x-x'))$. In this report, we will focus on these two kernels.

To implement this non-parametric normalizing flow Eq. (3), Liu and Wang (2016) approximate the optimal direction Eq. (1) using Monte Carlo samples. Specifically, one may draw n samples $\{x_i^{(0)}\}_{i=1}^n$ from the reference distribution p_0 and apply the empirical version of Eq. (3) to the set of particles, i.e., for $i = 0, 1, \ldots, n$ and learning rate $\gamma_k > 0$, the numerical iteration at k-th iteration is as follows,

$$\begin{aligned} &x_i^{(k+1)} \leftarrow x_i^{(k)} + \gamma_k\hat{\phi}_k^\star\left(x_i^{(k)}\right), \quad \text{where}\\ &\hat{\phi}_k^\star(x) := \frac{1}{n}\sum_{j=1}^n\left[\kappa\left(x_j^{(k)}, x\right)\nabla \log p\left(x_j^{(k)}\right) + \nabla_{x_j^{(k)}}\kappa\left(x_j^{(k)}, x\right)\right].\end{aligned} \tag{4}$$

Ideally, this iterative procedure will move the set of particles to the target distribution, which will eventually return samples from p. The detailed description of the standard SVGD is presented in Algorithm 1. In practice, to achieve a faster numerical convergence, Liu and Wang (2016) suggest



(a) (b)

Figure 1: Annealing schedules. Fig. 1a shows two annealing schedules with 2 cycles and 4 cycles respectively. Fig. 1b shows annealing schedules across $p = 0.5, 1, 5$.

using the AdaGrad updates with momentum—also known as RMSprop— instead of vanilla gradient descent update described in Eq. (4). Specifically, RMSprop performs the update at k-th iteration as follows: for $i = 1, 2, \ldots, n$,

$$\begin{aligned} g_{k+1} &\leftarrow \eta g_k + (1-\eta)\left(\hat{\phi}_k^\star\left(x_i^{(k)}\right)\right)^2, \\ x_i^{(k+1)} &\leftarrow x_i^{(k)} + \frac{\gamma_k}{\sqrt{g_{k+1}+\epsilon}}\hat{\phi}_k^\star\left(x_i^{(k)}\right), \end{aligned} \tag{5}$$

where $g_0 = 0 \cdot \mathbb{1}$ and all operations on vectors are elementwise. By default, the momentum factor η is set to 0.9 and the smoothing factor ϵ is set to 10^{-8}. Empirically, we do observe significantly faster convergence of the above update rule than the naive gradient descent update Eq. (4).

3. Annealed Stein variational gradient descent

A key limitation of SVGD is the mixing issue when the target distribution $p(x)$ has multiple modes, where the particles collapse at a particular mode and fail to explore other high density regions. This is also known as the *mode collapse* phenomenon (D'Angelo and Fortuin, 2021; Zhuo et al., 2018) and appears often when the modes are distant from each other. This problem tends to be more severe in high-dimensional settings, but can also be observed on a simple 2D Gaussian mixture target. In this section, we provide a brief introduction of the annealed SVGD (A-SVGD), which is developed to address the mixing issue (D'Angelo and Fortuin, 2021), and use a few synthetic examples to identify a key limitation of the current work. The design of the synthetic example aims to understand the influence of the kernel bandwidth on the behaviour of SVGD or A-SVGD and help us to build the intuition of picking a proper value. The experiments performed in this report use the RMSprop update rule Eq. (5). We run both SVGD and A-SVGD with standard RBF kernel for 1000 iterations under a constant learning rate $\gamma_k = 0.1$.

We start with the development of A-SVGD. Considering the updates Eq. (4), the optimal descent direction $\hat{\phi}^\star(x)$ for the particle x includes two parts: the driving force that moves the particle to

