Stein Variational Importance Sampling

Sky

1 RBF Kernel

According to section 5 of the paper, the authors use the RBF kernel defined as $k(x, x') = \exp(-||x - x'||^2/h)$ with h being the kernel bandwidth. h is defined as $\text{med}^2/(2\log(|A|+1))$ where med is the median of the pairwise distances between the leader particles and $|A|$ is the number of leader particles. (Note: The authors took the median of the squared pairwise distances between the leader particles.)

1.1 Construct mapping for leaders

We first construct the map using the leader particles x_A^ℓ with ℓ representing the ℓth iteration. The pairwise distances $||x - x'||$ are already computed when defining the kernel.

Compute the kernel by expanding the numerator that is exponentiated:

$$||x_A - x'_A||^2 = x_A^T x_A - 2x_A^T x'_A + x_A'^T x'_A \tag{1}$$

Find the median pairwise distance by taking the square root of $||x_A - x'_A||^2$ and checking if it is odd or even. If it is odd, pick the median else take the mean of the two middle values. Square the median and divide by $2\log(|A|+1)$. With the above, we have the kernel of the leader particles.

Assume that we have $\nabla \log p(x_A^\ell)$ obtained using automatic differentiation. Automatic differentiation in Tensorflow is implemented such that it accumulates the variable we are differentiating by. Since the operation $\nabla \log p(x_A^\ell)$ only accumulates each particle once, it is safe to employ automatic differentiation here.

However, computing the Jacobian of the kernel $\nabla_{x_A} k(x_A, x'_A)$ leads to accumulation of each particle $|A|$ times so we have to do it by hand:

$$\nabla_{x_A} k(x_A, x'_A) = \nabla_{x_A} \exp(-(x_A^T x_A - 2x_A^T x'_A + x_A'^T x'_A)/h) \text{ with } h \text{ being constant} \tag{2}$$

$$= -\frac{2(x_A - x'_A)}{h} \exp(-||x_A - x'_A||^2/h) \tag{3}$$

Since the update $\phi_A^{\ell+1}$ consist of two terms being added together and we are performing a sum over them, the operations are commutative so we can split it up into adding the sum of one term to the sum of the other. Hence $\phi_A^{\ell+1}(\cdot)$ is equivalent to $\frac{1}{|A|}\{k(x_A, \cdot)^T \nabla \log p(x_A) + \sum_j -\frac{2(x_{A_j} - \cdot)}{h} \exp(-||x_{A_j} - \cdot||^2/h)\}$.

1.2 Construct mapping for followers

We do the same for the followers:

- $||x_A - x_B||^2 = x_A^T x_A - 2x_A^T x_B + x_B^T x_B$
- $\nabla_{x_A} k(x_A, x_B) = -\frac{2(x_A - x_B)}{h} \exp(-||x_A - x_B||^2/h)$
- $\phi_B^{\ell+1}(\cdot) = \frac{1}{|A|}\{k(x_A, \cdot)^T \nabla \log p(x_A) + \sum_j -\frac{2(x_{A_j} - \cdot)}{h} \exp(-||x_{A_j} - \cdot||^2/h)\}$

1.3 Update leaders and followers

Update ϕ_A and ϕ_B to both leader and follower particle by adding $\epsilon * \phi$ to them.

1.4 Calculate density values of followers

We now compute $\nabla_{x_{B_i}} \phi_B(x_{B_i}) = \frac{1}{A} \sum_A [\nabla_{x_A} \log p(x_{B_i})^T \nabla_{x_{B_i}} k(x_A, x_{B_i}) + \nabla_{x_A} \nabla_{x_{B_i}} k(x_A, x_{B_i})]$. Like before but with a slight tweak, $\nabla_{x_{B_i}} k(x_A, x_{B_i})$ is:

$$\nabla_{x_{B_i}} \exp(-(x_A^T x_A - 2x_A^T x_{B_i} + x_{B_i}^T x_{B_i})/h) = \frac{2(x_A - x_{B_i})}{h} \exp(-||x_A - x_{B_i}||^2/h) \tag{4}$$

