University of Pittsburgh
CS1675 Introduction to Machine Learning — Handout 10
Professor Milos Hauskrecht — November 9, 2017

Problem assignment 9

Due: Thursday, November 16, 2017

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 accross rows.

To start the analysis, we need to define a measure of the agreement between the two data partionings that let you compare them. Propose the measure of agreement that averages pointwise agreements for clusters and class labels. Please note that for 2 partitionings (with two classes) there are two ways to match/mismatch individual partitions. Your score should measure the better of the two. After that please use the measure to compare the clustering found in Part a. (the best clustering from 30 kmeans with default seeds) and the class label and report the results. Do you think the clustering and class labels agree?

Part d. The analysis of the agreement score may be tricky. One way to calculate the agreement is to use Kappa agreement score (see e.g. wiki for the description). However, this measure may be tricky to interpret and and it vary a lot depending on the number of labels. Another way to see if the labels make sense is to use the permutation-based analysis. Basically what we do is we first calculate and store the agreement score for the clustering and the class labels. After that we randomly permute class labels, (class labels get randomly assigned to other datapoints), calculate the agreement for the clustering and permuted labels, and store the score. We repeat the permutation of class labels N times (say 1000) which gives us a baseline distribution of agreement scores under the random class labels. Given this distribution of 'random' agreement scores we can estimate the chance the agreement scores under random labels and true labels are the same (or different). To do this, calculate the proportion (probability) of agreement scores that are higher or equal to the score on the true labels. If this probability is small than the true score is different from 'random' scores at that probability level and the clustering is related to the class labels.

Please write a code for implementing the above permutation test. Print a histogram of agreement scores under the random permutation of class labels. Based on the result argue wheter the clustering is related to the class label or this relation is random.