A KSD Variational Inference

Kernelized Stein discrepancy (KSD) provides a discrepancy measure between distributions and can be in principle used as a variational objective function in replace of KL divergence. In fact, thanks to the special form of KSD ((11)-(12)), one can derive a standard stochastic gradient descent for minimizing KSD without needing to estimate $q_\eta(z)$ explicitly, which provides a conceptually simple wild variational inference algorithm. Although this work mainly focuses on amortized SVGD which we find to be easier to implement and tend to perform superior to KSD variational inference in practice (see Figure 6), we think the KSD approach is of theoretical interest and hence give a brief discussion here.

Specifically, take q_η to be the density of the random output $z = f(\xi;\ \eta)$ when $\xi \sim q_0$, and we want to find η to minimize $\mathbb{D}(q_\eta \,||\, p)$. Assuming $\{\xi_i\}$ is i.i.d. drawn from q_0, we can approximate $\mathbb{D}^2(q_\eta \,||\, p)$ unbiasedly using the U-statistics in (12), and derive a standard gradient descent

$$\eta \leftarrow \eta - \epsilon \frac{2}{n(n-1)} \sum_{i \neq j} \partial_\eta f(\xi_i;\ \eta)\, \nabla_{z_i} \kappa_p(z_i, z_j), \quad (24)$$

where $z_i = f(\xi_i;\ \eta)$. This enables a wild variational inference method based on directly minimizing η with standard (stochastic) gradient descent. We call this algorithm *amortized KSD*. Note that (24) is similar to (15) in form, but replaces $\boldsymbol{\phi}^*(z_i)$ with

$$\bar{\boldsymbol{\phi}}^*(z_i) \stackrel{def}{=} -2 \sum_{j:\ i \neq j} \nabla_{z_i} \kappa_p(z_i, z_j)/(n(n-1)).$$

Here $\bar{\boldsymbol{\phi}}^*$ depends on the second order derivative of $\log p$ because $\kappa_p(z, z')$ depends on $\nabla \log p$, which makes it more difficult to implement amortized KSD than amortized SVGD.

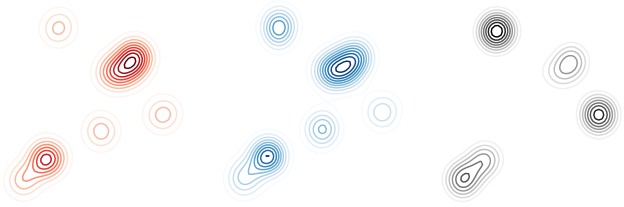
Intuitively, minimizing KSD can be viewed as seeking a stationary point of KL divergence under SVGD updates. To see this, recall that $q_{[\epsilon\boldsymbol{\phi}]}$ denotes the density of $z' = z + \epsilon\boldsymbol{\phi}(z)$ when $z \sim q$. From (4), we have for small ϵ,

$$\mathbb{D}^2(q \,||\, p) \approx \frac{1}{\epsilon} \max_{\boldsymbol{\phi} \in \mathcal{F}} \big\{ \mathrm{KL}(q \,||\, p) - \mathrm{KL}(q_{[\epsilon\boldsymbol{\phi}]} \,||\, p) \big\}.$$

That is, KSD measures the maximum degree of decrease in the KL divergence when we update the particles along the optimal SVGD perturbation direction $\boldsymbol{\phi}^*$. If $q = p$, then the decrease of the KL divergence equals zero and $\mathbb{D}^2(q \,||\, p)$ equals zero. In fact, KSD can be explicitly represented as the magnitude of the functional gradient of the KL divergence w.r.t. $\boldsymbol{\phi}$ in RKHS (Liu & Wang, 2016),

$$\mathbb{D}(q \,||\, p) = \Big|\Big| \nabla_{\boldsymbol{\phi}} F(0) \Big|\Big|_{\mathcal{H}^d}, \qquad F(\boldsymbol{\phi}) \stackrel{def}{=} \mathrm{KL}(q_{[\epsilon\boldsymbol{\phi}]} \,||\, p),$$

where $\nabla_{\boldsymbol{\phi}} F(\boldsymbol{\phi})$ denotes the functional gradient of the function $F(\boldsymbol{\phi})$ w.r.t. $\boldsymbol{\phi}$ defined in RKHS $\mathcal{H}^d$, and $\nabla_{\boldsymbol{\phi}} F(\boldsymbol{\phi})$ is also an element in $\mathcal{H}^d$. Therefore, in contrast to amortized SVGD which attends to minimize the KL objective $F(\boldsymbol{\phi})$, KSD variational inference minimizes the gradient magnitude $||\nabla_{\boldsymbol{\phi}} F(0)||_{\mathcal{H}^d}$ of KL divergence.