Shifting our focus to the $\nabla_{x_A} \nabla_{x_{B_i}} k(x_A, x_{B_i})$ term, we begin with the above result[1]We

$$\begin{aligned}
&\nabla_{x_A} \nabla_{x_{B_i}} k(x_A, x_{B_i}) && (5)\\
&= \nabla_{x_A} \frac{2(x_A - x_{B_i})}{h} \exp(-||x_A - x_{B_i}||^2/h) && (6)\\
&= \frac{2}{h}[\nabla_{x_A} x_A \exp(-||x_A - x_{B_i}||^2/h) + \nabla_{x_A} x_{B_i} \exp(-||x_A - x_{B_i}||^2/h)] && (7)\\
&= \frac{2}{h}[I - \frac{2(x_A - x_{B_i})}{h} x_A^T - \frac{2(x_A - x_{B_i})}{h} x_{B_i}^T] \exp(-||x_A - x_{B_i}||^2/h) && (8)\\
&= \frac{2}{h}[I - \frac{2}{h}(x_A - x_{B_i})(x_A - x_{B_i})^T] \exp(-||x_A - x_{B_i}||^2/h) && (9)
\end{aligned}$$

[1] `http://mlg.eng.cam.ac.uk/mchutchon/DifferentiatingGPs.pdf`

Hence, we have

$$\nabla_{x_{B_i}}\phi_B(x_{B_i}) = \frac{1}{|A|}\sum_{j\in A}[\nabla_{x_{A_j}}\log p(x_{A_j})^T\frac{2(x_{A_j}-x_{B_i})}{h}\exp(-||x_{A_j}-x_{B_i}||^2/h)) + \frac{2}{h}(I-\frac{2}{h}(x_{A_j}-x_{B_i})(x_{A_j}-x_{B_i})^T)\exp(-||x_{A_j}-x_{B_i}||^2/h)] \tag{10}$$

$$= \frac{1}{|A|}[\nabla_{x_A}\log p(x_A)^T\cdot\mathrm{diag}(\exp(-||x_A-x_{B_i}||^2/h))\cdot\frac{2(x_A-x_{B_i})}{h} + \left(\frac{2}{h}\cdot\sum_{j\in A}I\cdot\exp(-||x_{A_j}-x_{B_i}||^2/h)\right) - \frac{2(x_A-x_{B_i})}{h}^T\cdot\mathrm{diag}(\exp(-||x_A-x_{B_i}||^2/h))\cdot\frac{2(x_A-x_{B_i})}{h}] \tag{11}$$

1.5 Update density values of followers

With $\nabla_{x_{B_i}}\phi_B(x_{B_i})$ in hand, we update the density values using the last equation of the Stein IS algorithm by using Tensorflow to compute the absolute determinant of $I+\epsilon\nabla_{x_{B_i}}\phi_B(x_{B_i})$ and multiplying the current density values by the inverse of the result.

2 Fisher Kernel

Instead of the RBF kernel whose weaknesses are detailed in Zhuo et. al[2], use the Fisher kernel which requires less parameter tuning (and could possibly sidestep the issues highlighted in Zhuo et. al) and might have better properties. The Fisher kernel is formally defined as $k(x, x') = \nabla_\theta \log(p(x|\theta) I^{-1} \nabla_\theta \log(p(x'|\theta)^T$ where I is the Fisher matrix. In practice, the Fisher matrix is not computed to reduce computational cost, leading to the requirement that resulting kernel be normalised[3] and made PSD as SVGD exploits the RHKS property for closed-form updates. We follow the format defined by the previous section.

2.1 Construct mapping for leaders

Like before, we can assume that $\nabla_x \log p(x_A^\ell|\theta)$ is obtained through automatic differentiation. However the issue this time is TensorFlow being unable to differentiate through a mixture model which requires analytical expressions for each parameter for the Gaussian Mixture Model[4] and $\nabla_x \nabla_\theta \log p(x_A^\ell|\theta)$ for $\nabla k(x, x')$.

