CS 613 - Machine Learning

Assignment 1 - Dimensionality Reduction & Clustering
Fall 2016

Introduction

In this assignment you'll work on visualizing data, reducing its dimensionality and clustering it.

You may not use any function from the Matlab ML library in your code. Look at the *Matlab Functions* section on Blackboard for a list of functions that are ok to use.

In particular for this assignment you **MAY NOT** use Matlab functions like:

- pca
- k-nearest neighbors functions
- entropy

But you **MAY** use basic statistical functions like:

- std
- mean
- cov
- eig

As a reminder, make sure to clear out old variables prior to running your script.

Grading

Part 1 (Theory)	25pts
Part 2 (PCA)	30pts
Part 3 (k-Means)	30pts
Report	10pts
TOTAL	95pts
Code doesn't generalize as requested	-5pts

Table 1: Grading Rubric

DataSets

Pima Indians Diabetes Data Set In this dataset of 768 instances of testing Pima Indians for diabetes each row has the following information

1. Class Label (-1=negative,+1=positive)
2. Number of times pregnant
3. Plasma glucose concentration
4. Diastolic blood pressure (mm Hg)
5. Triceps skin fold thinkness (mm)
6. Insulin (mu U/ml)
7. Body mass index (kg/m^2)
8. Diabetes pedigree function
9. Age (yrs)

Data obtained from: https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes

1 Theory Questions

1. Why do we like to use quadratic error functions (say over a 4th degree polynomial function) (2pts)?

2. Consider the following data:

$$\begin{bmatrix} -2 & 1 \\ -5 & -4 \\ -3 & 1 \\ 0 & 3 \\ -8 & 11 \\ -2 & 5 \\ 1 & 0 \\ 5 & -1 \\ -1 & -3 \\ 6 & 1 \end{bmatrix}$$

   (a) Find the principle components of the data (you must show the math, including how you compute the eivenvectors and eigenvalues). Make sure you standardize the data first and that your principle components are normalized to be unit length (5pts).

   (b) Project the data onto the principal component corresponding to the largest eigenvalue found in the previous part (3pts).

3. Consider the following data:

$$\text{Class 1} = \begin{bmatrix} -2 & 1 \\ -5 & -4 \\ -3 & 1 \\ 0 & 3 \\ -8 & 11 \end{bmatrix}, \text{Class 2} = \begin{bmatrix} -2 & 5 \\ 1 & 0 \\ 5 & -1 \\ -1 & -3 \\ 6 & 1 \end{bmatrix}$$

   (a) Compute the information gain for each feature. You could standardize the data overall, although it won't make a difference. (5pts).

   (b) Which feature is more discriminating based on results in part a (1pt)?

   (c) Using LDA, find the direction of projection (you must show the math). Normalize this vector to be unit length.
   Note: You don't not have to standardize the data since your computations should take into account the mean and standard deviations of the classes separately. (5pts).

   (d) Project the data onto the principal component found in the previous part (3pts).

   (e) Does the projection you performed in the previous part seem to provide good class separation? Why or why not (1pt)?

2 Dimensionality Reduction via PCA

Download the dataset *diabetes.csv* from Blackboard. This dataset has eight features ($D = 8$) and 768 samples ($N = 78$). The first column is the class label {-1, 1}. **However** your script should be able to work on any dataset that lacks a header row and then has an arbitrary number of data observations, N, one per row, in the format:

$$(y_i, x_{i,1}, x_{i,2}, ..., x_{i,D})$$

where $x_{i,j}$ are real valued numbers, $r_i \in \{-1, 1\}$, and D is the number of features.

Write a script that:

1. Reads in the data
2. Standardizes the data (except for the first column of course)
3. Reduces data (except for the first column of course) to 2D using PCA
4. Graphs the data for visualization
   (a) Even though we're not using class labels to do the dimensionality reduction, plot the -1 data in blue and the +1 data in red

Your graph should end up looking similar to Figure 1.