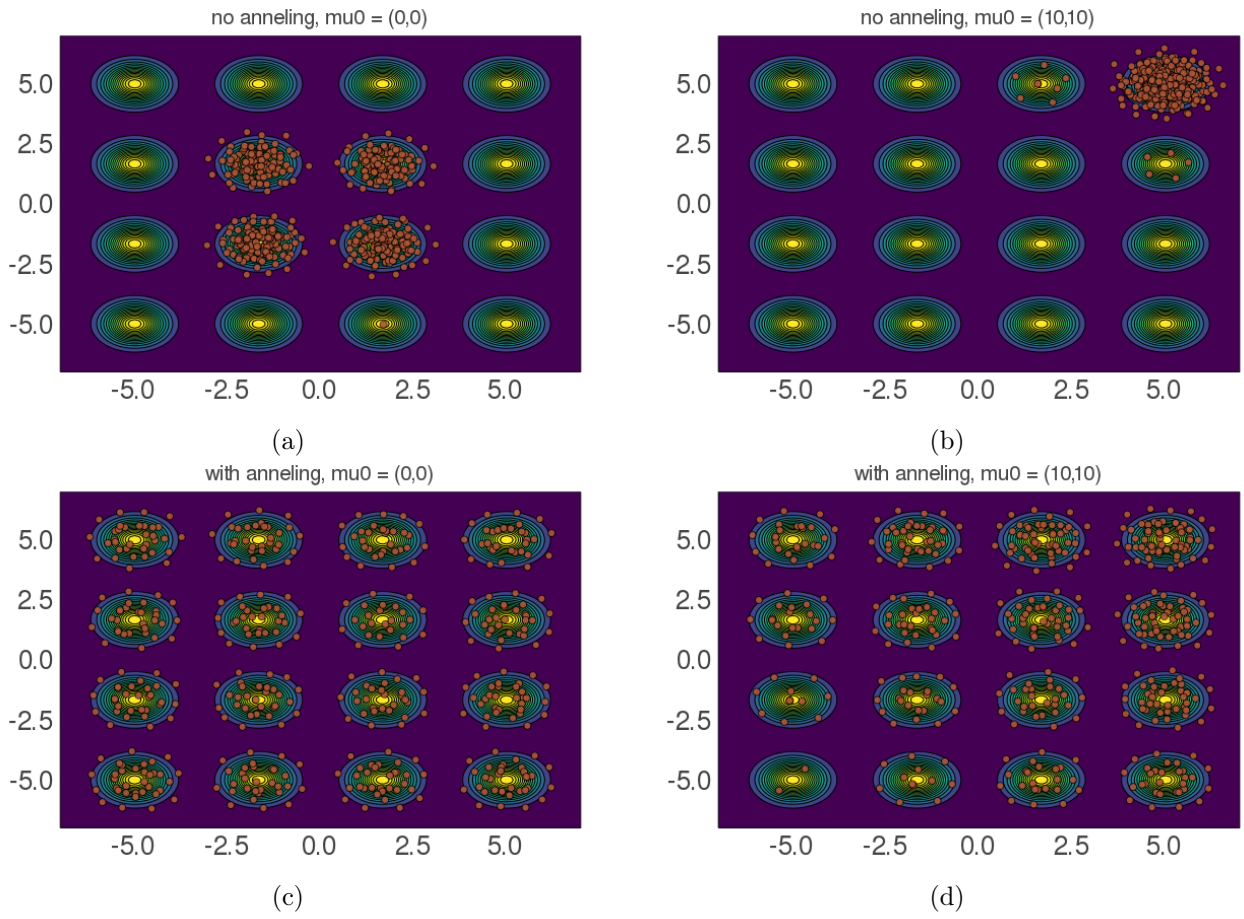


Figure 2: Comparison of SVGD (Figs. 2a and 2b) and A-SVGD (Figs. 2c and 2d) on the synthetic Gaussian mixture with 16 components with $\sigma = 0.5$ across two set of initial particles.

a high density region of $p(x)$, and a repulsion that prevents the particles collapsing together. The specific derivation is as follows,

$$\hat{\phi}_k^{\star}(x) = \frac{1}{n}\sum_{j=1}^{n}\left[\underbrace{\kappa\left(x_j^{(k)}, x\right)\nabla \log p\left(x_j^{(k)}\right)}_{\text{driving force}} + \underbrace{\nabla_{x_j^{(k)}}\kappa\left(x_j^{(k)}, x\right)}_{\text{repulsion}}\right].$$

When mode collapse occurs, particles produced by SVGD are trapped at some region and fail to explore the full support of $p(x)$. This often happens when the driving force dominates the repulsion. An intuitive solution is to boost the repulsion or scale down the driving force to enable a wider spread of the particles. D'Angelo and Fortuin (2021) follows this intuition and modify the descent direction by introducing an annealing term $\alpha(k) \in [0, 1]$:

$$\hat{\phi}_{\alpha_k}(x) = \frac{1}{n}\sum_{j=1}^{n}\left[\alpha(k)\kappa\left(x_j^{(k)}, x\right)\nabla \log p\left(x_j^{(k)}\right) + \nabla_{x_j^{(k)}}\kappa\left(x_j^{(k)}, x\right)\right]. \tag{6}$$

