

# Clustering

Shusen Wang

# Clustering Task

Tasks

clustering

Methods

K-means

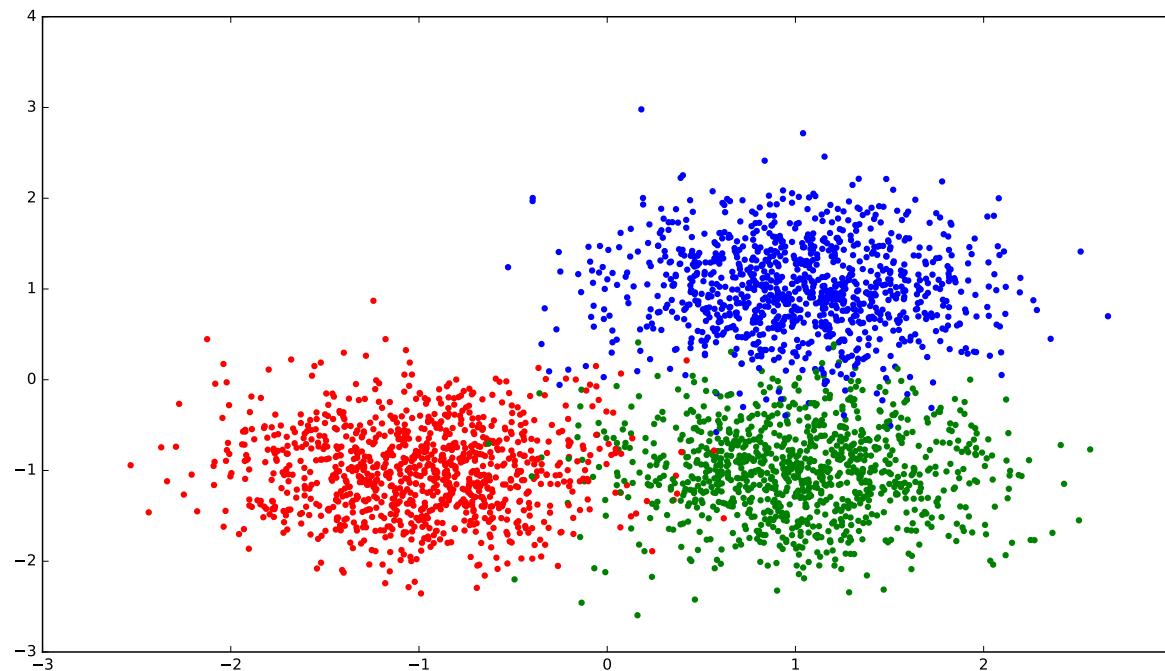
Algorithms

Lloyd's algorithm

# Clustering Task

**Input:** vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$  and cluster number  $k$  ( $\ll n$ ).

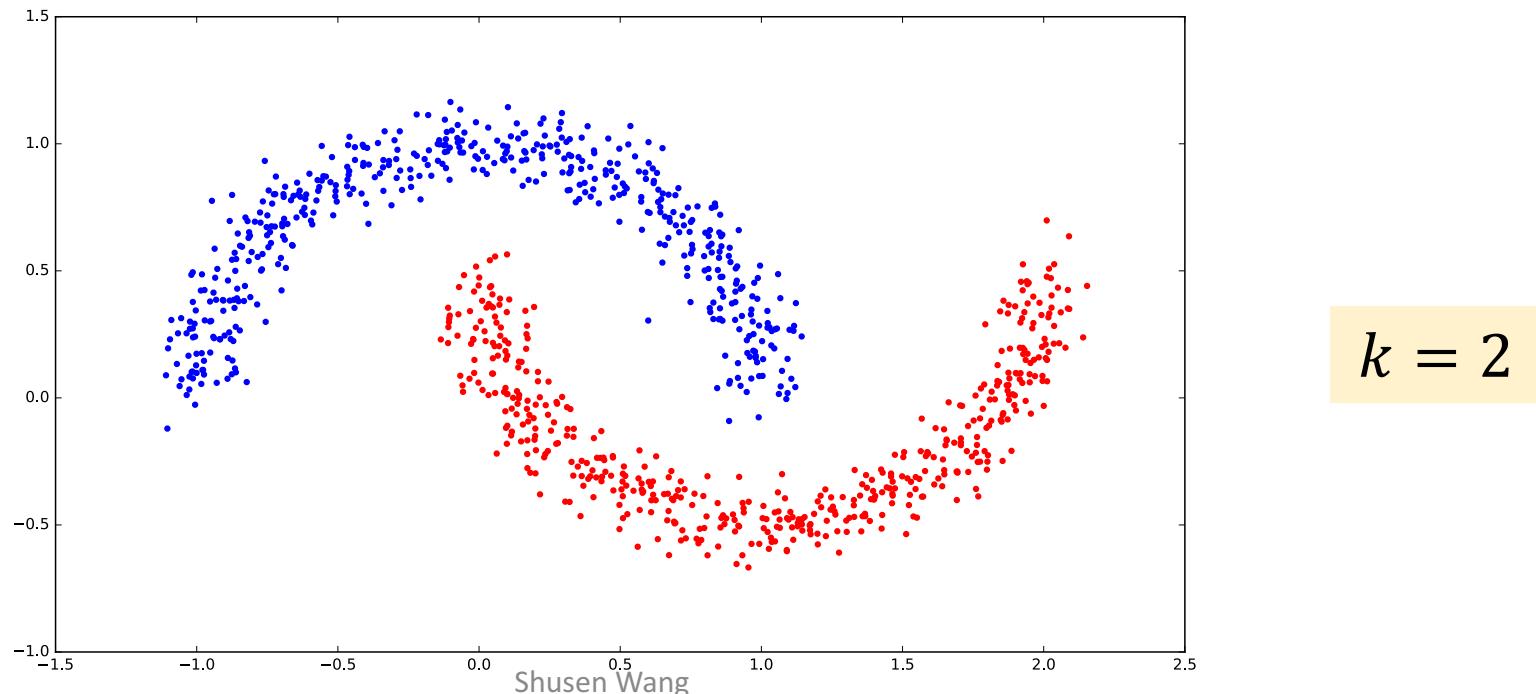
**Output:** predicted class labels  $y_1, \dots, y_n \in \{1, 2, \dots, k\}$ .



