Variational Inference (wrap-up)

CS772A: Probabilistic Machine Learning

Piyush Rai

Plan

- VI for non-conjugate models
  - Black-box VI (BBVI) for general purpose VI
  - Reparametrization Trick for general purpose VI
  - Some model-specific tricks

- Some other recent advances in VI
  - Amortized VI
  - Structured VI
  - Automatic Differentiation VI (ADVI)

VI for Non-conjugate Models

(1) Black-Box VI

Black-Box Variational Inference (BBVI)

- Black-box Var. Inference* (BBVI) approximates ELBO derivatives using Monte-Carlo
- Uses the following identity for the ELBO's derivative

$$\begin{aligned}\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \mathbb{E}_q[\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)] \\ &= \mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] \quad \text{(proof on next slide)}\end{aligned}$$

- Thus ELBO gradient can be written solely in terms of expec. of gradient of $\log q(\boldsymbol{Z}|\boldsymbol{\phi})$
  - Required gradients don't depend on the model; only on chosen var. distribution (hence "black-box")
- Given S samples $\{Z_s\}_{s=1}^S$ from $q(\boldsymbol{Z}|\boldsymbol{\phi})$, we can get (noisy) gradient as follows

$$\nabla_\phi \mathcal{L}(q) \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)(\log p(\mathbf{X},\mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi))$$

- Above is also called the "score function" based gradient (also REINFORCE method)

Gradient of a log-likelihood or log-probability function w.r.t. its params is called score function; hence the name

*Black Box Variational Inference - Ranganath et al (2014)

Proof of BBVI Identity

- The ELBO gradient can be written as

$$\begin{aligned}
\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \int (\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))q(\mathbf{Z}|\phi)d\mathbf{Z} \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))q(\mathbf{Z}|\phi)]d\mathbf{Z} \qquad (\nabla \text{ and } \int \text{ interchangeable; dominated convergence theorem}) \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]q(\mathbf{Z}|\phi) + \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]d\mathbf{Z} \\
&= \mathbb{E}_q[-\nabla_\phi \log q(\mathbf{Z}|\phi)] + \int \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]d\mathbf{Z}
\end{aligned}$$

- Note that $\mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)] = \mathbb{E}_q\left[\frac{\nabla_\phi q(\mathbf{Z}|\phi)}{q(\mathbf{Z}|\phi)}\right] = \int \nabla_\phi q(\mathbf{Z}|\phi)d\mathbf{Z} = \nabla_\phi \int q(\mathbf{Z}|\phi)d\mathbf{Z} = \nabla_\phi 1 = 0$
- Also note that $\nabla_\phi q(\mathbf{Z}|\phi) = \nabla_\phi[\log q(\mathbf{Z}|\phi)]q(\mathbf{Z}|\phi)$, using which

$$\begin{aligned}
\int \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]d\mathbf{Z} &= \int \nabla_\phi \log q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]q(\mathbf{Z}|\phi)d\mathbf{Z} \\
&= \mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]
\end{aligned}$$

- Therefore $\nabla_\phi \mathcal{L}(q) = \mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]$

Benefits of BBVI

- Recall that BBVI approximates the ELBO gradients by the Monte Carlo expectations

$$\nabla_\phi \mathcal{L}(q) \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)(\log p(\mathbf{X}, \mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi))$$

- Enables applying VI for a wide variety of probabilistic models

- Can also work with small minibatches of data rather than full data

- BBVI has very few requirements
  - Should be able to sample from $q(\boldsymbol{Z}|\boldsymbol{\phi})$ (usually sampling routines exists!)
  - Should be able to compute $\nabla_{\boldsymbol{\phi}} \log q(\boldsymbol{Z}|\boldsymbol{\phi})$ (automatic differentiation methods exist!)
  - Should be able to evaluate $p(\boldsymbol{X}, \boldsymbol{Z})$ and $\log q(\boldsymbol{Z}|\boldsymbol{\phi})$ for any value of $\boldsymbol{Z}$

