ACM/IDS 104 APPLIED LINEAR ALGEBRA
PRACTICE PROBLEMS FOR LECTURE 5: SOLUTIONS

KONSTANTIN M. ZUEV

Problem 5A. K-Means Algorithm: Toy Example
Whenever you learn (or develop) a new algorithm, it is always very useful to explore how the algorithm works in simple, particular cases where the solution to the problem the algorithm is intended to solve is known. This illuminates the algorithm, makes it more intuitively clear, and serves as a common-sense check.

Suppose we want to partition the 12 blue points shown in Fig. 1 into $K = 4$ clusters. Four of the points, marked with red crosses, are the initial representatives of the clusters. Perform the k-means algorithm by hand and determine the resulting clusters.

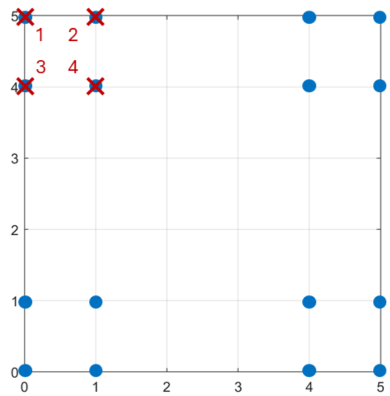

Figure 1. A cloud of (blue) points and four cluster representatives (red crosses).

Solution: The successive iterations of the k-means algorithm are shown in Fig. 2. Groups of points enclosed by red curves represent the clusters at each iteration. The algorithm terminates when two successive clusterings are identical.

Problem 5B. Convergence of the K-Means Algorithm
Suppose that, given n points $v_1, \dots, v_n$ in $\mathbb{R}^d$, we want to partition the points into K disjoint clusters using the K-means algorithm. Let's recall how the K-means algorithm works.

(1) Choose cluster representatives $r_1^{(0)}, \dots, r_K^{(0)}$ at random among $v_1, \dots, v_n$.
(2) Compute the first clustering $c^{(1)} = (c_1^{(1)}, \dots, c_n^{(1)})$ by assigning v_i to cluster number $c_i^{(1)}$, where

$$c_i^{(1)} = \arg \min_{k=1,\dots,K} \|v_i - r_k^{(0)}\|, \quad i = 1, \dots, n. \tag{1}$$

(3) Update representatives as follows:

$$r_k^{(1)} = \frac{1}{n_k} \sum_{i:c_i^{(1)}=k} v_i, \quad k = 1, \dots, K, \tag{2}$$

where n_k is the number of points in cluster k.

Then compute the second clustering $c^{(2)}$ using the updated representatives, and continue iterating Steps (2) and (3) until two consecutive clusterings are identical, that is, until $c^{(m)} = c^{(m-1)}$ for some m.

Prove the K-Means algorithm always converges in a finite number of steps.

Solution: Let $c^{(1)}, c^{(2)}, \dots$ denote the sequence of clusterings obtained while executing the algorithm. First of all, note that the set of all possible clusterings $\mathcal{C}$ is finite.

E-mail address: `kostia@caltech.edu`.

Since $\mathcal{C}$ is finite, the sequence $c^{(1)}, c^{(2)}, \ldots \in \mathcal{C}$ must eventually revisit a clustering it has already visited. In other words, there exists an $m \in \mathbb{N}$ and some $s \in 1, 2, \ldots, m-1$ such that $c^{(m)} = c^{(m-s)}$. Our goal is to prove that $s = 1$, because if $s > 1$, the algorithm would enter a cycle and continue running forever.

Let $\rho^{(l)}$ be the value of the objective function that is minimized by the K-means algorithm, corresponding to clustering $c^{(l)}$ and vector of representatives $r^{(l)}$. That is,

$$\rho^{(l)} = \rho(c^{(l)}, r^{(l)}) = \frac{1}{n} \sum_{i=1}^{n} \|v_i - r^{(l)}_{c_i^{(l)}}\|^2 \tag{3}$$

The key observation is that if $c^{(l)} \neq c^{(l-1)}$, then $\rho^{(l)} < \rho^{(l-1)}$. Let's prove this. Each iteration of the algorithm consists of two steps: in the assignment Step (2), each point v_i is assigned to its nearest representative, and in the update Step (3), each representative is updated to be the mean of the points in its cluster.

Let $r^{(l-1)} = (r_1^{(l-1)}, \ldots, r_K^{(l-1)})$ be the representatives after iteration $l-1$, which have been obtained by averaging points according to clustering $c^{(l-1)}$. In the next assignment step, the algorithm computes $c^{(l)}$ by minimizing the objective function

$$\rho(c, r) = \frac{1}{n} \sum_{i=1}^{n} \|v_i - r_{c_i}\|^2 \tag{4}$$

over all clusterings $c \in \mathcal{C}$, with r fixed as $r^{(l-1)}$. Therefore,

$$\rho(c^{(l)}, r^{(l-1)}) = \min_{c \in \mathcal{C}} \rho(c, r^{(l-1)}) \leq \rho(c^{(l-1)}, r^{(l-1)}). \tag{5}$$

Moreover, if $c^{(l)} \neq c^{(l-1)}$, then the new clustering $c^{(l)}$ is better than the old one, and the inequality is strict[1]:

$$c^{(l)} \neq c^{(l-1)} \quad \Rightarrow \quad \rho(c^{(l)}, r^{(l-1)}) < \rho(c^{(l-1)}, r^{(l-1)}). \tag{6}$$

