

# CS 7641 CSE/ISYE 6740 Homework 2

Le Song

Deadline: 10/17 Thursday, 11:55 **am**

- **Please read the following submission rules carefully. We will strictly follow these rules starting from this assignment.**
- No unapproved extension of deadline is allowed. Late submission window will open immediately after the deadline for 12 hours, but with a  $-40$  points penalty. No submission via email will be accepted.
- We recommend writing your report with Latex. Other text editors such as MS Word are also accepted. Hand-written report must be clear. Unreadable handwriting is subject to zero credit. Report must be **a single pdf** file named with the format **HW $\langle number \rangle$ - $\langle your\_GT\_account \rangle$ .pdf**. For example, “HW2\_yyang123.pdf”. Any additional document is ignored.
- For programming assignments, please strictly follow the provided function APIs, as we will be grading your code by running the homework2.m/homework2.py file with required arguments. You will be responsible for any runtime error due to the change of the API or code file name. You should implement your programs in the given .m/.py file. Any code pasted to the report or other files will not be graded.
- Matlab submissions will be graded in Matlab Online R2019b environment<sup>12</sup>. Python submission will be graded in Python 3.6 environment on a regular laptop. Please make sure your program is runnable in those environments. Any program that cannot run is subject to zero credit. Situations include but not restricted to exceeding 5 minutes of running time, exceeding memory limit and invalid text character.
- Put your report and code files into a single flat folder and **submit it as a single .zip file**.
- Explicitly mention your collaborators if any.
- Recommended reading: PRML<sup>3</sup> Section 1.5, 1.6, 2.5, 9.2, 9.3

## 1 EM for Mixture of Gaussians

Mixture of  $K$  Gaussians is represented as

$$p(x) = \sum_{k=1}^K \pi_k \mathcal{N}(x | \mu_k, \Sigma_k), \quad (1)$$

where  $\pi_k$  represents the probability that a data point belongs to the  $k$ th component. As it is probability, it satisfies  $0 \leq \pi_k \leq 1$  and  $\sum_k \pi_k = 1$ . In this problem, we are going to represent this in a slightly different manner with explicit latent variables. Specifically, we introduce 1-of- $K$  coding representation for latent

---

<sup>1</sup><https://matlab.mathworks.com/>

<sup>2</sup>You can also use the environment for developing by registering online with your university email.

<sup>3</sup>Christopher M. Bishop, Pattern Recognition and Machine Learning, 2006, Springer.

variables  $z^{(k)} \in \mathbb{R}^K$  for  $k = 1, \dots, K$ . Each  $z^{(k)}$  is a binary vector of size  $K$ , with 1 only in  $k$ th element and 0 in all others. That is,

$$\begin{aligned} z^{(1)} &= [1; 0; \dots; 0] \\ z^{(2)} &= [0; 1; \dots; 0] \\ &\vdots \\ z^{(K)} &= [0; 0; \dots; 1]. \end{aligned}$$

For example, if the second component generated data point  $x^n$ , its latent variable  $z^n$  is given by  $[0; 1; \dots; 0] = z^{(2)}$ . With this representation, we can express  $p(z)$  as

$$p(z) = \prod_{k=1}^K \pi_k^{z_k},$$

where  $z_k$  indicates  $k$ th element of vector  $z$ . Also,  $p(x|z)$  can be represented similarly as

$$p(x|z) = \prod_{k=1}^K \mathcal{N}(x|\mu_k, \Sigma_k)^{z_k}.$$

By the sum rule of probability, (1) can be represented by

$$p(x) = \sum_{z \in Z} p(z)p(x|z). \quad (2)$$

where  $Z = \{z^{(1)}, z^{(2)}, \dots, z^{(K)}\}$ .

**(a) Show that (2) is equivalent to (1). [5 pts]**

**(b) In reality, we do not know which component each data point is from. Thus, we estimate the responsibility (expectation of  $z_k^n$ ) in the E-step of EM. Since  $z_k^n$  is either 1 or 0, its expectation is the probability for the point  $x_n$  to belong to the component  $z_k$ . In other words, we estimate  $p(z_k^n|x_n)$ . Derive the formula for this estimation by using Bayes rule. Note that, in the E-step, we assume all other parameters, i.e.  $\pi_k$ ,  $\mu_k$ , and  $\Sigma_k$ , are fixed, and we want to express  $p(z_k^n|x_n)$  as a function of these fixed parameters. [10 pts]**

**(c) In the M-Step, we re-estimate parameters  $\pi_k$ ,  $\mu_k$ , and  $\Sigma_k$  by maximizing the log-likelihood. Given  $N$  i.i.d (Independent Identically Distributed) data samples, derive the update formula for each parameter. Note that in order to obtain an update rule for the M-step, we fix the responsibilities, i.e.  $p(z_k^n|x_n)$ , which we have already calculated in the E-step. [15 pts]**

*Hint:* Use Lagrange multiplier for  $\pi_k$  to apply constraints on it.

