Machine Learning Homework 2

Instructor: Prof. Jun Zhu & Prof. Jie Tang

October 30, 2024

Requirements:

- We recommend that you typeset your homework using appropriate software such as LaTeX. If you submit your handwritten version, please make sure it is cleanly written up and legible. The TAs will not invest undue effort to decrypt bad handwriting.
- We have programming tasks in each homework. Please submit the source code along with your homework. Please include experiment results using figures or tables in your homework, instead of asking TAs to run your code.
- There are optional problems in the assignments. We will give bonus points to those who succeed in solving these problems.
- You can use any modern language model, like ChatGPT, ChatGLM, and so on, to help you complete this assignment. If you have, please attach your prompt and the results of the language model to the assignment.
- Please finish your homework independently. In addition, you should write in your homework the set of people with whom you collaborated.
- If you have any questions, please contact me via *ycy21 [AT] mails [DOT] tsinghua [DOT] edu [DOT] cn*.

1 EM for mixture of multinomials (4pts)

Recall a multinomial distribution with the parameter $\boldsymbol{\mu} = (\mu_i)_{i=1}^d$:

$$P(\boldsymbol{x} \mid \boldsymbol{\mu}) = \frac{n!}{\prod_i x_i!} \prod_i \mu_i^{x_i}, \quad i = 1, \cdots, d \tag{1.1}$$

where $x_i \in \mathbb{N}$, $\sum_i x_i = n$, and $0 < \mu_i < 1$, $\sum_i \mu_i = 1$.

Consider the following mixture-of-multinomials model to analyze a corpus of documents that are represented in the bag-of-words model.

Specifically, assume we have a corpus of D documents and a vocabulary of W words from which every word in the corpus is token. We are interested in counting how many times each word appears in each document, regardless of their positions and orderings. We denote by $T \in \mathbb{N}^{D \times W}$ the word occurrence matrix where the w-th word appears T_{dw} times in the d-th document. According to the mixture-of-multinomials model, each document is generated i.i.d. as follows. We first choose for each document d a latent "topic" c_d (analogous to choosing for each data point a component z_n in the mixture-of- Gaussians) with

$$P\left(c_d = k\right) = \pi_k, k = 1, 2, \cdots, K; \tag{1.2}$$

And then given this "topic" $\boldsymbol{\mu}_k = (\mu_{1k}, \ldots, \mu_{Wk})$ which now simply represents a categorical distribution over the entire vocabulary, we generate the word bag of the document from the corresponding multinomial distribution [1]

$$P\left(d \mid c_d = k\right) = \frac{n_d!}{\prod_w T_{dw}!} \prod_w \mu_{wk}^{T_{dw}}, \tag{1.3}$$

[1] Make sure you understand the difference between a categorical distribution and a multinomial distribution. You may think about a Bernoulli distribution and a binomial distribution for reference.

where $n_d = \sum_w T_{dw}$. Hence in summary

$$P(d) = \sum_{k=1}^{K} P(d \mid c_d = k) P(c_d = k) = \frac{n_d!}{\prod_w T_{dw}!} \sum_{k=1}^{K} \pi_k \prod_w \mu_{wk}^{T_{dw}}. \tag{1.4}$$

Problem 1 (2pts)**.** Given the corpus T, design and derive an EM algorithm to learn the parameters $\{\pi, \mu\}$ of this mixture model.

Problem 2 (2pts)**.** Implement the EM algorithm on the Newsgroups dataset. *[Hint: When implementing EM algorithm, you may encounter the problem of overflow or numerical instability. You can use off-the-shelf functions for scientific computations like* `scipy.special.softmax` *and* `scipy.special.logsumexp` *to avoid this.]*

Set the number of topics K to be 10, 20, 30, 50 respectively and show the most-frequent words in each topic for each case.

Problem 3 (Bonus, 1pts)**.** Observe the result and try to find the "best" K value for this dataset and analyze this reason.

2 Minimum Error Formulation of PCA (2pts)

We have a set of data points $\{\mathbf{x}_n\}, n = 1, ..., N$, and each $\mathbf{x}_n \in \mathbb{R}^p$. Also, we have a set of complete orthogonal basis $\mu_i, i = 1, ..., p$, where $\mu_i^T \mu_j = \delta_{ij}$. To do dimension reduction, we want to approximate $\mathbf{x}_n$ by

$$\tilde{\mathbf{x}}_n = \sum_{i=1}^{d} z_{ni}\mu_i + \sum_{i=d+1}^{p} b_i\mu_i,$$

and the objective function is the approximation error:

$$J = \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2.$$

Problem 4 (2pts)**.** You need to prove that optimal solutions of z_{ni} and b_i which minimize J are as follows ($\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_x$):

$$z_{ni} = \mathbf{x}_n^T \mu_i, i = 1, 2, ..., d, n = 1, 2, ..., N,$$
$$b_i = \bar{\mathbf{x}}^T \mu_i, i = d + 1, ..., p.$$

3 Deep Generative Model (4pts)

Problem 5 (Gaussian VAE vs Bernoulli VAE, 3pts)**.** Consider two types of implicit generative models (aka the"decoder") in Variational Auto-Encoders (VAEs). The first one defines $p(\mathbf{x}|\mathbf{z})$ as a Bernoulli distribution:

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \mathbf{x}|\mathbf{z} \sim \text{Bernoulli}(\mu_\theta(\mathbf{z})).$$

The second one defines $p(\mathbf{x}|\mathbf{z})$ as a Gaussian distribution:

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \mathbf{x}|\mathbf{z} \sim \mathcal{N}(\mu_\theta(\mathbf{z}), \sigma^2\mathbf{I}).$$

For this problem, the dimension of the latent variable $\mathbf{z}$ is fixed at 40. We will use the MNIST dataset[2] and choose a Multilayer Perceptron (MLP) as the function $\mu_\theta(\mathbf{z})$.

1. Implement a Bernoulli VAE and plot samples generated by the Bernoulli VAE. *[Hint: the data should be binarized before training or testing.]*

2. Implement a Gaussian VAE, report your selected σ and plot samples generated by the Gaussian VAE. You can try it under different σ.

3. Compare the sample quality between Bernoulli VAE and Gaussian VAE and analyze the reason. (There is no standard answer; feel free to express your thoughts.)

[2] http://yann.lecun.com/exdb/mnist/