Next, in the update step, the algorithm updates the representatives by computing, for each k,

$$r_k^{(l)} = \frac{1}{n_k} \sum_{i: c_i^{(l)} = k} v_i. \tag{7}$$

This choice minimizes the objective function (4) over all possible representatives r, with fixed $c = c^{(l)}$. Hence,

$$\rho(c^{(l)}, r^{(l)}) \leq \rho(c^{(l)}, r^{(l-1)}). \tag{8}$$

Combining the two steps, we obtain

$$c^{(l)} \neq c^{(l-1)} \quad \Rightarrow \quad \rho^{(l)} = \rho(c^{(l)}, r^{(l)}) \leq \rho(c^{(l)}, r^{(l-1)}) < \rho(c^{(l-1)}, r^{(l-1)}) = \rho^{(l-1)} \tag{9}$$

Let's briefly summarize what we have proved about the sequence of consecutive clusterings $c^{(1)}, c^{(2)}, \ldots$. This sequence traverses a finite set $\mathcal{C}$, and while it visits new clusterings, the value of the objective function is strictly decreasing: $\rho^{(1)} > \rho^{(2)} > \rho^{(3)} > \ldots$. Eventually, it will revisit a clustering it has already visited, but we cannot have $c^{(m)} = c^{(m-s)}$ for $s > 1$, since this would mean that $\rho^{(m)} = \rho^{(m-s)}$, which contradicts the strictly decreasing behavior of ρ. Therefore, there exists an m such that $c^{(m)} = c^{(m-1)}$ and the K-means algorithm is guaranteed to converge in a finite number of steps.

[1] We assume that the data points $v_1, \ldots, v_n$ are in general positions, so that there are no distance ties.

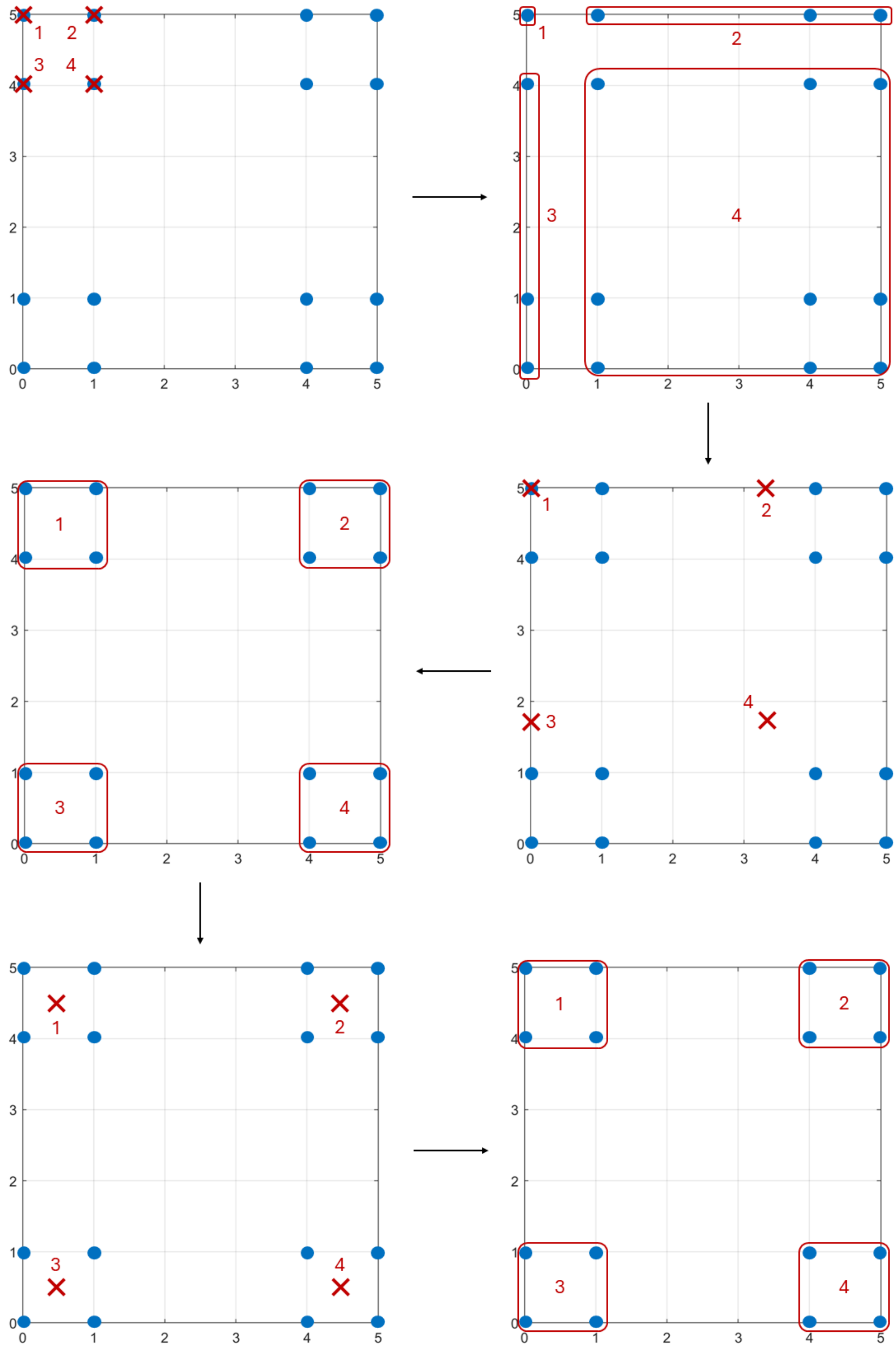

FIGURE 2. K-means algorithm in action.