Related Topic: Clustering

Input: n sites: $S = \{s_1, s_2, ..., s_n\}$

Output: Locations of k centers: $C = \{c_1, c_2, ..., c_k\}$

Objective: Minimize the total squared distance from each site to the nearest center.

$$\text{Minimize} \quad \sum_{s \in S} dist(s, C)^2$$

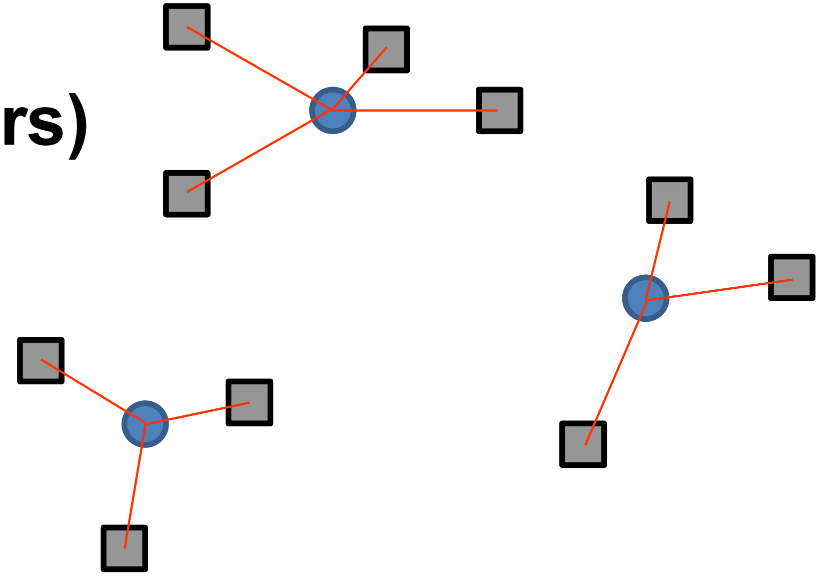
$dist(s, C)$: Distance from s to the nearest center.

$$dist(s, C) = \underset{c \in C}{\text{Min}} \{dist(s, c)\}$$

Reformulation

site (n sites)

center (k centers)



(1) Divide the n sites into k clusters based on the nearest center.

$$S = \{s_1,\ s_2,\ \ldots,\ s_n\}$$

$$S = S_1 \bigcup S_2 \bigcup \cdots \bigcup S_k$$

(2) Reformulate the objective function as follows:

$$\text{Minimize} \sum_{j=1}^{k} \sum_{s \in S_j} dist(s,\ c_j)^2$$

$$\text{Minimize} \sum_{j=1}^{k} \sum_{s \in S_j} dist(s, c_j)^2$$

$$S = \{s_1, s_2, \ldots, s_n\}$$

$$\Downarrow$$

$$S = S_1 \cup S_2 \cup \cdots \cup S_k$$

s and c_j: Points in the 2D space.

k-means Algorithm: Iterate the following two steps from a random partition of *S* into *k* subsets: S_1, S_2, ..., S_k

(i) $c_j = \dfrac{1}{|S_j|} \sum_{s \in S_j} s, \quad j = 1, 2, \ldots, k.$

(ii) $S_j = \{s \mid dist(s, c_j) = \min_{l=1,\ldots,k} dist(s, c_l)\}, \; j = 1, 2, \ldots, k.$

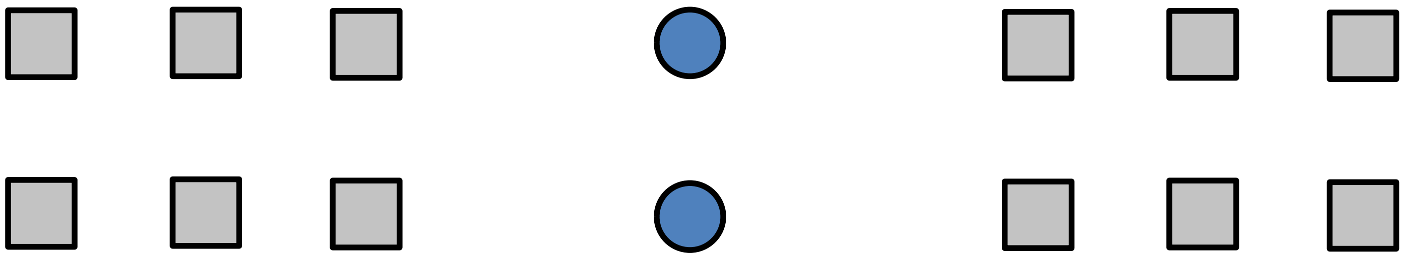
Exercise 7-1:

In the k-means algorithm, we can start with (i) using an initial partition $\{S_1, S_2, \ldots, S_k\}$ or with (ii) using initial centers $\{c_1, c_2, \ldots, c_k\}$. Design a good initialization method for the k-means algorithm with (i) or (ii).

Example (12 sites and 2 centers)

If we start with the following partitions:

If we start with the following centers:

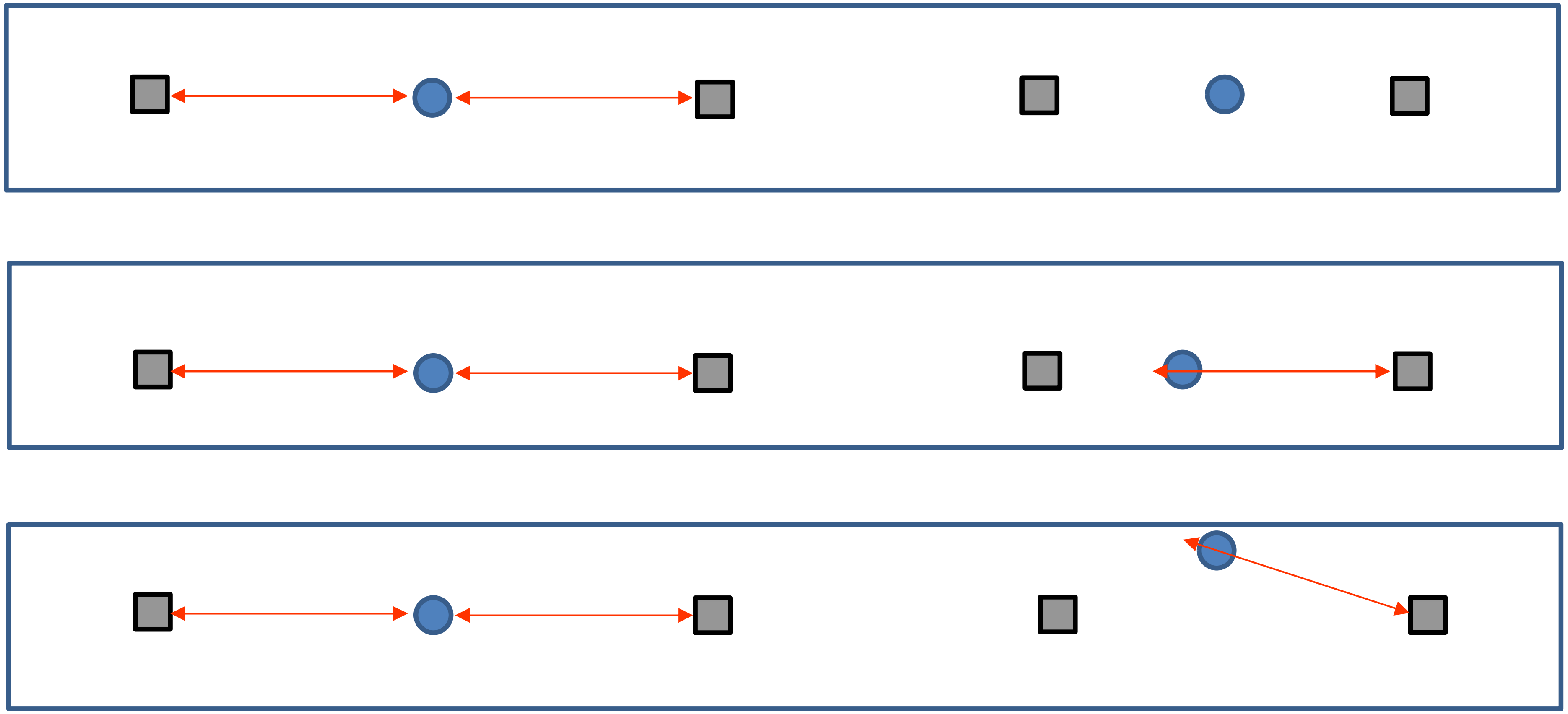


Difficulty of "Min-Max" objective function:
("minimize the worst case" objective function)

Minimization of the maximum distance from each site to the nearest center.

$$\text{Minimize} \ \operatorname*{Max}_{s \in S} \ dist(s, C)$$

All the following solutions are optimal (for *k* = 2).



Comparison of Problems:

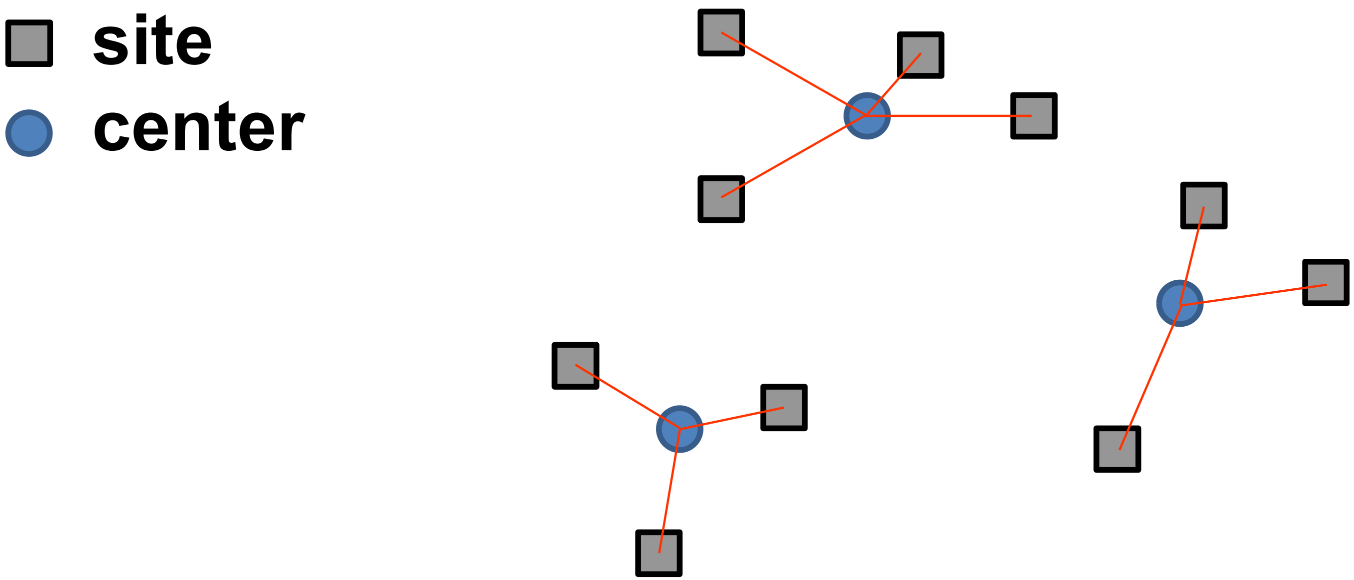
(1) Minimization of the maximum distance from each site to the nearest center.

$$\text{Minimize } \operatorname*{Max}_{s \in S} dist(s, C)$$

(2) Minimization of the total squared distance from each site to the nearest center

$$\text{Minimize } \sum_{s \in S} dist(s, C)^2$$

Q. Which is a better problem formulation?

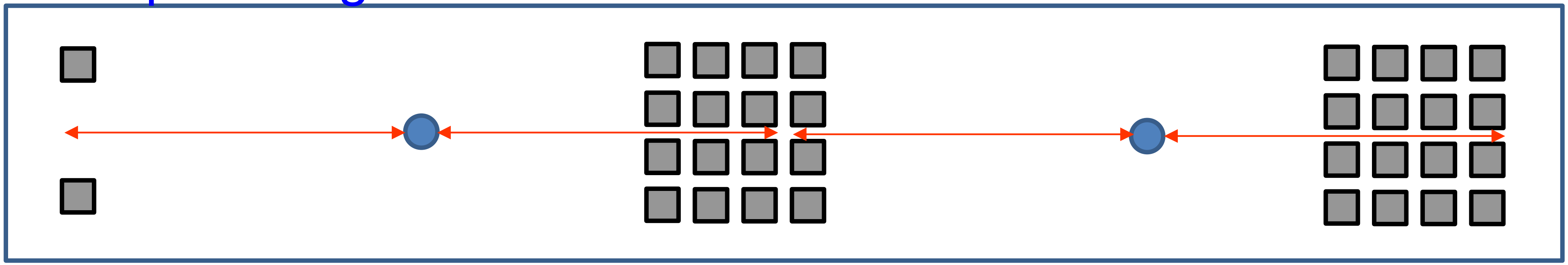


Comparison of Problems:

