DSC 190: Homework 3

January 25, 2021

Instructions

Please submit your solution **by the beginning of the week 7 lecture (9:30 AM PT, Feb 16)**. Submissions should be made on **gradescope**. Please complete homework **individually**. Please include the code of your solutions in the submission with a write-up describing how to run the code.

You are allowed to use any third-party libraries.

You will need the following files for this Homework:

Iris.csv (available on canvas)

1. **DBSCAN Algorithm (10 points):**
Consider the following figure:

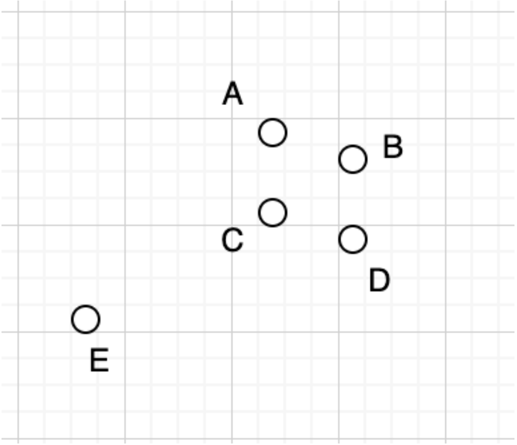

Figure 1: Points

There are 5 points: A, B, C, D, E. The distance matrix between these points is as follows:

	A	B	C	D	E
A	0	2	2	5	6
B	2	0	2	3	7
C	2	2	0	3	6
D	5	3	3	0	7
E	6	7	6	7	0

Table 1: Distance matrix

If we cluster the above points using the DBSCAN algorithm with $\epsilon = 4$ and `minimum points` = 3,

(a) (5 points) How many clusters are formed? Draw the outline of your clusters. Explain your reasoning.

(b) (5 points) With respect to the above figure, state one advantage of DBSCAN over k-means algorithm.

DSC 190: Homework 3

January 25, 2021

2. **Association Rule Mining (20 points):** Consider the following table:

Id	Movies watched
1	Titanic, A star is born, Crazy Rich Asians
2	Titanic, Inception, Crazy Rich Asians
3	Titanic, Crazy Rich Asians, Avatar, Iron Man
4	A star is born, Inception, Crazy Rich Asians, Avengers
5	A star is born, Inception, Crazy Rich Asians, Avatar, Avengers

Table 2: Movies

(a) (10 points) Find all frequent patterns (i.e., movie combinations) whose support ≥ 0.5.

(b) (10 points) Find all the rules `(X -> Y) (s, c)` where `s` represents support and `c` represents confidence such that `s` ≥ 0.5, `c` ≥ 0.6.

3. **PCA (20 points):** Consider the following two plots. These are plots of training data points X in $\mathbb{R}^2$ belonging to 2 classes.

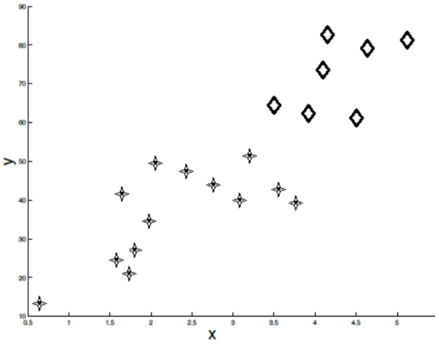

Figure 2: Dataset-1

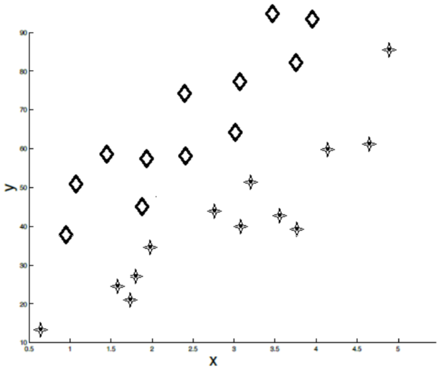

Figure 3: Dataset-2

Answer the following questions for each dataset.

(a) (10 points) Draw all two principal components in the picture (you can take a screenshot). You are expected to draw the rough directions of the principal components instead of accurate computations.

(b) (10 points) After projecting all the points onto one of the principal components, is it possible to correctly classify all the points by just a threshold function? If yes, which principal component should we project onto and why? If no, please explain your reasoning.

4. **PCA, k-Means and GMM clustering (50 points):**

In this question, we will using the Iris dataset to predict `Species` of the iris plant. ('Iris.csv' in Canvas)

(a) (5 points) In the data preparation step do the following:

(i) Split data into features (X) and label (y). Our label is the column `Species` and features include all the other columns except `Species` and `Id`.

(ii) Standardize the features (X_standardized) by removing the mean (i.e mean=0) and scaling to unit variance. (Hint: use `sklearn.preprocessing.StandardScaler()`)

(b) (15 points) Project the 4-dimensional standardized data (X_standardized) onto 2 dimensions using PCA (`sklearn.decomposition.PCA()`). Visualize the scatterplot of the first two principal components of the data. In the scatterplot assign each data point a color based on its species with the following dictionary:

`{'Iris-setosa':'r', 'Iris-versicolor':'g', 'Iris-virginica':'b'}`.

(c) (10 points) Cluster the standardized data (X_standardized) into 3 clusters using GMM clustering. Score the clustering accuracy with `sklearn.metrics.cluster.adjusted_rand_score()`. (Hint: use `sklearn.mixture.GaussianMixture()`)

(d) (10 points) Cluster the standardized data (X_standardized) into 3 clusters using K-means clustering. Score the clustering accuracy with `sklearn.metrics.cluster.adjusted_rand_score()`. (Hint: use `sklearn.cluster.KMeans()`)

(e) (10 points) Briefly compare the result from part (c) and part (d). Explain why Gaussian Mixture algorithm performs better than k-Means algorithm.

Suggested reading: http://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html