Problem Set 2

October 9, 2014

Due date: Wed, Oct 24, 2014 at 1pm; before class.

Exercise 1 (20 points): The aglomerative hierarchical clustering produces a hierarchical clustering of n points by starting with all points being in separate clusters and merging at every step the two clusters that are the *closest*.

Independently of how the distance between two clusters is computed (minimum, maximum or average) the *Naive* implementation of the agglomerative clustering algorithm works as follows: first compute all pairwise distances among all pairs of points. At every step where clusters C_i and C_j are merged the distance between the merged cluster C_{ij} and all existing clusters is recomputed in order to decide the next merge that is going to be selected. This *Naive* algorithm requires $O(n^3)$ distance computations.

Design an algorithm that improves the running time of *Naive* to $O(n^2 \log n)$.

Exercise 2 (30 points): We consider a set X of n points in $\mathbb{R}^d$. The following algorithm aims to cluster the points in X and *at the same time* discover the outliers of X. The algorithm takes as input two integers k and ℓ, such that $k + \ell \leq n$. It produces a clustering of X into k clusters and it reports ℓ outliers. The algorithm, which is named (k, ℓ)-means, is inspired by the k-means algorithm and it works as follows.

1. Select k points in X, uniformly at random, and call them centers
2. Until convergence:
   - 2.1 Assign each point in X to its closest center
   - 2.2 Consider as outliers the ℓ points in X that have the largest distance to their center
   - 2.3 Consider the cluster C_i consisting of all non-outlier points assigned to center i
   - 2.4 Recompute the center i as the mean of all points in C_i

Given a partition of X specified by k centers and ℓ outliers we can define the error of the partition to be the sum of square of distances of each non-outlier point to its closest center (that is, similar to the error of k-means but with excluding the outliers).

- **2.1 [10 points]**: Provide an example in which the (k, ℓ)-means algorithm does not perform well. Your example should be a dataset, which intuitively has k clusters and ℓ outliers, and the (k, ℓ)-means algorithm fails to discover the correct clusters and correct outliers, even though it uses the correct values of k and ℓ.

- **2.2 [10 points]**: Prove formally that in each iteration of the (k, ℓ)-means algorithm the error does not increase, and thus the (k, ℓ)-means algorithm converges to a local optimum.

- **2.3 [10 points]**: Propose a new algorithm for the problem of detecting k clusters and ℓ outliers, which is inspired by the k-means++ algorithm.

Exercise 3 (25 points): In the k-center problem the input consists of a set of n d-dimensional points $X = \{x_1, \ldots, x_n\}$ and the goal is to partition the points into k groups $C_1, \ldots, C_k$ such that:

$$\max_{i=1\ldots k} \max_{x,x' \in C_i} L_2(x - x')$$

is minimized. The problem for $d \geq 2$ is NP-hard. However, for $d = 1$ it has a polynomial-time algorithm. The goal of this excersice is to give an optimal polynomial-time algorithm that solves the k-center problem for 1-dimensional points in time $O(n^2)$; your running time should not depend on k. Write the pseudocode of your algorithm, prove that it is optimal and give a running-time analysis.

Exercise 4 (25 points): Consider the edit distance between two labeled graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ with the same set of nodes to be the number of edges in E_1 that are not in E_2 plus the number of edges in E_2 that are not in E_1. That is,

$$\Delta(G_1, G_2) = |E_1 \setminus E_2| + |E_2 \setminus E_1|.$$

- **4.1 [10 points]**: Prove that $\Delta()$ is a metric.
- **4.2 [15 points]**: Given a set of graphs $G_1, G_2, \ldots, G_n$ consisting of n graphs all sharing the same set of labeled nodes design an algorithm for finding the centroid of the set of clusters, when distance Δ is used as a distance function between graphs.