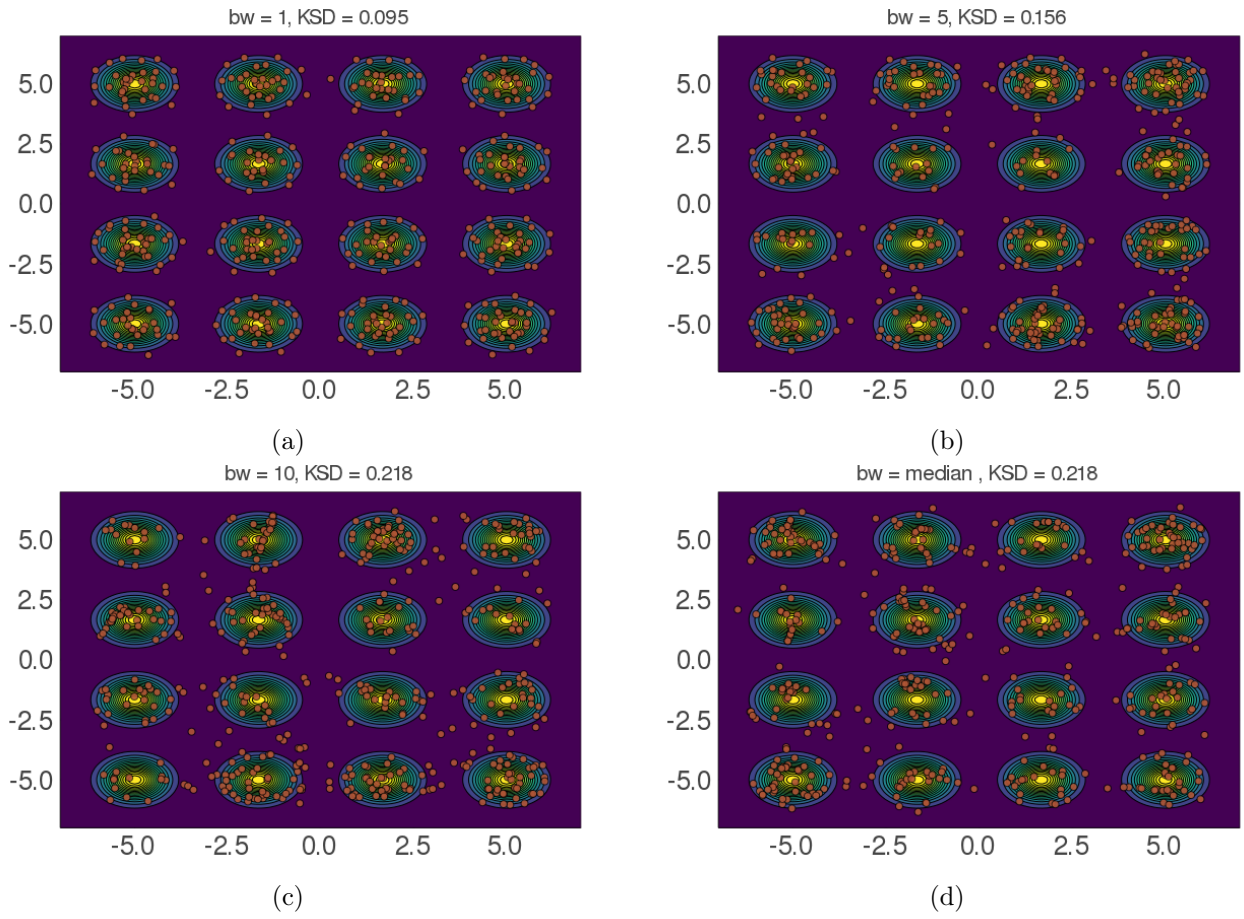


Figure 3: Comparison of A-SVGD on the synthetic Gaussian mixture with 16 components across different bandwidth. The KSDs are estimated using the same kernel.

In particular, the authors suggest using a cyclical annealing schedule, i.e.,

$$\alpha(k) = \left(\frac{\mathrm{mod}(k, K/C)}{K/C} \right)^p, \tag{7}$$

where K is the total number of iterations, C is the number of cycles, and $p > 0$ is an exponent factor that controls the speed of the transition between two phases. As demonstrated in Fig. 1, smaller p corresponds to a shorter period of the exploratory phase. For the two synthetic Gaussian mixture examples of this section, we set $p = 1$ and $C = 2$ for A-SVGD.

Now we compare the performance of A-SVGD and SVGD on a synthetic Gaussian mixture (Fig. 2). The target distribution is a 2D mixture of 16 Gaussian component with means equally distributed on a 4×4 grid with identical covariance $0.25I$. We consider two different initialization particles: Figs. 2a and 2c are initialized with 500 i.i.d. samples from $\mathcal{N}(0, 0.25I)$; Figs. 2b and 2d are initialized with 500 i.i.d. samples from $\mathcal{N}(10, 0.25I)$. In this example, we use RBF kernel with $h = 0.5$. Note that in both initialization schemes, A-SVGD is able to explore all the modes while all the particles produced by