(1) Minimization of the maximum distance from each site to the nearest center.

$$\text{Minimize } \operatorname*{Max}_{s \in S} dist(s, C)$$

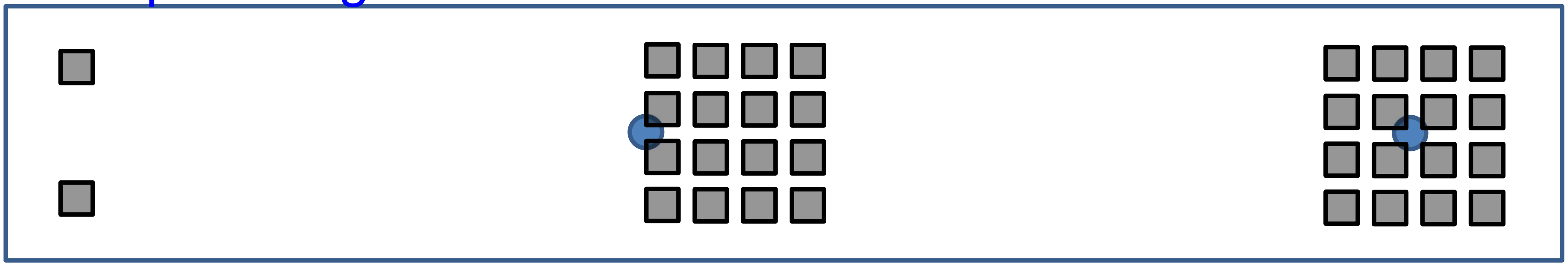
Example of a good solution



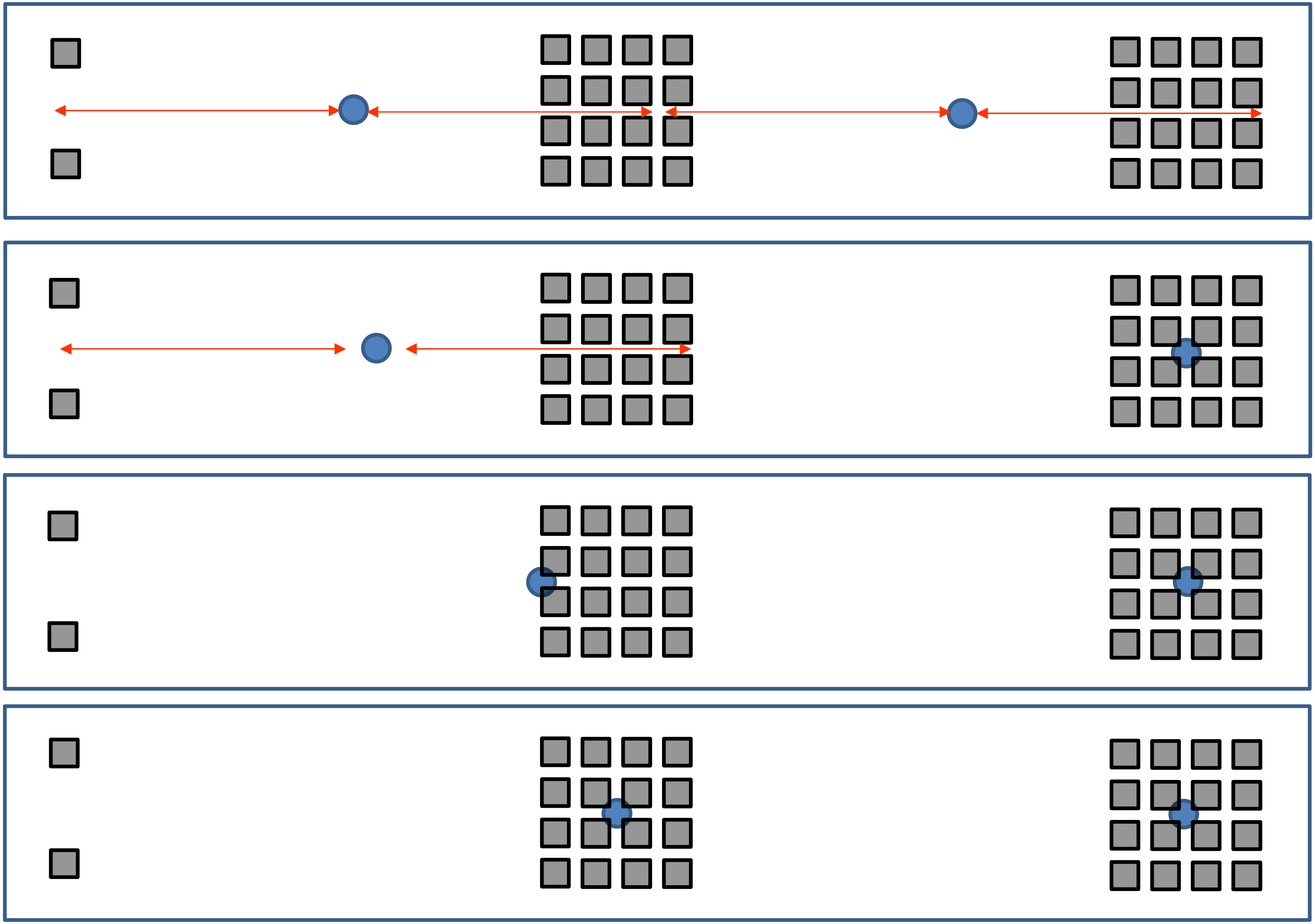
(2) **Minimization of the total squared distance** from each site to the nearest center

$$\text{Minimize } \sum_{s \in S} dist(s, C)^2$$

Example of a good solution



Which is the best solution?



