Variational Bayes for latent variable models

Reetam Majumder

STAT 740 Guest Lecture, Fall 2022

NC STATE UNIVERSITY

Some housekeeping

- We'll focus mainly on latent variable models
- Variational Bayes (VB) often term them local variables.
- Parameters are global variables
- The intuition (usually) is that the number of local variables grow with the data, while global variables have fixed dimension
- General notation:

$$\begin{aligned} y, x &:= \text{observations/covariates} \\ s &:= \text{hidden/latent variables} \\ \theta &:= \text{parameters} \\ z &:= (s, \theta) \\ p(\cdot) &:= \text{prior/likelihood/posterior} \\ q(\cdot) &:= \text{variational posterior} \end{aligned}$$

VB vs. MCMC

- MCMC used when you don't have a closed form for the posterior, but can sample from it[1]
- Idea: Get samples to approximately reconstruct the exact posterior.
- Pros: Uncertainty, theoretical guarantees. Cons: s l o w
- What if we consider an approximate posterior in a 'nice' family that we can work with analytically?
- Might be good enough if all we care are about point estimates (posterior means, in particular)

[1] *https://www4.stat.ncsu.edu/~bjreich/ST740/MixNormal.html*

Example: Gaussian mixture model

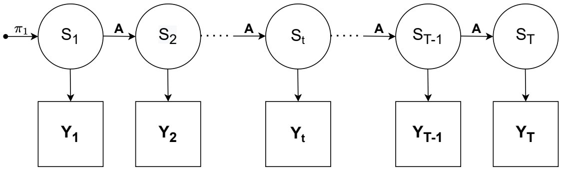
$$
\begin{aligned}
p(y_i|s_i,\theta) &= \sum_{j=1}^{K} c_j \cdot \text{Normal}(y_i|\mu_j,\sigma^2), i = 1:n \\
p(s_i|c_{1:K}) &= \text{Categorical}(s_i|c_1,\ldots,c_K) \\
p(\mu_j) &= \text{Normal}(\mu_j|m_j,\tau^2) \\
p(c_{1:K}) &= \text{Dirichlet}(c_1\ldots,c_K|\alpha_1\ldots,\alpha_K)
\end{aligned}
$$

- Global variables: $\theta = (\mu_{1:K}, c_{1:K})$ tend to usually be of fixed dimension
- Local variables: S_i control the cluster assignments, dimension grows with size of data

The posterior is:

$$
p(\mu,s,c|y) = \frac{p(c_{1:K})\prod_{j=1}^{K} p(\mu_j)\prod_{i=1}^{n} p(s_i)p(y_i|s_i,\theta)}{\int_{\mu_{1:K}}\sum_{z_{1:n}} p(c_{1:K})\prod_{j=1}^{K} p(\mu_j)\prod_{i=1}^{n} p(s_i)p(y_i|s_i,\theta)}
$$

Example: Hidden Markov model



- $p(y_t|s_{tj}, \theta) = \text{Categorical}(y_t|c_{j1}, \ldots, c_{jM})$, where $s_{tj} = \mathbb{I}(S_t = j)$
- $S_{1:T}$ is a Markov chain parameterized by $\pi_1 = Pr[s_1 = j]$, and $A := ((a_{jk}))$, where $a_{jk} = Pr[s_{t+1} = k|s_t = j]$, $j, k = 1 : K$
- $p(C) = \prod_{j=1}^{K} \text{Dirichlet}(c_{j,1:M}|\zeta_1, \ldots, \zeta_M)$
- $p(A) = \prod_{j=1}^{K} \text{Dirichlet}(a_{j,1:K}|\alpha_1, \ldots, \alpha_K)$
- Global variables $\theta = (C, A)$, local variables $S_{1:T}$

Example: Text prediction. MCMC for HMMs is non-trivial at best and prohibitive for many real cases.

Models without latent variables

Linear regression

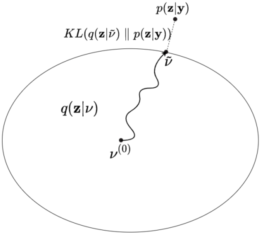
- Global variables (θ_j, σ^2)
- $p(\theta_j) = \text{Normal}(\theta_j | \mu_j, \tau^2)$

Logistic regression

- Global variable θ_j
- $p(\theta_j) = \text{Normal}(\theta_j | \mu_j, \tau^2)$

Bayesian neural networks aren't necessarily latent variable models, they're just plain intractable.

