STAT 161/261: Homework 3
Due Wednesday, May 11 in class.

1. Problem 3, parts (c) and (d) from Homework 2.

2. Parameter estimation: Suppose that we are given data samples $\mathbf{x} = (x_1, \dots, x_N)$, which we model as i.i.d. generated from an exponential density

$$p(x_i|\theta) = \theta e^{-\theta x_i}, \quad x_i \geq 0.$$

   (a) Suppose that θ has a Gamma prior: $\theta \sim \Gamma(\alpha, \beta)$ for some known α and β. Show that the posterior density of θ given $\mathbf{x}$ is a Gamma density $\Gamma(\alpha', \beta')$ for some α' and β'. What are α' and β'?

   The Gamma distribution is called a *conjugate prior* of the exponential. A family of distributions A is called a conjugate prior for the family of distributions B if: when the prior is from A and the likelihood is from B, then the posterior is also from family A.

   (b) Using the results from above, write the posterior mean and variance $\mathbb{E}(\theta|\mathbf{x})$ and $\text{var}(\theta|\mathbf{x})$ in terms of the prior mean $\mathbb{E}(\theta)$ and variance $\text{var}(\theta)$ as well as $\mathbf{x}$.

3. Consider the k-medoids algorithm, where, instead of defining the cluster center as the centroid of the data points in the cluster, the cluster centers must be chosen from the data points. The goal of the k-medoids algorithm is the same as k-means: minimize the total sum of distances from each data point to its cluster center. Construct a simple 1-d example where k-medoids outputs a different clustering compared to k-means (starting with the same initial clustering).

4. Given training data, $(\mathbf{x}_i, r_i)$, $i = 1, \dots, N$, suppose we wish to fit a model of the form

$$r_i = w_0 + w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i1} x_{i2} + \epsilon_i. \tag{1}$$

   (a) Find a matrix $\mathbf{A}$ such that the model can be rewritten as

$$\mathbf{r} = \mathbf{A}\mathbf{w} + \epsilon,$$

   where $\mathbf{r}$ and $\mathbf{w}$ are the vectors of the responses r_i and weights w_j.

   (b) What is the linear regression estimate $\widehat{\mathbf{w}}$ of $\mathbf{w}$?

   (c) Suppose that the data r_i is actually generated from the model (1) for some true weights $\mathbf{w}$ and ϵ_i is i.i.d. Gaussian with $\epsilon_i \sim N(0, \sigma_\epsilon^2)$. What are the expected value and covariance of the estimate $\widehat{\mathbf{w}}$,

$$\mathbb{E}(\widehat{\mathbf{w}}|\mathbf{w}) \quad \text{and} \quad \text{var}(\widehat{\mathbf{w}}|\mathbf{w})?$$

5. Suppose data $\{\mathbf{x}_i\}$ are a set of p-dimensional vectors. Each vector is of the form $\mathbf{x}_i = a\mathbf{e}_k$, where $\mathbf{e}_k$ is the unit vector with 1 in the k-th position, and a and k are independent random variables with k uniformly distributed over $\{1, \ldots, p\}$ and $P(a)$ is arbitrary. Calculate the covariance matrix of the data $\mathbf{x}_i$. Show that it has one eigenvector of form (1, 1, . . . , 1) and that the other eigenvectors all have the same eigenvalue. Discuss whether PCA is a good way to select features for this problem. Hint: The covariance matrix $\mathbf{C}$ of the vectors $\mathbf{x}$ is of form $C_{ij} = \lambda + \mu\delta_{ij}$ for some λ, μ.

6. We would like to perform clustering on the following two simple datasets: `simple1.csv` and `simple2.csv`. The actual cluster solutions are given in `simple1_clusters.csv` and `simple2_clusters.csv`, but do your clustering on the two simple original datasets.

   (a) Create a scatter plot of the data set `simple1.csv`.

   (b) Cluster the dataset into three clusters using k-means. Plot the data with three colors.

   (c) Repeat using Gaussian mixture modeling with 3 clusters. Plot again.

   (d) Compare (b) and (c) with the true clusters. Explain.

7. Repeat problem 5 for the dataset `simple2.csv`.

8. PCA and LDA on gene expression data. Use MATLAB, R or equivalent. In this problem, we use PCA and LDA to analyze gene expression data. We will use the data collected in the experiments described in this paper:

   - C. Higuera, K.J. Gardiner, K.J. Cios, "Self-organizing feature maps identify proteins critical to learning in a mouse model Of Down syndrome," *PLoS ONE* 10(6):e0129126, 2015.

   The data is publicly available in the UCI repository:

   `http://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression`

   This experiment seeks to identify subsets of proteins responsible for Down syndrome and measure the effect of drugs in reducing the effect of Down syndrome. There are 38 control mice and 34 trisomic mice (Down syndrome), for a total of 72 mice. The 72 mice are divided into eight classes:

   - c-CS-s: control mice, stimulated to learn, injected with saline (9 mice)
   - c-CS-m: control mice, stimulated to learn, injected with memantine (10 mice)
   - c-SC-s: control mice, not stimulated to learn, injected with saline (9 mice)
   - c-SC-m: control mice, not stimulated to learn, injected with memantine (10 mice)
   - t-CS-s: trisomy mice, stimulated to learn, injected with saline (7 mice)
   - t-CS-m: trisomy mice, stimulated to learn, injected with memantine (9 mice)
   - t-SC-s: trisomy mice, not stimulated to learn, injected with saline (9 mice)
   - t-SC-m: trisomy mice, not stimulated to learn, injected with memantine (9 mice)

Some of the mice have been injected with mematime, a drug that may recover learning ability with Down syndrome; the others are injected with a control saline solution. Now, a gene is a region of the DNA that encodes proteins, and gene expression is the process by which a gene synthesizes a protein or other gene product. This experiment measures the expression levels of 77 proteins/protein modifications from a region of the DNA known to be involved in Down syndrome. There are 15 measurements made over time from each of the 72 mice producing 1080 samples. The samples may be considered independent. The aim is to identify which of the 77 proteins levels that can discriminate between Down syndrome and the control group.

