Scaling Up Variational Inference and Expectation Propagation

Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

March 22, 2018

Quick Recap and An Important Fact

- The updates for mean-field VI with $q(\mathbf{Z}|\phi) = \prod_{i=1}^{M} q(\mathbf{Z}_i|\phi_i)$ are of the following form

$$\boxed{\log q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i\neq j}[\log p(\mathbf{X}, \mathbf{Z})] + \text{const}}$$

  where $\mathbb{E}_{i\neq j}$ denotes expectation w.r.t. current optimal q_i's of all other groups
- The above solution can also be written as $q_j^*(\mathbf{Z}_j) = \frac{\exp(\mathbb{E}_{i\neq j}[\log p(\mathbf{X},\mathbf{Z})])}{\int \exp(\mathbb{E}_{i\neq j}[\log p(\mathbf{X},\mathbf{Z})])d\mathbf{Z}_j}$
- Important fact: $\log q_j^*(\mathbf{Z}_j) \propto \mathbb{E}_{i\neq j}[\log p(\mathbf{X}, \mathbf{Z})] = \mathbb{E}_{i\neq j}[\log p(\mathbf{X}, \mathbf{Z}_j, \mathbf{Z}_{-j})]$ can also be written as

$$\boxed{\log q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i\neq j}[\log p(\mathbf{Z}_j|\mathbf{X}, \mathbf{Z}_{-j})] + \text{const}}$$

- For locally conjugate models, $p(\mathbf{Z}_j|\mathbf{X}, \mathbf{Z}_{-j})$ is easy to find, and usually an exp. fam. dist.

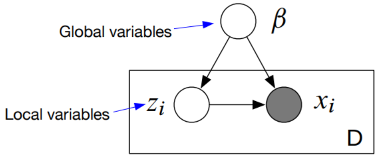
$$p(\mathbf{Z}_j|\mathbf{X}, \mathbf{Z}_{-j}) \propto h(\mathbf{Z}_j) \exp\left[\eta(\mathbf{X}, \mathbf{Z}_{-j})^\top \mathbf{Z}_j - A(\eta(\mathbf{X}, \mathbf{Z}_{-j}))\right]$$

  .. therefore (verify) $q_j^*(\mathbf{Z}_j) \propto h(\mathbf{Z}_j) \exp\left[\mathbb{E}_{i\neq j}[\eta(\mathbf{X}, \mathbf{Z}_{-j})]^\top \mathbf{Z}_j\right]$; only requires finding $\mathbb{E}_{i\neq j}[\eta(\mathbf{X}, \mathbf{Z}_{-j})]$
- In VI, ELBO/its derivatives in general may be difficult. But tricks exist (e.g., reparam., BBVI, etc)

Scaling Up VI for Large Datasets via Stochastic Variational Inference (SVI)

A Generic Probabilistic Model

- Assume D data points $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_D\}$ generated via a probabilistic model



- Assume local latent variables $\mathbf{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_D\}$, one per data point ($\boldsymbol{z}_i$ for $\boldsymbol{x}_i$, $i = 1, \ldots, D$)
- Assume global latent variables β shared by all data points
- Probability distribution of data point $\boldsymbol{x}_i$ only depends on $\boldsymbol{z}_i$ and β
- The joint distribution for this model admits the following factorization

$$p(\mathbf{X}, \mathbf{Z}, \beta) = p(\beta) \prod_{i=1}^{D} p(x_i, z_i | \beta) = p(\beta) \prod_{i=1}^{D} p(x_i | z_i, \beta) p(z_i)$$

Mean-Field VI

- Our goal is to infer the posterior distribution $p(\mathbf{Z}, \beta|\mathbf{X})$. Intractable in general.
- Let's approximate $p(\mathbf{Z}, \beta|\mathbf{X})$ using a mean-field distribution $q(\mathbf{Z}, \beta)$

$$q(\mathbf{Z}, \beta) = q(\beta) \prod_{i=1}^{D} q(\boldsymbol{z}_i)$$

- The log joint probability has a simple form as a summation over all data points

$$\ln p(\boldsymbol{x}, \boldsymbol{z}, \beta) = \sum_{i=1}^{D} \ln p(x_i, z_i|\beta) \; + \; \ln p(\beta)$$

