HW 3

CS6510: Applied Machine Learning
IIT-Hyderabad
Aug-Nov 2016

Max Points: 60
Due: 18th Nov 2016 11:59 pm

Instructions

- Please use Google Classroom to upload your submission by the deadline mentioned above. Your submission should comprise of a single ZIP file with all your solutions, including code.

- For late submissions, 10% is deducted for each day (including weekend) late after an assignment is due. Note that each student begins the course with 5 grace days for late submission of assignments. Late submissions will automatically use your grace days balance, if you have any left. You can see your balance on the CS6510 Marks and Grace Days document under the course Google drive.

- You should use Python for the programming assignments.

- Please read the department plagiarism policy. Do not engage in any form of cheating - strict penalties will be imposed for both givers and takers. Please talk to instructor or TA if you have concerns.

1 Theory [26 points]

1. **(6 points) Hierarchical Clustering:** Given below is the distance matrix for 6 data points

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0					
x_2	0.12	0				
x_3	0.51	0.25	0			
x_4	0.84	0.16	0.14	0		
x_5	0.28	0.77	0.70	0.45	0	
x_6	0.34	0.61	0.93	0.20	0.67	0

(a) Draw a dendrogram for the final result of hierarchical clustering with single link. [*2 points*]

(b) Draw a dendrogram for the final result of hierarchical clustering with complete link. [*2 points*]

(c) Change two values from the matrix so that the answer to the last two questions is same. [*2 points*]

2. **(8 points) Principal Component Analysis:** Suppose each of the data elements $\mathbf{x}$ is an M-dimensional vector. The vectors are of the form $\mathbf{x} = a\delta_k = (0, ..., 0, a, 0, ...)^T$, where a is in the k^{th} slot, and k,a are random variables. k is uniformly distributed over $1, \cdots, M$ and $P(a)$ is arbitrary.

   (a) Calculate the covariance matrix. [*2 points*]

   (b) Show that it has one eigenvector of form $(1, ..., 1)$ and that the other eigenvectors all have the same eigenvalue. [*3 points*]

   (c) Discuss whether PCA is a good way to select features for this problem. [*3 points*]

   (**Hint:** Use expectation to compute the covariance matrix: $C = E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^T]$. You should get C of form $C_{ij} = \lambda + \mu\delta_{i,j}$ for some λ, μ.)

3. **(12 points) Logistic Regression:**

   (a) Plot the sigmoid function $1/(1 + e^{-w\mathbf{x}})$ vs $\mathbf{x} \in \mathbb{R}$ or increasing weight $w \in \{1, 5, 100\}$. A qualitative sketch is enough. Use these plots to argue why a solution with large weights can cause logistic regression to overfit. [*2 points*]

(b) To prevent overfitting, we want the weights to be small. To achieve this, instead of maximum likelihood estimation MLE for logistic regression:

$$\max_{w_0,\cdots,w_d} \prod_{i=1}^{n} P(Y_i|X_i, w_0, \cdots, w_d)$$

we can consider maximum a posterior (MAP) estimation:

$$\max_{w_0,\cdots,w_d} \prod_{i=1}^{n} P(Y_i|X_i, w_0, \cdots, w_d)P(w_0, \cdots, w_d)$$

where $P(w_0, \cdots, w_d)$ is a prior on the weights. Assuming a standard Gaussian prior $\mathcal{N}(0, I)$ for the weight vector (I = Identity matrix), derive the gradient ascent update rules for the weights. [*4 points*]

(c) One way to extend logistic regression to multi-class (say, K class labels) setting is to consider $(K-1)$ sets of weight vectors and define:

$$P(Y = y_k|X) \propto \exp(w_{k0} + \sum_{i=1}^{d} w_{ki}X_i) \text{ for } k = 1, \cdots, K-1$$

What model does this imply for $P(Y = y_k|X)$? What would be the classification rule in this case? [*4 points*]

(d) Draw a set of training data with three labels and the decision boundary resulting from a multi-class logistic regression. (The boundary does not need to be quantitatively correct but should qualitatively depict how a typical boundary from multi-class logistic regression would look like.) [*2 points*]

2 Programming (34 points)

1. **(13 points)** DBSCAN is density based clustering algorithm. In this question you need to implement your own DBSCAN algorithm. You can read more about it from paper that proposed this method [link]

   (a) Use the Kmeans clustering algorithm from python's sklearn package and find the number of clusters in dataset1. Plot the data points with different colors for different clusters. [*3 points*]

(b) Implement your own DBSCAN algorithm on the same dataset1 and plot the data points. [*5 points*]

(c) What differences do you see between the DBSCAN and k-means methods, and why? [*2 points*]

(d) Consider the dataset2 with three clusters. Use (a) and (b) for dataset2, and compare the performance. List your observations clearly, and make conclusions on pros and cons of DBSCAN and k-means. [*3 points*]

2. **(13 points)** One of the earliest methods used for face recognition was based on Principal Component Analysis (PCA), and was called *Eigenfaces*[1]. You can understand how this works by using the facial data set of Tom Mitchell provided on this link.

   (a) Read all the faces from the data set, flatten each face, and collect the faces in an ($d \times n$ matrix, where d is the dimension of the flattened face, and n the total number of faces. For the given data set, you should find that you end up with (d, n) = (960, 624). This means that the number of observations is significantly lower than the dimension of the observations. What impact does this situation have on your investigation? [*2 points*]

   (b) Find the PCA components of this face data set. Plot the singular values (scaled so that the largest equals one). How many principal components are needed to explain $90\%, 80\%, 50\%$ of the variance? [*4 points*]

   (c) The eigenfaces mentioned above are the principal components. Display several of the principal components as images, starting at the lowest ones. What, if anything, do you learn from looking at them? Hint: How much detail do you see in the eigenfaces? [*3 points*]

   (d) Choose one of the faces in the data set and project it onto 1, 2, 10, and 100 PCs. Display the results. What do you observe? [*4 points*]

[1] Turk, Matthew, and Alex Pentland. "Eigenfaces for recognition." Journal of cognitive neuroscience 3.1 (1991): 71-86.

3. **(8 points)** Download the Reuters-21578 dataset. It is a collection of documents on different topics (classes). The .SGM files contain the documents. You can read more about the file format and other details from Section VI of README.txt.

   (a) Use the documents that occur in the following classes as the dataset for this question: acquisitions, corn, crude, earn, grain, interest, money-fx, ship, trade, and wheat. Represent each document in vector form using the tf-idf notation.[information] [*2 points*]

   (b) Use the vector form of documents created above in (b) to run K-means clustering on these documents into 10 clusters.(You can use the inbuilt k-means algorithm). [*2 points*]

   (c) Compute purity, normalized mutual information, F_1 and Adjusted RI(*Rand Index*) for the clustering with respect to the 10 classes. [*2 points*]

   (d) Compare the performance with your own DBSCAN implementation from Programming Question 1, and report your observations. [*2 points*]