CS534 — Implementation Assignment 4 — Due Dec 4th, 2015

General instruction.

1. The following languages are acceptable: Java, C/C++, Matlab, Python and R.
2. You can work in team of up to 3 people. Each team will only need to submit one copy of the source code and report.
3. Your source code and report will be submitted through the TEACH site `https://secure.engr.oregonstate.edu:8000/teach.php?type=want_auth`

   Please clearly indicate your team members' information.
4. Be sure to answer all the questions in your report. You will be graded based on your code as well as the report. So please write your report in clear and concise manner. Clearly label your figures, legends, and tables.
5. In your report, the results should always be accompanied by discussions of the results. Do the results follow your expectation? Any surprises? What kind of explanation can you provide?

Dimension Reduction & Clustering

In this assignment you will implement 1) Principle Component Analysis (PCA); 2) Linear Discriminant Analysis (LDA); and 3) Kmeans clustering. You will test your implementation on the provided USPS digit data sets, which contains 2200 handwriting images of number 7 and 9 (1100 each). Each image is a 16×16 matrix that leads to 256 features in the data. You will apply PCA and LDA on the digits data to reduce the dimension and then Kmean clustering will be applied to the reduced data and compare the results. You will use the purity measure to measure the resulting clustering performance. Specifically, to measure purity using ground truth class labels, you first assign each cluster (and all of its instances) a class label based on the majority class it contains. Purity simply measures the accuracy of the resulting assignment. You need to submit source code of your implementations (PCA, LDA Kmean, and the experimental evaluation of them). Be sure that your code is properly commented to enhance its readability.

Specifically you need to conduct the following experiments with your implementation.

1. (15 pts) Fix $k = 2$, apply kmeans to the original data (256 dimension). Measure and report the class purity of the resulting clustering solution. Specifically, your will need to measure purity using ground truth class labels. First assign each cluster (and all of its instances) a class label based on the majority class it contains. Purity simply measures the accuracy of the resulting assignment. Because Kmeans is sensitive to random initialization, you will need to randomly restart kmeans 10 times, and pick the solution with the best Sum Squared Error objective and measure its class purity.

2. (10 pts) Compute the principal components of the data. To do this, you need to first compute the covariance matrix of your data using the equation on page 15 of the dimension reduction slides. Then compute the eigen vectors of the covariance matrix and sort them according to the eigen values (in decreasing order). Note that you don't need to implement your own eigen decomposition function. Feel free to use any numerical package for this purpose. For example, in matlab, function eig can be used. Use the results to answer the following question: what is the smallest dimension we can reduce to if we wish to retain at least 80% and 90% of the total variance respectively?

3. (10 pts) Use the principal components computed in (2) to reduce the data from 256 to 1, 2 and 3 dimensions respectively. For each choice, apply kmeans with $k = 2$ to the resulting reduced data and report their purity measures. How do they compare to the results reported in (1)?

4. (15 pts) Compute the project direction that best separates digit 7 and digit 9 by applying Linear discriminant analysis. For this part, you will use the class label as LDA is a supervised dimension reduction technique. First you compute the mean for each digit separately (say m_1 and m_2). You will then compute the within-class scatter matrix, assuming x_i is a column vector of 256 dimensions representing the i-th image in the data, using the following equation:

$$S = \sum_{y_i=7} (x_i - m_1) * (x_i - m_1)^T + \sum_{y_i=9} (x_i - m_2) * (x_i - m_2)^T$$

The projection vector is then computed as $w = S^{-1} * (m_1 - m_2)$. Note that similarly you don't need to implement the inversion function. Use existing numerical package (e.g., inv for matlab) for this purpose is fine. Project the data onto this direction, and then apply kmeans to the resulting 1-d data to find 2 clusters. Measure and report its class purity. How does this compare to the results you obtained in (3)?

5. (10 pts) Provide a discussion on the suitability of PCA and LDA for this particular dataset.