DD2434/FDD3434 Machine Learning, Advanced Course
Module 3 Exercise

November 2023

Contents

3 Variational Inference – Theory

A deterministic approximate inference algorithm.

3.1 Motivation

We can't compute the posterior for many interesting models. For example for the Bayesian mixture of Gaussian, we draw $z_i \sim Categorical(\pi)$ and $x_i \sim N(\mu_{zi}, \sigma^2)$ resulting in the following posterior [1]:

$$p(\mu_{1:K}, z_{1:n}|x_{1:n}) = \frac{\prod_k p(\mu_k) \prod_i p(z_i)p(x_i|z_i, \mu_{1:K})}{\int_{\mu_{1:K}} \sum_{z_{1:n}} \prod_k p(\mu_k) \prod_i p(z_i)p(x_i|z_i, \mu_{1:K})} \tag{1}$$

The denominator is the problem; even if we take the summation outside, this is intractable when n is reasonably large.

3.2 Main idea

In DD1420, the EM-algorithm is used for point estimate for the model parameters; now we want to find the posterior distribution for the **unknown** model parameters and hidden variables. For a DGM with observations X, hidden variables Z and model parameters Θ, want to pick an approximation $q(Z, \Theta)$ to the distribution from some tractable family, and then to try to make this approximation as close as possible to the true posterior.

$$p(Z, \theta|X) \approx q(Z, \theta) \tag{2}$$

This reduces inference to an optimization problem [2]. We measure the closeness of the two distributions q and p with Kullback-Leibler (KL) divergence.

$$\mathbb{KL}(q||p) = \sum_{Z,\Theta} q(Z, \Theta) log \frac{q(Z, \Theta)}{p(Z, \Theta|X)} \tag{3}$$

If we call the set of latent variables and parameters, Ψ, we can rewrite Eq. 3 as:

$$\sum_{\Psi} q(\Psi) log \frac{q(\Psi)}{p(\Psi|X)} = -E_\Psi\big[\log p(X, \Psi)\big] + E_\Psi\big[\log q(\Psi)\big] + \log p(X) \tag{4}$$

We cannot actually minimize KL divergence in Eq. 3 but since we have Eq. 4, we maximize lower bound of log marginal likelihood, the Evidence Lower Bound (ELBO):

$$\text{ELBO}(q) = E_\Psi\big[\log P(X, \Psi)\big] - E_\Psi\big[\log q(\Psi)\big]. \tag{5}$$

Note that we can use ELBO for convergence test at each iteration i.e. the difference of the ELBO of current value and the previous one should be smaller than some small epsilon.

In mean field Variational Inference, we assume that the variational family factorizes,

$$q(Z_1, ..., Z_n, \Theta_1, ..., \Theta_K) = \prod_i q(Z_i) \prod_k q(\Theta_k) \tag{6}$$

Now for each update equation of $q(z_i)$ we perform the expectation, over the log of the joint distribution, w.r.t. all the hidden variables except the one we are deriving the approximate

posterior for. For example, we obtain the update equation for z_j, $q(z_j)$, by calculating $E_{-z_j}\big(\log P(X, Z, \Theta)\big)$. We call this the coordinate ascent VI (CAVI) update equation, i.e:

$$\log q^*(\Psi_k) \stackrel{+}{=} E_{-\Psi_k}[\log P(\Psi, X)] \tag{7}$$

Algorithm 1 shows how (7) is used in the CAVI algorithm. To monitor convergence we can use the ELBO.

Convergence condition: not increasing the ELBO i.e. Eq. 5, or, reaching the max number of iterations. Note that to guarantee that the ELBO increases even after each update equation a proper coordinate ascend should be implemented. That is, the updates should be *sequentially*. For example, if ψ_1 is used in the calculation of ψ_2 then in the update of $q(\psi_2)$ the latest ψ_1 (hopefully from the current iteration) should be used. If the mean field VI is implemented by a *parallel* version of coordinate ascent it means that each $q(\psi_i)$ uses the variables values from the previous iteration.