# Clustering Task

**Input:** vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$  and cluster number  $k$  ( $\ll n$ ).

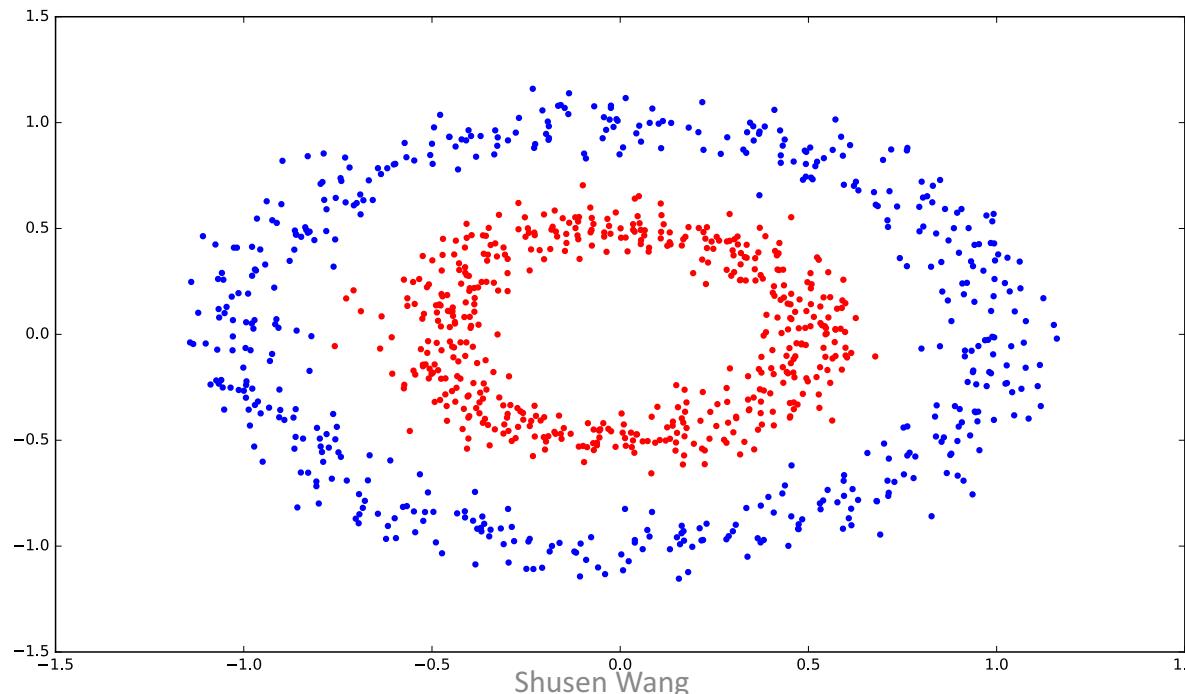
**Output:** predicted class labels  $y_1, \dots, y_n \in \{1, 2, \dots, k\}$ .



# Clustering Task

**Input:** vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$  and cluster number  $k$  ( $\ll n$ ).

**Output:** predicted class labels  $y_1, \dots, y_n \in \{1, 2, \dots, k\}$ .



$k = 2$

# Clustering Task

**Input:** vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$  and cluster number  $k$  ( $\ll n$ ).

**Output:** predicted class labels  $y_1, \dots, y_n \in [k]$ .

denote  $[k] = \{1, 2, \dots, k\}$

# Clustering Task

**Input:** vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$  and cluster number  $k (\ll n)$ .

**Output:** predicted class labels  $y_1, \dots, y_n \in [k]$ .

**Equivalent definition:**

**Clustering:** partition  $[n]$  to  $k$  disjoint subsets  $S_1, \dots, S_k$ , 将n个样本 划分到 不相交的子集S1...Sk

- $S_1 \cup \dots \cup S_k = [n]$ ,
- $S_i \cap S_j = \emptyset$ , for all  $i \neq j$ .

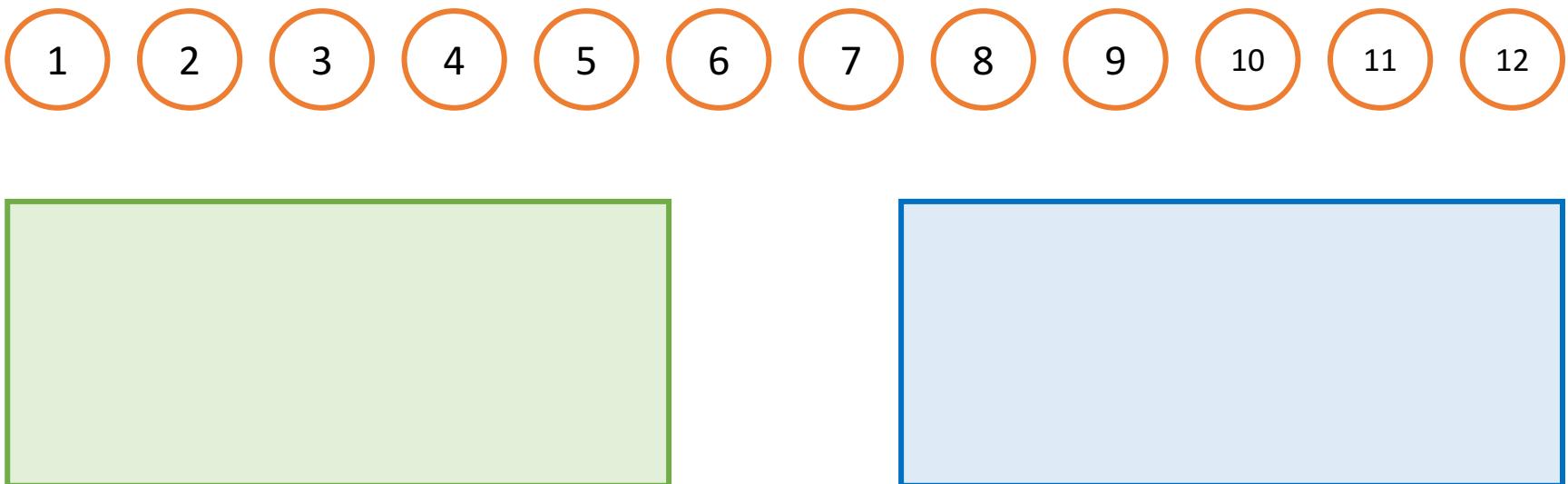
# Clustering Task

**Clustering:** partition  $[n]$  to  $k$  disjoint subsets  $S_1, \dots, S_k$ ,

- $S_1 \cup \dots \cup S_k = [n]$ ,
- $S_i \cap S_j = \emptyset$ , for all  $i \neq j$ .

Example

- $n = 12$
- $k = 2$



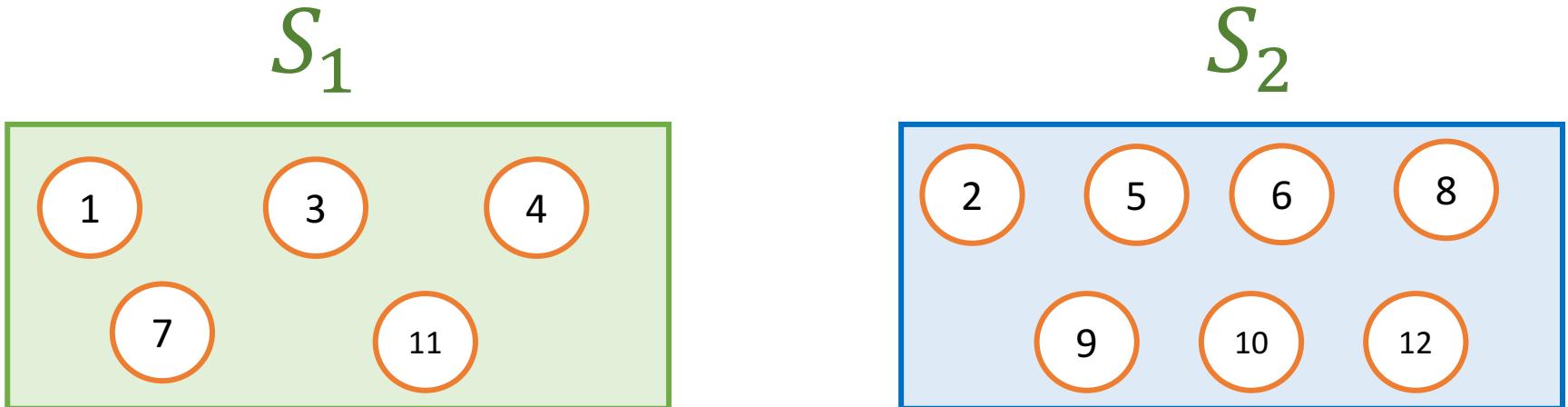
# Clustering Task

**Clustering:** partition  $[n]$  to  $k$  disjoint subsets  $S_1, \dots, S_k$ ,

- $S_1 \cup \dots \cup S_k = [n]$ ,
- $S_i \cap S_j = \emptyset$ , for all  $i \neq j$ .

