Problem assignment 9  
Due: Thursday, April 5, 2018

Problem 1. K-means clustering

Assume we have a 2-dimensional space and four points \((0, 0), (0, 5), (6, 7)\) and \((7, 0)\) and we want to cluster the examples into two groups using the k-means algorithm and the Euclidean distance.

Part a. Let us assume the algorithm is initialized with means \((0, 0)\) and \((7, 0)\). What are the values of the two means the algorithm converges to. How are datapoints divided into groups?

Part b. Now let us assume the algorithm is initialized from the means \((3, 3)\) and \((7, 0)\). What are the values of the two means the algorithm converges to. How are datapoints divided into groups?

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

Problem 2. Clustering experiments

Please load the dataset `clustering_data.txt`.

Part a. Run the k-means algorithm (implemented in Matlab in the function kmeans) for finding 2 clusters. Use Euclidean distance to define the differences in between the points. The kmeans procedure (if initial means seeds are not set) uses a random set of seeds in each run. Run the kmeans procedure (in the default mode) 30 times. Report the cluster sizes for these different runs? Use formula from Problem 1 Part c. to decide which clustering is the best.

Part b. kmeans procedure implemented in Matlab allows you to control initial means’ seeds. Propose, describe and implement a method for generating initial seeds and run the k-means procedure for 30 times with this new seed procedure. To determine the best clustering from 30 runs use again the formula from Problem 1 Part c. Is your clustering initialization
better in Part b than in Part a? Compare this by running the competition 100 times, such that each time, we initialize the k-means 30 times and compare the best clusters from each approach. Please analyze the results.

Part c. One reason for performing the clustering is to analyze data and see what datapoints fall into the same class. When performing this analysis you see the groups and you often try to make sense of the groups, that is, you try to see whether there is something common about the elements in the group. One way to evaluate the clustering is to use an additional class (group) label related to some aspect of the data and compare the clustering results to these labels, that is, you want to see whether clusters agree with these labels. We provide one such set of labels in class_labels.txt file. Our goal is to see if the clustering found matches well the labels in the class labels file. Both files are aligned across rows.

To start the analysis, we need to define a measure (or score) reflecting the agreement between the clusters and labels. Propose a score that reflects how well the clusters match labels. Hint: the best cluster should have the representatives of just one class.

After that please use the score to evaluate the clustering you found in Part 1. Do you think the clustering and the class labels agree?

**Problem 3. Feature/Input ranking**

Consider the dataset in file Data.txt. 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 function `Fisher_score(x, y)` that takes as arguments a vector of one-dimensional inputs `x` and a vector of 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.** Write and submit 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. You may use Matlab functions to calculate the area under the curve for this purpose. Similarly to part a, evaluate 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 from part a and part b and discuss your findings. Are the ordered lists the same? In general, do you expect them to be the same.