University of Ioannina,
Computer Science & Engineering Department

Data Mining – CS 059
Fall 2013

Assignment 2

The assignment should be handed in at the beginning of the class on Tuesday December 17. For late submissions the late policy on the page of the course will be applied. The details for the turn-in are on the page of the course.

Question 1 (Distance Metrics)

Consider a universe U of objects and a distance function $dist: U \times U \to \mathbb{R}$, such that for any pair of objects x, y we can compute their distance $dist(x, y)$. The distance function $dist$ is a **metric**. Consider a set $S \subseteq U$ of N objects. Let $m \in U$ be the object **in *U*** such that $m = \arg\min_{\mathrm{y} \in \mathrm{U}} \sum_{x \in S} dist(y, x)$. For example in the case where U is the $\mathbb{R}^d$ and $dist$ is the Euclidean distance, m is the mean of the points in S. Also, let y^* be the object **in *S*** such that $y^* = \arg\min_{\mathrm{y} \in \mathrm{S}} \sum_{x \in S} dist(y, x)$. Prove that

$$\sum_{x \in S} dist(x, y^*) \leq 2 \cdot \sum_{x \in S} dist(x, m)$$

Finding m is an NP-hard problem for some distance metrics, while finding y^* is easy. The above inequality shows that using y^* instead of m we get an object that has sum of distances close to the minimum.

Hint: It will be useful to consider the object $y_m \in S$ that is closest to the object m.

Question 2 (Min-Hashing)

Do **Exercise 3.3.2** from the textbook Mining Massive Datasets by Anand Rajaraman and Jeff Ullman. Hand in the output of the intermediate steps of the Min-Hashing signature computation (as we did in class, and as it is done in the book), the final signature with all four functions, and the estimated and true similarity for all pairs of columns.

Question 3 (Clustering)

In this exercise you will implement the k-means algorithm described in class. You can use any programming language for your implementation. Apply the algorithm on the iris and spambase datasets supplied on the material web page of the course. The class labels should not be part of the input, when running the algorithm. The number of clusters should be equal to the number of classes in the data.

Produce the confusion table between classes and clusters and use one of the external validation measures to evaluate the algorithm. Submit your code, the output of the clustering, the confusion tables, and the validation measures, and a short report on your results.

Question 4 (Clustering)

The goal of this exercise is to experiment with the application of the k-means algorithm on real data. For this question you will use the dataset of FourSquare tips that you utilized for Assignment 1. On top of the pre-processing you did for the tips in Assignment 1, you will also compute for each distinct word in the tips the tf-idf score of the word (the tf-idf score was defined in the second lecture of the course). Each tip will be a real vector with the tf-idf scores of words in the tip.

You can use your own implementation for the k-means algorithm from the previous question, or some other one (there is an implementation of k-means in WEKA, and in MATLAB). For the number of clusters plot the SSE curve and select the value of k that seems “correct”. For each cluster that you produce, take the centroid of the cluster and output the top-20 words (more if necessary) with the highest score. Examine the words, and try to argue about what the cluster represents.

Submit your code and a report where you discuss your findings. The report should contain the plot of the SSE curve, a discussion about the value of k, the top words of the centroids, and your evaluation about what each cluster means.