Comparison of Algorithms:

(1) Minimization of the maximum distance from each site to the nearest center.

$$\text{Minimize } \max_{s \in S} dist(s, C)$$

Center Selection Algorithm:

Simple heuristics (a greedy algorithm)

2-Approximation algorithm

(2) Minimization of the total squared distance from each site to the nearest center

$$\text{Minimize } \sum_{s \in S} dist(s, C)^2$$

K-means Algorithm

Iterative adjustment algorithm

(iterations of two greedy algorithms)

Not an exact optimization algorithm

Comparison of Algorithms:

(2) Minimization of the total squared distance from each site to the nearest center

$$\text{Minimize} \quad \sum_{s \in S} dist(s, C)^2$$

k-means Algorithm

Iterative adjustment algorithm
(iterations of two greedy algorithms)
Not an exact optimization algorithm

(3) Minimization of the total distance from each site to the nearest center

$$\text{Minimize} \quad \sum_{s \in S} dist(s, C)$$

k-medoids Algorithm

Iterative adjustment algorithm
(iterations of two greedy algorithms)
Not an exact optimization algorithm

k-means Algorithm (start with *k* centers):

Iterate the following two steps from k centers $c_1, c_2, \ldots, c_k$:

(i) $S_j = \{s \mid dist(s, c_j) = \min_{l=1,2,\ldots,k} dist(s, c_l)\}, \ \ j = 1, 2, \ldots, k.$

(ii) $c_j = \dfrac{1}{|S_j|} \sum_{s \in S_j} s, \quad j = 1, 2, \ldots, k.$

k-medoids Algorithm (start with *k* centers):

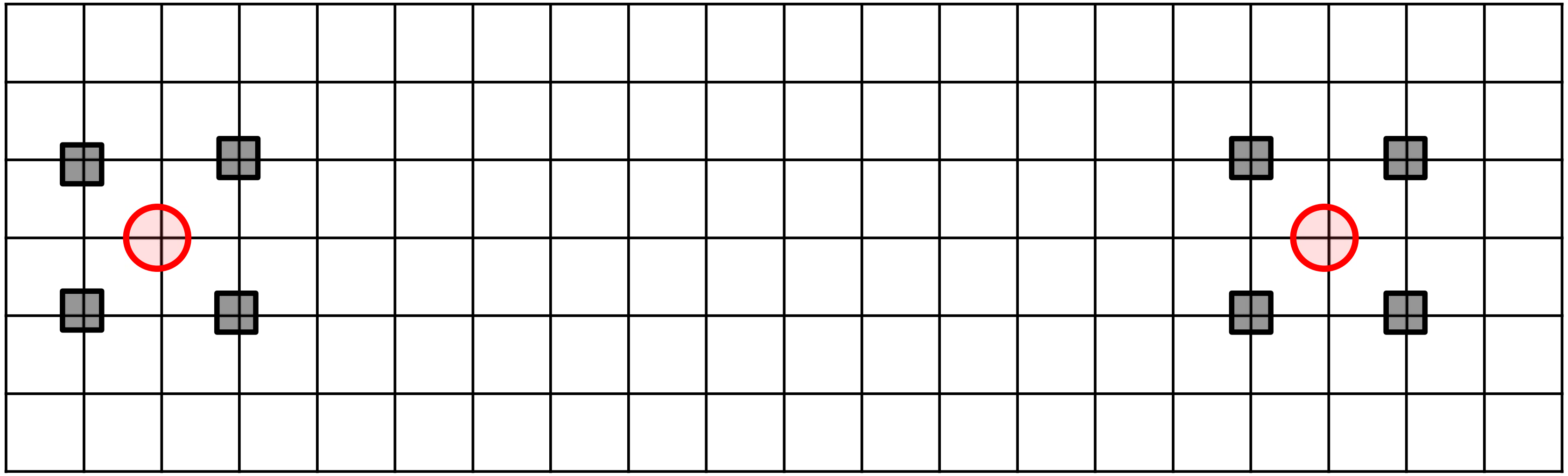
Iterate the following two steps from k centers $c_1, c_2, \ldots, c_k$:

(i) $S_j = \{s \mid dist(s, c_j) = \min_{l=1,2,\ldots,k} dist(s, c_l)\}, \ \ j = 1, 2, \ldots, k.$

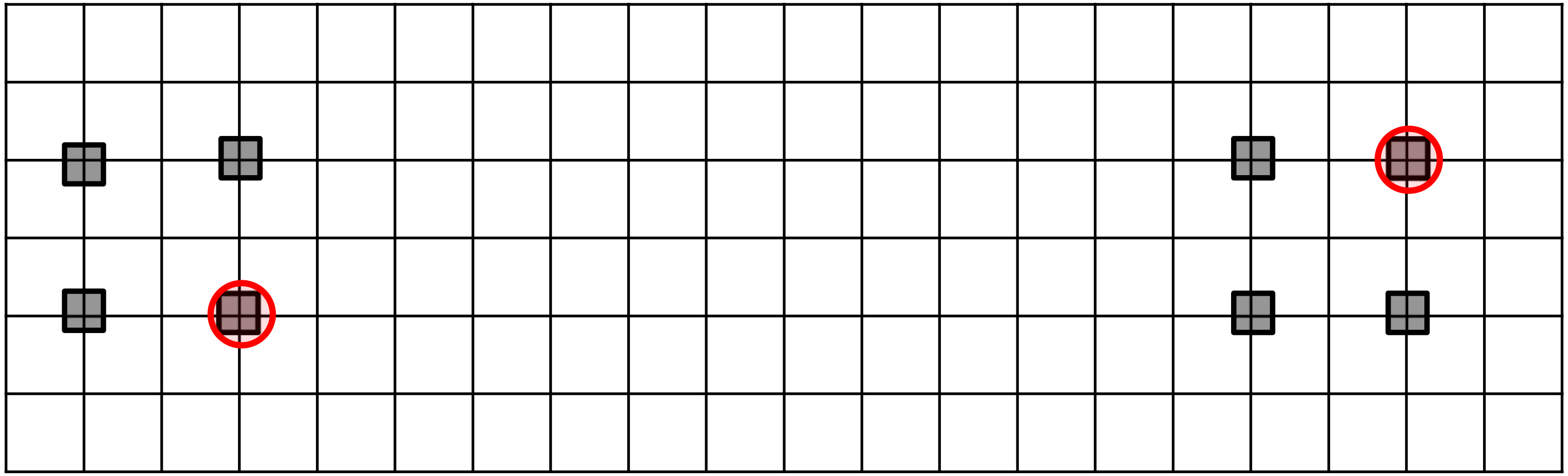
(ii) In each cluster S_j ($j = 1, 2, \ldots, k$), choose the site c_j which minimizes the total distance to all the other sites in cluster S_j.

