CAP6610 Machine Learning, Spring 2025

Midterm 1 Solution

Name: _____ **UFID:** _____

This is a 2-hour in-class midterm exam. You may not use any books or notes, but a double-sided cheat sheet is allowed. You will write your exam answers directly in this exam. You should use the attached scratch paper to do your rough work. Feel free to tear them away when submitting.

Problem	Score
1	/20
2	/25
3	/25
4	/30
bonus	/20

1. (20%) *Train vs test data sets.* Suppose you are building a classifier that identifies cats and dogs. You have a data set of 3,000 images containing cats, dogs, or other objects (neither cat nor dog). You randomly split the data into a 2,500 image training set and a 500 image test set.

   (a) Why is it important to "reserve" some images for the test data set? (Why shouldn't we use all 3,000 images to train the classifier?)

   (b) After training your classifier for a while, you observe it performs well on the training images, but poorly on the test images. What is one possible explanation?

   Solution.

   (a) The primary purpose of a machine learning classifier is to generalize from the training data to make accurate predictions on unseen data. By setting aside a separate test data set, you can evaluate how well your classifier generalizes to new, unseen examples.

   Besides, training a model on all available data can lead to overfitting. By evaluating your model on a separate test set, you can detect if your model is overfitting.

   (b) One possible explanation is that it is suffering from overfitting.

2. (25%) *Fitting a piecewise linear function to data.* We are given some samples $(x_i, y_i), i = 1, \ldots, n$, with x_i, y_i scalars, with a function of the form

$$\hat{y} = f(x) = \theta_1 + \theta_2 x + \theta_3 \max(0, x),$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ contains the parameters. The function is affine for $x < 0$ and also for $x > 0$, and it is continuous at $x = 0$ with value θ_1. Such a function is called *piecewise affine*, or more commonly *piecewise linear*, with a single knot or kink point at 0. We choose the parameters using the least squares regressionn on the given data points.

Consider the specific data set of (x, y) pairs

$$(-2, 3), (-1, 1), (0, 1), (1, 3), (2, 2)$$

Give the matrix $\boldsymbol{\Phi}$ and vector $\boldsymbol{\psi}$ for which the sum of the squares of the fitting errors is equal to $\|\boldsymbol{\Phi\theta} - \boldsymbol{\psi}\|^2$. We want the explicit numerical values of $\boldsymbol{\Phi}$ and $\boldsymbol{\psi}$.

Solution.

$$\boldsymbol{\Phi} = \begin{bmatrix} 1 & -2 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 2 \end{bmatrix}, \qquad \boldsymbol{\psi} = \begin{bmatrix} 3 \\ 1 \\ 1 \\ 3 \\ 2 \end{bmatrix}.$$

3. (25%) *Clustering with pre-assigned vectors.* Suppose that some of the vectors in $\phi_1, \ldots, \phi_n$ are assigned to specific groups. For example, we might insist that ϕ_{27} be assigned to cluster 5. Suggest a simple modification of the k-means algorithm that respects this requirement. Describe a practical example where this might arise, when each vector represents m features of a medical patient.

Solution. We modify k-means by over-ruling the partitioning in step 1. For any vector that is pre-assigned to a specific group, we use the given assignment. This might arise when the vectors represent patient feature vectors, and the groups (at least the ones we pre-assign) represent diagnoses of specific diseases. We have a collection of N patient vectors; for some of them, we have a known definitive diagnosis of the disease they have.

4. (30%) *Mixture of multinomials.* Consider the document classification problem. The training data is a set of documents with their correct labels. Suppose the docs are categorizated into k classes, and y_i denotes the class that doc i belongs to. Each doc is transformed into the "bag-of-words" representation, say $\boldsymbol{x}_i$ for the ith doc, meaning the jth entry of $\boldsymbol{x}_i$ represents the number of times term j appears in doc i. We want to estimate the probability $p(y|\boldsymbol{x})$ in order to design a document classifier. We use a generative approach: according to Bayes' rule, $p(y|\boldsymbol{x})$ is proportional to $p(y)p(\boldsymbol{x}|y)$:

   (a) Describe how to estimate $p(y)$

   (b) Assume $p(\boldsymbol{x}|y)$ follows a multinomial distribution; for each class c, the parameter for the multinomial distribution is a nonnegative vector $\boldsymbol{p}_c$ that sums to one. Describe how to estimate $\boldsymbol{p}_c$.

   (c) Write the resulting classifier in the form $\widehat{y} = \max_c(\boldsymbol{w}_c^\top \boldsymbol{x} + \beta_c)$. Explain how to obtain $\boldsymbol{w}_c$ and β_c from $p(y)$ and $p(\boldsymbol{x}|y)$.