- Some tricks needed to control the variance in the Monte Carlo estimate of the ELBO gradient (if interested in the details, please refer to the BBVI paper)

(2) VI using Reparametrization Trick

Reparametrization Trick

- Another Monte-Carlo approx. of ELBO grad (with often lower var than BBVI gradient)
- Suppose we want to compute ELBO's gradient $\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X},\mathbf{Z}) - \log q_\phi(\mathbf{Z})]$
- Assume a deterministic transformation g

$$\boldsymbol{Z} = g(\epsilon, \phi) \quad \text{where} \quad \epsilon \sim p(\epsilon)$$

Assumed to not depend on ϕ

- With this reparametrization, and using LOTUS rule, the ELBO's gradient would be

$$\nabla_\phi \mathbb{E}_{p(\epsilon)}[\log p(\mathbf{X}, g(\epsilon,\phi)) - \log q_\phi(g(\epsilon,\phi))] = \mathbb{E}_{p(\epsilon)} \nabla_\phi [\log p(\mathbf{X}, g(\epsilon,\phi)) - \log q_\phi(g(\epsilon,\phi))]$$

- Given S i.i.d. random samples $\{\epsilon_s\}_{s=1}^S$ from $p(\epsilon)$, we can get a Monte-Carlo approx.

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X},\mathbf{Z}) - \log q_\phi(\mathbf{Z})] \approx \frac{1}{S}\sum_{s=1}^{S}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon_s,\phi)) - \nabla_\phi \log q_\phi(g(\epsilon_s,\phi))]$$

- Such gradients are called pathwise gradients* (since we took a "path" from ϵ to $\boldsymbol{Z}$)

*Autoencoding Variational Bayes - Kingma and Welling (2013), Stochastic Backpropagation and Approximate Inference in Deep Generative Models- Rezende et al (2014)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q(\boldsymbol{w}|\boldsymbol{\phi}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, so $\boldsymbol{\phi} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$
- Suppose our ELBO has a difficult expectation term $\mathbb{E}_q[f(\boldsymbol{w})]$

Or $\boldsymbol{\phi} = \{\boldsymbol{\mu}, \mathbf{L}\}$ where $\mathbf{L} = \text{chol}(\Sigma)$

- However, note that we need ELBO gradient, not ELBO itself. Let's use the trick
- Reparametrize $\boldsymbol{w}$ as $\boldsymbol{w} = \boldsymbol{\mu} + \mathbf{L}\mathbf{v}$ where $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

Note that we will still have $q(\boldsymbol{w}|\phi) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$$\nabla_{\mu,\mathbf{L}}\mathbb{E}_{\mathcal{N}(\mathbf{w}|\mu,\Sigma)}[f(\mathbf{w})] = \nabla_{\mu,\mathbf{L}}\mathbb{E}_{\mathcal{N}(\mathbf{v}|0,\mathbf{I})}[f(\mu + \mathbf{L}\mathbf{v})] = \mathbb{E}_{\mathcal{N}(\mathbf{v}|0,\mathbf{I})}[\nabla_{\mu,\mathbf{L}} f(\mu + \mathbf{L}\mathbf{v})]$$

- The above is now straightforward
  - Easily take derivatives of $f(\boldsymbol{w})$ w.r.t. variational params $\boldsymbol{\mu}, \mathbf{L}$
  - Replace exp. by Monte-Carlo averaging using samples of $\mathbf{v}$ from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$

Often even one (or very few) samples suffice

$$\nabla_{\mu}\mathbb{E}_{\mathcal{N}(\mathbf{w}|\mu,\Sigma)}[f(\mathbf{w})] = \mathbb{E}_{\mathcal{N}(\mathbf{v}|0,\mathbf{I})}[\nabla_{\mu} f(\mu + \mathbf{L}\mathbf{v})] \approx \nabla_{\mu} f(\mu + \mathbf{L}\mathbf{v}_s)$$

$$\nabla_{\mathbf{L}}\mathbb{E}_{\mathcal{N}(\mathbf{w}|\mu,\Sigma)}[f(\mathbf{w})] = \mathbb{E}_{\mathcal{N}(\mathbf{v}|0,\mathbf{I})}[\nabla_{\mathbf{L}} f(\mu + \mathbf{L}\mathbf{v})] \approx \nabla_{\mathbf{L}} f(\mu + \mathbf{L}\mathbf{v}_s)$$