Medoid: (mathematics) A mathematically representative object in a set of objects; it has the smallest average dissimilarity to all other objects in the set.

k-means Algorithm



k-medoids Algorithm

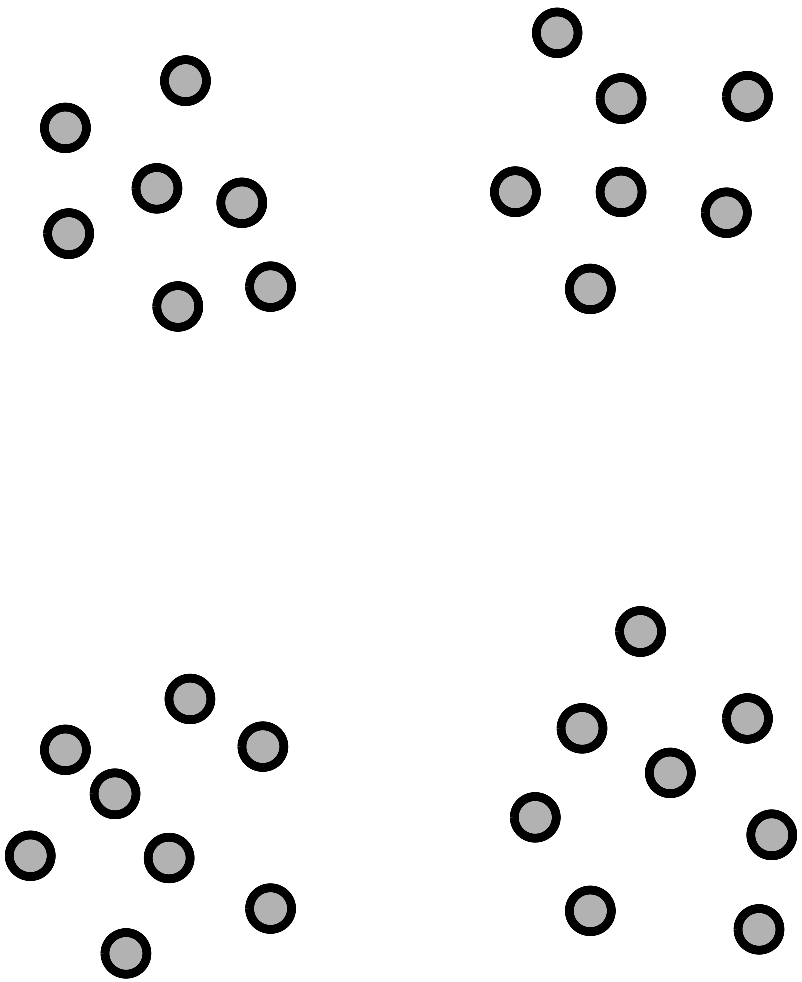


<u>Exercise 7-2</u>:

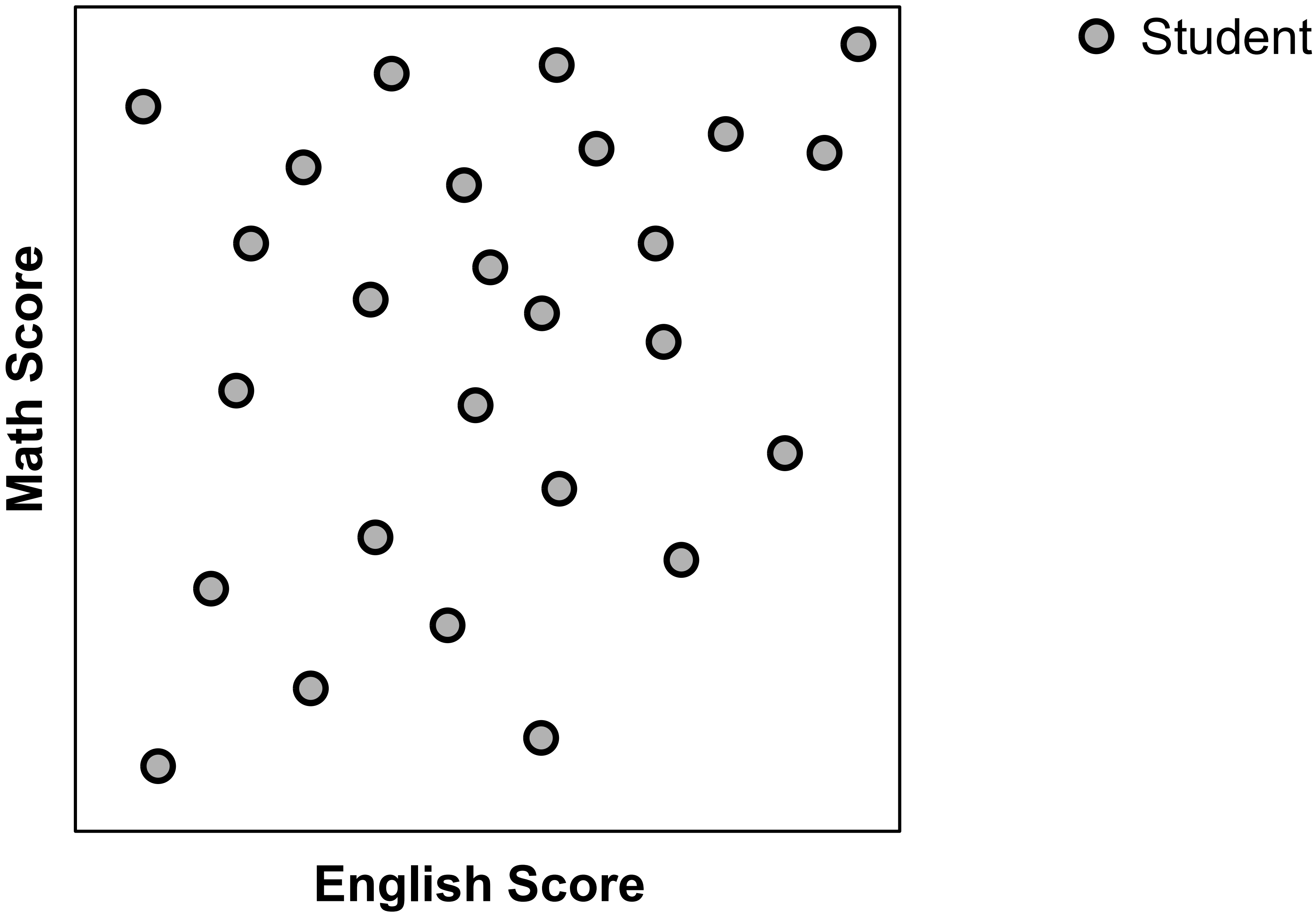
Clearly demonstrate the difference between the *k*-means algorithm and the k-medoids algorithm using a test data set (i.e., create a test data set which can be used for clearly demonstrating the difference between the *k*-means algorithm and the k-medoids algorithm).