- The ELBO $\mathcal{L} = \mathbb{E}_q \left[\ln \frac{p(\mathbf{X},\mathbf{Z},\beta)}{q(\mathbf{Z},\beta)} \right]$ for the model can be written as

$$\mathcal{L} = \sum_{i=1}^{D} \mathbb{E}_q \left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

- Can now do mean-field VI by optimizing w.r.t. $q(\boldsymbol{z}_i)$, $\forall i$, and $q(\beta)$, until convergence

Mean-Field VI

- The basic mean-field VB is a batch algorithm (each iteration operates on all the data)
- Each iteration has to look at every data point $\boldsymbol{x}_i$ and infer the corresponding $q(\boldsymbol{z}_i)$, $\forall i$

Batch variational inference

1. For $i = 1, \ldots, D$, optimize $q(z_i)$
2. Optimize $q(\beta)$ $\longleftarrow$ Depends on all the $q(z_i)$
3. Repeat

- This can be very slow is the number of data points D is very large
    - Before updating the global variables β, we must update all the local variables $\boldsymbol{z}_i$'s
- Would like to have a faster inference algorithm that scales nicely with number of data points

Stochastic Variational Inference (SVI)

- Based on the idea of stochastic optimization
- In each VB iteration, let's use only a small mini-batch of data points (chosen randomly)

Stochastic variational inference

1. Randomly select a subset of local data, $S_t \subset \{1, \dots, D\}$
2. Construct the scaled variational objective function

$$\mathcal{L}_t = \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}_q \left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

3. Optimize each $q(z_i)$ in $\mathcal{L}_t$ *only*
4. Update the parameters of $q(\beta|\psi)$ using a gradient step

$$q(\beta|\psi) \quad \rightarrow \quad \psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$$

5. Repeat

- Similar in spirit to online EM
- ρ_t is a learning rate. M_t (a "pre-conditioner") will be defined later
- Note: In step-4, instead of solving for $q(\beta|\psi)$ analytically, we are using a gradient method

Stochastic Variational Inference - Hoffman et al (2013)

ELBO on the Mini-Batch

- The ELBO on the full data

$$\mathcal{L} = \sum_{i=1}^{D} \mathbb{E}_q \left[\ln \frac{p(x_i, z_i | \beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

- The (scaled) ELBO on the random subset (mini-batch) of the data chosen in iteration t

$$\mathcal{L}_t = \frac{D}{|S_t|} \sum_{i=1}^{D} \mathbb{1}(d \in S_t) \mathbb{E}_q \left[\ln \frac{p(x_i, z_i | \beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

- The expectation of the mini-batch ELBO (expectation w.r.t. the random selection of mini-batch)

$$\mathbb{E}_P[\mathcal{L}_t] = \frac{D}{|S_t|} \sum_{i=1}^{D} \underbrace{\mathbb{E}_P[\mathbb{1}(d \in S_t)]}_{P(d \in S_t)} \mathbb{E}_q \left[\ln \frac{p(x_i, z_i | \beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

- Note that $P(d \in S_t) = \frac{|S_t|}{D}$. Therefore $\mathbb{E}_P[\mathcal{L}_t] = \mathcal{L}$ (this is good news!)

Stochastic Updates for SVI

- For our stochastic update of ψ for updating $q(\beta|\psi)$, i.e., $\psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$

$$\mathbb{E}_P[\psi_t] = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathbb{E}_P[\mathcal{L}_t] (= \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L})$$

- Therefore the stochastic gradient computed using S_t is unbiased
- We now basically have an stochastic gradient method to update ψ
- Note: The learning rate ρ_t must satisfy $\sum_{t=1}^{\infty} \rho_t = \infty$ and $\sum_{t=1}^{\infty} \rho_t^2 < \infty$
  - Setting $\rho_t = \frac{1}{(t_0+t)^\kappa}$ with $\kappa \in (0.5, 1)$ ensures this. t_0 is some positive number.
- Note that SVI also gives us all the "local" $q(\boldsymbol{z}_i)$ distributions in the normal way. It's only the $q(\beta)$ distribution (which depends on the local distributions) that is inferred in an online fashion

SVI for Exponential Family Distributions

- Assuming the joint distribution of data $\boldsymbol{x}_i$ and local latent var. $\boldsymbol{z}_i$ to be exponential family dist.

