CSCI 347
Project 03: Dimensinality Reduction and Clustering

This project may be completed individually or with group of up to size three. Turn in the code and written responses in both Brightspace and Gradescope.

Choose a data set that you are interested in from the UCI Machine Learning Repository that has at least five numerical attributes, and that you believe may contain clusters. Only use the numerical attributes for this project. *Note*: if you are planning to complete the extra credit portion of this project, you will need to use a data set that has class labels (ground truth cluster labels), i.e., a classification data set, in order to compute the accuracy of the clustering. If you would like to use a data set from a different source, please discuss this with me.

Problem 1: Think about the data

In a well-written paragraph, answer the following questions:

1. (1 point) Why are you interested in this data set?
2. (1 point) How many numerical attributes and categorical attributes are there in the data set?
3. (1 point) Are there any missing values? If there are missing values, how are you planning to handle these? (Will all data instances with missing values be removed? Will all attributes with missing values be removed? Will missing values be imputed? If so, how?)
4. Before doing any analysis, answer the following questions:
   (a) (1 point) Why do you expect clusters to be present in the data?
   (b) (1 point) Why might finding clusters in this data set be helpful? How might this help us understand or analyze the data?
   (c) (1 point) How many clusters do you expect to see in the data? Provide a range of values to answer this question. For example, 2 to 4. Why do you expect a number of clusters in this range?
   (d) (1 point) Do you expect that the clusters will be of similar size (i.e., cluster 1 is about the same size as cluster 2, is about the same size as cluster 3, etc..)? Why or why not?

Part 2: Write Python code for clustering

Write the following functions in Python. You may use scikit-learn or other packages to check the correctness of your implementation, but you may not use any existing clustering algorithm implementation in your code.

1. (10 points) A function that implements the k-means clustering algorithm. The function should take a data matrix, a number of clusters k, and a convergence parameter ϵ, as input, and return the representatives (means) as well as the clusters found using k-means. If the distance is the same between a point and more than one representative (mean), then assign the point to the mean corresponding to the cluster with the lowest index.

2. (10 points) A function that implements the DBSCAN clustering algorithm. The function should take a data matrix and the parameters *minpts* and ϵ, as input, and return the clusters found using DBSCAN, and for each data point a label of core, border, or noise point.

3. (Extra Credit - 5 points): A function that computes the precision of a clustering. The function should take a list of true cluster labels and a list of the cluster labels returned by some clustering algorithm, and return the precision of the clustering.

Part 3: Analyze your data

Report the following, using tables or figures as appropriate. You may use scikit-learn's implementation of k-means and DBSCAN, but you are encouraged to first try using your own implementations on real-world data.

1. (4 points) Use sklearn's PCA implementation to linearly transform the data to two dimensions. Create a scatter plot of the data, with the x-axis corresponding to coordinates of the data along the first principal component, and the y-axis corresponding to coordinates of the data along the second principal component. Does it look like there are clusters in these two dimensions? If so, how many would you say there are?

2. (3 points) Use sklearn's PCA implementation to linearly transform the data, without specifying the number of components to use. Create a plot with r, the number of components (i.e., dimensionality), on the x-axis, and $f(r)$, the fraction of total variance captured in the first r principal components, on the y-axis. Based on this plot, choose a number of principal components to reduce the dimensionality of the data. Report how many principal components will be used as well as the faction of total variance captured using this many components.

3. (5 points) For both the original and the reduced-dimensionality data obtained using PCA in question 3.2, do the following: Experiment with a range of values for the number of clusters, k, that you pass as input to the k-means function, to find clusters in the chosen data set. Use at least 5 different values of k. For each value of k, report the value of the objective function for that choice of k.

4. (5 points) For both the original and the reduced-dimensionality data obtained using PCA in question 3.2, do the following: Experiment with a range of values for the *minpts* and ϵ input parameters to the DBSCAN function to find clusters in the chosen data set. First, keep ϵ fixed and try out a range of different values for *minpts*. Then keep *minpts* fixed, and try a range of values for ϵ. Use at least 5 values of ϵ and at least 5 values of *minpts*. Report the number of clusters found for each (*minpts*, ϵ) pair tested.

5. (Extra credit - 3 points): Create a plot of clustering precision for each value of k used in question 3.3, each value of ϵ used in question 3.4, and each value of *minpts* used in question 3.4, for both the original and reduced-dimensionality data.

Tips and Acknowledgements

Make sure to submit your answer as a PDF on Gradscope and Brightspace. Make sure to show your work. Include any code snippets you used to generate an answer, using comments in the code to clearly indicate which problem corresponds to which code.

Acknowledgements: Project adapted from assignments of Veronika Strnadova-Neeley.