Example

- $n = 12$
- $k = 2$



# K-Means Clustering Method

Tasks

clustering

Methods

K-means

Algorithms

Lloyd's algorithm

# K-Means Clustering Method

**Clustering:** partition  $[n]$  to  $k$  disjoint subsets  $S_1, \dots, S_k$ ,

- $S_1 \cup \dots \cup S_k = [n]$ ,
- $S_i \cap S_j = \emptyset$ , for all  $i \neq j$ .

**K-Means Model:**

$$\min_{S_1, \dots, S_k} \sum_{i=1}^k \sum_{j \in S_i} \left\| \mathbf{x}_j - \frac{1}{|S_i|} \sum_{l \in S_i} \mathbf{x}_l \right\|_2^2.$$

- $\mathbf{x}_j$  indexed by  $S_i$

# K-Means Clustering Method

**Clustering:** partition  $[n]$  to  $k$  disjoint subsets  $S_1, \dots, S_k$ ,

- $S_1 \cup \dots \cup S_k = [n]$ ,
- $S_i \cap S_j = \emptyset$ , for all  $i \neq j$ .

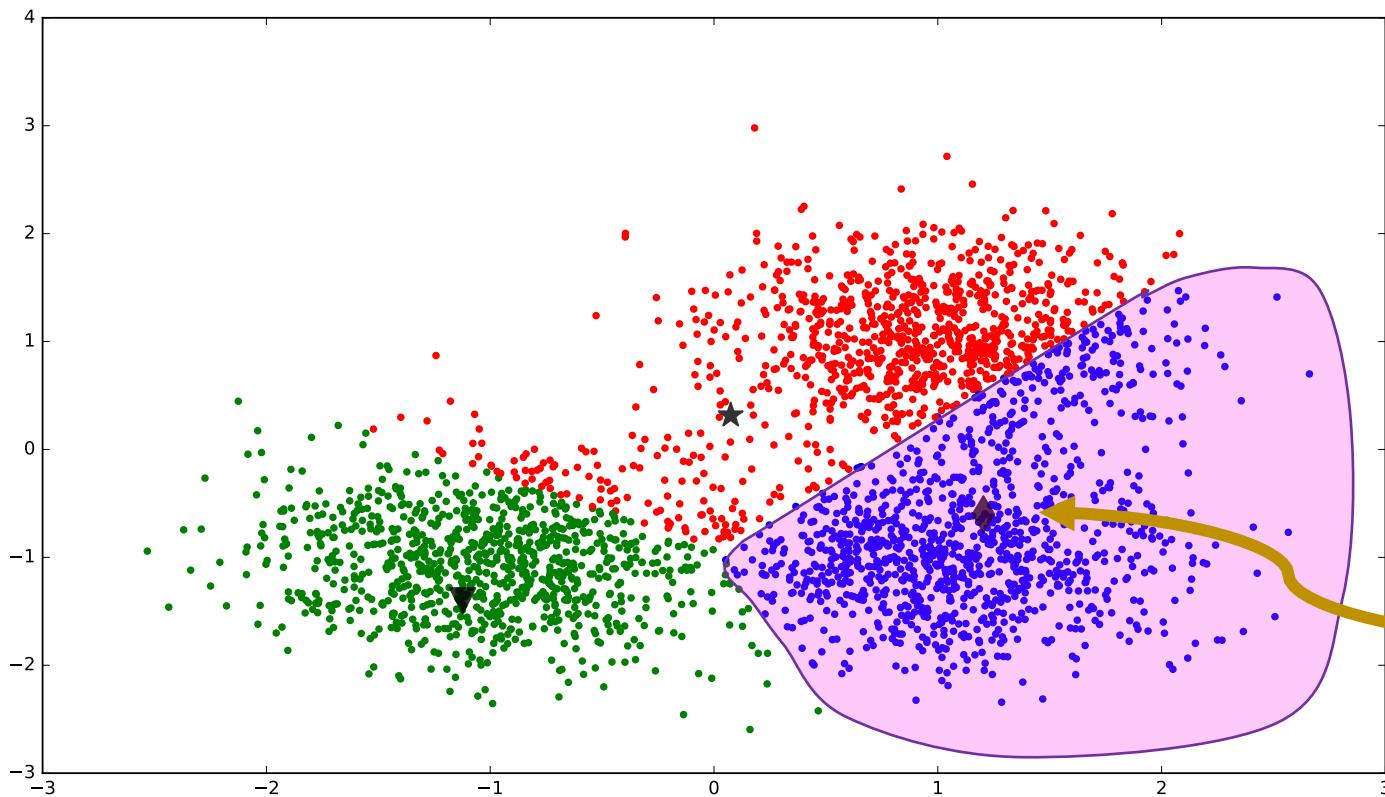
**K-Means Model:**

$$\min_{S_1, \dots, S_k} \sum_{i=1}^k \sum_{j \in S_i} \left\| \mathbf{x}_j - \frac{1}{|S_i|} \sum_{l \in S_i} \mathbf{x}_l \right\|_2^2.$$

- $\mathbf{x}_j$  indexed by  $S_i$
- The centroid of the  $i$ -th cluster.

第*i*个簇的中心

# K-Means Clustering Method



- $\mathbf{x}_j$  indexed by  $S_i$
- The centroid of the  $i$ -th cluster.

# K-Means Clustering Method

**Clustering:** partition  $[n]$  to  $k$  disjoint subsets  $S_1, \dots, S_k$ ,

- $S_1 \cup \dots \cup S_k = [n]$ ,
- $S_i \cap S_j = \emptyset$ , for all  $i \neq j$ .

**K-Means Model:**

$$\min_{S_1, \dots, S_k} \sum_{i=1}^k \sum_{j \in S_i} \left\| \mathbf{x}_j - \frac{1}{|S_i|} \sum_{l \in S_i} \mathbf{x}_l \right\|_2^2.$$

- Squared Euclidean distance between  $\mathbf{x}_j$  and its cluster centroid.

$\mathbf{x}_j$  和 聚类中心 之间的 欧式距离

# K-Means Clustering Method

**Clustering:** partition  $[n]$  to  $k$  disjoint subsets  $S_1, \dots, S_k$ ,

- $S_1 \cup \dots \cup S_k = [n]$ ,
- $S_i \cap S_j = \emptyset$ , for all  $i \neq j$ .

**K-Means Model:**

$$\min_{S_1, \dots, S_k} \sum_{i=1}^k \sum_{j \in S_i} \left\| \mathbf{x}_j - \frac{1}{|S_i|} \sum_{l \in S_i} \mathbf{x}_l \right\|_2^2.$$

- It is a combinatorial optimization problem.
- NP-hard!

P : Polynomial 多项式

这是一个组合优化问题  
NP-hard 难题

一种是 $O(1), O(\log(n)), O(n^a)$ 等，我们把它叫做多项式级的复杂度，因为它的规模 $n$ 出现在底数的位置；另一种是 $O(a^n)$ 和 $O(n!)$ 型复杂度，它是非多项式级的，其复杂度计算机往往不能承受。  
如果一个问题验证解的正确性需要花费多项式时间，而求解花费多少时间不知道，它就是NP问题

# Lloyd's Algorithm

Tasks

clustering

Methods

K-means

Algorithms

Lloyd's algorithm

# Lloyd's Algorithm

劳埃德算法

**Input:** vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$  and cluster number  $k$  ( $\ll n$ ).

1. Initialize cluster centroids  $\mathbf{c}_1, \dots, \mathbf{c}_k \in \mathbb{R}^d$ .

**Example 1:** Randomly select  $k$  points from  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  as the cluster centroids .