Recent Hot Topic: Fairness in Clustering

Problem: Find two clusters (k = 2)



Classification Problem: Accept 10 Students



Related Topic: Fuzzy Clustering

Input: n sites: $S = \{s_1, s_2, \ldots, s_n\}$, and a constant m ($m > 1$)

Output: Locations of k centers: $C = \{c_1, c_2, \ldots, c_k\}$

Membership of s_i to c_j: μ_{ij} ($i = 1, \ldots, n$; $j = 1, \ldots, k$)

Objective: Minimize the weighted total squared distance from each site to each center.

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{k}(\mu_{ij})^m dist(s_i, c_j)^2$$

where

$$0 \le \mu_{ij} \le 1, \; i = 1, 2, \ldots, n; \; j = 1, 2, \ldots, k$$

$$\sum_{j=1}^{k}\mu_{ij} = 1, \; i = 1, 2, \ldots, n$$

Related Topic: Fuzzy Clustering

Input: n sites: $S = \{s_1, s_2, \ldots, s_n\}$, and a constant m ($m > 1$)

Output: Locations of k centers: $C = \{c_1, c_2, \ldots, c_k\}$

Membership of s_i to c_j: μ_{ij} ($i = 1, \ldots, n$; $j = 1, \ldots, k$)

Objective: Minimize the weighted total squared distance from each site to each center.

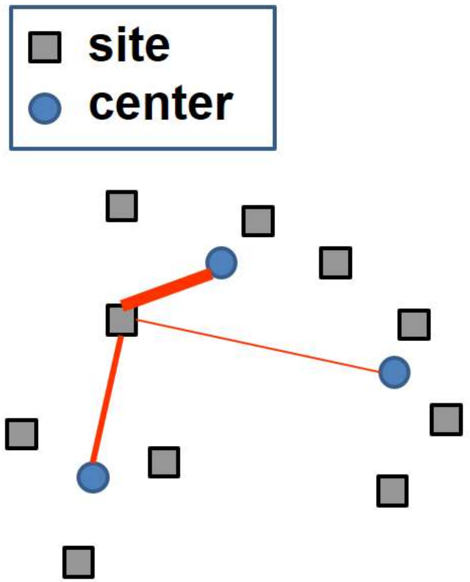
$$\text{Minimize} \quad \sum_{i=1}^{n} \sum_{j=1}^{k} (\mu_{ij})^m \, dist(s_i, c_j)^2$$

Membership grade

where

$$0 \le \mu_{ij} \le 1, \; i = 1, 2, \ldots, n; \; j = 1, 2, \ldots, k$$

$$\sum_{j=1}^{k} \mu_{ij} = 1, \; i = 1, 2, \ldots, n$$



fuzzy *c*-means

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{k}(\mu_{ij})^{m} dist(s_i, c_j)^2$$

$$0 \le \mu_{ij} \le 1,\ i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, k$$

$$\sum_{j=1}^{k}\mu_{ij} = 1,\ i = 1, 2, \ldots, n$$

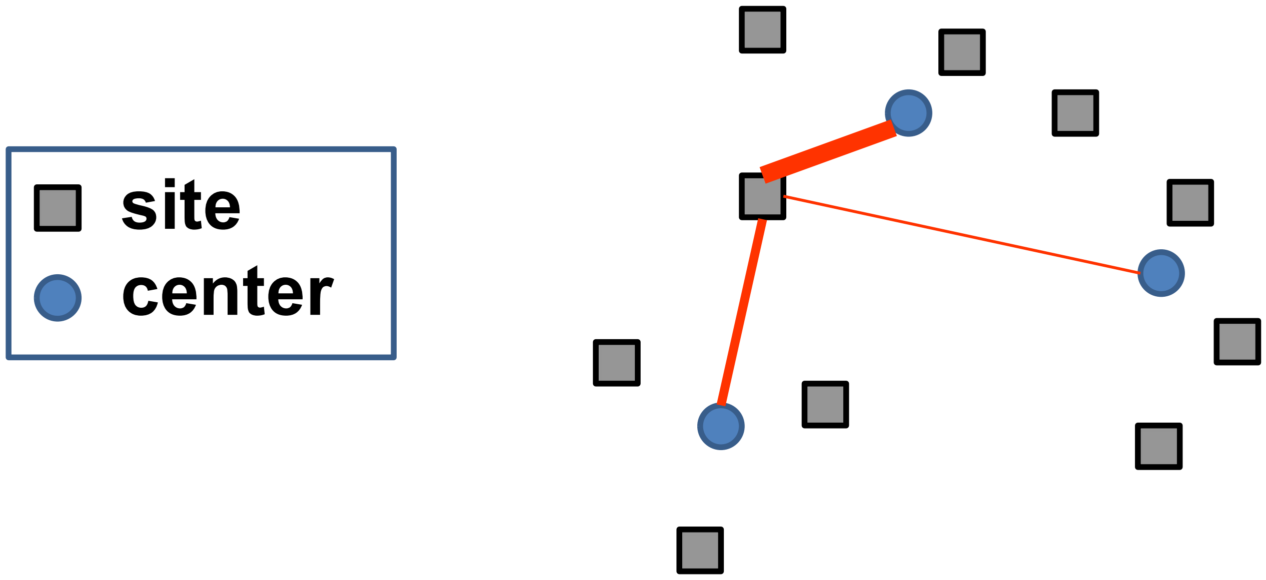
k-means

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{k}(\mu_{ij})^{1} dist(s_i, c_j)^2$$