Algorithm 1 Coordinate ascent VI

procedure APPROXIMATE POSTERIOR(X)
Initialise $q(\Psi_i)$ for $i = 1, \ldots, l$
  repeat
    for $i = 1, \ldots, l$ **do**
      Update: $\log q^*(\Psi_i) \propto E_{-\Psi_i}[\log P(\Psi, X)]$
  until convergence
  Approximate posterior: $q(\Psi) = \prod_i q^*(\Psi_i)$
return $q(\Psi)$

3.3 Alternative proof of CAVI update equation (Optional)

In the video lectures, Jens proves how (7) is derived using an argument with KL-divergence. Here we present an alternative proof, where we rewrite the ELBO and then take the partial derivative and set it to zero.

If we use the chain rule, we can write $p(X, \Psi) = p(X) \prod_j p(\Psi_j | \Psi_{1:j-1}, X)$. Also because of the independence we have: $\log q(\Psi) = \log\big(q(\Psi_1)...q(\Psi_l)\big) = \sum_j \log q(\Psi_j)$. Now the ELBO becomes the following. Note that in the last expectation term in Eq. 11, the expectation w.r.t. Ψ can be reduced to expectation w.r.t. Ψ_j since there is only one variable, i.e. Ψ_j, in the brackets, i.e. $\log q(\Psi_j)$.

$$\text{ELBO}(q) = \log P(X) + \sum_j \Big(E_\Psi\big[\log P(\Psi_j | \Psi_{1:j-1}, X)\big] - E_{\Psi_j}\big[\log q(\Psi_j)\big]\Big) \tag{8}$$

Now if you write the ELBO as a function of the last variable in the chain, say Ψ_k, you get:

$$\mathcal{L}(q(\Psi_k)) = const + E_\Psi\big[\log P(\Psi_k | \Psi_{-k}, X)\big] - E_{\Psi_k}\big[\log q(\Psi_k)\big] = \tag{9}$$

$$\int q(\Psi_k) E_{-\Psi_k}[\log P(\Psi_k | \Psi_{-k}, X)] d\Psi_k - \int q(\Psi_k) \log q(\Psi_k) d\Psi_k \tag{10}$$

Taking the partial derivative of the integrals in 10 and setting it to zero (and using lagrange multiplier) you get the update equation for each posterior during the coordinate ascent algorithm:

$$\log q^*(\Psi_k) \stackrel{+}{=} E_{-\Psi_k}[\log P(\Psi_k|\Psi_{-k}, X)] = E_{-\Psi_k}[\log \underbrace{\frac{P(\Psi_k, \Psi_{-k}, X)}{P(\Psi_{-k}, X)}}_{\text{see } **}] \stackrel{+}{=} E_{-\Psi_k}[\log P(\Psi, X)] \tag{11}$$

** The term $P(\Psi_{-k}, X)$ is constant w.r.t. Ψ_k.

3.3.1 Proof continued (Optional part)

We now prove the result of the partial derivative set to zero. To maximize the objective function $\mathcal{L}$ with constraint, as shown below, we need to find the *stationary function* accounting for the lagrangian term. That is solved by solving the Euler–Lagrange equation which includes the constraint, i.e., $\frac{\partial \mathcal{L} - \lambda G}{\partial q(\Psi_k)} - \frac{d}{d\Psi_k}\Big[\frac{\partial \mathcal{L} - \lambda G}{\partial q'(\Psi_k)}\Big] = 0$; G is the constraint.

$$\begin{aligned} &\mathcal{L}(q(\Psi_k)) = \\ &\int q(\Psi_k) E_{-\Psi_k}[\log P(\Psi_k|\Psi_{-k}, X)] d\Psi_k - \int q(\Psi_k) \log q(\Psi_k) d\Psi_k \\ &s.t. \quad \int q(\Psi_k) d\Psi_k = 1 \quad \forall k \end{aligned} \tag{12}$$

Opening up the Euler–Lagrange equation with constraint, we achieve:

$$\frac{\partial \mathcal{L}}{\partial q(\Psi_k)} - \frac{d}{d\Psi_k} \cancelto{0}{\frac{\partial \mathcal{L}}{\partial q'(\Psi_k)}} - \lambda_k \Big[\cancelto{1}{\frac{\partial \int q(\Psi_k) d\Psi_k}{\partial q(\Psi_k)}} - \frac{d}{d\Psi_k} \Big(\cancelto{0}{\frac{\partial \int q(\Psi_k) d\Psi_k}{\partial q'(\Psi_k)}} \Big) \Big] \tag{13}$$

