Homework 5
M1522.001300 Probabilistic Graphical Models (2024 Spring)

Due: June 27 Monday 11:59PM.

READ CAREFULLY

There are problems that require programming. Do not attach your code to the writeup. Put your **writeup** and **code** into a directory called "`(studentid)-(yourname)-HW5`" and tar it into a `tar.gz` named "`(studentid)-(yourname)-HW5`".

For example, `202012345-gildonghong-HW5.tar.gz`.

Your homework should be formatted as the following:

```
(studentid)-(yourname)-HW5
|
+- writeup.pdf
|
+- data
|   |
|   +- 20newsgroup
|   |   |
|   |   +- test.data
|   |   |
|   |   +- test.label
|   |   |
|   |   +- test.map
|   |   |
|   |   +- train.data
|   |   |
|   |   +- train.label
|   |   |
|   |   +- train.map
|   |   |
|   |   +- vocabulary.txt
|   |
|   +- 2d_gauss.txt
|
+- code
    |
    +- README.txt
    |
    +- your-lda-implementation.py
```

Upload your `tar.gz` file to ETL before the due date. You can discuss the problem with others, but you have to provide your own answer.

If your submission does not follow the above format, you will receive penalty.

1 Factorized Distributions [20 points]

1. [5 pts] Show that the log marginal distribution of the observed data $\ln p(\mathbf{X})$ can be decomposed into two terms $\mathcal{L}(q)$ and $KL(q||p)$.

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} d\mathbf{Z} \tag{1}$$

$$KL(q||p) = - \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right\} d\mathbf{Z} \tag{2}$$

2. [7 pts] Let us assume a factorized variational distribution $q(\mathbf{Z})$ as

$$q(\mathbf{Z}) = \prod_{i=1}^{M} q(\mathbf{Z}_i). \tag{3}$$

Use the method of Lagrange multipliers and prove that minimization of the Kullback-Leibler divergence $KL(p||q)$ with respect to one of the factors $q_i(\mathbf{Z}_i)$, keeping all other factors fixed, leads to

$$q_j(\mathbf{Z}_j) = \int p(\mathbf{Z}) \prod_{i \neq j} d\mathbf{Z}_i = p(\mathbf{Z}_j). \tag{4}$$

3. [8 pts] Assume a model where the set of all hidden (stochastic) variables are denoted as $\mathbf{Z}$. It includes some latent variables $\mathbf{z}$ and the model parameters θ. We will use a variational distribution that factorizes with latent variables and model parameters as the following: $q(\mathbf{z}, \theta) = q_{\mathbf{z}}(\mathbf{z}) q_\theta(\theta)$. The distribution $q_\theta(\theta)$ will be approximated by a point estimate of the form $q_\theta(\theta) = \delta(\theta - \theta_0)$ where θ_0 is a vector of free parameters. Verify that the variational optimization of this factorized distribution is equivalent to an EM algorithm, where the E-step optimizes $q_{\mathbf{z}}(\mathbf{z})$, and the M-step maximizes the expected complete-data log posterior distribution of θ with respect to θ_0.

2 Rejection Sampling [20 points]

1. [10 pts] In this exercise, we show more carefully that rejection sampling does indeed draw samples from the desired distribution $p(z)$. Suppose the proposal distribution is $q(z)$ and show that the probability of a sample value z being accepted is given by $\widetilde{p}(z)/kq(z)$ where $\widetilde{p}$ is any unnormalized distribution that is proportional to $p(z)$, and the constant k is set to the smallest value that ensures $kq(z) \geqslant p(z)$ for all values of z. Note that the probability of drawing a value z is given by the probability of drawing that value from $q(z)$ times the probability of accepting that value given that it has been drawn. Make use of this, along with the sum and product rules of probability, to write down the normalized form for the distribution over z, and show that it equals $p(z)$.

2. [10 pts] Write a rejection sampling algorithm to uniformly generate points within a sphere of radius r, centered at the origin of a three-dimensional coordinate system. Assume you have the capability to uniformly sample from any closed interval $[a, b]$. (1) Describe the algorithm in detail and (2) calculate the rejection ratio of the algorithm.

3 Gibbs Sampling [20 points]

1. [10 pts] For a target density $p(x)$ and proposal density $q(x' \leftarrow x)$, the Metropolis–Hastings transition operator is given by

$$T(x' \leftarrow x) = q(x' \leftarrow x) \min\left\{1, \frac{\widetilde{p}(x')\, q(x \leftarrow x')}{\widetilde{p}(x)\, q(x' \leftarrow x)}\right\}$$

where $\widetilde{p}(x)$ is the unnormalized target density.

Show that this transition operator satisfies detailed balance.

2. [10 pts] Show that Gibbs sampling is a special case of the Metropolis-Hastings algorithm. More precisely, provide a particular proposal distribution Q_i for each local transition T^{Q_i} that induces precisely the same distribution over the transitions taken as the associated Gibbs transition distribution T_i.

4 Variational Inference in Latent Dirichlet Allocation (LDA) [40 points]

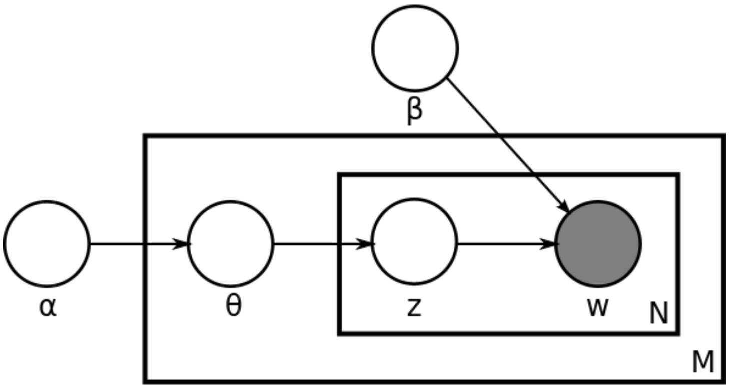

Figure 1: The LDA graphical model

The LDA graphical model (Figure 1) was discussed in class. The most popular use for LDA is in modeling a document collection by topics.

You will implement the Latent Dirichlet Allocation using variational inference. The dataset is the Newsgroup 20Newsgroup corpora. You can use the preprocessed data 20news-bydate-matlab.tgz, which is in the `data` directory. If you hope to use a different version, submit your preprocessed data with your code. When submitting your assignment, set the path correctly and make it self-executable. You will need to do some preprocessing to turn things into word counts. Removing punctuation, lowercasing, and tokenizing the text is recommended. You may remove stop-words and very rare words. You need to implement it in the `code` directory as self-executable codes and add a `README` file there.

Provide analysis and discussion of your implementation and results.

For example:

- What were the most common topics?
- What were the most common words across each topic?
- If you had metadata, how did this data relate to the topics?
- How many topics did you use and how did you arrive at this choice?
- What preprocessing did you carry out?
- How many iterations did you run and how long did this take?
- Did the implementations find interestingly different solutions?
- How did you set the hyperparameters?
- Did you try alternative values?
- Did you do any quantitative analysis of your model?

Plots are welcome when showing your results!

This is the final assignment of this semester. We're glad you made it :D