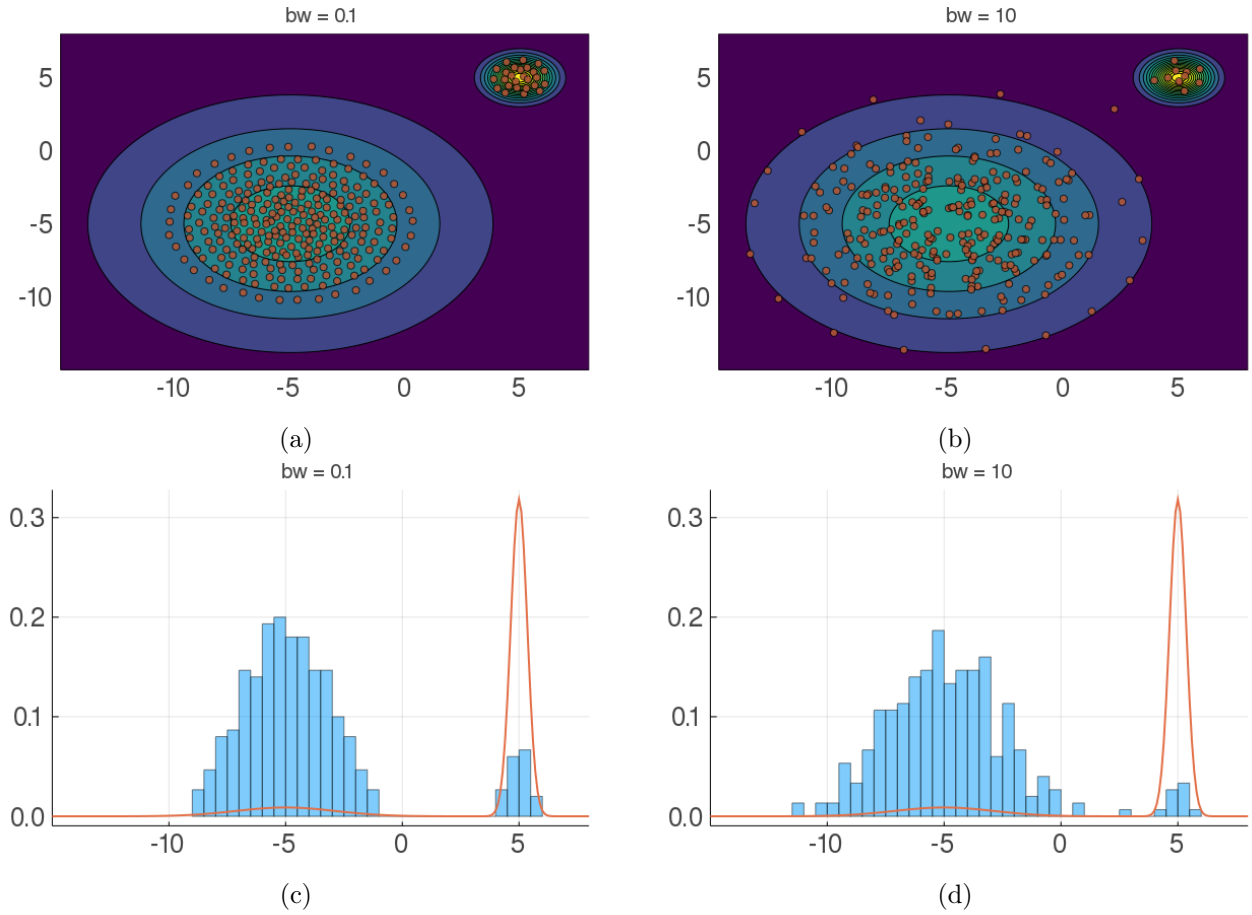


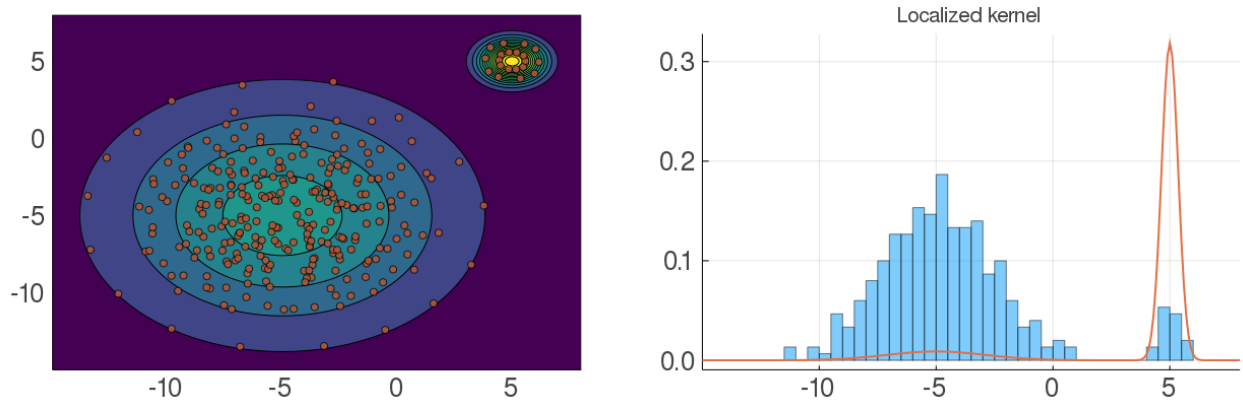
Figure 4: Comparison of A-SVGD on a 2-component Gaussian mixture with different bandwidth. Figs. 4a and 4b show scatter plots of particles at the last iteration. Figs. 4c and 4d show histograms of the particles by projecting the locations to the line $y = x$, where the orange curve denotes the density of Gaussian mixture along the slice $y = x$.

SVGD collapse at the components close by the initialization. Comparing Figs. 2c and 2d, one may notice that the performance of A-SVGD still depend on the location of the initial particles.

3.1 Bandwidth selection

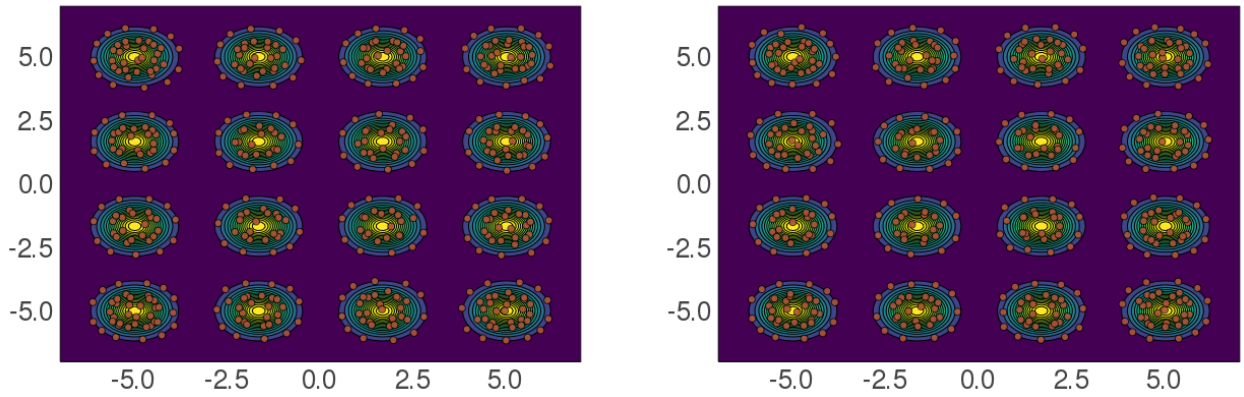
Although A-SVGD manages to mix well, its performance relies heavily on a good choice of kernel bandwidth. Generally, to optimize the performance of SVGD and its variants, one must tune the kernel bandwidth. The original paper (Liu and Wang, 2016) suggests using the median trick for the typical RBF kernel,

$$\kappa(x, x') = \exp(-\frac{1}{h}\|x - x'\|^2).$$



(a) (b)

Figure 5: A-SVGD with localized Gaussian kernel on a 2-component Gaussian mixture.