VB as optimization



Aim : Approximate the exact posterior $p(\mathbf{z}|\mathbf{y})$

1. Posit a family of approximate distributions $\mathbb{Q}$ with its own variational parameters
2. Optimize over this family to find the parameter settings which minimize the KL divergence from the exact posterior

$$q(\tilde{\mathbf{z}}) = \arg\min_{q(\mathbf{z}|\nu)\in\mathbb{Q}} KL\big(q(\mathbf{z}|\nu) \parallel p(\mathbf{z}|\mathbf{y})\big)$$

Review of variational inference

- Minimizing KL-divergence $\iff$ maximizing evidence lower bound (ELBO)

$$\text{ELBO}(q) = \mathbb{E}[\log p(\mathbf{z}, \mathbf{y})] - \mathbb{E}[\log q(\mathbf{z})]$$

- Analysis often restricted to a mean-field variational family $\mathbb{Q}$, where the latent variables and the parameters are all mutually independent

$$q(\mathbf{z}) \approx \prod_i q_i(z_i)$$

  Each latent component z_i has its own variational marginal posterior, with free parameters/variational parameters that are optimized

More on the ELBO

$$
\begin{aligned}
\log p(y) &= \log \int_z p(y, z) \\
&= \log \int_z q(z) \frac{p(y, z)}{q(z)} \\
&= \log \mathbb{E}_q \left[\frac{p(y, z)}{q(z)} \right] \\
&\geq \mathbb{E}_q[\log p(y, z)] - \mathbb{E}_q[\log q(z)]
\end{aligned}
$$

- How did that last inequality happen?
- Other divergence metrics are also possible
- Using KL breaks this optimization problem into nice, manageable chunks

One last assumption before we we get to the optimization bit.

The mean field assumption

- At the very least, it assumes that the variational posteriors for the local and global variables are independent, i.e.

$$q(\theta, s) \approx q_\theta(\theta) q_s(s)$$

- Typically, the more you factorize, the simpler the optimization becomes, e.g. for the GMM example,

$$q(\mu, s, c) \approx q(\mu_{1:K}) q(s_{1:K}) q(c_{1:K})$$

- The optimization is straightforward if things are in the conjugate-exponential family

VB optimization for conjugate exponential families

Most classical VB approaches lean on this[2]. Given that,

Condition 1: The complete data likelihood is in the exponential family:

$$p(y, s|\theta) = f(y, s)g(\theta)\exp\{\phi(\theta)^T u(y, s)\}$$

Condition 2: The parameter prior is conjugate to the complete data likelihood:

$$p(\theta|\nu, \eta) = h(\nu, \eta)g(\theta)^\eta \exp\{\phi(\theta)^T \nu\}$$

Note: $\phi(\theta)$ is the vector of natural parameters, η, ν are hyperparameters of the prior.

[2] *https://papers.nips.cc/paper/2000/file/77369e37b2aa1404f416275183ab055f-Paper.pdf*

Theorem (1)
Given an iid data set $y = (y_1, \ldots, y_n)$, if the model satisfies the stated conditions, then at the minima of $KL(q||p)$,

- *$q_\theta(\theta)$ is conjugate and of the form:*

$$q_\theta(\theta) = h(\tilde{\eta}, \tilde{\nu}) g(\theta)^{\tilde{\eta}} \exp\{\phi(\theta)^T \tilde{\nu}\},$$

  where $\tilde{\eta} = \eta + n$, $\tilde{\nu} = \nu + \sum_{i=1}^n \bar{u}(y_i)$, and $\bar{u}(y_i) = \mathbb{E}_q u(y_i, s_i)$.
- *$q_s(s) = \prod_{i=1}^n q_{s_i}(s_i)$ and $q_{s_i}(s_i)$ is of the same form as the known parameter posterior:*

$$q_{s_i}(s_i) \propto f(y_i, s_i) \exp\{\bar{\phi}(\theta)^T u(y_i, s_i)\} = p(s_i | y_i, \bar{\phi}(\theta)),$$

  where $\bar{\phi}(\theta) = \mathbb{E}_q(\theta)$.

The VBEM algorithm

- **VE Step**: Compute the expected sufficient statistics $t(y) = \sum_i \bar{u}(y_i)$ under the hidden variable distributions $q_{s_i}(s_i)$.
- **VM Step**: Compute the expected natural parameters $\bar{\phi}(\theta)$ under the parameter distribution given by $\tilde{\eta}$ and $\tilde{\nu}$

Connection with Gibbs sampling: It's easy to show that a valid alternative expression for $q_{\theta_i}(\theta_i)$ is

$$q_{\theta_i}(\theta_i) \propto \exp\{\mathbb{E}_{-\theta_i} \log p(\theta_i|\theta_{-i}, y, s),$$

viz, the full conditionals. A similar optimal density form can be see for $q_{s_i}(s_i)$ too. In situations where Gibbs sampling is viable, analytical VB posteriors are available under conjugacy.

Coordinate ascent VB

What would the VBEM algorithm look like for the GMM?