**(d) EM and K-Means [10 pts]**

K-means can be viewed as a particular limit of EM for Gaussian mixture. Considering a mixture model in which all components have covariance  $\epsilon I$ , show that in the limit  $\epsilon \rightarrow 0$ , maximizing the expected complete data log-likelihood for this model is equivalent to minimizing objective function in K-means:

$$J = \sum_{n=1}^N \sum_{k=1}^K \gamma_{nk} \|x_n - \mu_k\|^2,$$

where  $\gamma_{nk} = 1$  if  $x_n$  belongs to the  $k$ -th cluster and  $\gamma_{nk} = 0$  otherwise.

## 2 Density Estimation

Consider a histogram-like density model in which the space  $x$  is divided into fixed regions for which density  $p(x)$  takes constant value  $h_i$  over  $i$ th region, and that the volume of region  $i$  is denoted as  $\Delta_i$ . Suppose we have a set of  $N$  observations of  $x$  such that  $n_i$  of these observations fall in regions  $i$ .

(a) What is the log-likelihood function? [8 pts]

(b) Derive an expression for the maximum likelihood estimator for  $h_i$ . [10 pts]

*Hint:* This is a constrained optimization problem. Remember that  $p(x)$  must integrate to unity. Since  $p(x)$  has constant value  $h_i$  over region  $i$ , which has volume  $\Delta_i$ . The normalization constraint is  $\sum_i h_i \Delta_i = 1$ . Use Lagrange multiplier by adding  $\lambda(\sum_i h_i \Delta_i - 1)$  to your objective function.

(c) Mark  $T$  if it is always true, and  $F$  otherwise. Briefly explain why. [12 pts]

- Non-parametric density estimation usually does not have parameters.
- The Epanechnikov kernel is the optimal kernel function for all data.
- Histogram is an efficient way to estimate density for high-dimensional data.
- Parametric density estimation assumes the shape of probability density.

## 3 Information Theory

In the lecture you became familiar with the concept of entropy for one random variable and mutual information. For a pair of discrete random variables  $X$  and  $Y$  with the joint distribution  $p(x, y)$ , the *joint entropy*  $H(X, Y)$  is defined as

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y) \quad (3)$$

which can also be expressed as

$$H(X, Y) = -\mathbb{E}[\log p(X, Y)] \quad (4)$$

Let  $X$  and  $Y$  take on values  $x_1, x_2, \dots, x_r$  and  $y_1, y_2, \dots, y_s$  respectively. Let  $Z$  also be a discrete random variable and  $Z = X + Y$ .

(a) Prove that  $H(X, Y) \leq H(X) + H(Y)$  [4 pts]

(b) Show that  $I(X; Y) = H(X) + H(Y) - H(X, Y)$ . [2 pts]

(c) Under what conditions does  $H(Z) = H(X) + H(Y)$ . [4 pts]

## 4 Programming: Text Clustering

In this problem, we will explore the use of EM algorithm for text clustering. Text clustering is a technique for unsupervised document organization, information retrieval. We want to find how to group a set of different text documents based on their topics. First we will analyze a model to represent the data.

## Bag of Words

The simplest model for text documents is to understand them as a collection of words. To keep the model simple, we keep the collection unordered, disregarding grammar and word order. What we do is counting how often each word appears in each document and store the word counts into a matrix, where each row of the matrix represents one document. Each column of matrix represent a specific word from the document dictionary. Suppose we represent the set of  $n_d$  documents using a matrix of word counts like this:

$$D_{1:n_d} = \begin{pmatrix} 2 & 6 & \dots & 4 \\ 2 & 4 & \dots & 0 \\ \vdots & & \ddots & \end{pmatrix} = T$$

This means that word  $W_1$  occurs twice in document  $D_1$ . Word  $W_{n_w}$  occurs 4 times in document  $D_1$  and not at all in document  $D_2$ .

## Multinomial Distribution

The simplest distribution representing a text document is multinomial distribution (Bishop Chapter 2.2). The probability of a document  $D_i$  is:

$$p(D_i) = \prod_{j=1}^{n_w} \mu_j^{T_{ij}}$$

Here,  $\mu_j$  denotes the probability of a particular word in the text being equal to  $w_j$ ,  $T_{ij}$  is the count of the word in document. So the probability of document  $D_1$  would be  $p(D_1) = \mu_1^2 \cdot \mu_2^6 \cdot \dots \cdot \mu_{n_w}^4$ .