(a) $\{x_i^{(0)}\}_{i=1}^{500} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0\mathbb{1}, 0.25I)$ (b) $\{x_i^{(0)}\}_{i=1}^{500} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(10\mathbb{1}, 0.25I)$

Figure 6: A-SVGD with localized Gaussian kernel on a 4×4 Gaussian mixture.

The bandwidth is selected based on the median of the pairwise Euclidean distances between the current particles, i.e., $h = \text{med}^2 / \log n$, where med is the median of the Euclidean distances. This procedure will update the bandwidth automatically at each iteration and works generally well for unimodal target distribution. However, we will show that the median trick can be problematic when the local geometry of $\log p$ is complicated.

Fig. 3 present the results of A-SVGD on the same target across 4 different kernel bandwidth: $h = 1, 5, 10$ and the median heuristic (will be introduced in Section 4). The initial particles are 500 i.i.d. samples from $\mathcal{N}(0, 0.25I)$. We find that as h increases (not even a large amount), the convergence of A-SVGD gets ruined. Moreover, the median trick suggested by Liu and Wang (2016) completely fails. For quantitative characterization of the sample qualities, we compute the kernelized Stein discrepancy,which shows a decreasing sample quality as h increases. The key reason is that as the annealing schedule enables the points to traverse over different modes in the exploratory phase, the particles need to quickly converge to the high-density region around them in the converging phase.

As we mentioned previously, the driving force will push the samples along the $\nabla \log p(x)$. However, the driving force is a weighted summation of $\nabla \log p(x_i), i = 1, 2, \ldots, n$ and weights are controlled by the kernel. Adopting a large bandwidth tends to include misleading information from the particles that converge to other modes.

Even though the previous example prefers a smaller bandwidth, it does not mean that smaller h is always better. We then create a pathological yet simple example such that neither small nor large bandwidth is satisfying. Considering a two-component Gaussian mixture:

$$\frac{1}{2}\mathcal{N}\left(-5\mathbb{1}, 9I\right) + \frac{1}{2}\mathcal{N}\left(5\mathbb{1}, 0.25I\right).$$

Fig. 4 display the results of A-SVGD with 300 initial particles sampled from $\mathcal{N}(0\mathbb{1}, 25I)$ using different kernel bandwidth $h = 0.1, 10$. With small h, particles located in the wide Gaussian component are overly contracted; with large h, almost all particles converge to the wide mode. This is essentially due to the fact that the curvature of the local basin of two gaussian components is quite different. When using a wide kernel bandwidth, the driving force of those particles around the narrow Gaussian component is significantly influenced by gradient information $\log p(x)$ of the particles that are far away. As most particles are trapped in the wider Gaussian component, the driving force tends to point to the wide mode. Also, a large bandwidth leads to a stronger repulsion, which encourages the spread of particles. On the contrary, with small bandwidth—as shown in Figs. 4a and 4c—although the peaky Gaussian absorbs more particles, a weaker repulsion makes the particles collapsing together and underestimate the variance of the wide Gaussian. This hints at an important fact that using a universal kernel bandwidth for all particles is not ideal, and we should use different kernel bandwidths when updating each of the particles depending on its local basin of $\nabla \log p(x)$. In the next section, we propose a localized Gaussian kernel scheme that adaptively learns an anisotropic Gaussian kernel for each particle.

4. Localized Gaussian kernel

With the intuition obtained in the last section, we suggest using different bandwidth values for each particle and updates the bandwidth adaptively across the iteration. The goal of this section is to design an adaptive procedure that learns a proper kernel bandwidth for each particle automatically. As illustrated in Fig. 4, a careful choice of kernel bandwidth is necessary when the target distribution has multiple modes and one might consider using different bandwidth values at a different location. Intuitively, a proper choice of the bandwidth should reflect the local geometry of $\log p(x)$—the curvature of $\log p(x)$—which includes information of the Hessian $\nabla^2 \log p(x)$.

To generalize this idea, we consider the anisotropic Gaussian kernel with covariance Σ—a positive definite matrix,

$$\kappa(x, x'; \Sigma) = \exp\left(-\frac{1}{2}(x - x')^T \Sigma^{-1}(x)(x - x')\right).$$

Note that this anisotropic Gaussian kernel is *localized* as its covariance Σ depends on x. Intuitively, Σ^{-1} serves as a scaling matrix to the distance between particles and induces a deformed geometry structure in the particle space, which directly affects the gradient averaging when moving each particle and the repulsion among particles. To align with our intuition and previous synthetic experiments in Section 3, we want the particles are more influenced by neighbouring particles when

the curvature is huge and more spread when the local curvature is small. Therefore propose to set Σ^{-1} to abs $\left(\text{diag}\left(\nabla^2 \log p(x)\right)^{-1}\right)$. We use the absolute value because the diagonal element of $\nabla^2 \log p(x)$ can be negative if p is not log-concave. The complete procedure of A-SVGD using adaptive local bandwidth is described in Algorithm 2.

With other settings fixed, we apply Algorithm 2 to the previous two synthetic examples. As demonstrated in Fig. 5, by using the local bandwidth, the samples in the wide component correctly characterize its variance and there is a certain amount of samples concentrate at the peak mode. This combines the advantages of both small h and large h. And for the 16-component Gaussian example, as shown in Fig. 6, A-SVGD using localized kernel effectively explores all the Gaussian component regardless of the initial particles; which even outperforms A-SVGD with hand-tuned bandwidth; and importantly this requires no user input of the kernel bandwidth. In the next section, we will further illustrate the advantages of incorporating localized kernel to SVGD and A-SVGD in Bayesian logistic regression model and Bayesian Gaussian mixture model (GMM).