随机选择  $k$  个点，作为聚类中心

# Lloyd's Algorithm

**Input:** vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$  and cluster number  $k$  ( $\ll n$ ).

1. Initialize cluster centroids  $\mathbf{c}_1, \dots, \mathbf{c}_k \in \mathbb{R}^d$ .

**Example 2:**

- Randomly select one point from  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  as the 1<sup>st</sup> centroid,  $\mathbf{c}_1$ .
- Select the point from  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  as  $\mathbf{c}_2$  by maximizing  $\left\| \mathbf{x}_j - \mathbf{c}_1 \right\|_2^2$ .
- Select the point from  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  as  $\mathbf{c}_3$  by maximizing  $\left\| \mathbf{x}_j - \mathbf{c}_1 \right\|_2^2 + \left\| \mathbf{x}_j - \mathbf{c}_2 \right\|_2^2$ .
- And so on...

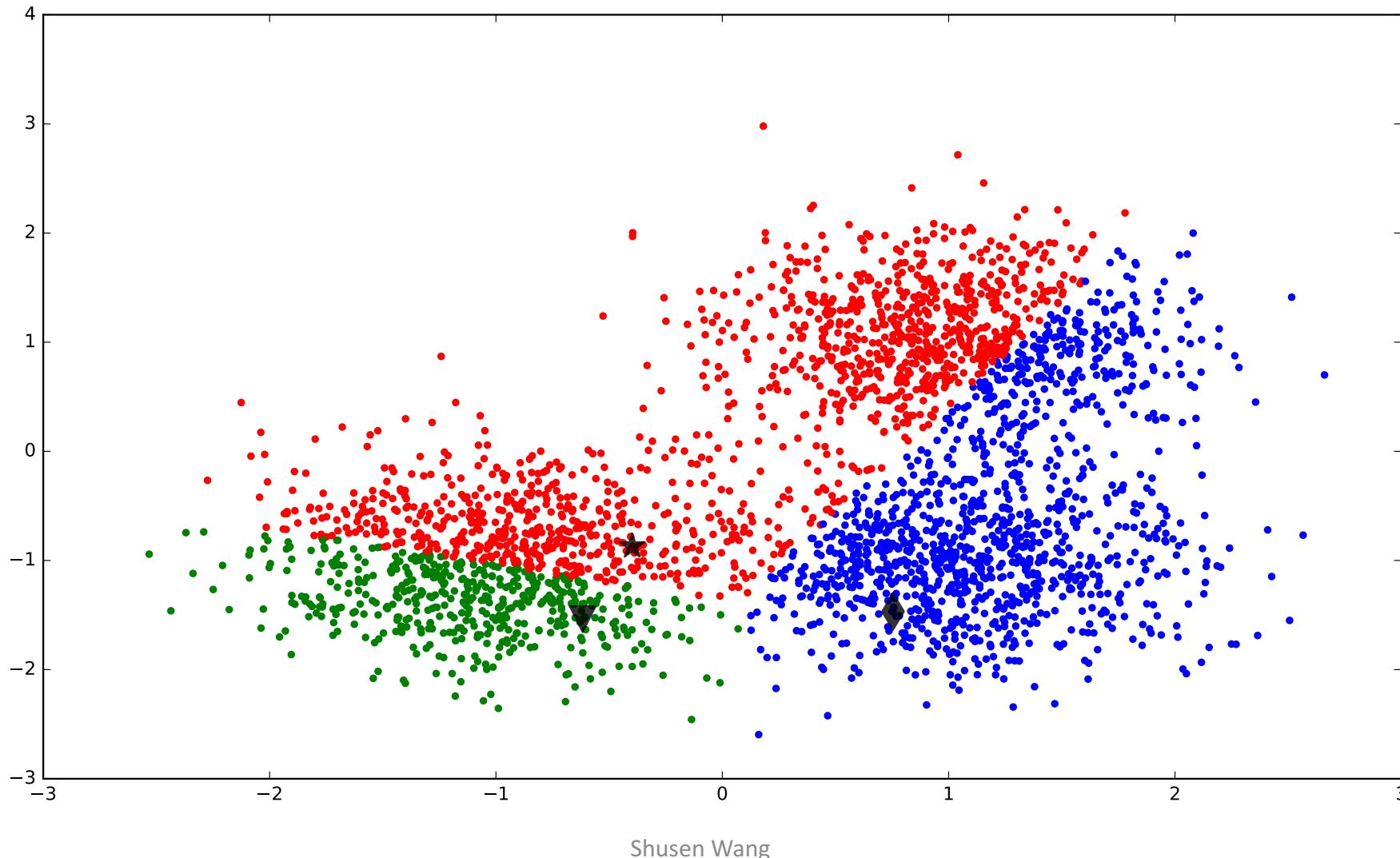
# Lloyd's Algorithm

**Input:** vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$  and cluster number  $k$  ( $\ll n$ ).

1. Initialize cluster centroids  $\mathbf{c}_1, \dots, \mathbf{c}_k \in \mathbb{R}^d$ .
2. Repeat
  - i. For  $j = 1$  to  $n$ , move  $j$  to set  $S_i$ , where  $i$  optimizes
$$\min_{i \in [k]} \left\| \mathbf{x}_j - \mathbf{c}_i \right\|_2^2.$$

Assign  $\mathbf{x}_j$  to its nearest centroid.