   (d) (bonus 20%) Now suppose we are in the unsupervised setting, i.e., only the docs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are provided but not their class labels. Based on the previously derived method, can you come up with an expectation-maximization algorithm for the unsupervised case?

Solution.

(a) $p(y)$ is simply obtained from counting the number of documents of a specific class divided by the total number of docs, i.e.,

$$p(c) = \pi_c = \frac{n_c}{\sum_{s=1}^k n_s}.$$

(b) The vector $\boldsymbol{p}_c$ represents the probability of each term appearing in a doc from class c. A "bag-of-words" representation of a document is a histogram of terms, and we have multiple of them from the same class. The maximum likelihood estimate amounts to combining all of them into a single histogram and normalize:

$$\boldsymbol{p}_c = \frac{1}{\sum_{i \in \text{class } c} \boldsymbol{1}^\top \boldsymbol{x}_i} \sum_{i \in \text{class } c} \boldsymbol{x}_i.$$

(c) The probability that a new document $\boldsymbol{x}$ belongs to class c is proportional to

$$\pi_c \prod_{j=1}^m p_{cj}^{x_j},$$

where the terms that involve the factorials are the same for all classes, therefore inconsequential in making the predictions. Taking the log of it results in the following classifier:

$$\begin{aligned} \widehat{y} &= \arg\max_c \left(\log \pi_c + \sum_{j=1}^m x_j \log p_{cj} \right) \\ &= \arg\max_c (\boldsymbol{w}_c^\top \boldsymbol{x} + \beta_c), \end{aligned}$$

where

$$\boldsymbol{w}_c(j) = \log p_{cj}, \qquad \beta_c = \log \pi_c.$$

(d) Assume $p(y)$ and $p(\boldsymbol{x}|y)$ follow the following distributions

$$p(y_i = c) = \pi_c, \quad p(\boldsymbol{x}_i | y_i = c) = \frac{L_i!}{\prod_{j=1}^d x_{ij}!} \prod_{j=1}^d {p_{cj}}^{x_{ij}},$$

where L_i denotes the total length of doc i. The parameters we try to learn (in an unsupervised way) are π_c and p_{cj} for $c = 1, \ldots, k$ and $j = 1, \ldots, m$.

First we fix the model parameters π_c and $\boldsymbol{p}_c, c = 1, \ldots, k$ and compute the conditional mean $\mathrm{E}[y_i|\boldsymbol{x}_i]$ by $E(y_i|\boldsymbol{x}_i) = \frac{p(\boldsymbol{x}_i|y_i)p(y_i)}{p(\boldsymbol{x}_i)} = \frac{p(\boldsymbol{x}_i|y_i)p(y_i)}{\sum_{y_i} p(\boldsymbol{x}_i|y_i)p(y_i)}$

$$\begin{aligned}\psi_{ic} = \mathrm{E}[y_{ic}|\boldsymbol{x}_i] &= \frac{\pi_c \frac{L_i!}{\prod_{j=1}^{d} x_{ij}!} \prod_{j=1}^{m} p_{cj}^{x_{ij}}}{\sum_{s=1}^{k} \pi_s \frac{L_i!}{\prod_{j=1}^{m} x_{ij}!} \prod_{j=1}^{m} p_{sj}^{x_{ij}}} \\ &= \frac{\pi_c \prod_{j=1}^{m} p_{cj}^{x_{ij}}}{\sum_{s=1}^{k} \pi_s \prod_{j=1}^{m} p_{sj}^{x_{ij}}}, \qquad c = 1, \ldots, k, \quad i = 1, \ldots, n.\end{aligned} \tag{1}$$

Next we maximize $\mathrm{E}[-\log p(y, \boldsymbol{x})]$, where the expectation is taken over $p(y_i|\boldsymbol{x}_i)$ using our current estimate of this distribution, which in this case is

$$\sum_{i=1}^{n} \sum_{c=1}^{k} \psi_{ic} \left(\log \pi_c + \sum_{j=1}^{m} x_{ij} \log(p_{cj}) \right).$$

To maximize it, we get

$$\pi_c = \frac{\sum_{i=1}^{n} \psi_{ic}}{\sum_{s=1}^{k} \sum_{i=1}^{n} \psi_{is}}, \quad p_{cj} = \frac{\sum_{i=1}^{n} \psi_{ic} x_{ij}}{\sum_{i=1}^{n} \psi_{ic} L_i}, \qquad c = 1, \ldots, k, \quad j = 1, \ldots, m. \tag{2}$$

Notice the similarity between this step and how you would do it if $y_1, \ldots, y_n$ were provided to you as in part (a) and (b).

The EM algorithm for this model alternates between (1) and (2).