Lecture 16, Apr 9, 2024

Stochastic Variational Inference (SVI)

- *Stochastic variational inference* is the technique of approximating the true conditional $p(\boldsymbol{z}|\boldsymbol{x}) = \dfrac{p(\boldsymbol{x},\boldsymbol{z})}{p(\boldsymbol{x})}$ by a simpler distribution, $q(\boldsymbol{z}|\boldsymbol{\theta})$
  - We want $q(\boldsymbol{z}|\boldsymbol{\theta})$ to be "close to" $p(\boldsymbol{z}|\boldsymbol{x})$; to do this we need to define "closeness" of distributions
  - We can choose $q(\boldsymbol{z}|\boldsymbol{\theta})$ to come from a known family of distributions, e.g. Gaussians

Definition

The *Kullback-Leibler (KL) divergence* of two distributions $p(\boldsymbol{z})$ and $q(\boldsymbol{z})$ is

$$KL(q \parallel p) = \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}\left[\log \frac{q(\boldsymbol{z})}{p(\boldsymbol{z})}\right]$$

with the following properties:

- $KL(q \parallel p) \geq 0$
- $KL(q \parallel p) = 0 \iff q = p$
- $KL(q \parallel p) \neq KL(p \parallel q)$

- KL divergence is always positive and zero when distributions are equal, however it is not symmetric!
  - For *reverse-KL* (aka *information projection*), we take $KL(q \parallel p)$, which penalizes q having mass where p has none
    - When p is large where q is small, the KL divergence is small
    - When p is small where q is large, the KL divergence is large
    - This will compress q so it fits to one of the peaks of p
  - For *forward-KL* (aka *moment projection*), we take $KL(p \parallel q)$, which penalizes q missing mass where p has some
    - When p is large where q is small, the KL divergence is large
    - When p is small where q is large, the KL divergence is small
    - This will stretch out q to cover all the peaks of p
  - The choice of which KL divergence to optimize leads to different fits
    - In practice however we normally use reverse KL for computational reasons

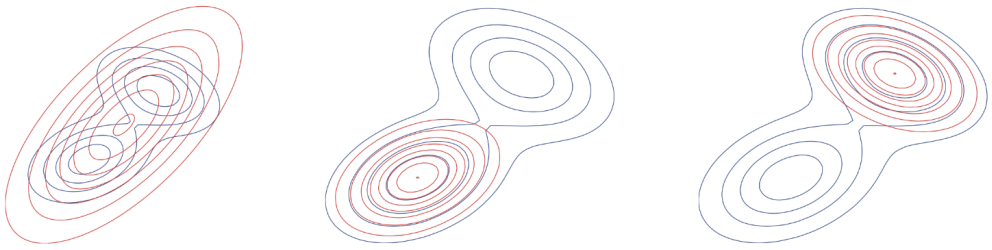

Figure 1: Approximating a bimodal distribution by a unimodal distribution; (a) minimizes forward KL, (b) and (c) minimize reverse KL.

- SVI tries to minimize the KL divergence of p and q

- $KL(q(\boldsymbol{z}|\boldsymbol{\theta}) \parallel p(\boldsymbol{z}|\boldsymbol{x})) = \mathbb{E}_{\boldsymbol{z}\sim q}\left[\log \frac{q(\boldsymbol{z}|\boldsymbol{\theta})}{p(\boldsymbol{z}|\boldsymbol{x})}\right]$
$= \mathbb{E}_{\boldsymbol{z}\sim q}\left[\log\left(q(\boldsymbol{z}|\boldsymbol{\theta})\frac{p(\boldsymbol{x})}{p(\boldsymbol{z},\boldsymbol{x})}\right)\right]$
$= \mathbb{E}_{\boldsymbol{z}\sim q}\left[\log\left(\frac{q(\boldsymbol{z}|\boldsymbol{\theta})}{p(\boldsymbol{z},\boldsymbol{x})}\right]\right) + \log p(\boldsymbol{x})$
$= -\mathcal{L}(\boldsymbol{\theta},\boldsymbol{x}) + \log p(\boldsymbol{x})$
- $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{x}) = -\mathbb{E}_{\boldsymbol{z}\sim q}\left[\log \frac{q(\boldsymbol{z}|\boldsymbol{\theta})}{p(\boldsymbol{z},\boldsymbol{x})}\right]$ is the *evidence lower bound* (ELBO)
- Since $-\mathcal{L}(\boldsymbol{\theta},\boldsymbol{x}) + \log p(\boldsymbol{x}) \geq 0$ (since KL is positive), the ELBO is a lower bound for $\log p(\boldsymbol{x})$
- As $\log p(\boldsymbol{x})$ is constant, to minimize the KL divergence we have to maximize the ELBO; therefore we do not have to compute the normalization, which is infeasible to do