$$p(\mathbf{X}, \mathbf{Z}|\beta) = \prod_{i=1}^{D} p(x_i, z_i|\beta) = \left[\prod_{i=1}^{D} h(x_i, z_i)\right] \mathrm{e}^{\beta^T \sum_{i=1}^{D} t(x_i, z_i) - DA(\beta)}$$

- Let's assume a conjugate prior on the global variables β

$$p(\beta) = f(\xi, \nu) e^{\beta^\top \xi - \nu A(\beta)}$$

- Let's assume our variational distribution $q(\beta)$ to have the same form as the prior $p(\beta)$

$$q(\beta) = f(\xi', \nu') e^{\beta^\top \xi' - \nu' A(\beta)}$$

- SVI will basically do stochastic optimization to learn the parameters ξ', ν' of variational distribution
- We will now look at the general form of these updates for a generic model with exp. family dist.

SVI for Exponential Family Distributions

- If we were doing full batch updates for $q(\beta|\psi)$ (where $\psi = [\xi, \nu]$) using gradient methods then

$$\begin{bmatrix} \xi' \\ \nu' \end{bmatrix} \leftarrow \begin{bmatrix} \xi' \\ \nu' \end{bmatrix} + \rho_t M_t \nabla_{(\xi',\nu')} \mathcal{L}$$

- For this update, the part of $\mathcal{L}$ that depends on β is

$$\mathcal{L}_\beta = \sum_{i=1}^{D} \mathbb{E}_q[\ln p(x_i, z_i|\beta)] + \mathbb{E}_q[\ln p(\beta)] - \mathbb{E}_q[\ln q(\beta)]$$

- Plugging in the expressions for exponential family distributions (given on the previous slide)

$$\mathcal{L}_\beta = \mathbb{E}_q[\beta]^T \left(\sum_{i=1}^{D} \mathbb{E}[t(x_i, z_i)] + \xi - \xi' \right) - \mathbb{E}_q[A(\beta)](D + \nu - \nu') - \ln f(\xi', \nu') + \text{const.}$$

- Note that it requires computing two expectations $\mathbb{E}_q[\beta]$ and $\mathbb{E}_q[A(\beta)]$

SVI for Exponential Family Distributions

- The expectations $\mathbb{E}_q[\beta]$ and $\mathbb{E}_q[A(\beta)]$ can be computed as follows

$$q(\beta) = f(\xi', \nu')\mathrm{e}^{\beta^T \xi' - \nu' A(\beta)}$$

$$\int \nabla_{\xi'} q(\beta) d\beta = 0, \qquad \int \frac{\partial}{\partial \nu'} q(\beta) d\beta = 0$$

$$\mathbb{E}_q[\beta] = -\nabla_{\xi'} \ln f(\xi', \nu'), \qquad \mathbb{E}_q[A(\beta)] = \frac{\partial \ln f(\xi', \nu')}{\partial \nu'}$$

- Exercise: Verify the above (using the fact that $q(\beta)$ is an exp-family dist.)
- These can then be plugged into the expression of $\mathcal{L}_\beta$

SVI for Exponential Family Distributions

- ELBO gradient (batch case)

$$\nabla_{(\xi',\nu')}\mathcal{L}_\beta = \begin{bmatrix} \nabla_{\xi'}\mathcal{L}_\beta \\ \frac{\partial}{\partial\nu'}\mathcal{L}_\beta \end{bmatrix} = -\begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi',\nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'\partial\xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'\partial\xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'^2} \end{bmatrix} \begin{bmatrix} \sum_{i=1}^D \mathbb{E}_q[t(x_i,z_i)] + \xi - \xi' \\ D + \nu - \nu' \end{bmatrix}$$

- Since preconditioning matrix is p.s.d., setting gradient to zero gives closed-form batch VB update

$$\xi' = \xi + \sum_{i=1}^{D} \mathbb{E}[t(x_i,z_i)], \qquad \nu' = \nu + D$$

- The SVI uses the stochastic gradients and the updates will be $\psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$

$$\begin{bmatrix} \xi'_t \\ \nu'_t \end{bmatrix} = \begin{bmatrix} \xi'_{t-1} \\ \nu'_{t-1} \end{bmatrix} - \rho_t M_t \begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi',\nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'\partial\xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'\partial\xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'^2} \end{bmatrix} \begin{bmatrix} \frac{D}{|S_t|}\sum_{i\in S_t} \mathbb{E}[t(x_i,z_i)] + \xi - \xi'_{t-1} \\ D + \nu - \nu'_{t-1} \end{bmatrix}$$

- M_t can be set to $\mathbf{I}$. However we can choose M_t sensibly to get a clean update.