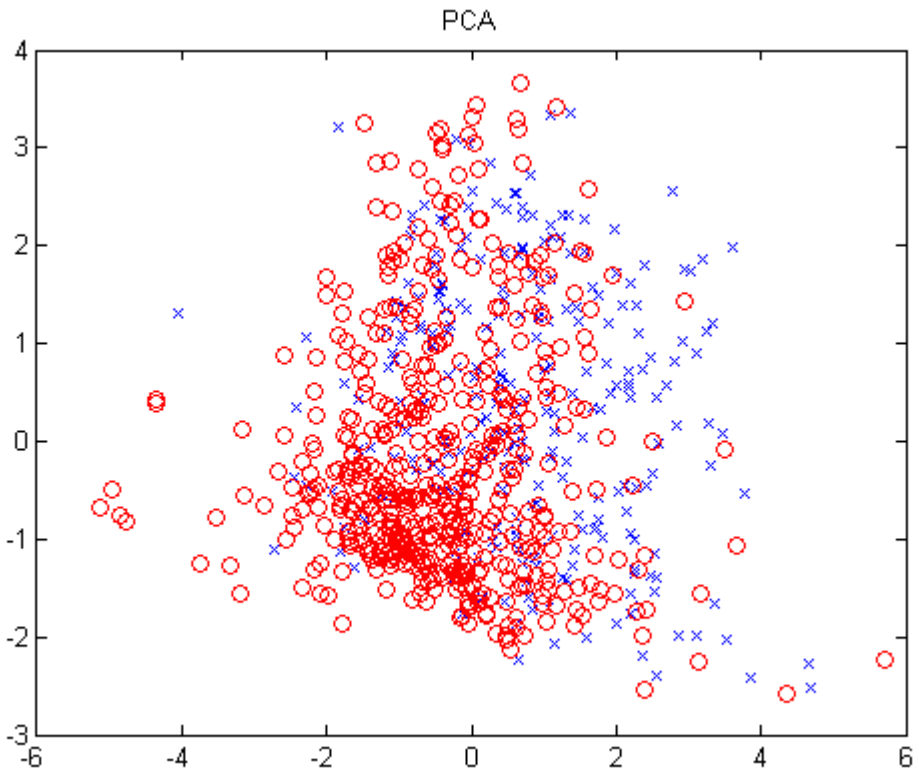

Figure 1: 2D PCA Projection of data

3 Clustering

Next we're going to cluster this same data using k-means!

Write a script that:

1. Reads in the data
2. Standardizes the data
3. Performs k-means clustering **using just the 6th and 7th feature of the data with k=2**

In addition, in your k-means code you'll want to visualize the progress of the algorithm (this will be part of your report):

1. Plot the initial setup
   (a) Data points are red 'x'
   (b) Cluster centers are blue 'o'
2. Plot the initial cluster assignments
   (a) Cluster 1 = red
   (b) Cluster 2 = blue
   (c) Data points are as 'x' (according the their assigned color)
   (d) Cluster centers are as 'o' (according the their assigned color)
3. Plot the final cluster assignments
   (a) Cluster 1 = red
   (b) Cluster 2 = blue
   (c) Data points are as 'x' (according the their assigned color)
   (d) Cluster centers are as 'o' (according the their assigned color)
   (e) Title should indicate how many iterations it took to get there

Your figures should end up similar to Figures 2-4.

Implementation Details

1. Seed the random number generator with zero (do this right before running your k-means). You can use Matlab's **rng** function to do this.
2. Randomly select two data instances and use them for the initial seeds (since we'll do $k = 2$)
3. Terminate the EM process when the sum of the L1 distance between the prior seeds and the new ones is less than *eps* (which is a Matlab defined variable related to the possible precision). *If need be, refer to the Blackboard section on "Similarity and Distance Functions" for the definition of the L1 distance.*

4. Write your code in such a way that it could work for any value of positive integer k, and any number of features, D. However you only have to plot the first two features and only have to plot *two* clusters.