That is, to calculate $\frac{\partial \mathcal{L}}{\partial q(\Psi_k)} - \lambda_k = 0$. The zero cancellations above are due to $q'(\Psi_k)$ not appearing explicitly in the objective.

$$
\begin{aligned}
&1. \quad \frac{\partial \mathcal{L}}{\partial q(\Psi_k)} - \lambda_k = 0 \implies \\
&E_{-\Psi_k}[\log P(\Psi_k|\Psi_{-k}, X)] - \Big(\log q(\Psi_k) + \frac{q(\Psi_k)}{q(\Psi_k)}\Big) - \lambda_k = 0 \\
&\implies \log q(\Psi_k) = \Big[E_{-\Psi_k}[\log P(\Psi_k|\Psi_{-k}, X)] - 1 - \lambda_k\Big] \\
&\implies \lambda_k = E_{-\Psi_k}[\log P(\Psi_k|\Psi_{-k}, X)] - 1 - \log q(\Psi_k) \\
&2. \quad \partial\mathcal{L}/\partial\lambda_k = 0 \implies \int q(\Psi_k) d\Psi_k = 1 \\
&\downarrow replace \quad q(\Psi_k) \\
&\int e^{\Big[E_{-\Psi_k}[\log P(\Psi_k|\Psi_{-k}, X)] - 1 - \lambda_k\Big]} d\Psi_k = 1 \\
&\frac{1}{e^{\lambda_k}} \int e^{\Big[E_{-\Psi_k}[\log P(\Psi_k|\Psi_{-k}, X)] - 1\Big]} d\Psi_k = 1 \\
&e^{\lambda_k + 1} = \int e^{\Big[E_{-\Psi_k}[\log P(\Psi_k|\Psi_{-k}, X)]\Big]} d\Psi_k \\
&\downarrow from \quad 1 \quad and \quad 2 \\
&q(\Psi_k) = \frac{e^{E_{-\Psi_k}[\log P(\Psi_k|\Psi_{-k}, X)]}}{e^{1+\lambda_k}} = \frac{e^{E_{-\Psi_k}[\log P(\Psi_k|\Psi_{-k}, X)]}}{\int e^{\Big[E_{-\Psi_k}[\log P(\Psi_k|\Psi_{-k}, X)]\Big]} d\Psi_k}
\end{aligned}
\tag{14}
$$

We can see that the denominator is the normalizing constant for the distribution $q(\Psi_k)$. Therefore, we can formulate the log distribution as $\log q(\Psi_k) \propto E_{-\Psi_k}[\log P(\Psi_k|\Psi_{-k}, X)]$.

Algorithm 1, summarizes the VI considering mean field (factorization of the latent variables). Similar to EM, it is crucial to initialize the algorithm, i.e., the better initialization, the better local optima.

4 Variational Inference – Exercises

4.1 Beta and Binomial model

Let $X = (X_1, .., X_N)$ be i.i.d. where $X_n|m, \theta \sim Binomial(m, \theta)$ and $\theta \sim Beta(\alpha, \beta)$.
a) Derive the CAVI updates for $q(\theta)$ using equation 7.
b) How does this compare to the posterior in exercise 1.1 of Module 1? Describe qualitatively in one sentence why this is the case.

4.2 Gaussian Mixture Model - light

Here we will examine an simpler version of the Gaussian Mixture model. Still $p(X_n|Z_n = k, \mu_k, \tau_k) = Normal(\mu_k, \frac{1}{\tau_k})$, $p(Z_n|\pi) = Categorical(\pi)$, but we assume π and τ_k are given and let $p(\mu_k) = Normal(\nu_k, \sigma_k)$.
a) Write the DGM/Bayes net for the model.
b) Write out $\log p(X, Z, \mu)$.
c) Apply and state the mean-field approximation for Z and μ.
d) Derive the associated CAVI updates using 7.
e) Implement the CAVI algorithm 1 and apply it to simulated data using the generative model (If you are unfamiliar with this, it will be shown in the Exercise session of module 3). Try simulating data for different K, N, ν_k, τ and π - under what circumstances does it have trouble finding all clusters?
f) In how many iterations does it converge?