SVI for Exponential Family Distributions

- Our stochastic gradient updates had the form

$$\begin{bmatrix} \xi'_t \\ \nu'_t \end{bmatrix} = \begin{bmatrix} \xi'_{t-1} \\ \nu'_{t-1} \end{bmatrix} - \rho_t M_t \begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi', \nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu'^2} \end{bmatrix} \begin{bmatrix} \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}[t(x_i, z_i)] + \xi - \xi'_{t-1} \\ D + \nu - \nu'_{t-1} \end{bmatrix}$$

- Suppose we set M_t as

$$M_t = - \begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi', \nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu'^2} \end{bmatrix}^{-1}$$

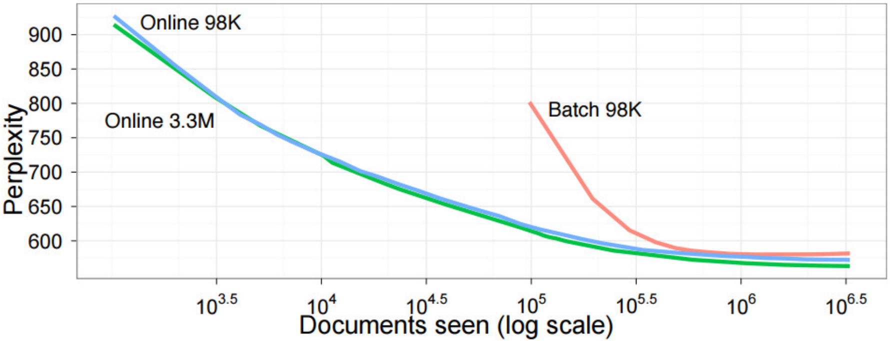
- Then the updates will be

$$\boxed{\xi'_t = (1-\rho_t)\xi'_{t-1} + \rho_t \left(\xi + \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}_{q(z_i)}[t(x_i, z_i)] \right) \qquad \nu'_t = (1-\rho_t)\nu'_{t-1} + \rho_t(\nu + D)}$$

- Note: The above choice of M_t is not arbitrary. It is actually equivalent to $M_t = -\mathbb{E}_q[\nabla^2 \ln q(\beta)]$
  - This choice makes our stochastic gradient a "natural gradient" (as opposed to Euclidean gradient)

SVI vs Batch Inference

- Online inference methods (e.g., SVI) usually have a faster convergence than batch inference
- Shown below is a plot comparing batch and online inference for topic model (LDA)



(Pic courtesy: David Blei)

Minimizing KL by Moment Matching

- VB minimizes $\text{KL}(q||p)$ w.r.t. q. Consider minimizing the "reverse", i.e., $\text{KL}(p||q)$
- Assume $p(\mathbf{Z})$ fixed (but unknown) and $q(\mathbf{Z})$ to be an exponential family dist.

$$q(\mathbf{Z}) = h(\mathbf{Z})g(\boldsymbol{\eta})\exp(\boldsymbol{\eta}^\top T(\mathbf{Z}))$$

- Then $KL(p||q) = \int p(\mathbf{Z}) \log \frac{p(\mathbf{Z})}{q(\mathbf{Z})} d\mathbf{Z}$ can then be written as

$$KL(p||q) = -\log g(\boldsymbol{\eta}) - \boldsymbol{\eta}^\top \mathbb{E}_{p(\mathbf{Z})}[T(\mathbf{Z})] + \text{const}$$

- Minimizing w.r.t. q means minimizing w.r.t. $\boldsymbol{\eta}$, and is attained when

$$-\nabla \log g(\boldsymbol{\eta}) = \mathbb{E}_{p(\boldsymbol{z})}[T(\mathbf{Z})]$$

- Note that $-\nabla \log g(\boldsymbol{\eta})$ is equal to $\mathbb{E}_{q(\mathbf{Z})}[T(\mathbf{Z})]$, we will have

$$\mathbb{E}_{q(\mathbf{Z})}[T(\mathbf{Z})] = \mathbb{E}_{p(\mathbf{Z})}[T(\mathbf{Z})]$$

  which is a simple moment matching problem. How about using this idea for approximate inference?