(a) Go to the above URL and load the data. The expression levels are in the first 77 columns, and the class labels are in the 82nd column. After you have loaded the data, you should have a 1080×77 data matrix $\mathbf{X}$ and a 1080×1 class label vector $\mathbf{y}$. (In MATLAB, for example, you can use the `xlsread` command to load the data.)

- Some of the data entries in $\mathbf{X}$ are missing. Handling missing data is a large area in itself. Here, use a simple method where we replace the missing values with sample mean of the non-missing values in the same column. (For example, if you used the MATLAB `xlsread` command, missing values will be indicated by a `NaN` value in the matrix $\mathbf{X}$.)
- For the label vector, $\mathbf{y}$, the class labels in the spreadsheet are represented as strings. You will want to convert them to numeric labels from 1 to 8 or 0 to 7, to represent the eight classes.

(b) Normalize the data by removing the mean and scaling by the standard deviation in each column. Then, take the PCA of the normalized data. Feel free to use any built-in routines or to write your own. (For example, MATLAB has a `pca` method.) Plot the proportion of variation (PoV) as a function of the number of PCs. How much variation is explained by the first two PCs?

(c) Create a scatter plot of the coefficients of the data samples on the first two PCs. Use a different color for each class. You should see that when we use the PCs, the classes are completely overlapping and there is no way to discriminate the class, at least from the first two PCs.

(d) Now, we want to use LDA instead. First, for this data set, find $\mathbf{S}_B$, the matrix representing the between-class scatter and $\mathbf{S}_W$, the total in-class scatter.

(e) Now find the LDA vectors. If you recall, in class we said the LDA vectors are the eigenvectors from a matrix of the form $\mathbf{S}_W^{-1}\mathbf{S}_B$. You will find this matrix inverse is not well-conditioned. To overcome this, use a matrix of the form

$$\left[\mathbf{S}_W + \frac{\epsilon}{N}\mathrm{Tr}(\mathbf{S}_W)\right]^{-1}\mathbf{S}_B,$$

where ϵ is a small positive constant that ensures that the matrix is invertible. Use $\epsilon = 10^{-6}$. (If your software includes an LDA package or command, you can use it to check whether your work above was correct.)

(f) Now create a scatter plot of the coefficients of the samples along the first two LDA vectors. (So our 1080 samples will now each be represented in the 2-dimensional space.) Plot the classes in different colors. You should see a nice separation between classes for all but one class. Are the clusters easily distinguishable?

(g) Normally, the LDA vector does not show anything. Plot the first two LDA vectors (i.e., plot the coefficients v_j versus j for the $p \times 1$ LDA vectors). You will see that the vectors have most of their magnitude concentrated on a few dominant proteins. This implies that the difference between classes appears most strongly in these proteins – a useful piece of information for geneticists. Modern techniques try to find sparse LDA vectors. Sparse LDA is beyond the scope of this class, but the interested reader can read the paper:

- Line Clemmensen, Trevor Hastie, Daniela Witten, and Bjarne Ersboll, "Sparse discriminant analysis," *Technometrics*, 53(4):406–413, 2011.

9. Color image segmentation using k-means. MATLAB, R or equivalent. In this problem, we will use k-means for color image segmentation. As we will see, this is not the best method for segmenting images – better methods use graph-based techniques. But, it will illustrate the basic principles of k-means.

(a) Load the image `birds.jpg`. (Use the image load command or image reading command from your software. In MATLAB, you can use the `imread` command.) This should create an $n_x \times n_y \times 3$ matrix with the RGB color values over the $n_x \times n_y$ pixels. Plot the image. (In MATLAB, you can use `imshow`.)

(b) Instead of having a three-dimensional matrix, convert the image to a matrix $\mathbf{X}$ of size $n_x n_y \times 3$ so that each of the $n_x n_y$ pixels are stored as a 3×1 vectors of colors. (In MATLAB, you will also need to convert this matrix from `uint8` to `double` using the `double` command since k-means will expect double precision data.)

(c) Run k-means on the data matrix $\mathbf{X}$ with $n_c = 3$ clusters. In MATLAB, you can use the `kmeans` function. Otherwise, you will have to write a k-means routine yourself.

(d) Create a "color-blocked" image, $\mathbf{Y}$, where the RGB values of each pixel are replaced by the RGB value of the cluster center that the pixel belongs to. Reshape this back to an $n_x \times n_y \times 3$ matrix. (In MATLAB, you will also need to round the values and convert back to `uint8`.) Use the `subplot` command to plot the original image with the 3 clusters. Redo this for $n_c = 5$ clusters.

(e) In what ways were the image segmentations successful and in what ways were they not? What is the limitation of RGB-based image segmentation?