# Lloyd's Algorithm



# Lloyd's Algorithm

**Input:** vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$  and cluster number  $k$  ( $\ll n$ ).

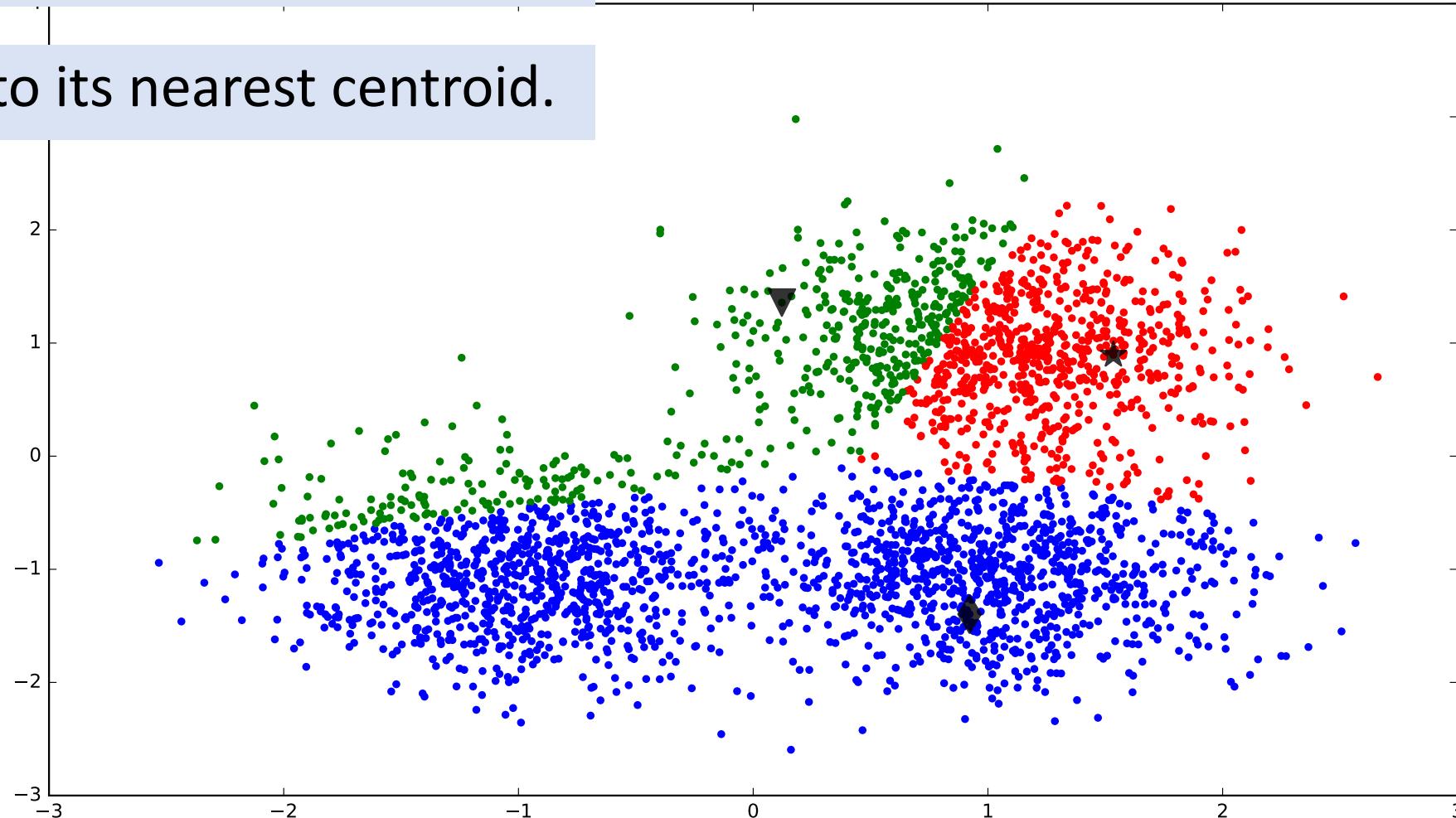
1. Initialize cluster centroids  $\mathbf{c}_1, \dots, \mathbf{c}_k \in \mathbb{R}^d$ .
2. Repeat
  - i. For  $j = 1$  to  $n$ , move  $j$  to set  $S_i$ , where  $i$  optimizes
$$\min_{i \in [k]} \left\| \mathbf{x}_j - \mathbf{c}_i \right\|_2^2.$$
  - ii. For  $i = 1$  to  $k$ , re-compute centroids by  $\tilde{\mathbf{c}}_i = \frac{1}{|S_i|} \sum_{j \in S_i} \mathbf{x}_j$ .
    - New centroid  $\mathbf{c}_i$ : the feature vector (among  $\mathbf{x}_1, \dots, \mathbf{x}_n$ ) closest to  $\tilde{\mathbf{c}}_i$ .

move centroids

# Lloyd's Algorithm

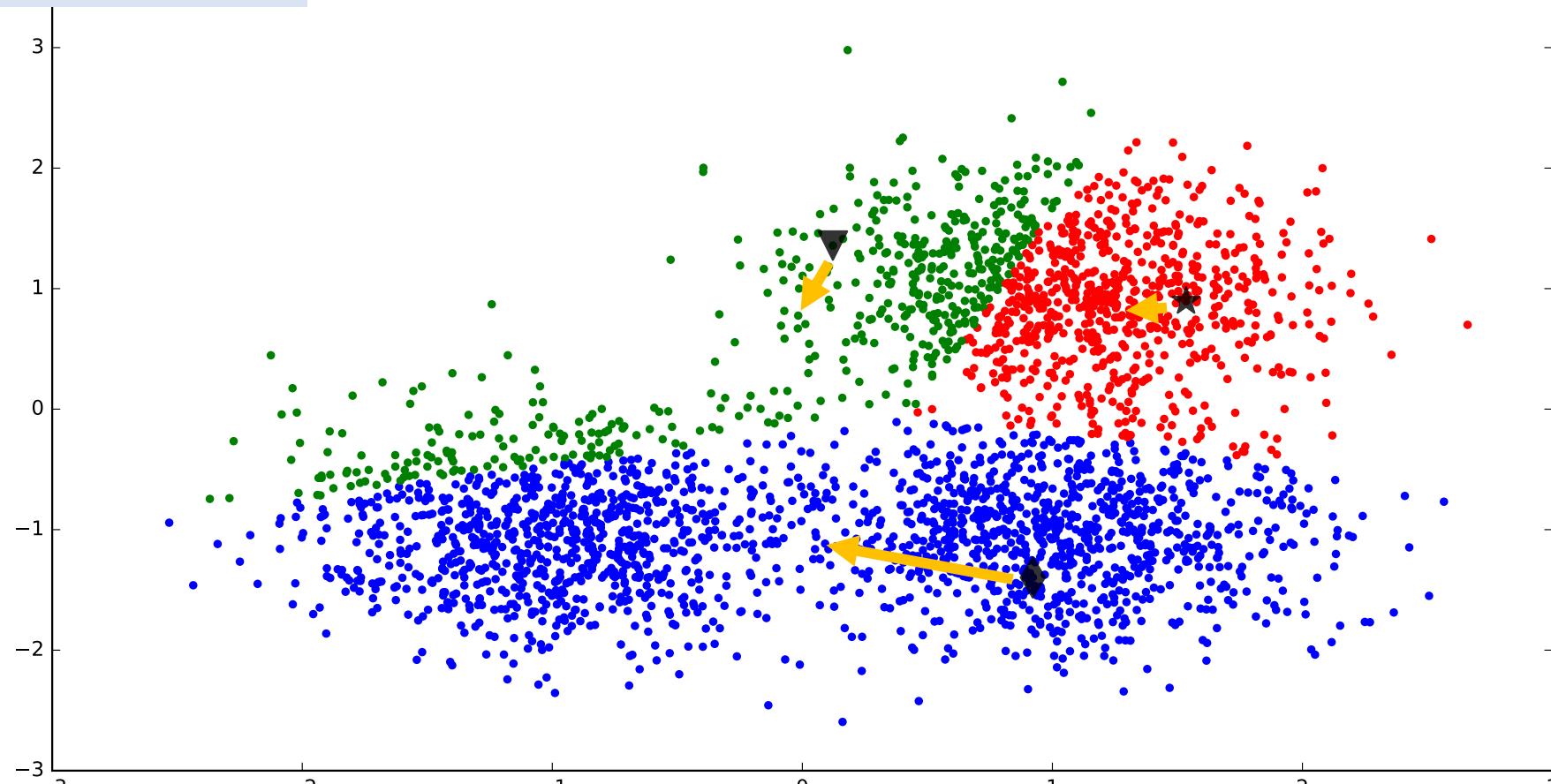
Random initialization.

Assign  $x_j$  to its nearest centroid.