Differentials of GMM loglikelihood
Take $p(x|\theta) = \sum_m w_i p(x|\theta_i)$ where $\theta = \{w, \mu, \sigma^2\}$ with m Gaussians.
Correspondingly, $\log p(x|\theta) = \log \sum_m w_i p(x|\theta_i)$.
Differentiating w.r.t. w_i,

$$\nabla_{w_i} \log p(x|\theta) = \frac{1}{\sum_m w_i p(x|\theta_i)} \nabla_{w_i} \sum_m w_i p(x|\theta_i) \tag{12}$$

$$= \frac{w_i p(x|\theta_i)}{\sum_m w_i p(x|\theta_i)} \frac{1}{w_i} \tag{13}$$

$$\tag{14}$$

Differentiating w.r.t. μ_i,

$$\nabla_{\mu_i} \log p(x|\theta) = \frac{1}{\sum_m w_i p(x|\theta_i)} \nabla_{\mu_i} \sum_m w_i p(x|\theta_i) \tag{15}$$

$$= \frac{w_i p(x|\theta_i)}{\sum_m w_i p(x|\theta_i)} \left(\frac{x - \mu_i}{\sigma_i^2} \right) \tag{16}$$

$$\tag{17}$$

[2] https://arxiv.org/pdf/1711.04425.pdf
[3] http://qwone.com/~jason/writing/normalizeKernel.pdf
[4] Checked against Matrix Cookbook

Differentiating w.r.t. σ_i^2 [5][6],

$$\nabla_{\sigma_i^2} \log p(x|\theta) = \frac{1}{\sum_m w_i p(x|\theta_i)} \nabla_{\sigma_i^2} \sum_m w_i p(x|\theta_i) \tag{18}$$

$$= \frac{w_i p(x|\theta_i)}{\sum_m w_i p(x|\theta_i)} \underbrace{\frac{1}{2}\left(-\frac{1}{\sigma_i^2} + \left(\frac{x-\mu_i}{\sigma_i^2}\right)^2\right)}_{\xi} \tag{19}$$

Second Order Differentials of GMM loglikelihood
To ensure that the second order differentials are right, we start by differentiating an entry in the Fisher kernel and then generalise from the obtained results.

Differentiating $k_{w_i}(x_1, x_2)$ w.r.t. x,

$$\nabla_{x_1} k_{w_i}(x_1, x_2) = \nabla_{x_1} \frac{w_i^2 p(x_1|\theta_i) p(x_2|\theta_i)}{\sum w_i p(x_1|\theta_i) \sum w_i p(x_2|\theta_i)} \tag{20}$$

$$= \frac{\sum \left(\frac{x_1-\mu_i}{\sigma_i^2}\right) w_i p(x_1|\theta_i)}{(\sum w_i p(x_1|\theta_i))^2} \frac{w_i^2 p(x_1|\theta_i) p(x_2|\theta_i)}{\sum w_i p(x_2|\theta_i)} \tag{21}$$

$$- \frac{w_i^2 p(x_2|\theta_i)}{\sum w_i p(x_1|\theta_i) \sum w_i p(x_2|\theta_i)} \left(\frac{x_1-\mu_i}{\sigma_i^2}\right) p(x_1|\theta_i) \tag{22}$$

We then separate the contribution of $\nabla_{w_i} \log \sum w_i p(x_2|\theta_i)$ from the above to get $\nabla_{x_1} \nabla_{w_i} \log \sum w_i p(x_1|\theta_i)^T$

$$\nabla_{x_1} \nabla_{w_i} \log \sum w_i p(x_1|\theta_i)^T$$

$$= \frac{w_i p(x_1|\theta_i)}{\sum w_i p(x_1|\theta_i)} \left[\frac{\sum \left(\frac{x_1-\mu_i}{\sigma_i^2}\right) w_i p(x_1|\theta_i)}{\sum w_i p(x_1|\theta_i)} - \left(\frac{x_1-\mu_i}{\sigma_i^2}\right) \right] \tag{23}$$

Its transpose is

$$\nabla_{x_1} \nabla_{w_i} \log \sum w_i p(x_1|\theta_i)$$

$$= \frac{w_i p(x_1|\theta_i)}{\sum w_i p(x_1|\theta_i)} \left[\frac{\sum \left(\frac{x_1-\mu_i}{\sigma_i^2}\right) w_i p(x_1|\theta_i)}{\sum w_i p(x_1|\theta_i)} - \left(\frac{x_1-\mu_i}{\sigma_i^2}\right) \right]^T \tag{24}$$