- VBM step:

$$q_{\mu_i}(\mu_i) \propto \exp\{\mathbb{E}_{-\mu_i} \log p(\mu_i|\cdot)\} \tag{1}$$

$$q_{c_i}(c_i) \propto \exp\{\mathbb{E}_{-c_i} \log p(c_i|\cdot)\} \tag{2}$$

- VBE step:

$$q_{s_i}(s_i) \propto \exp\{\mathbb{E}_{-s_i} \log p(s_i|\cdot)\} \tag{3}$$

ELBO guaranteed to increase at every step, and like the EM, will converge to a local maximum.

Questions:

1. Why is it called coordinate ascent?
2. What's the connection between this and the theorem before?
3. How does this lead to a stochastic implementation?

The Gaussian mixture model

Likelihood:

$$p(y_i|s_i,\theta) = \sum_{j=1}^{K} c_j \cdot \mathrm{Normal}(y_i|\mu_j,\sigma^2), i = 1:n$$

$$p(s_i|c_i) = \prod_{j=1}^{K} c_j^{\mathbb{I}(s_i=j)}$$

Priors:

$$p(\mu_j) = \mathrm{Normal}(m_j,\tau^2)$$

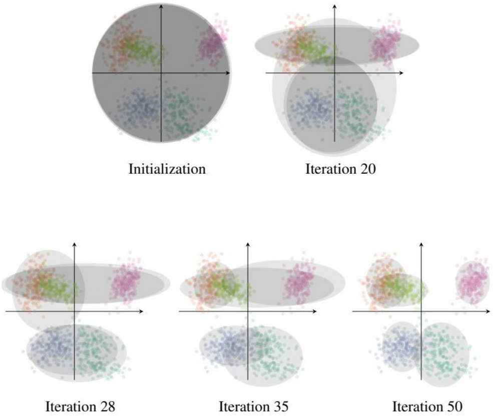
$$p(c_{1:K}) = \mathrm{Dirichlet}(c_1\ldots,c_K|\alpha_1\ldots,\alpha_K)$$

Variational posteriors:

$$q(\mu_j) = \mathrm{Normal}(\tilde{m}_j,\tilde{\tau}^2)$$

$$q(c_{1:K}) = \mathrm{Dirichlet}(c_1\ldots,c_K|\tilde{\alpha}_1\ldots,\tilde{\alpha}_K)$$

The Gaussian mixture model



Source: Blei *et al*. Variational inference: a review for statisticians. 2017.
Code examples (RStudio/RPubs): Linear regression, probit regression, GMM.

What do we actually get out of this?

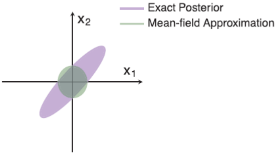


Figure 1. Visualizing the mean-field approximation to a two-dimensional Gaussian posterior. The ellipses show the effect of mean-field factorization. (The ellipses are 2σ contours of the Gaussian distributions.)

Source: Blei *et al.* Variational inference: a review for statisticians. 2017.

- Posterior means - the full variational posterior is not always a good representation of the true posterior
- (Approximate) predictive distribution, posterior covariances[3]
- The more we relax the mean field assumption, the better the approximation gets, with increasing computational cost

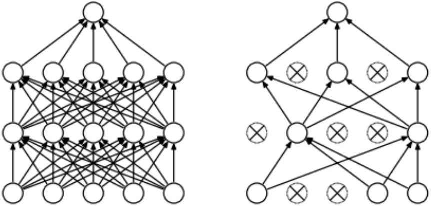
[3] Giordano *et al.* Covariances, Robustness, and Variational Bayes. 2018.

Related reading and extensions

- M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. **An Introduction to Variational Methods for Graphical Models.** 1999.
- D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. **Variational inference: A review for statisticians**. 2017.
- M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. **Stochastic variational inference**. 2013.
- R. Ranganath, S. Gerrish, and D. M. Blei. **Black Box Variational Inference**. 2013.
- Y. Yang, D. Pati, and A. Bhattacharya. $\alpha-$**variational inference with statistical guarantees**. 2017.
- Y. Gal and Z. Ghahramani. **Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning**. 2015.

Dropout as a Bayesian approximation

- Bayesian NNs can get intractable very easily
- Using dropout in your NN architecture is equivalent to a variational approximation
- Implementation is pretty straightforward. But first some basics.



(a) Standard Neural Net (b) After applying dropout.

Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.

Source: Srivastava *et al.* **Dropout: a simple way to prevent neural networks from overfitting**. 2014.

Dropout as Bayesian approximation

- How is dropout actually implemented in NNs?
    - Sample iid Bernoulli(p_i) variables for every input point in layer i
    - A unit is dropped if the Bernoulli variable takes value 0
- The dropout objective minimizes KL divergence between an approximate distribution and the posterior of a deep Gaussian process
- Predictive distribution moments:
    - Perform T *stochastic* forward passes through the network
    - Average the results - that's the first moment (and so on).