5. Experiment

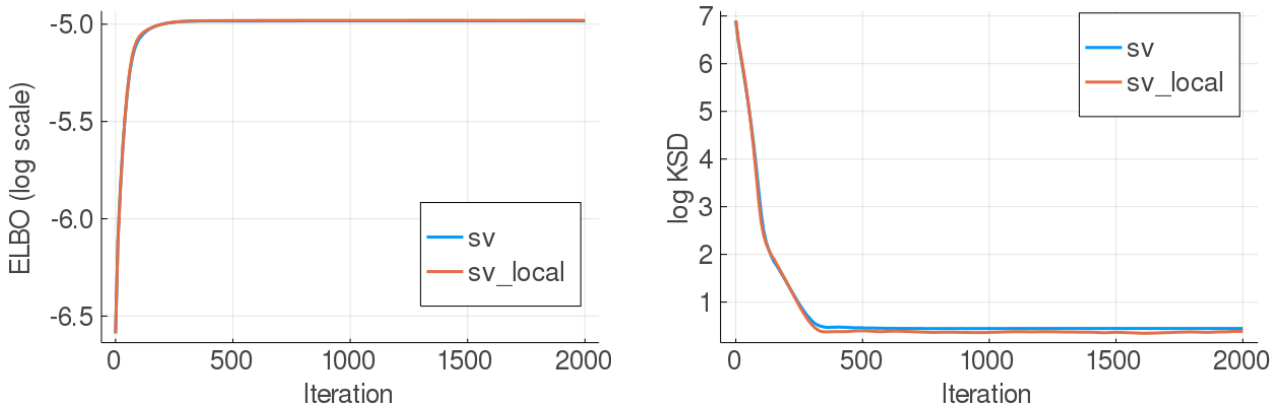
In this section, we measure the sample quality of SVGD and A-SVGD using both RBF kernel with median heuristic bandwidth and localized anisotropic Gaussian kernel on two common Bayesian models—the Bayesian logistic regression and the Bayesian Gaussian mixture model. The variants of the algorithm are denoted by SV, SV_local, ASV, and ASV_local, where "local" denotes the localized anisotropic Gaussian kernel. We run all algorithms with RMSprop updates described in Eq. (5). Also, in each example, algorithms share the same set of initial particles sampled from the prior distribution, and share same learning rates.

We focus on quantitative characterizations of sample quality for these algorithms. Choosing a reasonable metric that measures the performance of a sampling method is difficult. In fact, a major problem to the empirical results in D'Angelo and Fortuin (2021); Liu and Wang (2016) is the lack of insightful quantitative comparisons between different methods. Liu and Wang (2016) considers the test accuracy of the model with estimated parameters, while D'Angelo and Fortuin (2021) rely on visualizations of the particles positions; both methods are not straightforward and can be misleading in certain cases. In this section, we keep track of two quantities as algorithms run: *expected lower bound*(ELBO) (Blei et al., 2017)—which is equivalent to the negative KL divergence between the posterior and variational distribution up to a constant—and the KSD between the posterior and variational distribution. A larger ELBO corresponds to a better approximation, while a smaller KSD means that the empirical distribution of the particles are closer to the target distribution. As we have no access to the exact variational distribution, both the ELBO and KSD are estimated using current particles. And we choose the standard Gaussian kernel for KSD by default.

5.1 Bayeisan logistic regression

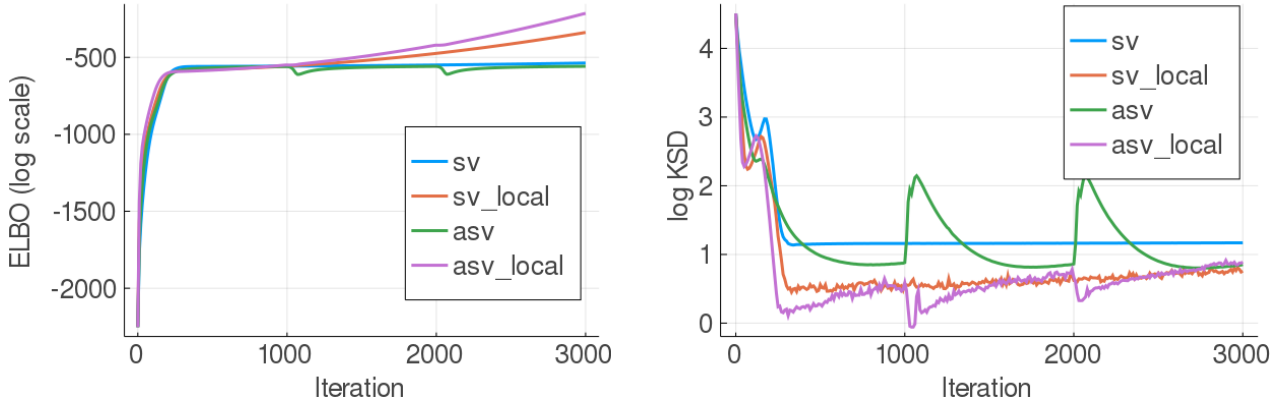
We first investigate the performance of localized Gaussian kernel on Bayesian logistic regresion for a library borrower dataset [1]. The goal is to infer the regression weights $w \in \mathbb{R}^d$ and a hyperparameter $\log \alpha \in \mathbb{R}$ in a binary classification problem. We are given a set of data points $(x_n, y_n)_{n=1}^N$ each

1. This dataset contains information on library borrowers, and is available at https://cran.r-project.org/web/packages/ISLR/index.html. We downsample the original dataset into 1000 data points.