$\frac{\partial f}{\partial \boldsymbol{w}}\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{\mu}}$

Chain Rule

$\frac{\partial f}{\partial \boldsymbol{w}}\frac{\partial \boldsymbol{w}}{\partial \mathbf{L}}$

- Std. reparam. trick assumes differentiability (recent work on removing this req).

*Autoencoding Variational Bayes - Kingma and Welling (2013), Stochastic Backpropagation and Approximate Inference in Deep Generative Models- Rezende et al (2014)

Reparametrization Trick: Some Comments

- Standard Reparametrization Trick assumes the model to be differentiable

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X},\mathbf{Z}) - \log q_\phi(\mathbf{Z})] = \mathbb{E}_{p(\epsilon)}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon,\phi)) - \nabla_\phi \log q_\phi(g(\epsilon,\phi))]$$

- In contrast, BBVI (score function gradients) only required $\boldsymbol{q(Z)}$ to be differentiable
- Thus rep. trick often isn't applicable, e.g., when $\boldsymbol{Z}$ is discrete (e.g., binary /categorical)
  - Recent work on continuous relaxation† of discrete variables† (e.g., Gumbel Softmax for categorical)
- The transformation function $\boldsymbol{g}$ may be difficult to find for general distributions
  - Recent work on generalized reparametrizations*
- Also, the transformation function $\boldsymbol{g}$ needs to be invertible (difficult/expensive)
  - Recent work on implicit reparametrized gradients#
- Assumes that we can directly draw samples from $\boldsymbol{p(\epsilon)}$. If not, then rep. trick isn't valid@

†Categorical Reparameterization with Gumbel-Softmax (Jang et al, 2017), * The Generalized Reparameterization Gradient (Ruiz et al, 2016), # Implicit Reparameterization Gradients (Figurnov et al, 2018), @ Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms (Naesseth et al, 2016)

(3) Some Model-Specific Tricks

Some Model-Specific Tricks for Difficult Cases

- In some cases, we can use tricks specific to the model to simplify ELBO/its derivatives
- A common approach is to replace each difficult term by a tight lower bound, e.g.
  - Assuming $q(a,b) = \prod_i q(a_i)q(b_i)$ the expec. below can be replaced by a lower bound

$$\mathbb{E}_q\left[\log \sum_i a_i b_i\right] = \mathbb{E}_q\left[\log \sum_i p_i \frac{a_i b_i}{p_i}\right] \geq \mathbb{E}_q\left[\underbrace{\sum_i p_i \log \frac{a_i b_i}{p_i}}_{\text{via Jensen's inequality}}\right] = \sum_i p_i \mathbb{E}_q[\log a_i + \log b_i] - \sum_i p_i \log p_i$$

Now easy to compute expectations

where p_i is an auxiliary variable (depends on a_i and b_i) that we also need to optimize

  - For models with logistic lik., can use the following trick by Jaakkola and Jordan (2000)

$$-\mathbb{E}_q[\log(1+\exp(-y_n \mathbf{w}^\top \mathbf{x}_n))] \geq \log \sigma(\xi_n) + \mathbb{E}_q\left[\frac{1}{2}(y_n \mathbf{w}^\top \mathbf{x}_n - \xi_n) - \lambda(\xi_n)(\mathbf{w}^\top \mathbf{x}_n \mathbf{x}_n^\top \mathbf{w} - \xi_n^2)\right]$$