Expectation Propagation

- Denote the unknowns by θ. Assume the true posterior distribution over θ given data $\mathcal{D}$

$$p(\theta|\mathcal{D}) = \frac{1}{p(\mathcal{D})}\prod_i f_i(\theta)$$

- For many problems $p(\theta|\mathcal{D})$ has this form: one factor $f_n(\boldsymbol{x}|\theta)$ per data point $\boldsymbol{x}_n$, plus one factor $f_0(\theta) = p(\theta)$ for the prior θ (total $N+1$ factors)
- Assume we approximate it with another product of total $N+1$ factors

$$q(\theta) = \frac{1}{Z}\prod_i \tilde{f}_i(\theta)$$

- Expectation Propagation (EP) is based on minimizing the following KL divergence

$$\mathrm{KL}\left(p||q\right) = \mathrm{KL}\left(\frac{1}{p(\mathcal{D})}\prod_i f_i(\boldsymbol{\theta}) \middle\| \frac{1}{Z}\prod_i \widetilde{f}_i(\boldsymbol{\theta})\right)$$

- But the above KL minimization is a hard problem in general
  - Reason: Since $KL(p||q) = \int p(\theta|\mathcal{D}) \log \frac{p(\theta|\mathcal{D})}{q(\theta)} d\theta$, this requires averaging over the true posterior

Expectation Propagation

- EP is an iterative scheme of solving the above problem by matching each $\tilde{f}_j(\theta)$ with $f_j(\theta)$
  - But instead of doing it independently for each $\tilde{f}_j(\theta)$, we do it in the context of all other $\tilde{f}_i(\theta)$, $i \neq j$
- The idea is to refine $\tilde{f}_j(\theta)$ s.t. the new approx. posterior

$$q^{\text{new}}(\boldsymbol{\theta}) \propto \tilde{f}_j(\boldsymbol{\theta}) \prod_{i \neq j} \tilde{f}_i(\boldsymbol{\theta})$$

  is as close as possible to the distribution $\propto f_j(\theta) \prod_{i \neq j} \tilde{f}_i(\theta)$

- Define $q^{\setminus j} = \frac{q(\theta)}{\tilde{f}_j(\theta)}$; can get it easily by subtracting off the natural parameters of $\tilde{f}_j$ from q
- Now solve the following *simpler* KL minimization problem w.r.t. $q^{new}(\theta)$

$$\mathrm{KL}\left(\frac{f_j(\boldsymbol{\theta}) q^{\setminus j}(\boldsymbol{\theta})}{Z_j} \middle\| q^{\text{new}}(\boldsymbol{\theta})\right)$$

  where $Z_j = \int f_j(\theta) q^{\setminus j}(\theta) d\theta$

- The above can be solved by matching the moments of q^{new} with $\frac{f_j(\theta) q^{\setminus j}(\theta)}{Z_j}$

Expectation Propagation

- From $q^{new}(\theta)$, we can get the required factor $\tilde{f}_j(\theta)$

$$\tilde{f}_j(\theta) = K\frac{q^{new}(\theta)}{q^{\setminus j}(\theta)}$$

  where $K = \int \tilde{f}_j(\theta)q^{\setminus j}(\theta)d\theta$

- Finally, using the fact

$$\int \tilde{f}_j(\theta)q^{new}(\theta)d\theta = \int f_j(\theta)q^{new}(\theta)d\theta$$

  we get $K = Z_j$

- This is repeated over each factor $\tilde{f}_j(\theta)$, for several passes
- Look at the clutter problem in PRML (sec 10.7.1) for a concrete example

Summary

- VI is a deterministic approximate inference method (unlike sampling methods)
- Finds the best q by maximizing the ELBO (or minimizing $\text{KL}(q||p)$)
- VI is guaranteed to converge to a local optima (in finite time)
- Simplifying assumption on q (e.g., mean field VB) can make VB updates very easy to derive
- More advanced VI methods can handle richer forms of $q(\mathbf{Z})$, likelihood and priors
  - E.g., Monte-Carlo approximations of ELBO and its derivatives
- Combination of methods like BBVI and SVI can help us develop scalable approximate inference algorithms for a large class of models
- Implementations of many classic/advanced VI methods available in Stan, Edward, etc.