(a) (b)

Figure 7: Results for Bayesian logistic regression



(a) (b)

Figure 8: Results for GMM

consisting a feature $x_n \in \mathbb{R}^d$ and a label $y_n \in \{0, 1\}$; the generative model is as follows:

$$\alpha \sim \mathsf{Gamma}(a, b), \quad w \mid \alpha \sim \mathcal{N}(0, \alpha^{-1} I)$$
$$y_n \mid x_n, w \overset{\text{indep}}{\sim} \text{Bern}\left(\frac{1}{1 + e^{-x_n^T w}}\right).$$

We adopt the same setting as Liu and Wang (2016) by picking the shape and rate parameters as $a = 1, b = 0.01$. Note that since the posterior distirbution on this problem is unimodal and this is no risk of mode collapse, we only compare the standard SVGD using median bandwidth (SV) and its counterpart with localized Gaussian kernel (SV_local). For both algorithms, we initialize 300 particles from the prior, and use a constant learning rate 0.01 for 2000 iterations. Fig. 7 presents the of this example. As this is a simple problem, these two methods give almost identical results in terms of the convergence speed and final sample qualities. Though the final particles given by SV_local seem to yield a slightly lower KSD to the posterior distribution, the difference is not significant enough.

5.2 Bayesian Gaussian mixture model

Next we compare the methods on a Bayesian Gaussian mixture model applied to a synthetic dataset. In the Bayesian Gaussian mixture model, we are given N observations $(x_n)_{n=1}^N \subseteq \mathbb{R}^d$, each following a k-component Gaussian mixture distribution:

$$x_n|\theta_{1:K}, \mu_{1:K}, \sigma_{1:K,1:D} \sim \sum_{k=1}^{K} \theta_k \mathcal{N}\left(\mu_k, \mathrm{diag}(\sigma_{k1}, \ldots, \sigma_{kD})\right).$$

The goal is to infer the posterior distribution of all the latent parameters $(\theta_{1:k}, \mu_{1:k}, \sigma_{1:K,1:D})$, on which we place a Dirichlet prior on the mixture proportions, a Gaussian prior on the Gaussian means, and a lognormal prior on the standard deviations, i.e.,

$$\begin{aligned}(\theta_1, \ldots, \theta_K) &\sim \mathsf{Dir}(\alpha_0) \\ \mu_k &\sim \mathcal{N}(0, I), \quad k = 1, 2, \ldots, K \\ \sigma_{kd} &\sim \mathsf{LogNormal}(0, 1), \quad k = 1, 2, \ldots, K, d = 1, 2, \ldots D.\end{aligned}$$

One may notice that the support of θ_k and σ_{kd} is constrained; the presented SVGD and A-SVGD with Gaussian kernel are limited to target distribution with full support. Thus, we first apply transformations such that the transformed variables have unconstrained support. The details of the transformation can be found in Section B.1.

The synthetic dataset contains $N = 400$ datapoints, equally generated from 4 isotropic bivariate Gaussian distributions with equal covariance $0.6^2 I$. The mean of those Gaussian distributions are randomely generated within $[-5, 5]^2$. We fit this dataset with $K = 2$; this misspecified setting creates multiple meaningful modes on posterior distribution, which makes standard SVGD/A-SVGD harder to mix. For the inference, we use 300 initial particles, and set a constant learning rate 0.02 over 3000 iterations. For A-SVGD, we use the cyclical annealing schedule Eq. (7) with $p = 1$ and $C = 3$.

Unlike the last example, the posterior distribution of GMM is complicated enough to differentiate these methods. As demonstrated in Fig. 8, using localized Gaussian kernel notably improves the convergence speed—both KSD and KL divergence drops faster. And is able to produce better approximation to the posterior, as shown in the higher ELBO value and smaller KSD estimated from the final particles. It is also surprising that SV_local outperforms A-SVGD with median heuristic bandwidth, and shows similar performance to ASV_local.

This result also reveals a potential problem with A-SVGD using cyclical annealing schedule. One may notice that at 1000-th and 2000-th iteration, when the annealing function starts a new cycle, A-SVGD using median heuristic bandwidth displays significant drop in sample quality. This is essentially because the particles leaves their local basin due to low driving force and spread aggressively. However, we do not observe such phenomenon when employ localized Gaussian kernel to A-SVGD; there is still oscillation of the KSD in the exploratory phase, but it does not change the sample quality drastically. One thing that is not clear from Fig. 8b is that ASV_local shows opposite oscillation trend to ASV at the beginning of each cycle; the KSD of ASV_local even drops. This may indicate that the set of particles may approximate the posterior better if it is slightly more spread. We might want to introduce a dimension-dependent scaling factor to the kernel covariance Σ; but this idea has not been experimented yet due to time constraint.

6. Discussion

In this report, we provide a comprehensive summarization of (annealed) Stein variational gradient descent. We also propose a novel kernel learning procedure, by allowing using a localized anisotropic Gaussian kernel for each particles. The covariance of the kernel function include the local curvature of $\log p(x)$ and improves the performance of (A-)SVGD when the target distribution has several modes. Multiple experiments show that compared with (A-)SVGD with universal RBF kernel, the localized kernel strategy leads to faster convergence and better sample qualities.