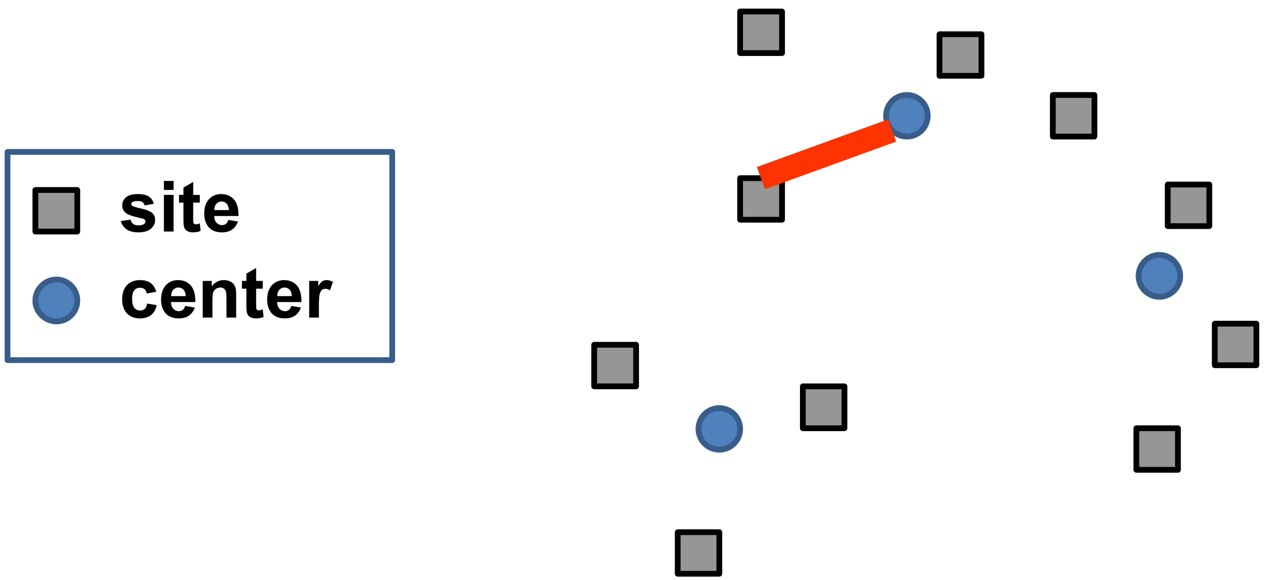
$$\boxed{\mu_{ij} = 0 \text{ or } 1,\ i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, k}$$

$$\sum_{j=1}^{k}\mu_{ij} = 1,\ i = 1, 2, \ldots, n$$

fuzzy *c*-means



k-means



Fuzzy *c*-means Algorithm: **Iterate the following two steps from randomly specified values of μ_{ij}**

$$\boxed{\sum_{j=1}^{k} \mu_{ij} = 1,\ i = 1, 2, \ldots, n}$$

(i) $$c_j = \frac{\sum_{i=1}^{n} (\mu_{ij})^m s_i}{\sum_{i=1}^{n} (\mu_{ij})^m},\ j = 1, 2, \ldots, k$$

(ii) $$\mu_{ij} = \left[\sum_{h=1}^{k} \left(\frac{dist(s_i, c_j)}{dist(s_i, c_h)} \right)^{\frac{2}{m-1}} \right]^{-1} \quad \text{for all } i \text{ and } j$$

$$\mu_{ij} = \left[\sum_{h=1}^{k} \left(\frac{dist(s_i, c_j)}{dist(s_i, c_h)} \right)^{\frac{2}{m-1}} \right]^{-1} \quad \text{for all } i \text{ and } j$$

When $m \to \infty$

$$\mu_{ij} \to \left[\sum_{h=1}^{k} \left(\frac{dist(s_i, c_j)}{dist(s_i, c_h)} \right)^{0} \right]^{-1} = 1/k$$

When $m \to 1$

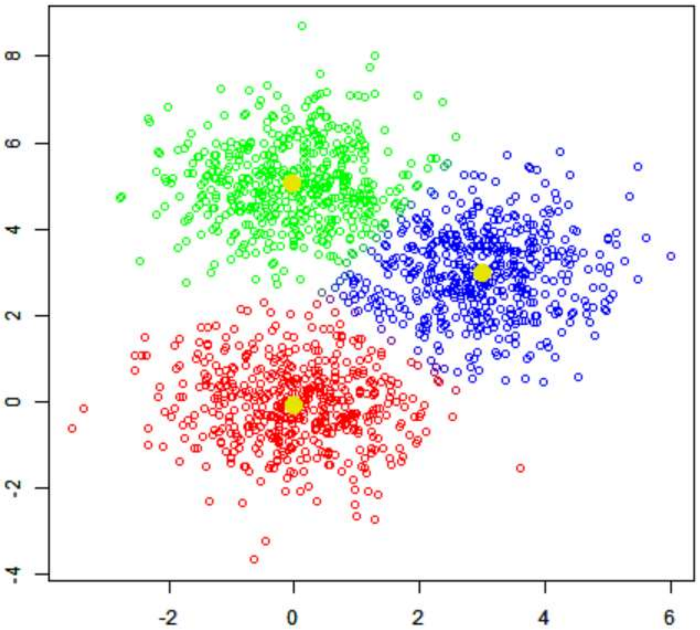
$$\mu_{ij} \to \left[\sum_{h=1}^{k} \left(\frac{dist(s_i, c_j)}{dist(s_i, c_h)} \right)^{\infty} \right]^{-1} = 0 \text{ or } 1$$