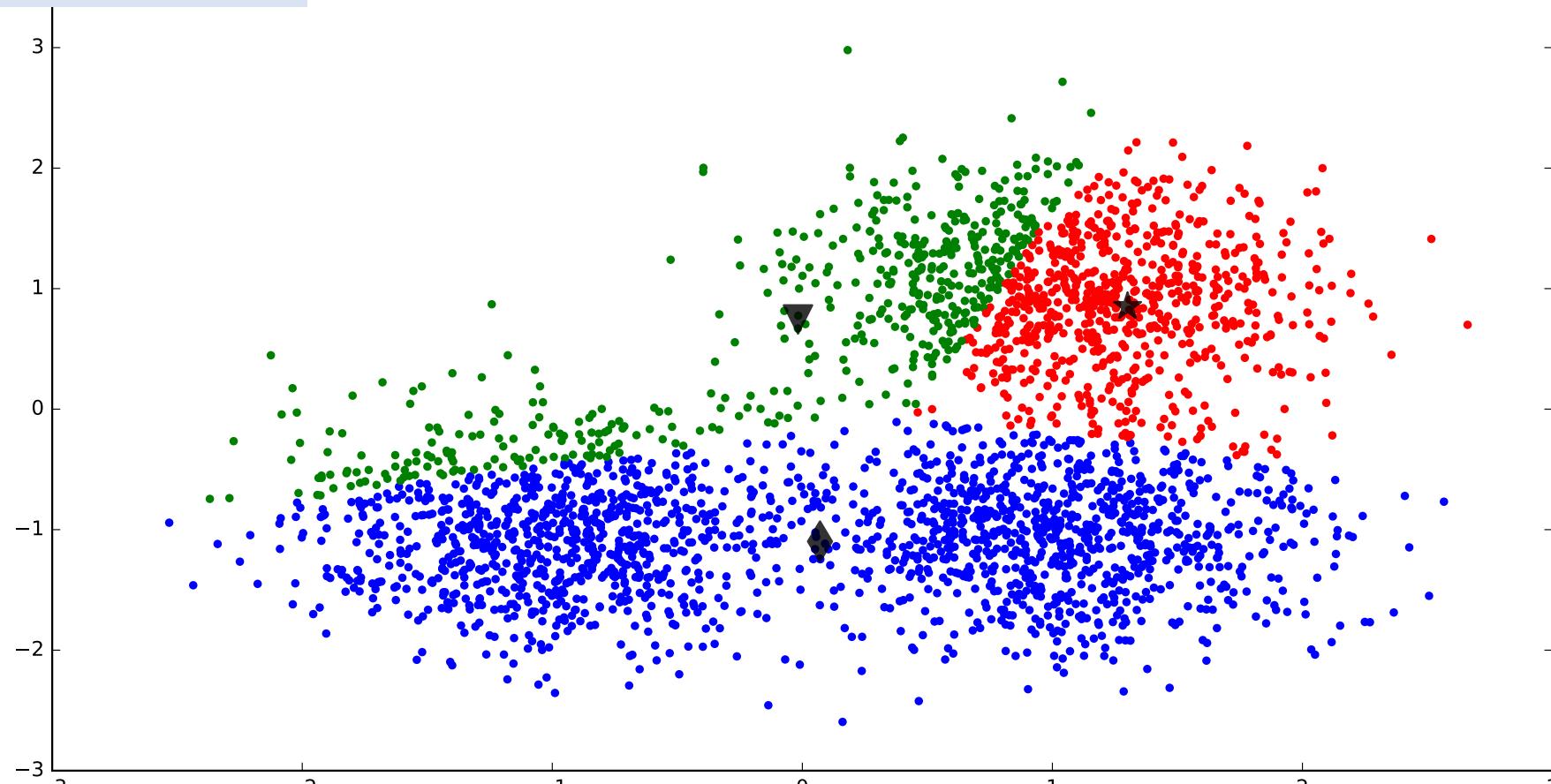
# Lloyd's Algorithm

move centroids



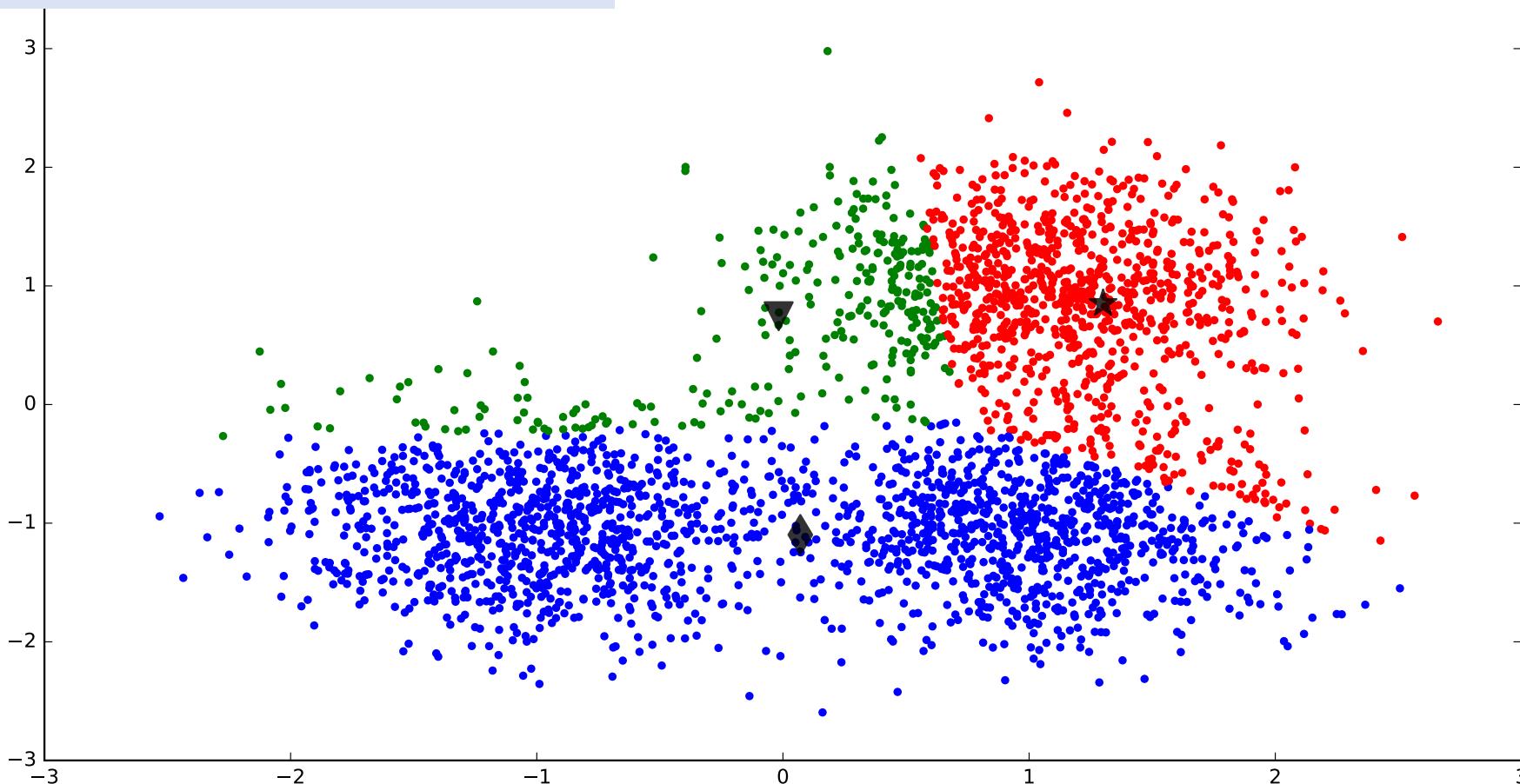
