Variational Inference (Wrap-up), Inference via Sampling

Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

Feb 27, 2019
Recap: VI using Monte-Carlo based Gradients of ELBO

- **VI = ELBO optimization.** Requires ELBO gradients: 
  \[ \nabla_\phi \mathcal{L}(\phi) = \nabla_\phi \mathbb{E}_q[\log p(X, Z) - \log q(Z|\phi)] \n\]

- Looked at two approaches that optimize ELBO using its Monte-Carlo based gradients
  - Black-box VI (a.k.a. score-function gradients): No model-specific gradient calculations required
    \[
    Z_s \sim q(Z|\phi) \quad s = 1, \ldots, S \quad \nabla_\phi \mathcal{L}(q) \approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\phi \log q(Z_s|\phi)[\log p(X, Z_s) - \log q(Z_s|\phi)]
    \]
  - Reparameterization trick (a.k.a. pathwise gradients)
    \[
    Z = g(\epsilon, \phi) \\
    \epsilon_s \sim p(\epsilon) \quad s = 1, \ldots, S \\
    \nabla_\phi \mathcal{L}(q) \approx \frac{1}{S} \sum_{s=1}^{S} [\nabla_\phi \log p(X, g(\epsilon_s, \phi)) - \nabla_\phi \log q_\phi(g(\epsilon_s, \phi))] \]

- Note: We can use minibatches of data (instead of all \(X\)) to compute the above gradients
Automatic Differentiation Variational Inference (ADVI)

- Auto. Diff. (AD): A way to automate differentiation of functions with unconstrained variables

- 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 can somehow transform our distributions to unconstrained ones, we can use AD for VI

ADVI transforms the variables to real-valued and then does VI with Gaussian variational approx.

\[
T : \text{supp}(p(\theta)) \rightarrow \mathbb{R}^K
\]

\[
\zeta = T(\theta)
\]

\[
p(x, \zeta) = p(x, T^{-1}(\zeta)) \mid \det J_{T^{-1}}(\zeta)
\]

Transformed density  \hspace{1cm} Original density  \hspace{1cm} Jacobian of inverse of \( T \)

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

- Many latent variable models have one latent variable $z_n$ for each data point $x_n$
- VI finds the optimal $\phi_n$ for each $q(z_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)
- Also slow at test time: Given a new $x_*$, finding $\phi_*$ requires iterative updates
  - Update local $\phi_*$, update global $\lambda$, and repeat until convergence
- Amortized VI: Learn an “inference network” or “recognition model” to directly get $\phi_n$, e.g.,
  - A neural network to directly map $x_n$ to $\phi_n$
    $$q(z_n|\phi_n) \approx q(z_n|\hat{\phi}_n) \quad \text{where} \quad \hat{\phi}_n = \text{NN}_\phi(x_n)$$
- The inference network params $\phi$ can be learned along with the other global vars
- Popular in deep probabilistic models such as variational autoencoders, deep Gaussian Processes, etc
Structured Variational Inference
Structured Variational Inference

- Here “structured” may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions

- To remove the mean-field assumption, various approaches exist
  - Structured mean-field (Saul et al, 1996)
  - Hierarchical VI (Ranganath et al, 2016): Variational params $\phi_1, \ldots, \phi_M$ “tied” via a shared prior

$$q(z_1, \ldots, z_M|\theta) = \int \left[ \prod_{m=1}^M q(z_m|\phi_m) \right] p(\phi|\theta) d\phi$$

- To learn more expressive variational approximations, various approaches exist, e.g.,
  - Boosting or mixture of simpler distributions, e.g., $q(z) = \sum_{c=1}^C \rho_c q_c(z|\phi_c)$
  - Normalizing flows: ‘Turn a simple $q(z)$ into a complex one via series of invertible transformations
Other Divergence Measures
Other Divergence Measures

- VI minimizes $KL(q\|p)$ but other divergences can be minimized as well.
- A general form of divergence is Renyi’s $\alpha$-divergence defined as
  \[
  D_{\alpha}^R(p(x)\|q(x)) = \frac{1}{\alpha - 1} \log \int p(x)^\alpha q(x)^{1-\alpha} \, dx
  \]
- $KL(p\|q)$ is a special case with $\alpha \to 1$ (can verify using L’Hopital rule of taking limits).
- An even more general form of divergence is $f$-Divergence
  \[
  D_f(p(x)\|q(x)) = \int q(x) f \left( \frac{p(x)}{q(x)} \right) \, dx
  \]
- Many recent inference algorithms are based on minimizing such divergences.
Many probabilistic models (deep/non-deep) nowadays rely on VI to do tractable inference