- The ELBO gradient is $\vec{\nabla}_{\boldsymbol{\theta}} = \vec{\nabla}_{\boldsymbol{\theta}} \int q(\boldsymbol{z}|\boldsymbol{\theta}) \log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{\theta})}\,\mathrm{d}\boldsymbol{z}$, which must be estimated since we cannot compute this high-dimension integral
  - The *score function* (aka *REINFORCE*) gradient estimator
    - $\vec{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}\sim q}\left[\vec{\nabla}_{\boldsymbol{\theta}} \log q(\boldsymbol{z}|\boldsymbol{\theta}) \log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{\theta})}\right]$
    - Using Monte Carlo, $\vec{\nabla}_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{x}) \approx \frac{1}{B}\sum_{i=1}^{B} \vec{\nabla}_{\boldsymbol{\theta}} \log q(\boldsymbol{z}^{(i)}|\boldsymbol{\theta}) \log \frac{p(\boldsymbol{x},\boldsymbol{z}^{(i)})}{q(\boldsymbol{z}^{(i)}|\boldsymbol{\theta}}$
      - B is the number of samples
      - This is an unbiased estimator and easy to compute
    - In practice, this has higher variance than the pathwise gradient estimator
    - Use in specific domains such as reinforcement learning
  - The *pathwise* (aka *reparametrization*) gradient estimator factors out all the randomness of the distribution into a parameterless fixed source of noise, $p(\boldsymbol{\varepsilon})$
    - Find $T(\boldsymbol{\varepsilon},\boldsymbol{\theta})$ such that for $\boldsymbol{\varepsilon} \sim p(\boldsymbol{\varepsilon})$, then $\boldsymbol{z} = T(\boldsymbol{\varepsilon},\boldsymbol{\theta}) \implies \boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{\theta})$
      - e.g. for a Gaussian, $\boldsymbol{\theta} = \{\mu, \sigma\}$, let $\varepsilon \sim \mathcal{N}(\varepsilon|0,1)$ and $T(\varepsilon,\boldsymbol{\theta}) = \sigma\varepsilon + \mu$, then $z \sim \mathcal{N}(z|\mu,\sigma)$
    - Using the above, $\vec{\nabla}_{\boldsymbol{\theta}}(\boldsymbol{\theta},\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\varepsilon}\sim p(\boldsymbol{\varepsilon})}\left[\vec{\nabla}_{\boldsymbol{\theta}} \log \frac{p(\boldsymbol{x},T(\boldsymbol{\varepsilon},\boldsymbol{\theta}))}{q(T(\boldsymbol{\varepsilon},\boldsymbol{\theta})|\boldsymbol{\theta})}\right]$
    - This can then be estimated using Monte Carlo
- The main drawback of SVI is the challenge of determining how good the approximation is after the optimization terminates

Monte Carlo and Importance Sampling

- So far we've examined methods of estimating the full distribution $p(\boldsymbol{x})$, but sometimes we're only interested in the expectation of some function $\phi(\boldsymbol{x})$ under the distribution, i.e. $I = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[\phi(\boldsymbol{x})]$
- The Monte Carlo approximation is given by $I = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[\phi(\boldsymbol{x})] \approx \hat{I} = \frac{1}{R}\sum_{i=1}^{R} \phi(\boldsymbol{x}^{(i)})$
  - This is unbiased, with a standard deviation proportional to $\frac{1}{\sqrt{R}}$, independent of the dimension of $\boldsymbol{x}$
- If we only need the expectation, we only need to be able to sample from the distribution, and apply Monte Carlo to find the expectation
  - However, sampling is hard because we typically only have the unnormalized distribution, $\tilde{p}(\boldsymbol{x}) = Zp(\boldsymbol{x})$; even if we did have the full distribution, sampling from a high-dimension distribution is hard
- *Importance sampling* is a method for approximating the expectation when we only have the unnormalized distribution
  - A notable example is the particle filter for state estimation in robotics
- Let $q(\boldsymbol{x})$ be the *sampler density*, a simpler density function that we can easily sample from

- $I = \int p(\boldsymbol{x})\phi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$
  $= \int \phi(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$
  $= \frac{\int \frac{\phi(\boldsymbol{x})p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}{\int \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}$
  $= \frac{\int \frac{\phi(\boldsymbol{x})\frac{1}{Z}\tilde{p}(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}{\int \frac{\frac{1}{Z}\tilde{p}(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}$
  $= \frac{\int \frac{\phi(\boldsymbol{x})p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}{\int \frac{\tilde{p}(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}$
  $= \frac{\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[\frac{\phi(\boldsymbol{x})p(\boldsymbol{x})}{q(\boldsymbol{x})}\right]}{\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\left[\frac{\tilde{p}(\boldsymbol{x})}{q(\boldsymbol{x})}\right]}$
  - Now we can use Monte Carlo to approximate the expectations
  - $\hat{I} = \frac{\frac{1}{R}\sum_{r=1}^{R}\frac{\phi(\boldsymbol{x}^{(r)})\tilde{p}(\boldsymbol{x}^{(r)})}{q(\boldsymbol{x}^{(r)})}}{\frac{1}{R}\sum_{r=1}^{R}\frac{\tilde{p}(\boldsymbol{x}^{(r)})}{q(\boldsymbol{x}^{(r)})}} = \frac{\sum_r w_r\phi(\boldsymbol{x}^{(r)})}{\sum_r w_r}$
  - Each $w_r = \frac{\tilde{p}(\boldsymbol{x}^{(r)})}{q(\boldsymbol{x}^{(r)})}$ is referred to as the *importance weight*
    * Intuitively, if at a point $p(\boldsymbol{x}^{(r)}) > q(\boldsymbol{x}^{(r)})$, then sampling from q will under-represent this point; therefore the points are weighted more in the sum, since w_r will be larger
    * Conversely $p(\boldsymbol{x}^{(r)}) < q(\boldsymbol{x}^{(r)})$ means q over-represents the point, so in this case w_r will be small and less weight is applied to it
    * When $p(\boldsymbol{x}^{(r)}) = q(\boldsymbol{x}^{(r)})$ we can show that $\hat{I}$ applies no reweighing to samples
- The sampler density should have heavy tails (e.g. a Cauchy distribution instead of a Gaussian), since we need to compensate for the difference between distribution
- If the sampler is chosen improperly, the variance of the result can be extremely high
- In high dimensions, if the sampler distribution is not a near-perfect approximation of the target, then the entire sum will likely be dominated by a few samples with a huge weight, leading to a very bad estimate