# Lloyd's Algorithm

move centroids



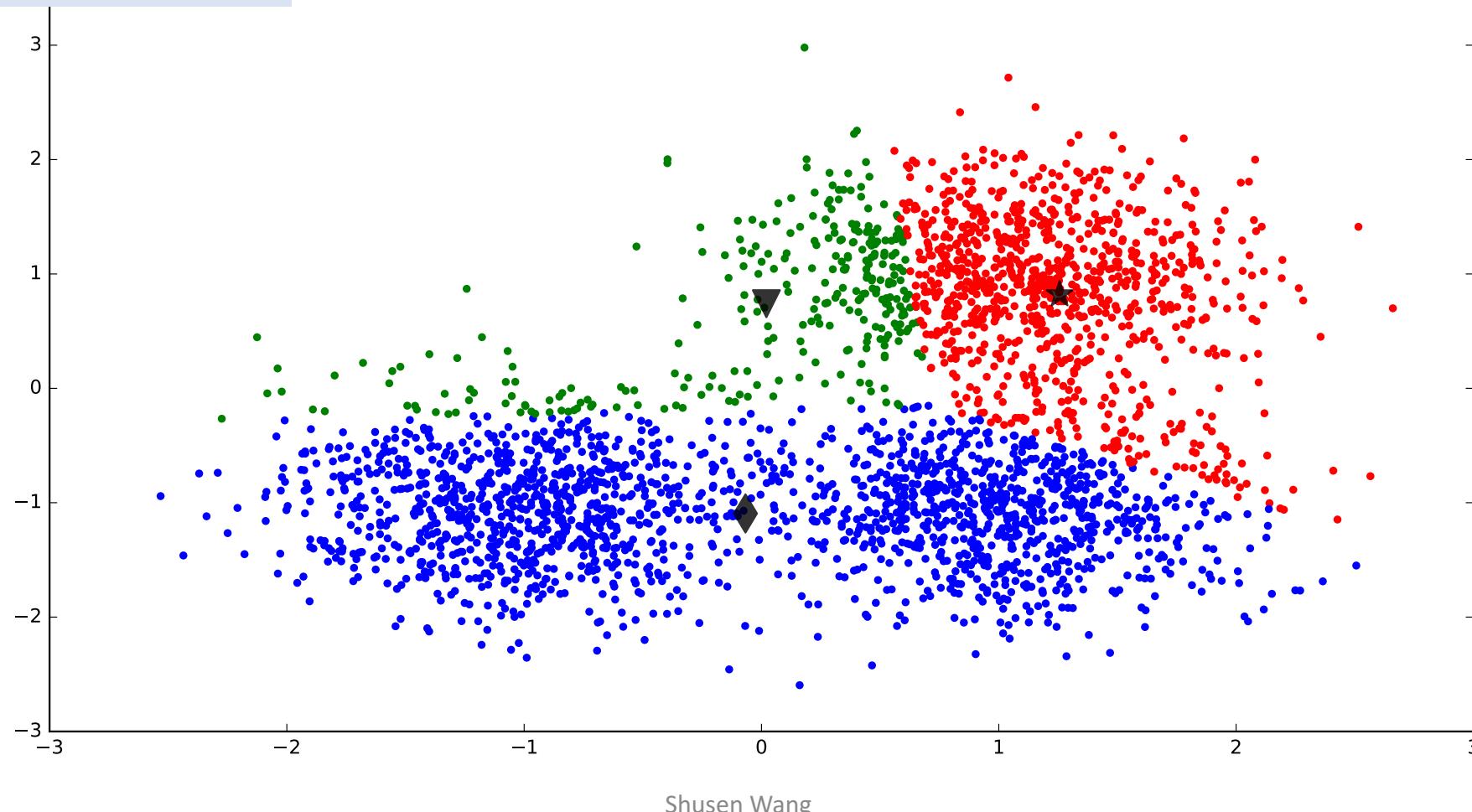
# Lloyd's Algorithm

Assign  $x_j$  to its nearest centroid.



