University of Pittsburgh
CS2750 Machine Learning
Professor Milos Hauskrecht

Handout 9
March 31, 2022

Problem assignment 9

Due: Thursday, April 6, 2022

Problem 1. K-means clustering

Part a. Write and submit a python program *kmeans_clustering_code.py* that loads the dataset in *clustering_data.csv* and run the k-means algorithm (implemented in scikit-learn, more specifically in sklearn.cluster) for finding 3 clusters. Please use Euclidean distance to define the distances in between the points and default initialization of cluster centers. After that the program should report the sizes of the three groups found by the kmeans algorithm and use scatter function to plot the data in the dataset and the means of the clusters. Please use colors to distinguish data that were assigned different groups. Use a separate color to show the cluster centers (means). Include the results, the center coordinates, number of instances in groups, and graphs in your report.

Part b. Modify the program from Part a. with the number of means equal to 4. Again report the centers, sizes of the groups, and plot the data (with different group colors) and the means found by kmeans in the report.

Part c. The kmeans procedure can be initialized randomly by setting its init parameter with value 'random'. Rerun your kmeans algorithm for $k = 4$ (the same as Part b) with init='random'. The means found are likely to change. If they did not, try to rerun the procedure again till you see the change in the means. Show the scatter plot of the results when the centers changed.

Part d. Let us assume the two runs of the k-means lead to two different clusterings. Write a math expression that would let you compare these different clusterings and pick the best one. Hint: what criterion does the k-means optimize?

Part e. Run the kmeans procedure (in the init='random' mode) with $k = 4$ 30 times. Report the cluster sizes found for these different runs? Use formula from Part d to decide which clustering is the best. Show the scatter plot of the best clustering and use different colors to distinguish instances in the different groups.

Problem 2. Hierarchical clustering

Part a. Write and submit a python program *agglomerative_clustering_code.py* that loads the dataset *clustering_data.csv*, and runs the hierarchical agglomerative clustering using the Euclidean and complete linkage (complete linkage is equal to the max linkage in the lecture) to obtain three clusters. Report the sizes of the three groups and use the scatter function to plot the data in the dataset. Use different colors to distinguish instances in the different groups. Include the results in the report. Also compare the results to kmeans in Problem 1. Part a. Are the clusters the same or they are different?

Part b. Modify the code from Part a to obtain four clusters. Compare the results to Problem 1. Part b. Are the clusters the same or different?

Part c. Now we try to change the linkage. Please run the hierarchical agglomerative clustering code using the Euclidean distance to generate four clusters with (1) Ward, (2) Average, and (3) single linkages. Analyze the differences from Part b in terms of the cluster sizes and scatter graphs showing the results of the clustering.

Optional (Extra credit). Plot the dendrogram one obtains using the hierarchical agglomerative clustering with the Euclidean distance and the complete linkage. A sample code for generating the dendrogram on a different dataset is included in code file.

Problem 3. Feature/Input ranking

Consider the dataset in file *FeatureSelectionData.csv*. The dataset consists of 259 examples (in rows) where each example is defined by 70 dimensional input vector (represented in columns) and an associated binary label (in last column).

Part a. Write and submit a python program $feature\_selection\_code.py$. The code should implement a function $Fisher\_score(x, y)$ that takes as arguments a vector of one-dimensional inputs x and a vector of binary outputs y and calculates the Fisher score as defined in the lecture. Use this function to evaluate the different dimensions of the input space (there are 70 dimensions) to estimate their individual predictive power. Please report the ordered list of dimensions with the top 20 Fisher scores, and their Fisher score values. The dimensions should be labeled from 1 to 70 depending on their position in the dataset.

Part b. Modify the code from Part a. by adding a function $AUROC\_score(x, y)$ that takes as arguments a one-dimensional vector of inputs x and a vector of outputs y and calculates the area under the ROC curve and by evaluating the different dimensions of the input space and their individual predictive power based on AUROC score. Again, report the ordered list of 20 dimensions with the top 20 AUROC scores, and their values. Compare the results

(top dimensions) from part a and part b in the report and discuss your findings. Are the ordered lists the same? In general, do you expect them to be the same?