Even mean-field for locally-conjugate models has many applications in lots of probabilistic models
  - This + SVI gives excellent scalability

Stoch. opt., auto. diff., Monte-Carlo gradient of ELBO, contributed immensely to the success

Note: Most of these ideas apply also to Variational EM

Many VI and advanced VI algorithms are implemented in probabilistic programming packages (e.g., Stan, Tensorflow Probability, etc), making VI a painless exercise 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
Inference via Sampling

(Note that we have already seen Gibbs sampling)
Sampling for Approximate Inference

• Some typical inference tasks
  
  • Compute a (possibly intractable) **posterior distribution**: \( p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d\theta} \)
  
  • Compute a difficult **expectation** of a random quantity w.r.t. a distribution (an integral), e.g.,
  
    - The **posterior predictive** (an expectation w.r.t the posterior over \( \theta \))
      
      \[
p(\mathcal{D}^{\text{new}}|\mathcal{D}) = \int p(\mathcal{D}^{\text{new}}|\theta)p(\theta|\mathcal{D})d\theta = \mathbb{E}_{p(\theta|\mathcal{D})}[p(\mathcal{D}^{\text{new}}|\theta)]
      \]
  
    - The **marginal likelihood** or “evidence” (an expectation over the prior)
      
      \[
p(\mathcal{D}|m) = \int p(\mathcal{D}|\theta)p(\theta|m)d\theta = \mathbb{E}_{p(\theta|m)}[p(\mathcal{D}|\theta)]
      \]
  
    - The **expected complete data log-likelihood** needed for doing MLE/MAP in LVMs (recall EM)
      
      \[
      \text{Exp-CLL} = \int p(z|\theta, x)p(x, z|\theta)dz = \mathbb{E}_{p(z|\theta, x)}[p(x, z|\theta)]
      \]
  
    - The **ELBO** in variational inference
      
      \[
      \mathcal{L}(q) = \mathbb{E}_{q}[\log p(x, z)] - \mathbb{E}_{q}[\log p(z)]
      \]

• **Sampling methods** provide a general way to (approximately) solve these problems
The Basic Idea

- Can approximate any distribution using a set of randomly drawn samples from it

- The samples can also be used for computing expectations (Monte-Carlo averaging)

- Usually straightforward to generate samples if it is a simple/standard distribution

- **The interesting bit:** Even if the distribution is “difficult” (e.g., an intractable posterior), it is often possible to generate random samples from such a distribution, as we will see.
Empirical Distribution

- Sampling based approximation of a distribution can be represented using an empirical distribution
- Given $L$ “points” $z^{(1)}, \ldots, z^{(L)}$, the empirical distribution of these points is defined as

$$p_L(A) = \sum_{\ell=1}^{L} w_{\ell} \delta_{z^{(\ell)}}(A)$$

- Here $w_1, \ldots, w_L$ are weights that sum to 1, i.e., $\sum_{\ell=1}^{L} w_{\ell} = 1$ (for uniform weights, $w_\ell = 1/L$)
- Here $\delta_z(A)$ denotes the Dirac distribution defined as

$$\delta_z(A) = \begin{cases} 
0 & \text{if } z \notin A \\
1 & \text{if } z \in A 
\end{cases}$$

- $p_L(A)$ is a discrete distribution with finite support $z^{(1)}, \ldots, z^{(L)}$ (can think of it as a histogram)
Approximate Inference: VI vs Sampling-based

- VI approximates a posterior distribution $p(Z|X)$ by another distribution $q(Z|\phi)$
- Sampling uses $S$ (typically large number) samples $\{Z_s\}_{s=1}^{S}$ to approximate $p(Z|X)$
- Sampling can be used within VI (already saw ELBO approximations using Monte-Carlo)
- Also possible (though less common) to use VI in sampling algorithms (will talk about it later)
- In terms of “comparison” between VI and sampling, a few things to be noted
  - **Convergence:** VI only has local convergence, sampling (in theory) can give posterior (more on it later)
  - **Storage requirements:** Sampling-based approximation requires more storage (why?)
  - **Prediction time cost (also related to storage requirement):** Sampling always requires Monte-Carlo averaging for posterior predictive; with VI, sometimes we can get closed form posterior predictive
    - Sampling based posterior predictive: $p(x^*|X) \approx \frac{1}{S} \sum_{s=1}^{S} p(x^*|\theta_s)p(\theta_s|X)$
    - VI based posterior predictive: $p(x^*|X) \approx \int p(x^*|\theta)q(\theta|\phi)d\theta$
  - There is some work on “compressing” sampling-based approximations (e.g., see “Compact approximations to Bayesian predictive distributions” by Snelson and Ghaharamani, 2005; and “Bayesian Dark Knowledge” by Korattikara et al, 2015)
Sampling: Some Basic Methods