[5] http://www.vlfeat.org/api/fisher-derivation.html
[6] https://hal.inria.fr/hal-00830491/PDF/journal.pdf

Differentiating $k_{\mu_i}(x_1, x_2)$ w.r.t. x,

$$\nabla_{x_1} k_{\mu_i}(x_1, x_2) = \nabla_{x_1} \frac{w_i^2 p(x_1|\theta_i) p(x_2|\theta_i)}{\sum w_i p(x_1|\theta_i) \sum w_i p(x_2|\theta_i)} \left(\frac{x_1 - \mu_i}{\sigma_i^2}\right)^T \left(\frac{x_2 - \mu_i}{\sigma_i^2}\right) \quad (25)$$

$$= \frac{\sum \left(\frac{x_1 - \mu_i}{\sigma_i^2}\right) w_i p(x_1|\theta_i)}{(\sum w_i p(x_1|\theta_i))^2} \frac{w_i^2 p(x_1|\theta_i) p(x_2|\theta_i)}{\sum w_i p(x_2|\theta_i)} \left(\frac{x_1 - \mu_i}{\sigma_i^2}\right)^T \left(\frac{x_2 - \mu_i}{\sigma_i^2}\right) \quad (26)$$

$$- \frac{w_i^2 p(x_2|\theta_i)}{\sum w_i p(x_1|\theta_i) \sum w_i p(x_2|\theta_i)} \left(\frac{x_1 - \mu_i}{\sigma_i^2}\right) p(x_1|\theta_i) \left(\frac{x_1 - \mu_i}{\sigma_i^2}\right)^T \left(\frac{x_2 - \mu_i}{\sigma_i^2}\right) \quad (27)$$

$$+ \frac{w_i^2 p(x_1|\theta_i) p(x_2|\theta_i)}{\sum w_i p(x_1|\theta_i) \sum w_i p(x_2|\theta_i)} \frac{1}{\sigma_i^2} \left(\frac{x_2 - \mu_i}{\sigma_i^2}\right) \quad (28)$$

We then separate the contribution of $\nabla_{\mu_i} \log \sum w_i p(x_2|\theta_i)$ from the above to get $\nabla_{x_1} \nabla_{\mu_i} \log \sum w_i p(x_1|\theta_i)^T$

$$\nabla_{x_1} \nabla_{\mu_i} \log \sum w_i p(x_1|\theta_i)^T$$

$$= \frac{w_i p(x_1|\theta_i)}{\sum w_i p(x_1|\theta_i)} \left[\left(\frac{\sum \left(\frac{x_1 - \mu_i}{\sigma_i^2}\right) w_i p(x_1|\theta_i)}{\sum w_i p(x_1|\theta_i)} - \left(\frac{x_1 - \mu_i}{\sigma_i^2}\right) \right) \left(\frac{x_1 - \mu_i}{\sigma_i^2}\right)^T + \frac{1}{\sigma_i^2} \right] \quad (29)$$

where σ_i^2 is a $d \times d$ diagonal matrix instead of a $d \times 1$ vector. Its transpose is

$$\nabla_{x_1} \nabla_{\mu_i} \log \sum w_i p(x_1|\theta_i)$$

$$= \frac{w_i p(x_1|\theta_i)}{\sum w_i p(x_1|\theta_i)} \left[\left(\frac{x_1 - \mu_i}{\sigma_i^2}\right) \left(\frac{\sum \left(\frac{x_1 - \mu_i}{\sigma_i^2}\right) w_i p(x_1|\theta_i)}{\sum w_i p(x_1|\theta_i)} - \left(\frac{x_1 - \mu_i}{\sigma_i^2}\right) \right)^T + \frac{1}{\sigma_i^2} \right] \quad (30)$$

