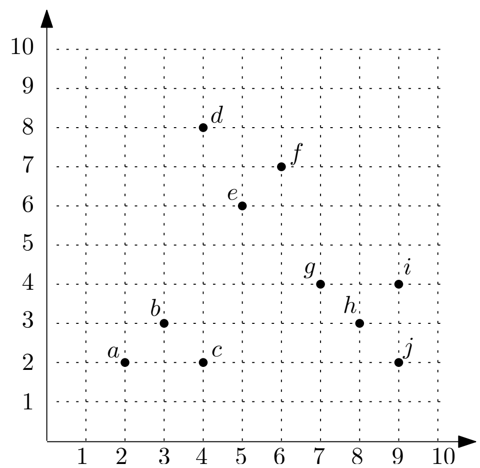
CMSC5724: Exercise List 8

Problem 1. Consider the execution of the k-center algorithm we discussed in the class on the following set P of points:



Suppose that $k = 3$ (i.e., we want to find 3 centers), and that the first center has been (randomly) decided to be f. Show what are the second and third centers found by the algorithm. The distance metric is Euclidean distance.

Answer. Let S be the set of centers that have been collected. $S = \{f\}$ currently and will eventually include 3 centers when the algorithm terminates. For each point $p \in P$, define:

$$d(p) \quad = \quad \min_{o \in S} dist(o, p)$$

where $dist(o, p)$ is the distance between o and p. Refer to $d(p)$ as the *center distance* of p.

In each iteration, the algorithm adds the point with the largest $d(p)$ to S. In the first iteration, $S = \{f\}$, the center distances of all the points are:

point	center distance
a	$\sqrt{41}$
b	5
c	$\sqrt{29}$
d	$\sqrt{5}$
e	$\sqrt{2}$
f	0
g	$\sqrt{10}$
h	$\sqrt{20}$
i	$\sqrt{18}$
j	$\sqrt{34}$

Hence, the center point added to S is a.

Since S has changed, the center distances become:

point	center distance
a	0
b	$\sqrt{2}$
c	2
d	$\sqrt{5}$
e	$\sqrt{2}$
f	0
g	$\sqrt{10}$
h	$\sqrt{20}$
i	$\sqrt{18}$
j	$\sqrt{34}$

Hence, the 3rd point added to S is j.

Problem 2. Let P be the set of points in Problem 1. What is the geometric center of the set $\{c, e, g\}$?

Answer. The geometric center of a set S of points is the point p whose x- (y-) coordinate x_p (y_p) is the mean of the x- (y-) coordinates of the points in S. Hence, the geometric center of $\{c, e, g\}$ is point (5.33, 4).

Problem 3. Let P be the set of points in Problem 1. Apply the k-means algorithm on P with $k = 3$ under Euclidean distance. Assume that the algorithm selects a set $S = \{c, g, h\}$ as the initial centroids. Recall that (i) the algorithm updates S iteratively, and (ii) the cost of S is defined to be $\phi(S) = \sum_{p \in P}(d_S(p))^2$ where $d_S(p) = \min_{q \in S} dist(p, q)$.

- Give the content of S after each iteration until the algorithm terminates.
- Show the value of $\phi(S)$ after every iteration.

Answer.
Iteration 1. Let $o_1 = c, o_2 = g, o_3 = h$, namely, the 3 centroids in the initial S. The algorithm divides P into 3 partitions P_1, P_2 and P_3, such that P_i ($1 \le i \le 3$) includes all the points in P that find o_i to be their closest centroids. Specifically, $P_1 = \{a, b, c\}, P_2 = \{d, e, f, g\}$, and $P_3 = \{h, i, j\}$. Then, the algorithm recomputes o_i as the centroid of P_i, for each $1 \le i \le 3$, giving $o_1 = (3, 2.33)$, $o_2 = (5.5, 6.25)$, and $o_3 = (8.67, 3)$. $\phi(S)$ is 19.08.

Iteration 2. The algorithm re-divides P into P_1, P_2 and P_3 based on the current centroids. Now, $P_1 = \{a, b, c\}, P_2 = \{d, e, f\}$, and $P_3 = \{g, h, i, j\}$. Accordingly, the centroids are re-computed as $o_1 = (3, 2.33)$, $o_2 = (5, 7)$, and $o_3 = (8.25, 3.25)$. $\phi(S) = 12.17$—the cost is lower than that of the previous iteration.

Iteration 3. After re-dividing P, $P_1 = \{a, b, c\}, P_2 = \{d, e, f\}$, and $P_3 = \{g, h, i, j\}$. The centroids are still $o_1 = (3, 2.33)$, $o_2 = (5, 7)$, and $o_3 = (8.25, 3.25)$, i.e., no change has occurred from the last iteration. The algorithm therefore terminates.

Problem 4. The goal of this problem is for you to understand why it suffices to consider a finite number of possible solutions to the k-means problem (recall that this was needed to argue that the algorithm terminates).

Consider the k-means problem defined in the lecture notes with $k = 2$. Suppose that we have a set P of n points in $\mathbb{R}^2$ (for simplicity, we assume that the dimensionality is 2). The goal is to find centroid points c_1, c_2 in $\mathbb{R}^2$ to minimize $\sum_{p \in P}(d(p))^2$, where $d(p) = \min_{i=1}^{2} dist(p, c_i)$, with *dist* representing Euclidean distance. Design an algorithm to solve this problem in $O(2^n \cdot n)$ time.

Answer. Each pair of c_1, c_2 defines two disjoint subsets of P:

- S_1 = the set of points in P closer to c_1;
- S_2 = the set of points in P closer to c_2.

Note: if a point is equi-distance to c_1, c_2, assign it to one of S_1, S_2 arbitrarily. In this way, we ensure that $S_1 \cup S_2 = P$.

How many different S_1 are there? At most 2^n — the number of all possible subsets of P.

Motivated by the above, we can solve the problem as follows. For each possible subset S_1, generate $S_2 = P \setminus S_1$. Then, take the geometric centers c_1, c_2 of S_1, S_2, respectively. Evaluate the *quality* $\sum_{p \in P}(d(p))^2$. Finally, return the c_1, c_2 with the best quality.

The running time is $O(2^n \cdot n)$ because the quality of a pair of c_1, c_2 can be obtained in $O(n)$ time.