4.3 Mixture Model with Bernoulli observations

In the video lectures the CAVI updates for a Mixture model with Gaussian observational model is introduced, i.e., $p(X_n|Z_n = k, \mu_k, \tau_k) = Normal(\mu_k, \frac{1}{\tau_k})$, $p(Z_n|\pi) = Categorical(\pi)$, $p(\pi) = Dirichlet(\alpha)$.
In this exercise we examine a similar model, but with $X_n = \{X_{n1}, ..., X_{nD}\}$ with observational model $p(X_{nd}|Z_n = k, \theta_k) = Bernoulli(\theta_k)$ with prior $p(\theta_k) = Beta(a, b)$.
a) Write the DGM/Bayes net for the model.
b) Write out $\log p(X, Z, \pi, \theta)$.
c) Apply and state the mean-field approximation for Z, π and θ.
d) Derive the associated CAVI updates using 7.
e) Implement the CAVI algorithm 1 and apply it to simulated data using the generative model (If you are unfamiliar with this, it will be shown in the Exercise session of module 3). Try simulating data for different K, N, θ_k and π - under what circumstances does it have trouble finding all clusters?
f) In how many iterations does it converge?

4.4 Cartesian Matrix Model (from assignment 1B, 2017)

The Cartesian Matrix Model (CMM) is defined as follows. There are R row distributions $\{N(\mu_r, \lambda_r^{-1}) : 1 \leq r \leq R\}$, each variance λ_r^{-1} is known and each μ_r has prior distribution $N(\mu, \lambda^{-1})$. There are also C column distributions $\{N(\xi_r, \tau_c^{-1}) : 1 \leq c \leq C\}$, each variance τ_c^{-1} is known and each ξ_c has prior distribution $N(\xi, \tau^{-1})$. All hyper-parameters are known. A matrix S is generated by, for each row $1 \leq r \leq R$ and each column $1 \leq c \leq C$, setting $S_{rc} = X_r + Y_c$ where X_r is sampled from $N(\mu_r, \lambda_r^{-1})$ and Y_c from $N(\xi_r, \tau_c^{-1})$. Use Variational Inference in order to obtained a variational distribution

$$q(\mu_1, \ldots, \mu_R, \xi_1, \ldots, \xi_C) = \prod_r q(\mu_r) \prod_c q(\xi_c)$$

that approximates $p(\mu_1, \ldots, \mu_R, \xi_1, \ldots, \xi_C | S)$. Tip: what distribution do you get from the sum of two Gaussian random variables? What is the relation between the means?

Question 15: *Present the algorithm written down in a formal manner (using both text and mathematical notation, but not pseudo code).*

Figure 1: From Assignment 2, 2017

4.5 Troll factories (from assignment, 1B 2022)

On a social media platform, K troll factories have posted N comments on a live news report from an ongoing war. A security agency wants to extract information on the troll factories, but due to integrity protection policies, the platform can only provide metadata of the posts, such as comment length X_n of each post as well as response time T_n. Together with the security agency's disinformation team, the newly employed ML expert develops a model which infers comment to factory assignment, Z_n, factory post volume fraction, π, troll factory specific response rate, λ_k, and average comment length, μ_k, and precision τ_k with the following distributions:

- $X_n | \mu_k, \tau_k, Z_n = k \sim Lognormal(\mu_k, \tau_k^{-1})$ - based on the assumptions that each troll factory has it's own strategy for comment length and variation in length and that comments are always of positive length.
- $\mu_k, \tau_k | \nu, \kappa, \alpha, \beta \sim NormalGamma(\nu, \kappa, \alpha, \beta)$
- $T_n | \lambda_k, Z_n = k \sim Exp(\lambda_k)$ - Comments are written as reactions to events with a factory specific response rate.
- $\lambda_k | a, b \sim Gamma(a, b)$ - The factory specific response rate is unkown, but the domain experts provide reasonable values for a and b.
- $Z_n | \pi \sim Categorical(\pi)$ - Each post is associated with K different factories.
- $\pi | \delta \sim Dirichlet(\delta)$

Note that comment length is a discrete entity, but we approximate the likelihood of observations with a continuous distribution in the model.

a) Provide a graphical model for the model described above.
b) Derive the CAVI update equations of each variational distribution.