Exercise 7-3:

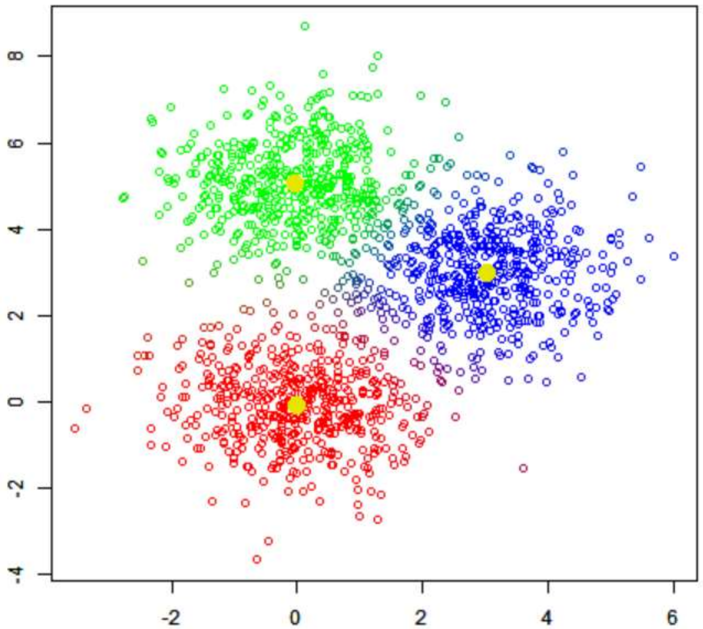
Clearly demonstrate the effects of *m* on the clustering results by the fuzzy c-means algorithm through computational experiments on a test data set (i.e., create a test data set which can be used for clearly demonstrating the effects of *m*). Try to create some beautiful figures.

Exercise 7-4:

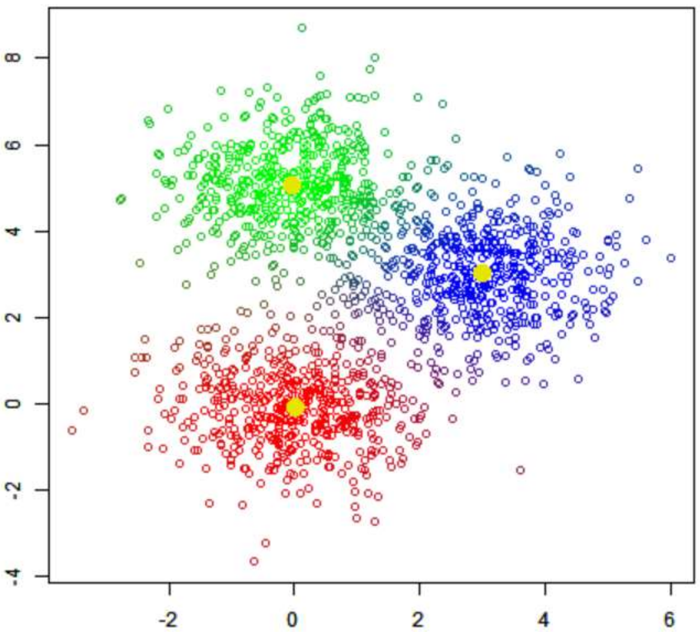
Clearly demonstrate the difference between the k-means algorithm and the fuzzy c-means algorithm through computational experiments on a test data set (i.e., create a test data set which can be used for clearly demonstrating the difference between the k-means algorithm and the fuzzy c-means algorithm). Try to create some beautiful figures.



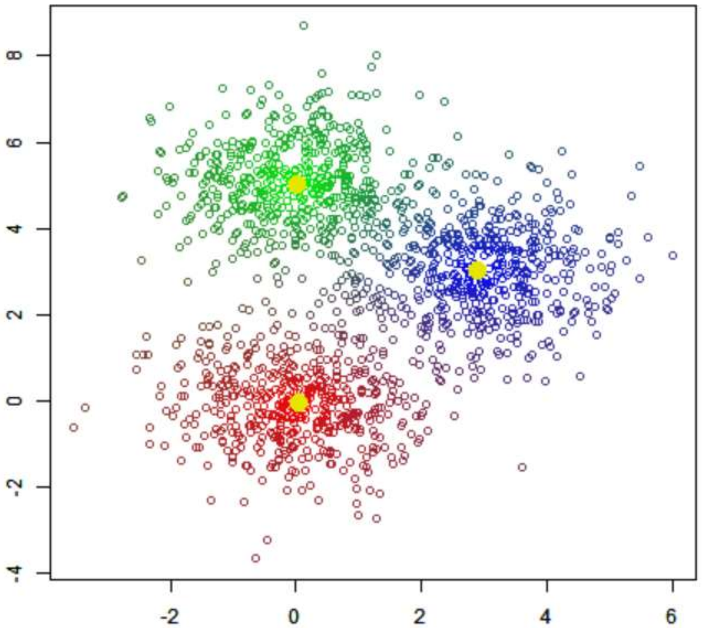
m = 1.1



m = 1.5



m = 2



m = 3

Use of a user-defined hyper parameter (*m* in fuzzy c-means)

Positive Aspects:

A more desirable result can be obtained by appropriately specifying the value of *m* (than the case of the fixed value of m)

Different results can be obtained by examining different values of *m* (we can choose one of them based on our preference).

Negative Aspects:

It is not always easy to appropriately specify the value of *m*.

Undesirable results can be obtained when the value of *m* is inappropriate.

Example: **If m is specified as $m = 1$, the algorithm does not work.**

(i) $$c_j = \frac{\sum_{i=1}^{n} (\mu_{ij})^m s_i}{\sum_{i=1}^{n} (\mu_{ij})^m}, \quad j = 1, 2, \ldots, k$$

(ii) $$\mu_{ij} = \left[\sum_{h=1}^{k} \left(\frac{dist(s_i, c_j)}{dist(s_i, c_h)} \right)^{\frac{2}{m-1}} \right]^{-1} \quad \text{for all } i \text{ and } j$$