Differentiating $k_{\sigma_i^2}(x_1, x_2)$ w.r.t. x,

$$\nabla_{x_1} k_{\sigma_i^2}(x_1, x_2) = \nabla_{x_1} \frac{w_i^2 p(x_1|\theta_i) p(x_2|\theta_i)}{\sum w_i p(x_1|\theta_i) \sum w_i p(x_2|\theta_i)} \xi_1^T \xi_2 \quad (31)$$

$$= \frac{\sum \left(\frac{x_1 - \mu_i}{\sigma_i^2}\right) w_i p(x_1|\theta_i)}{(\sum w_i p(x_1|\theta_i))^2} \frac{w_i^2 p(x_1|\theta_i) p(x_2|\theta_i)}{\sum w_i p(x_2|\theta_i)} \xi_1^T \xi_2 \quad (32)$$

$$- \frac{w_i^2 p(x_2|\theta_i)}{\sum w_i p(x_1|\theta_i) \sum w_i p(x_2|\theta_i)} \left(\frac{x_1 - \mu_i}{\sigma_i^2}\right) p(x_1|\theta_i) \xi_1^T \xi_2 \quad (33)$$

$$+ \frac{w_i^2 p(x_1|\theta_i) p(x_2|\theta_i)}{\sum w_i p(x_1|\theta_i) \sum w_i p(x_2|\theta_i)} \left(\mathrm{diag} \left(\frac{x_1 - \mu_i}{\sigma_i^2}\right) \frac{1}{\sigma_i^2} \right)^T \xi_2 \quad (34)$$

We then separate the contribution of $\nabla_{\sigma_i^2} \log \sum w_i p(x_2|\theta_i)$ from the above to get $\nabla_{x_1} \nabla_{\sigma_i^2} \log \sum w_i p(x_1|\theta_i)^T$

$$\nabla_{x_1} \nabla_{\sigma_i^2} \log \sum w_i p(x_1|\theta_i)^T$$
$$= \frac{w_i p(x_1|\theta_i)}{\sum w_i p(x_1|\theta_i)} \left[\left(\frac{\sum \left(\frac{x_1 - \mu_i}{\sigma_i^2} \right) w_i p(x_1|\theta_i)}{\sum w_i p(x_1|\theta_i)} - \left(\frac{x_1 - \mu_i}{\sigma_i^2} \right) \right) \xi_1^T + \frac{1}{\sigma_i^2} \text{diag} \left(\frac{x_1 - \mu_i}{\sigma_i^2} \right) \right] \tag{35}$$

where σ_i^2 is a $d \times d$ diagonal matrix instead of a $d \times 1$ vector. Its transpose is

$$\nabla_{x_1} \nabla_{\sigma_i^2} \log \sum w_i p(x_1|\theta_i)$$
$$= \frac{w_i p(x_1|\theta_i)}{\sum w_i p(x_1|\theta_i)} \left[\xi_1 \left(\frac{\sum \left(\frac{x_1 - \mu_i}{\sigma_i^2} \right) w_i p(x_1|\theta_i)}{\sum w_i p(x_1|\theta_i)} - \left(\frac{x_1 - \mu_i}{\sigma_i^2} \right) \right)^T + \text{diag} \left(\frac{x_1 - \mu_i}{\sigma_i^2} \right) \frac{1}{\sigma_i^2} \right] \tag{36}$$

3 Performance

The following target distributions are used to test the performance of the RBF and Fisher kernel:

- 2D Gaussian mixture model with 10 mixture components
- 6D Gaussian (Suggested by Wittawat to check how well SteinIS performs in an easy case for 6D)
- 6D Guassian mixture model with mixture component

In all experiments, both leader and follower particles are sampled from $\mathcal{N}(0, 2)$ and then transformed with SVGD. The number of leader particles are fixed to be 100 across experiments. Performance is measured by the normalised MSE of the target distributions' normalising constant approximated by the transformed particles.

For the first target distribution, both the RBF and Fisher kernel do well as shown below. The MSE is obtained over 500 independent runs with a different random seed set for each run. This ensures that the samples are the same across experiments for the two kernels, which enables fair comparison. α and β, the factors determining the learning rate, is set at 0.001 and 0.5 respectively.

Moving on to the 6D Gaussian, the MSE degenerates to the same order of magnitude as the normalising constant i.e. 0.9999 and 0.9812 respectively. The results are averaged over 50 independent runs with the random seed set done previously with α and β set at $5e-5$ and 0.5.