However, it is worth pointing out that leveraging the Hessian of $\log p(x)$ introduces additional computation complexity. Evaluating the Hessian for all particles costs $\mathcal{O}(d^2N)$, denoting a quadratic growth as d increases. This makes SVGD less practical in a high dimensional setting. In that case, we might consider using Gauss-Newton approximation to the Hessian matrix or other methods that approximate the second-order derivative using first-order information. Also, when n is large, evaluating $\nabla^2 \log p(x)$ for all the particles is impossible. But employing stochastic estimates of the exact Hessian matrix will a reasonable approach.

A more interesting problem in this line of work is to think wether there is a notion of optimal bandwidth or covariance matrix for Gaussian kernel. It could be fruitful by studying the influence of the kernel function from a theoretical perspective. Liu et al. (2019) uses a heat equation to characterize the distribution path produced by SVGD and propose to optimize the bandwidth for RBF kernel based on this equation. Detommaso et al. (2018) attempts to learn the kernel covariance based on second-order information of the functional derivative. However, their discussion is not rigorously justified.

References

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

D'Angelo, F. and Fortuin, V. (2021). Annealed Stein variational gradient descent. *arXiv preprint arXiv:2101.09815*.

Detommaso, G., Cui, T., Spantini, A., Marzouk, Y., and Scheichl, R. (2018). A stein variational newton method. *arXiv preprint arXiv:1806.03085*.

Kobyzev, I., Prince, S., and Brubaker, M. (2020). Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, C., Zhuo, J., Cheng, P., Zhang, R., and Zhu, J. (2019). Understanding and accelerating particle-based variational inference. In *International Conference on Machine Learning*, pages 4082–4092.

Liu, Q., Lee, J., and Jordan, M. (2016). A kernelized stein discrepancy for goodness-of-fit tests. In *International Conference on Machine Learning*, pages 276–284.

Liu, Q. and Wang, D. (2016). Stein variational gradient descent as gradient flow. In *Advances in Neural Information Processing Systems*, pages 2370–2378.

Zhuo, J., Liu, C., Shi, J., Zhu, J., Chen, N., and Zhang, B. (2018). Message passing Stein variational gradient descent. In *International Conference on Machine Learning*, pages 6018–6027.

Appendix A. Algorithms

Algorithm 1 SVGD

procedure SVGD($\{x_i^{(0)}\}_{i=1}^n$, $\nabla \log p(x)$, $\kappa(\cdot,\cdot)$,γ_k)
  for $k = 0, 1, \ldots, K-1$ **do**
    for $i = 1, 2, \ldots, n$ **do**
      $x_i^{(k+1)} \leftarrow x_i^{(k)} + \gamma_k \hat{\phi}_k^\star \left(x_i^{(k)}\right)$ where $\hat{\phi}_k^\star(\cdot)$ is defined in Eq. (4)
    end for
  end for
  return $\left\{x_i^{(K)}\right\}_{i=1}^n$
end procedure

Algorithm 2 A-SVGD with local bandwidth

procedure SVGD($\{x_i^{(0)}\}_{i=1}^n$, $\nabla \log p(x)$, $\nabla^2 \log p(x)$,γ_k, $\alpha(t)$)
  for $k = 0, 1, \ldots, K-1$ **do**
    for $i = 1, 2, \ldots, n$ **do**
      $\Sigma(x_i) \leftarrow \text{abs}\left(\text{diag}\left(\nabla^2 \log p(x_i)\right)^{-1}\right)$
      $\kappa(x, x'; \Sigma(x_i)) \leftarrow \exp\left(-\frac{1}{2}(x-x')^T \Sigma^{-1}(x_i)(x-x')\right)$
      $x_i^{(k+1)} \leftarrow x_i^{(k)} + \gamma_k \hat{\phi}_{\alpha_k} \left(x_i^{(k)}\right)$ where $\hat{\phi}_{\alpha_k}(\cdot)$ is defined in Eq. (6)
    end for
  end for
  return $\left\{x_i^{(K)}\right\}_{i=1}^n$
end procedure

Appendix B. Details of Experiments

B.1 Variable transformations of Bayesian GMM

To transform $(\theta_1, \ldots, \theta_K) \in \Delta(K)$ and $\sigma_{kd} \in \mathbb{R}_+$ into unconstrained space, we consider the change of random variables as below:

1. For $\sigma_{kd} \sim \mathsf{LogNormal}(0, 1)$, we consider
$$\tau_{kd} = \log(\sigma_{kd}) \sim \mathcal{N}(0, 1),$$
which has a full support on $\mathbb{R}$.

2. For $\theta \sim \mathsf{Dir}(\alpha_0)$, we consider using marginalized LogGamma random variables. Notice the relationship of Gamma distribution and Dirichlet distirbution as follows,
$$(\lambda_k)_{k=1}^K \overset{\text{i.i.d.}}{\sim} \mathsf{LogGamma}(\alpha_0, 1)$$
$$\left(\frac{\exp(\lambda_1)}{\sum_k \exp(\lambda_k)}, \ldots, \frac{\exp(\lambda_K)}{\sum_k \exp(\lambda_k)}\right) \sim \mathsf{Dir}(\alpha_0),$$
then λ_k is supported on $\mathbb{R}$. In our example, we set $\alpha_0 = 1$.

Therefore, instead of inferring the original parameters, we perform SVGD on the posterior distribution of $(\lambda_{1:k}, \mu_{1:k}, \tau_{11:kd})$.