CS 464: Homework Assignment 3

Due: May 10, 2017 17:00 pm

Instructions

- You will submit a hard copy for the answers of the write up questions (including the plots) and will upload the code online on Moodle by the due date. You may hand in the hard copy to Ali Burak or may drop it in the box in room EA427 (please stick to this submission routes) and please STAPLE your write up.
- You may code in any programming language you would prefer (however, using Matlab is strongly recommended for this homework). In submitting the code on Moodle, package your code as a ZIP file with the name convention explained in "How to Submit Programming Assignments" document available at Moodle.
- **If you are submitting the homework late (see the late submission policy in syllabus), prepare a soft copy for all the parts of the homework and submit it on Moodle. Moodle will allow late submissions until after 4 days of submission, but we will grade your homework based on the time stamp and your remaining late days.**
- Please refer to the syllabus for policies regarding collaboration, and extensions policy.
- Important: If you fail to follow the submission instructions, such as uploading your submission with a wrong name, you will receive **zero** for this assignment. There will be no exceptions, so please follow the instructions.

1 Multi-class Fisher Discriminant Analysis [20 pts]

In the lecture, you have seen the derivation of Fisher Discriminant Analysis (FDA) equations for two classes. Derive the multi-class FDA case. Show the steps of your derivation in detail.

2 Clustering [55 pts]

In this question, you will **implement** k-means and kernel k-means algorithms. You will work with a toy example to observe the difference of the two algorithms. Download `clustering.csv` from Moodle. It contains 200 samples and their labels. Note that clustering is an unsupervised algorithm, that is we do not have the labels of the samples; however, in this synthetic experiment we provide the labels in order to assess the difference of the two algorithms.

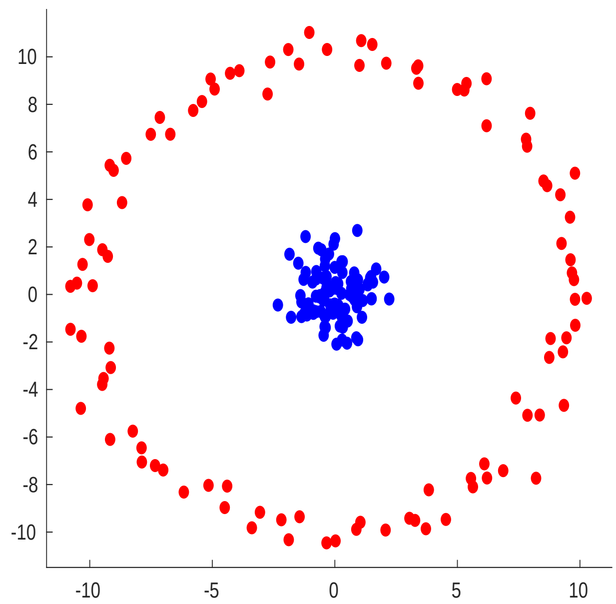

Figure 1: The visualization of the given dataset

a) [25 pts] Implement the k-means algorithm and apply it to the dataset. k-means may get stuck in local minima, you should repeat each clustering algorithm many times with random restarts and pick the best. Calculate the confusion matrix using your clustering results and the true labels. Plot the predicted clusters with two colors similiar to Figure 1 and mark the cluster centroids clearly. How does k-means perform, discuss the results?

Kernel k-means is kernelized version of the k-means algorithms and similarly aims to assign samples to the closest centroid such that the total distances of samples to their centroids are minimized. The difference between the two algorithms is in calculation of distance between samples. In k-means distance is calculated in the original feature space where as in the kernel k-means using the kernel trick in the transformed space. The kernel k-means objective is below:

$$\begin{aligned} \mathcal{J}(\mathcal{D}) &= \sum_{i=1}^{N} \sum_{k=1}^{K} \delta_{ik} \cdot \|\Phi(x_i) - m_k\|^2 \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} \delta_{ik} \cdot \left(\Phi(x_i)^T \Phi(x_i) - 2 \cdot \Phi(x_i)^T m_k + m_k^T m_k\right) \end{aligned} \quad (1)$$

where N is the number of samples, K is the number of clusters, $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$ is set of samples in input space. δ_{ik} is 1 if i^{th} sample is in k^{th} cluster, otherwise 0; Φ is a mapping function such that $\Phi : \mathcal{X} \to \mathcal{V}$ where $\mathcal{X}$ is the input space and $\mathcal{V}$ is a new feature space; m_k is the centroid of k^{th} cluster which can be calculated as follows:

$$m_k = \frac{\sum_{i=1}^{N} \delta_{ik} \cdot \Phi(x_i)}{\sum_{i=1}^{N} \delta_{ik}}. \quad (2)$$

Most of the time, we do not have the explicit mapping function to calculate $\mathcal{J}(\mathcal{D})$. This is where kernel function comes into play. Kernel function is a function such that:

$$\begin{aligned}\mathcal{K}(x_i, x_j) &= \langle \Phi(x_i), \Phi(x_j) \rangle \\ &= \Phi(x_i)^T \Phi(x_j)\end{aligned} \tag{3}$$

where $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is the mapping function from the input space to real numbers. $\mathcal{K}(x_i, x_j)$ will correspond to (i, j) and (j, i) indexed entries of the kernel matrix. Note that $\mathcal{K}$ is a symmetric function, that is $\mathcal{K}(x_i, x_j) = \mathcal{K}(x_j, x_i)$.

b) [**5 pts**] As you can see, $\mathcal{J}(\mathcal{D})$ has components which can be easily calculated by using the kernel matrix, the matrix where each entry is the kernel function evaluation between a pair of example. Rewrite the equation such that instead of Φ, kernel matrix is used in the calculation.

c) [**25 pts**] Implement the kernel k-means algorithm and apply it to the dataset using polynomial kernel with degree 2. Here to use random restarts. Give the resulting confusion matrix and calculate the accuracy. Plot the prediction of polynomial kernel k-means results by coloring the examples and marking the centroids. How does it perform? Is it better than k-means results? Discuss the results.

3 Principle Component Analysis [25 pts]

In this question, you are going to apply principal component analysis (PCA) on MNIST digit dataset. It contains 5000 images of handwritten digits from 0 to 9. You can find **digits.csv** file on Moodle. The first 400 columns are the features of the sample and the last column is the label of the sample, which you should ignore.

a) [**15 pts**] Calculate the principal components, you may use avaiable functions. Plot the resulting eigenvalues in descending order. Decide on a number of principal component, state your rationale of selection of number of principle components.

b) [**10 pts**] Display the top and bottom 5 principal components. Discuss the results. In Matlab, you can generate images of the digits using the following code snippet:

```
load digits.mat
i = 1; %select the digit (row) to display
I = digits( i, :  );
figure, imagesc( reshape( I, 20, 20 ) );
colormap( gray );
axis image;
```

NOTES

- This homework will be graded by your TA, Ali Burak Ünal. Please ask the clarifications on Moodle, for other things you may ask questions at his office hours or by e-mail.
- Do not forget to add a `README` file that contains execution details for your code.