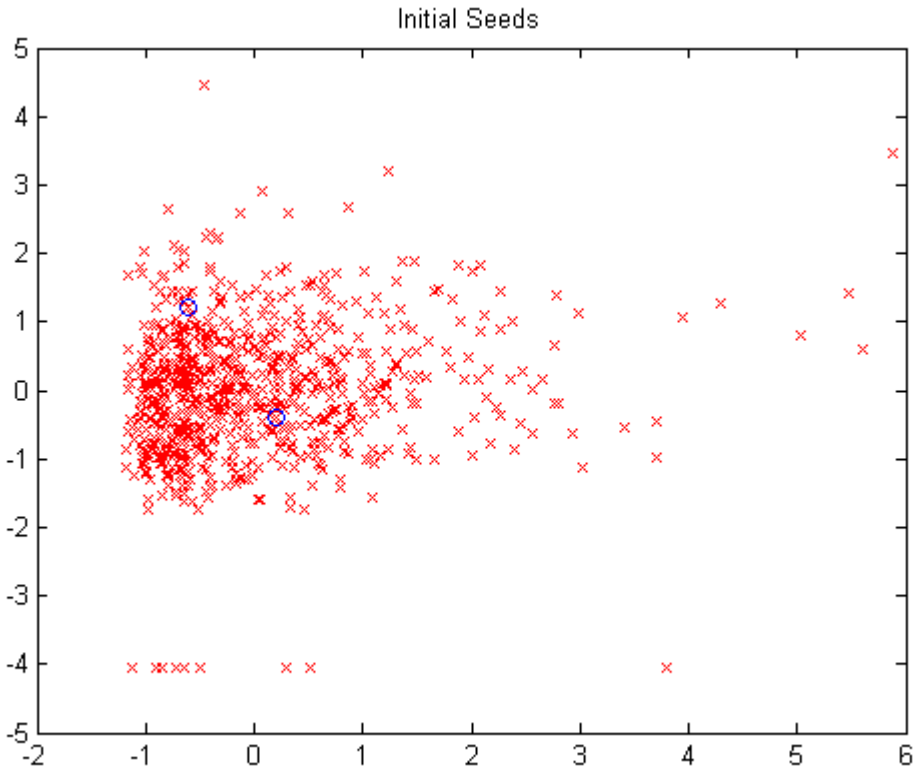

Figure 2: Seeds

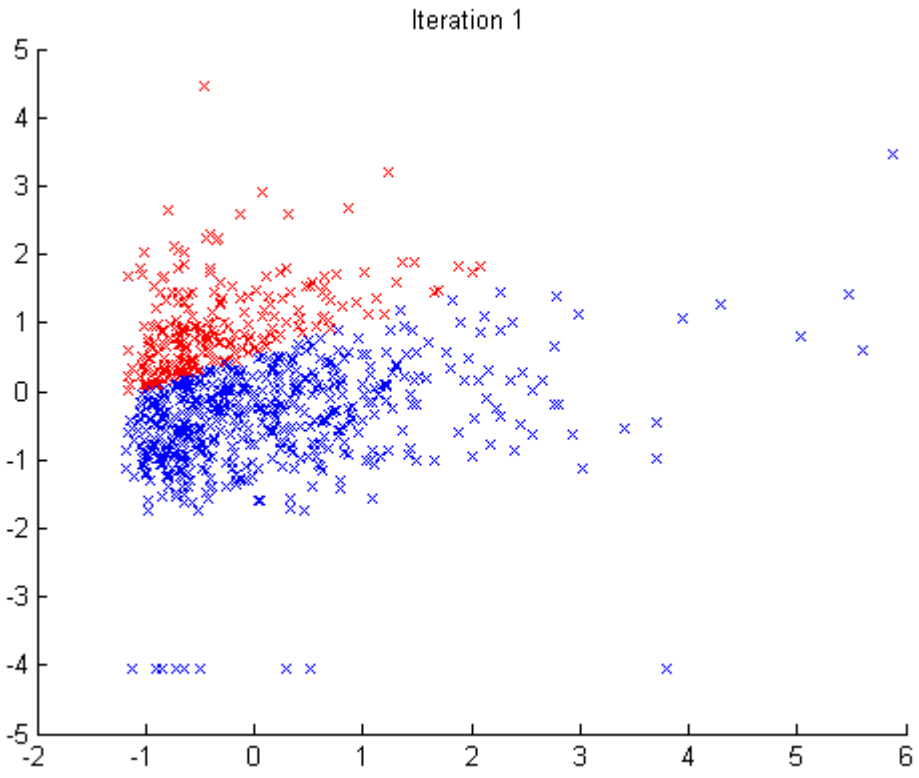

Figure 3: Initial Clustering

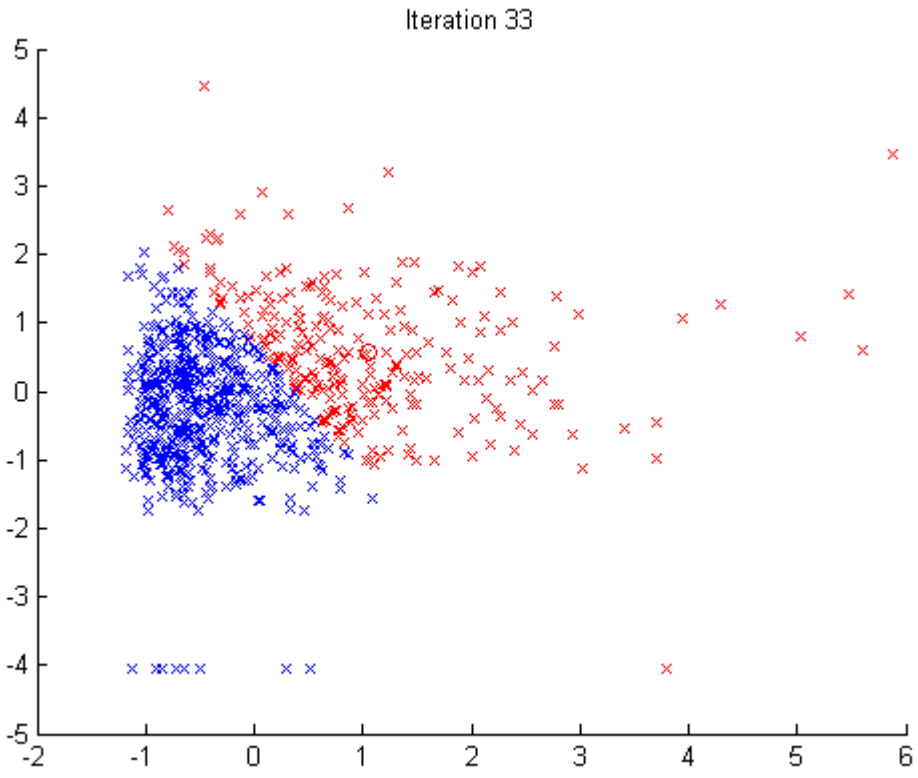

Figure 4: Final Clustering after 33 iterations

Submission

For your submission, upload to Blackboard a single zip file containing:

1. PDF Writeup
2. Source Code
3. readme.txt file

The readme.txt file should contain information on how to run your code to reproduce results for each part of the assignment.

The PDF document should contain the following:

1. Part 1: Your answers to the theory questions.
2. Part 2: The visualization of the PCA result
3. Part 3: The visualization of the k-means clustering process including:
   (a) The initial setup visualization
   (b) The initial cluster assignment visualization
   (c) The final cluster assignment visualization

   and report how many iterations it took for your algorithm to terminate.