Habib University
CSE 351 - Artificial Intelligence
Fall 2019
Assignment 2

Objective:

This assignment gives students a hands-on experience of some classification and clustering methods and familiarize them with some machine learning libraries in python.

Q1 – Classification [45 points]

Problem 1.1 – Implementing NaiveBayes [35 points]

You are given a dataset of textual summary of medical queries classified into five different categories. You have to build your own naïveBayes classifier to predict these categories for future queries. The task includes:

- **[15 points]** Implement Learn(..) method that takes training and test data (in the form of word vectors) and learns its NaïveBayes classifier. You have to do your own implementation of NaïveBayes here.
- **[05 points]** Some conditional probabilities may turn out to be zero in the training dataset. Modify your classifier such that it applies Laplacian smoothing while learning conditional probabilities to avoid such zero values.
- **[05 points]** Implement Predict(…) that takes a learned classifier and test data and returns predicted values.
- **[10 points]** Implement Evaluate(…) that takes actual and predicted labels and returns precision, recall, f-measure. The method will also display confusion matrix.

The code will be written in the attached Ass2_classification.ipynb. You can find here details of NaïveBayes classification.

Problem 1.2 - [10 Points]

You are given a Fashion-MNIST dataset comprising of 28x28 grayscale images of 60,000 fashion products from 10 categories, with 7,000 images per category. The training set has 60,000 images and the test set has 10,000 images. Some sample images are shown below:

The dataset is freely available here.

You have to build a scikit-learn based classifier for this dataset to classify each input image into one (out of 10) categories. You are required to experiment with three to four different classifiers and report accuracy of each of them.

Note: Your code will follow the same structure given in Problem 1.1.

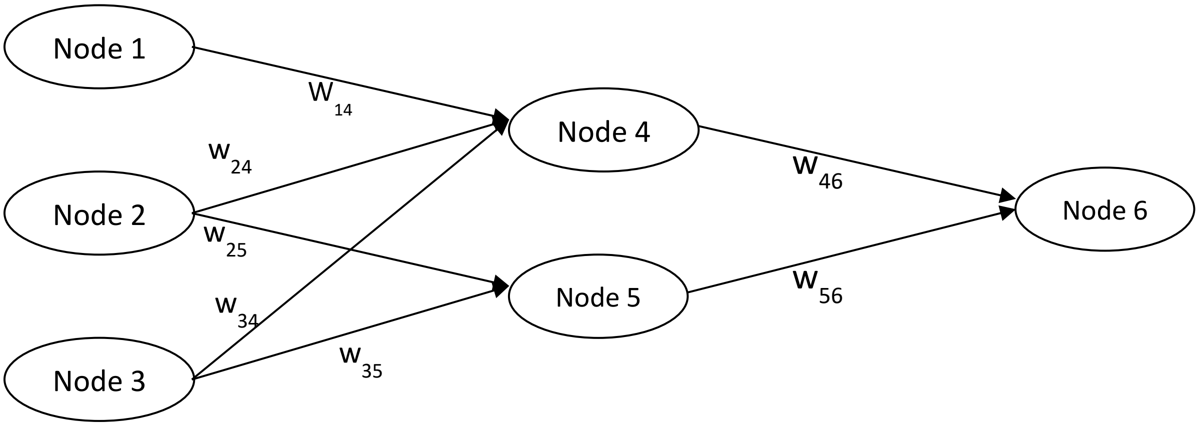
Q2 – Forward and Back propagation through ANN [10 points]

You have to manually run one complete iteration of feed-forward and back-propagation over the network given below. At the end of this iteration, you will get updated weights to be used in the next iteration. Sigmoid function will be used as activation function.

w_{14}=0.35, w_{24}=0.15, w_{25}=-0.10, w_{34}=-0.20, w_{35}=0.20, w_{46}=0.40, w_{56}=0.25

Actual Output(T)= 0.8, Learning Rate(r) = 0.8

Input : (0.5, 0.3,0.9)



The final result will be given in the following tabular format:

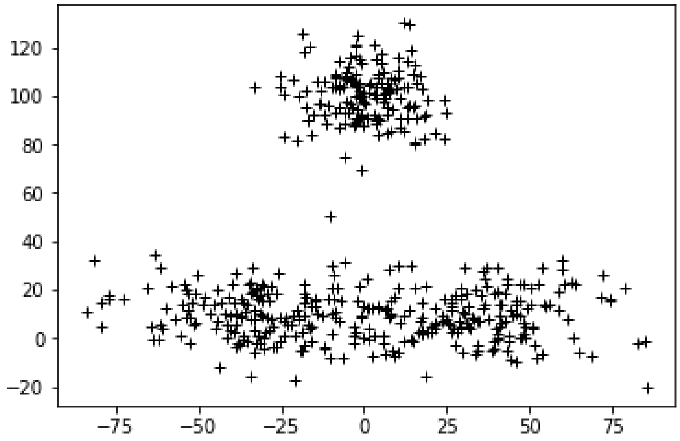
Input of Node 4		Error at Node 6	
Output of Node 4		Error at Node 4	
Input of Node 5		Error at Node 5	
Output of Node 5		Updated w_{46}	
Input of Node 6		Updated w_{56}	
Output of Node 6		Updated w_{14}	
		Updated w_{24}	
		Updated w_{34}	
		Updated w_{25}	
		Updated w_{35}	

Q 3 – Clustering [25 points]

You have to implement k-means algorithm to cluster a set of 1000 2D points. The data points (x,y) will be generated via Gaussian distribution using the sample code below:

```
points = []
def initializePoints(count):
     for i in range(int(count/3)):
         points.append([random.gauss(0,10),random.gauss(100,10)])
     for i in range(int(count/3)):
         points.append([random.gauss(-30,20),random.gauss(10,10)])
     for i in range(int(count/3)):
         points.append([random.gauss(30,20),random.gauss(10,10)])
```

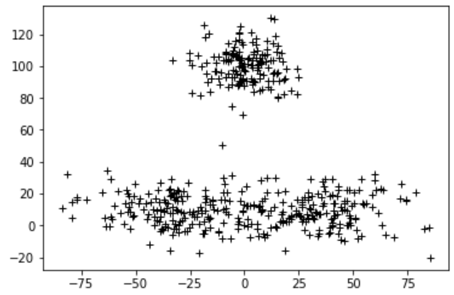
This code generates following data points:



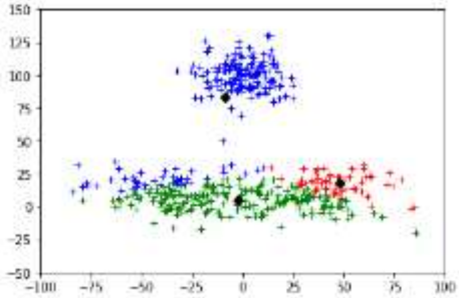
a) **[10 Point]** Implement K-means algorithm to cluster these points. K-means will take a set of points and the number of clusters (K) and will cluster the points into given number of clusters. The algorithm will stop if there is no significant change coming in positions of centroids.
You can try giving different mean and variance to gaussian distribution to have different input points and check your code.
b) **[04 Points]** Provide graphical visualization of the process of formation of clusters(as shown on next page).
c) **[06 Points]**Instead of running K-means once, run it 10 times and give the best clustering formation achieved. You need to identify the criteria to assess quality of clusters and then choose the best one.
d) **[05 Points]** Kmeans asks you to provide number of clusters (K) as input. Do some research to identify clustering methods that do not ask for K. Give short description of one such algorithm.

The code will be written in the attached Ass2_clustering.ipynb

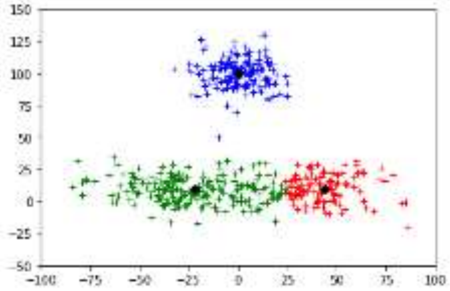
Iteration 0



Iteration 1



Iteration 2

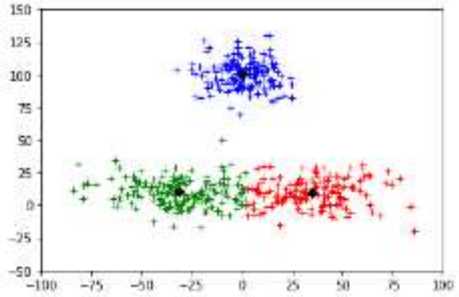


.

.

.

Iteration 9



Submission Instructions

Submissions will be made on the LMS by the due date (announced on LMS). No email submission will be accepted. The submitted file should be in the form of a ZIP file named as **<studentid>_Ass2** containing separate files/folders named Q1 and Q3(a-c) for the source code. The zip file will also contain a pdf answering theory questions (Q2 and Q3(d)).