## Mixture of Multinomial Distributions

In order to do text clustering, we want to use a mixture of multinomial distributions, so that each topic has a particular multinomial distribution associated with it, and each document is a mixture of different topics. We define  $p(c) = \pi_c$  as the mixture coefficient of a document containing topic  $c$ , and each topic is modeled by a multinomial distribution  $p(D_i|c)$  with parameters  $\mu_{jc}$ , then we can write each document as a mixture over topics as

$$p(D_i) = \sum_{c=1}^{n_c} p(D_i|c)p(c) = \sum_{c=1}^{n_c} \pi_c \prod_{j=1}^{n_w} \mu_{jc}^{T_{ij}}$$

## EM for Mixture of Multinomials

In order to cluster a set of documents, we need to fit this mixture model to data. In this problem, the EM algorithm can be used for fitting mixture models. This will be a simple topic model for documents. Each topic is a multinomial distribution over words (a mixture component). EM algorithm for such a topic model, which consists of iterating the following steps:

### 1. Expectation

Compute the expectation of document  $D_i$  belonging to cluster  $c$ :

$$\gamma_{ic} = \frac{\pi_c \prod_{j=1}^{n_w} \mu_{jc}^{T_{ij}}}{\sum_{c=1}^{n_c} \pi_c \prod_{j=1}^{n_w} \mu_{jc}^{T_{ij}}}$$

## 2. Maximization

Update the mixture parameters, i.e. the probability of a word being  $W_j$  in cluster (topic)  $c$ , as well as prior probability of each cluster.

$$\mu_{jc} = \frac{\sum_{i=1}^{n_d} \gamma_{ic} T_{ij}}{\sum_{i=1}^{n_d} \sum_{l=1}^{m_w} \gamma_{ic} T_{il}}$$

$$\pi_c = \frac{1}{n_d} \sum_{i=1}^{n_d} \gamma_{ic}$$

### Task [20 pts]

Implement the algorithm and run on the toy dataset `data.mat`. You can find detailed description about the data in the `homework2.m` file. Observe the results and compare them with the provided true clusters each document belongs to. Report the evaluation (e.g. accuracy) of your implementation.

*Hint:* We already did the word counting for you, so the data file only contains a count matrix like the one shown above. For the toy dataset, set the number of clusters  $n_c = 4$ . You will need to initialize the parameters. Try several different random initial values for the probability of a word being  $W_j$  in topic  $c$ ,  $\mu_{jc}$ . Make sure you normalized it. Make sure that you should not use the true cluster information during your learning phase.

### Extra Credit: Realistic Topic Models [20pts]

The above model assumes all the words in a document belongs to some topic at the same time. However, in real world datasets, it is more likely that some words in the documents belong to one topic while other words belong to some other topics. For example, in a news report, some words may talk about “Ebola” and “health”, while others may mention “administration” and “congress”. In order to model this phenomenon, we should model each word as a mixture of possible topics.

Specifically, consider the log-likelihood of the joint distribution of document and words

$$\mathcal{L} = \sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{V}} T_{dw} \log P(d, w), \quad (5)$$

where  $T_{dw}$  is the counts of word  $w$  in the document  $d$ . This count matrix is provided as input.

The joint distribution of a specific document and a specific word is modeled as a mixture

$$P(d, w) = \sum_{z \in \mathcal{Z}} P(z) P(w|z) P(d|z), \quad (6)$$

where  $P(z)$  is the mixture proportion,  $P(w|z)$  is the distribution over the vocabulary for the  $z$ -th topic, and  $P(d|z)$  is the probability of the document for the  $z$ -th topic. And these are the parameters for the model.

The E-step calculates the posterior distribution of the latent variable conditioned on all other variables

$$P(z|d, w) = \frac{P(z) P(w|z) P(d|z)}{\sum_{z'} P(z') P(w|z') P(d|z')}. \quad (7)$$

In the M-step, we maximizes the expected complete log-likelihood with respect to the parameters, and get the following update rules

$$P(w|z) = \frac{\sum_d T_{dw} P(z|d, w)}{\sum_{w'} \sum_d T_{dw'} P(z|d, w')} \quad (8)$$

$$P(d|z) = \frac{\sum_w T_{dw} P(z|d, w)}{\sum_{d'} \sum_w T_{d'w} P(z|d', w)} \quad (9)$$

$$P(z) = \frac{\sum_d \sum_w T_{dw} P(z|d, w)}{\sum_{z'} \sum_{d'} \sum_w T_{d'w'} P(z'|d', w')}. \quad (10)$$

## Task

Implement EM for maximum likelihood estimation and cluster the text data provided in the `nips.mat` file you downloaded. You can print out the top key words for the topics/clusters by using the `show_topics.m` utility. It takes two parameters: 1) your learned conditional distribution matrix, i.e.,  $P(w|z)$  and 2) a cell array of words that corresponds to the vocabulary. You can find the cell array `wl` in the `nips.mat` file. Try different values of  $k$  and see which values produce sensible topics. In assessing your code, we will use another dataset and observe the produces topics.