q is typically a Gaussian but the exp. still not tractable

ξ_n is an auxiliary variable that also needs to be optimized

Now easy to compute expectations

$$\lambda(\xi_n) = \frac{1}{2\xi_n}[\sigma(\xi_n) - 0.5]$$

Some Other Recent Advances in VI

Amortized Variational Inference

In addition to global variables

- Many latent variable models have a local latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI has to find the optimal $\boldsymbol{\phi}_n$ for each $q(\boldsymbol{z}_n|\boldsymbol{\phi}_n)$
- Expensive for large datasets (a similar issue which motivated SVI)
- Also slow at test time: Given a new $\boldsymbol{x}_*$, finding $\boldsymbol{\phi}_*$ requires iterative updates
  - Update local $\boldsymbol{\phi}_*$, update global $\boldsymbol{\lambda}$, and repeat until convergence
- Amortized VI: Learn a function to directly get $\boldsymbol{\phi}_n$ for any given $\boldsymbol{x}_n$

Since global variational params $\boldsymbol{\lambda}$ depend on all local var params $\boldsymbol{\phi}_n$'s

Amortization: We are shifting the cost of finding $\boldsymbol{\phi}_n$ for each data point to finding the network params $\boldsymbol{\phi}$ which are shared by all data points

$$q(\boldsymbol{z}_n|\phi_n) \approx q(\boldsymbol{z}_n|\hat{\phi}_n) \quad \text{where} \quad \hat{\phi}_n = \text{NN}_\phi(\boldsymbol{x}_n)$$

A neural network with parameters $\boldsymbol{\phi}$ (same network used for all data points)

- This function is usually called "inference network" or "recognition model"
  - Its parameters $\boldsymbol{\phi}$ are learned along with the other global vars of the model
- Popular in deep probabilistic models such as variational autoencoders (more later)

Structured Variational Inference

- Here "structured" may refer to anything that makes VI approx. more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - In general, learning more complex forms for the variational approximation family $q(\boldsymbol{Z}|\phi)$
- To remove the mean-field assumption in VI, various approaches exist
  - Structured mean-field (Saul et al, 1996)
  - Hierarchical VI (Ranganath et al, 2016): Variational params $\phi_1, \phi_2, \ldots, \phi_M$ "tied" via a shared prior

$$q(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M|\theta) = \int \left[\prod_{m=1}^{M} q(\boldsymbol{z}_m|\phi_m)\right] p(\phi|\theta) d\phi$$

- Recent work on learning more expressive variational approx. for general VI
  - Boosting or mixture of simpler distributions, e.g., $q(\boldsymbol{Z}) = \sum_{c=1}^{C} \rho_c q_c(\boldsymbol{Z})$

Even simple unimodal components will give a multimodal $q(\boldsymbol{Z})$

  - Normalizing flows*: Turn a simple var. distr. into a complex one via series of invertible transfor.

A much more complex (e.g., multimodal) variational distribution obtained via the flow idea

$$\mathbf{z}_K = f_K \circ \cdots \circ f_1(\mathbf{z}_0), \quad \mathbf{z}_0 \sim q_0(\mathbf{z}_0),$$

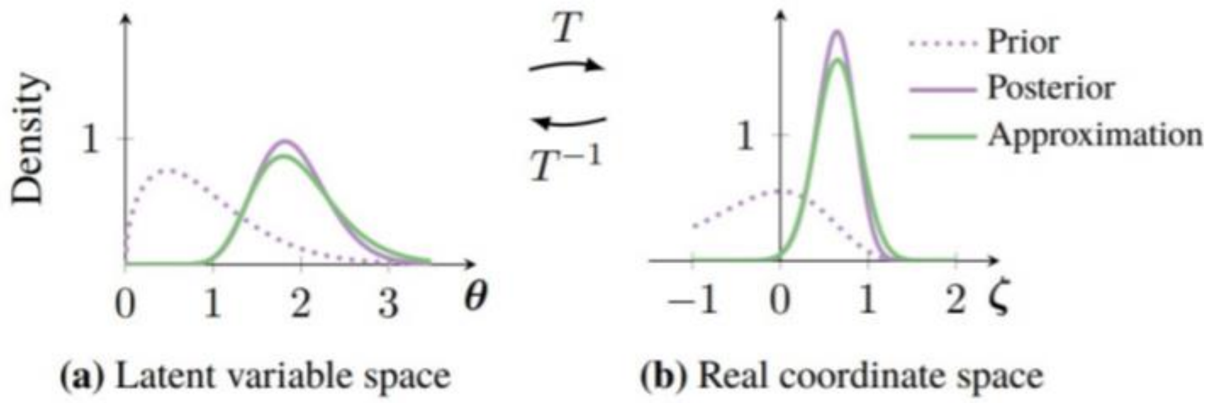
A simple unimodal variational distribution (e.g.. $\mathcal{N}(0, I)$)