- Most of these basic methods are based on the idea of transformation

- Given a sample $x$ from an “easy” distribution $p(x)$, transform it into a random sample $z$ from a “less easy” distribution $p(z)$

- Some popular examples of transformation methods
  - Inverse CDF method
    \[ x \sim \text{Unif}(0, 1) \Rightarrow z = \text{Inv-CDF}_{p(z)}(x) \sim p(z) \]
  - Reparametrization method
    \[ x \sim \mathcal{N}(0, 1) \Rightarrow z = \mu + \sigma x \sim \mathcal{N}(\mu, \sigma^2) \]
  - Box-Muller method: Given $(x_1, x_2)$ from Unif$(-1, +1)$, generate $(z_1, z_2)$ from 2D Gaussian $\mathcal{N}(0, I)$

- Transformation Methods are simple but have limitations
  - Mostly limited to standard distributions and/or distributions with very few variables
Rejection Sampling

- Want to sample from \( p(z) = \frac{\tilde{p}(z)}{Z_p} \). Suppose we can only evaluate the numerator \( \tilde{p}(z) \) at any \( z \).

- Suppose we have a proposal distribution \( q(z) \) that we can generate samples from, and \( Mq(z) \geq \tilde{p}(z) \) for all \( z \) (where \( M > 0 \) is some constant).

- Basic idea: Generate samples from the proposal \( q(z) \) and accept/reject based on some condition.
  1. Sample an r.v. \( z_* \) from \( q(z) \).
  2. Sampling a uniform r.v. \( u \sim \text{Unif}[0, Mq(z_*)] \).
  3. If \( u \leq \tilde{p}(z_*) \) then accept \( z_* \) else reject.
Why $z \sim q(z) +$ accept/reject rule is equivalent to $z \sim p(z)$?

Let’s look at the pdf of $z$’s that were accepted, i.e., $p(z|\text{accept})$

\[
p(\text{accept}|z) = \int_0^\tilde{p}(z) \frac{1}{Mq(z)} du = \frac{\tilde{p}(z)}{Mq(z)}
\]

\[
p(z, \text{accept}) = q(z) \frac{p(\text{accept}|z)}{p(\text{accept})} = \frac{\tilde{p}(z)}{M}
\]

\[
p(\text{accept}) = \int \frac{\tilde{p}(z)}{M} dz = \frac{Z_p}{M}
\]

\[
p(z|\text{accept}) = \frac{p(z, \text{accept})}{p(\text{accept})} = \frac{\tilde{p}(z)}{Z_p} = p(z)
\]
Sampling for Approximating Expectations

Suppose $f(z)$ is function of a random variable $z \sim p(z)$

Wish to compute $\mathbb{E}[f] = \mathbb{E}_{p(z)}[f(z)] = \int f(z)p(z)dz$

Given $L$ independent samples $\{z^{(\ell)}\}_{\ell=1}^L$ from $p(z)$, we can approximate the above as

$$\mathbb{E}[f] \approx \frac{1}{L} \sum_{\ell=1}^L f(z^{(\ell)})$$ (Monte Carlo sampling)

What if we can't generate samples from $p(z)$? Answer: Use Importance Sampling

If we can generate $L$ indep. samples $\{z^{(\ell)}\}_{\ell=1}^L$ from a different “proposal” distribution $q(z)$ then

$$\mathbb{E}[f] = \int f(z)p(z)dz = \int f(z)\frac{p(z)}{q(z)}q(z)dz \approx \frac{1}{L} \sum_{\ell=1}^L f(z^{(\ell)})\frac{p(z^{(\ell)})}{q(z^{(\ell)})}$$

IS only requires that we can evaluate $p(z)$ at any $z$ (in fact, with a small modification to the above, IS works even when we can evaluate $p(z)$ only up to a proportionality constant)

Note: IS is NOT a sampling method (doesn't generate samples from a desired distribution; just a way to approximate expectations)
Limitations of Basic Sampling Methods

- Transformation based methods: Usually limited to drawing from standard distributions
- Rejection Sampling and Importance Sampling: Require good proposal distributions

Difficult to find good prop. distr. especially when $z$ is high-dim. (e.g., models with many params)

- In high dimensions, most of the mass of $p(z)$ is concentrated in a tiny region of the $z$ space
- Difficult to \emph{a priori} know what those regions are, thus difficult to come up with good proposal dist.

A solution to these: MCMC methods