CSE/ISyE 6740-A Homework 2

Name: TBD
GTID: TBD

Deadline: Sep 29th 2024 11:59pm ET

- There are 2 sections in grade scope: Homework 2 and Homework 2 Programming. Submit your answers as a PDF file to Homework 2 (including report for programming) and also submit your code in a zip file to Homework 2 Programming.
- Late homework incurs a penalty of 20% for each 24 hours that it is late. Thus, right after the deadline it will only be worth 80% credit and after five days it will not be worth any credit.
- We recommend the use of LaTeX for typing up your solutions. No credit will be given to unreadable handwriting.
- List explicitly with whom in the class you discussed which problem, if any. Cite all external resources that you were using to complete the homework. For details, consult the collaboration policy in the class syllabus on Canvas.
- Recommended reading: PRML[1] Section 9.2, 9.3, 9.4, 12.1

1 PCA [10 pts]

Suppose we are given a set of points $x_1, \cdots, x_n$. Let us assume that we have as usual preprocessed the data to have zero-mean and unit variance in each coordinate. For a given unit-length vector v, let $f_v(x)$ be the projection of point x onto the direction given by v. I.e., if $\mathcal{V} = \{\alpha v : \alpha \in \mathbb{R}\}$, then

$$f_v(x) = \arg\min_{u \in \mathcal{V}} \|x - u\|^2$$

Show that the unit-length vector v that minimizes the mean squared error between projected points and original points corresponds to the first principal component for the data. I.e., show that

$$\arg\min_{\|v\|} \sum_{i=1}^{n} \|x_i - f_v(x_i)\|^2$$

gives the principle component.

[1] Christopher M. Bishop, Pattern Recognition and Machine Learning, 2006, Springer.

2 Density Estimation [15 pts]

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 in denoted as Δ_i (e.g. $\sum_i h_i \Delta_i = 1$). 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? [5 pts]

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

3 Bernoulli Mixture Model [20 pts]

In class, you have learned the Gaussian Mixture Model. For some discrete-valued problems, like binary images, the Bernoulli Mixture Model (BMM) is a good choice. In this model, we also have K components and parameters $\{\pi, \theta\}$. Each component i with prior π_i has a D-dimensional Bernoulli probability function parameterized by $\theta_i = [\theta_{i1}, \theta_{i2}, \ldots, \theta_{iD}]^T \in [0,1]^D$. Suppose we have N i.i.d samples $x_1, x_2, \ldots, x_N$, where each example is a D dimentional vector. Given the component i, the likelihood of observing an instance $x \in \{0,1\}^D$ is

$$P(x|i) = \prod_{d=1}^{D} \theta_{id}^{x_d} (1 - \theta_{id})^{(1-x_d)}$$

(a) Write down the log-likelihood $L(\pi, \theta)$ for N observations using BMM. If we use EM algorithm to find MLE, what are the latent variables? [4 pts]

(b) Given π and θ, derive the lower bound of your log-likelihood, and write down the update rule for E-step. (Hint: use Jensens inequality) [8 pts]

(c) Write down the M-step which maximizes your lower bound written above. To receive full credits, you should provide the answer step by step. [8 pts]

4 Feature Selection (Information Theory) [15 pts]

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

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

which can also be expressed as $H(X, Y) = -\mathbb{E}[\log p(X, Y)]$. Let X and Y take on values $x_1, x_2, \ldots, x_r$ and $y_1, y_2, \ldots, y_s$ respectively.

(a) Prove that $H(X, Y) \leq H(X) + H(Y)$. You can use the inequality: $\ln x \leq x - 1$ when $x > 0$. [10 pts]

(b) Let Z be a discrete random variable defined as $Z = X + Y$. Is the independence between X and Y a necessary condition, a sufficient condition, or both for the equality $H(Z) = H(X) + H(Y)$ to hold true? Justify your answer. [5 pts]

5 Naive Bayes [10 pts]

Let $X = \langle X_1, X_2, \ldots, X_n \rangle$ be a vector of n binary values where the random variable X_i denotes the i^{th} attribute of X. Suppose we are interested in estimating the parameters for the first attribute X_1. We denote $I\left(Y^j = y_k\right) = 1$, if $Y^j = y_k$; otherwise $I\left(Y^j = y_k\right) = 0$.

(a) Now suppose each X_i is distributed normally, i.e.,

$$P\left(X_i = x_{ij} \mid Y = y_k\right) = \frac{1}{\sqrt{2\pi}\sigma_{ik}} \exp\left(\frac{-\left(x_{ij} - \mu_{ik}\right)^2}{2\sigma_{ik}^2}\right).$$

Suppose the variance is independent of the class variable Y and X_i, i.e., $\sigma_{ik} = \sigma$. Derive the MLE estimator for μ_{ik}. [5 pts]

(b) If we model $P\left(X_1 \mid Y = y_k\right)$ with a Bernoulli distribution:

$$P\left(X_1 = x_{1j} \mid Y = y_k\right) = \theta_{1k}^{x_{1j}} \left(1 - \theta_{1k}\right)^{(1 - x_{1j})},$$

where $j = 1, \ldots, M$ refers to the j^{th} training instance (where M is the number of training samples), and where x_{1j} refers to the value of X_1 in the j^{th} training instance. Assume that the M training instances are independent and identically distributed (iid). Write down the MLE for $\hat{\theta}_{1k}$. [5 pts]

6 Logistic Regression [10 pts]

Logistic regression is named after the log-odds of success (the logit of the probability) defined as below:

$$\ln\left(\frac{P[Y = 1 \mid X = x]}{P[Y = 0 \mid X = x]}\right),$$

where

$$P[Y = 1 \mid X = x] = \frac{\exp\left(w_0 + w^T x\right)}{1 + \exp\left(w_0 + w^T x\right)}$$

(a) Show that log-odds of success is a linear function of X. [5 pts]

(b) Show that the logistic loss $\ell(z) = \log(1 + \exp(-z))$ is a convex function. [5 pts]

7 Programming: Text Clustering [20 pts + 10 pts bonus]

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:

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

Here, μ_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 $p\left(D_1\right) = \mu_1^2 \cdot \mu_2^6 \cdot \ldots \cdot \mu_{n_w}^4$.

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 $p\left(D_i \mid c\right)$ with parameters μ_{jc}, then we can write each document as a mixture over topics as

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

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.ipynb 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, μ_{jc}. Make sure you normalized it. Make sure that you should not use the true cluster information during your learning phase.

Bonus: Realistic Topic Models [10 pts]

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

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

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

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

where $P(z)$ is the mixture proportion, $P(w \mid z)$ is the distribution over the vocabulary for the z-th topic, and $P(d \mid z)$ 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

$$P(z \mid d, w) = \frac{P(z) P(w \mid z) P(d \mid z)}{\sum_{z'} P(z') P(w \mid z') P(d \mid z')}$$

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

$$P(w \mid z) = \frac{\sum_d T_{dw} P(z \mid d, w)}{\sum_{w'} \sum_d T_{dw'} P(z \mid d, w')}$$
$$P(d \mid z) = \frac{\sum_w T_{dw} P(z \mid d, w)}{\sum_{d'} \sum_w T_{d'w} P(z \mid d', w)}$$
$$P(z) = \frac{\sum_d \sum_w T_{dw} P(z \mid d, w)}{\sum_{z'} \sum_{d'} \sum_{w'} T_{d'w'} P(z' \mid d', w')}$$

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 `display_topics` utility. It takes two parameters: 1) your learned conditional distribution matrix, i.e., $P(w \mid z)$ 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. We will use another dataset to assess your code and observe the produced topics.