$$\mathbf{z}_K \sim q_K(\mathbf{z}_K) = q_0(\mathbf{z}_0) \prod_{k=1}^{K} \left| \det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}} \right|^{-1}$$

*Variational Inference with Normalizing Flows (Rezende and Mohamed, 2015)

Automatic Differentiation Variational Inference

- Auto. Diff. (AD): A way to automate diff. of functions with unconstrained variables
- These derivatives is all what we need to optimize the function (in our case, ELBO)
- VI is also optimization. However, often the variables are constrained, e.g.,
  - Gamma's shape and scale can only be non-negative
  - Beta's parameters can only be non-negative
  - Dirichlet's probability parameter sums to one
- If we could transform our distributions to unconstrained ones, AD can be used for VI



$$T : \operatorname{supp}(p(\boldsymbol{\theta})) \to \mathbb{R}^K$$

$$\zeta = T(\boldsymbol{\theta})$$

$$\boxed{p(\mathbf{x}, \zeta)} = \boxed{p\left(\mathbf{x}, T^{-1}(\zeta)\right)} \left|\det \boxed{J_{T^{-1}}(\zeta)}\right|$$

Transformed density — Original density — Jacobian of inverse of T

$\mathbf{x}$ is data, $\boldsymbol{\theta}$ is constrained param, ζ is unconst. param with a suitable distribution (e.g., Gaussian)

*Automatic Differentiation Variational Inference (Kucukelbir et al, 2017)

Other Divergence Measures

- VI minimizes $KL(q||p)$ but other divergences can be minimized as well
  - Recall that VI with minimization of $KL(q||p)$ leads to underestimated variances
- A general form of divergence is Renyi's $\boldsymbol{\alpha}$-divergence defined as

$$D_\alpha^R(p(\boldsymbol{Z})||q(\boldsymbol{Z})) = \frac{1}{\alpha - 1} \log \int p(\boldsymbol{Z})^\alpha q(\boldsymbol{Z})^{1-\alpha} d\boldsymbol{Z}$$

- $KL(p||q)$ is a special case with $\boldsymbol{\alpha} \rightarrow \mathbf{1}$ (can verify using L'Hopital rule of taking limits)
- An even more general form of divergence is f-Divergence

$$D_f(p(\boldsymbol{Z})||q(\boldsymbol{Z})) = \int q(\boldsymbol{Z}) f\left(\frac{p(\boldsymbol{Z})}{q(\boldsymbol{Z})}\right) d\boldsymbol{Z}$$

- Many recent variational inference algorithms are based on minimizing such divergences

Variational Inference: Some Comments

- Many probabilistic models nowadays rely on VI to do approx. inference
- Even mean-field with locally-conjugacy used in lots of models
  - This + SVI gives excellent scalability as well on large datasets
- Progress in various areas has made VI very popular and widely applicable
  - Stochastic Optimization (e.g., SGD)
  - Automatic Differentiation
  - Monte-Carlo gradient of ELBO

We covered many of the threads being explored in recent work but a lot of work still being done in this area

- Note: Most of these ideas apply also to Variational EM
- Many VI and advanced VI algos are implemented in probabilistic prog. packages (e.g., Tensorflow Probability, PyTorch, etc), making VI easy even for complex models
- Still a very active area of research, especially for doing VI in complex models
  - Models with discrete latent variables
  - Reducing the variance in Monte-Carlo estimate of ELBO gradients
  - More expressive variational distribution for better approximation