(a) Test Distribution (truth) (b) Amortized SVGD (15 steps) (c) Amortized KSD (15 steps)

Figure 6: Learning to sample from GMM. The Langevin sampler with step size trained by amortized SVGD (b) obtains close approximation with $T = 15$ is close to the true test distribution (a) while amortized KSD (c) which we use equation (24) to perform does not work as well as amortized SVGD.

This idea is closely related to the operator variational inference (Ranganath et al., 2016), which directly minimizes the variational form of Stein discrepancy in (4) and (8) with $\mathcal{F}$ replaced by sets of parametric neural networks. Specifically, Ranganath et al. (2016) assumes $\mathcal{F}$ consists of a neural network $\boldsymbol{\phi}_\tau(z)$ with parameter τ, and find τ jointly with η by solving a min-max game:

$$\min_\eta \max_\tau \mathbb{E}_{z \sim q_\eta}[\mathcal{T}_p \boldsymbol{\phi}_\tau(z)].$$

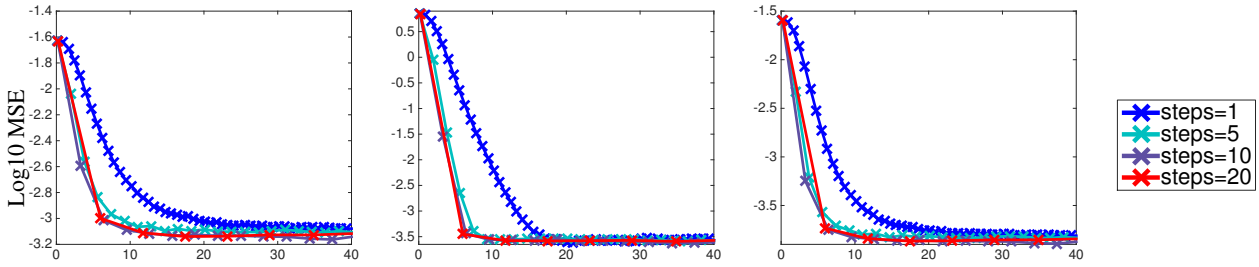
This yields a more challenging computation problem, although it is possible that the neural networks provide stronger discrimination than RKHS in practice. The main advantage of the KSD based approach is that it leverages the closed form solution in RKHS, yields a simpler optimization formulation based on standard gradient descent.

Figure 6 shows results of Langevin samplers trained by amortized SVGD and amortized KSD, respectively, for learning simple Gaussian mixtures under the same setting as that in Section 5.1. We find that amortized KSD tends to perform worse (Figure 6(c)) than amortized SVGD (Figure 6(b)); given that it is also less straightforward to implement amortized KSD (for requiring calculating $\nabla_z \kappa_p(z, z')$ in (24)), we did not test it in our other experiments.

B Solving the Projection Step Using Different Numbers Gradient Steps

Amortized SVGD requires us to solve the projection step using either (13) or (14) at each iteration. In practice, we approximately solve it using only one step of gradient descent starting from the old values of η for the sake of computational efficiency.

In order to study the trade-off of accuracy and computational cost here, we plot in Figure 7 the results when we



(a) $\mathbb{E}(x)$ (b) $\mathbb{E}(x^2)$ (c) $\mathbb{E}(\cos(wx+b))$

Figure 7: Results when using different numbers of gradient decent steps for solving (14). The setting is the same as that in Figure 2, but we conduct experiments using 1, 5, 10, 20 gradient steps when solving (14), and show their corresponding training time in the x-axis, and their mean square error for estimating $\mathbb{E}_p h$ (for the "testing" distributions) in the y-axis. The Langevin samplers we used have $T = 10$ layers (Langevin update steps). The results are evaluated by drawing 1,000 samples from the trained samplers at different iterations of SVGD. The dimension of the Gaussian Mixtures is $d = 50$.

VAE-CNN EVAE-CNN ESteinVAE-CNN

Figure 8: Images generated by VAE-CNN, EVAE-CNN, and ESteinVAE-CNN

solve Eq (14) using different numbers of gradient descent steps (the result is almost identical when we solve Eq (13) instead). We can see that when using more gradient steps, although the training time per iteration increases, the overall convergence speed may still improve, because it may take less iterations to converge. Figure 7 seems to suggest that using 5, 10, 20 steps gives better convergence than using a single step, but this may vary in different cases. We suggest to search for the best gradient step if the convergence speed is a primary concern. On the other hand, the number of gradient steps seems to have minor influence on the final result at the convergence as shown in Figure 7.

C Images Generated by Different VAEs

Figure 8 shows the images generated by the standard VAE-CNN, the entropy regularized VAE-CNN and ESteinVAE-CNN. We can see that both EVAE-CNN and ESteinVAE-CNN can generate images of good quality.