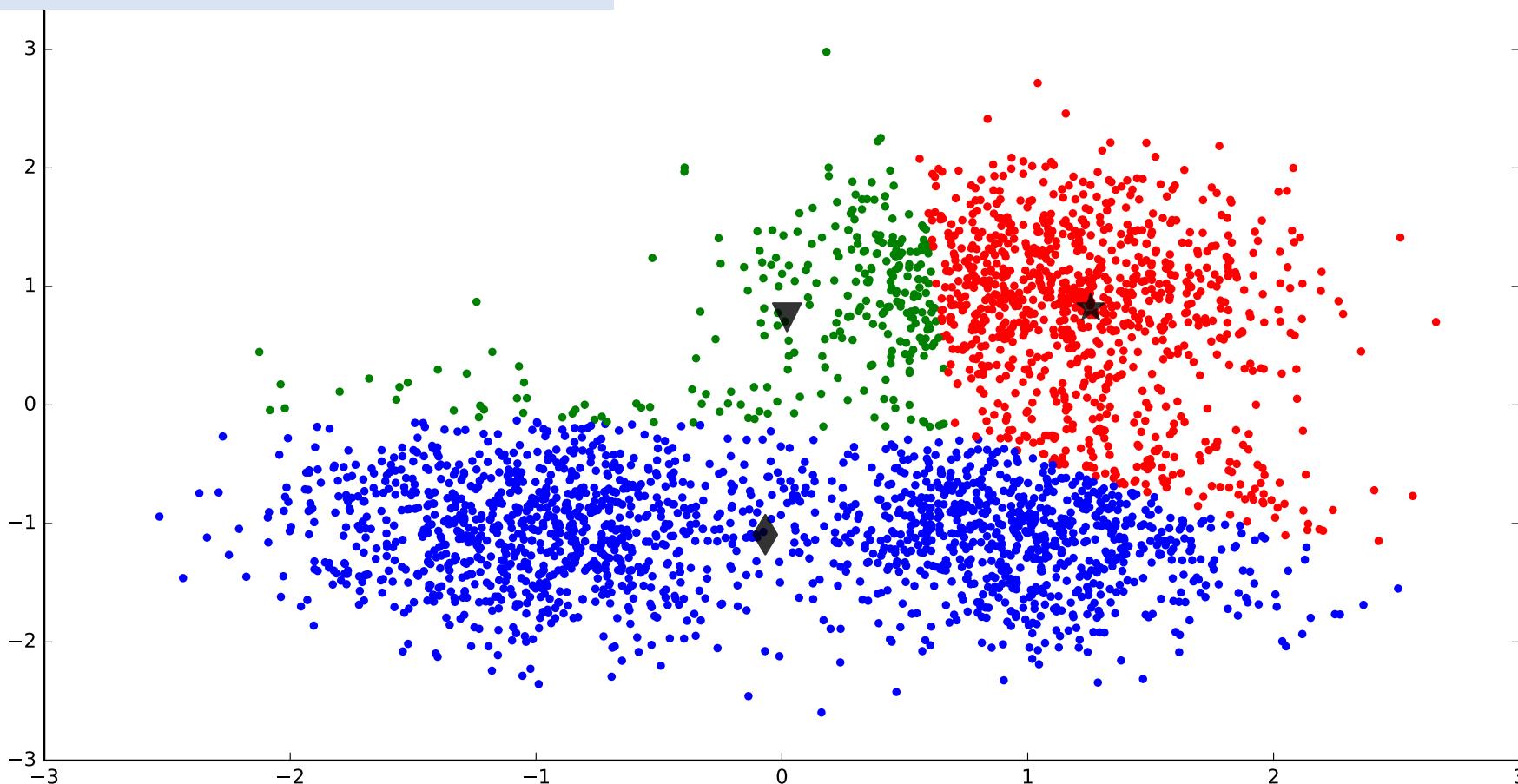
# Lloyd's Algorithm

move centroids



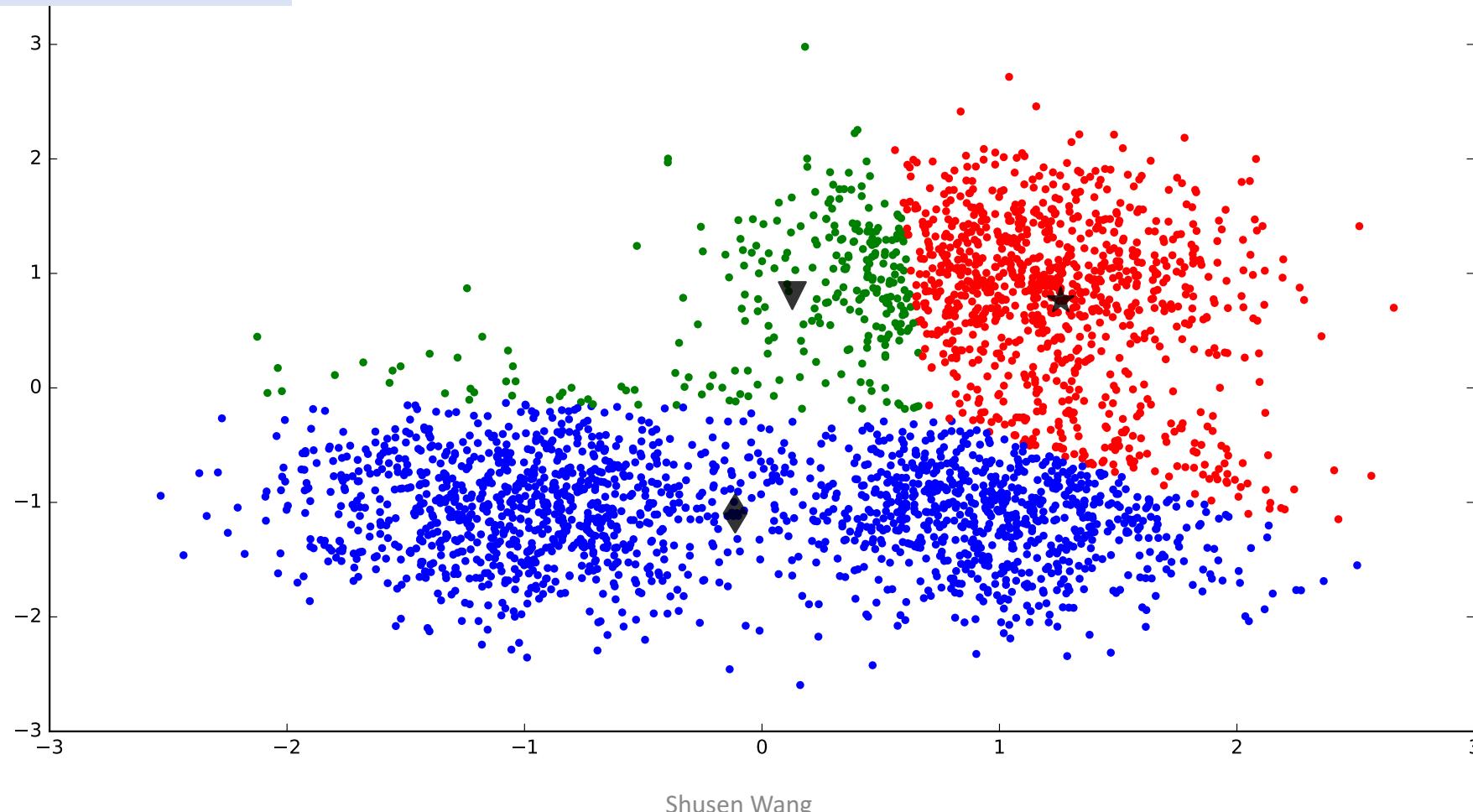
# Lloyd's Algorithm

Assign  $x_j$  to its nearest centroid.



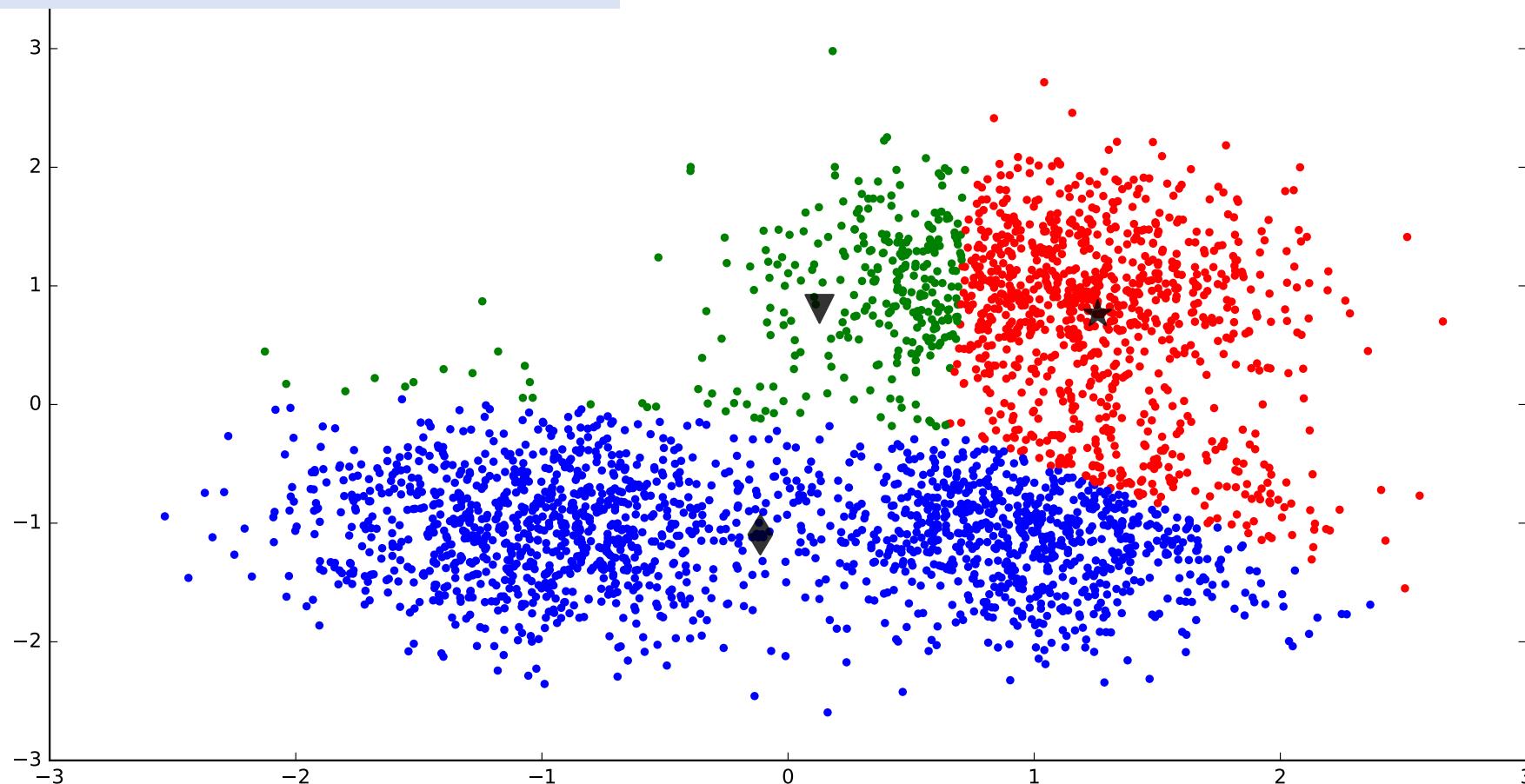
# Lloyd's Algorithm

move centroids



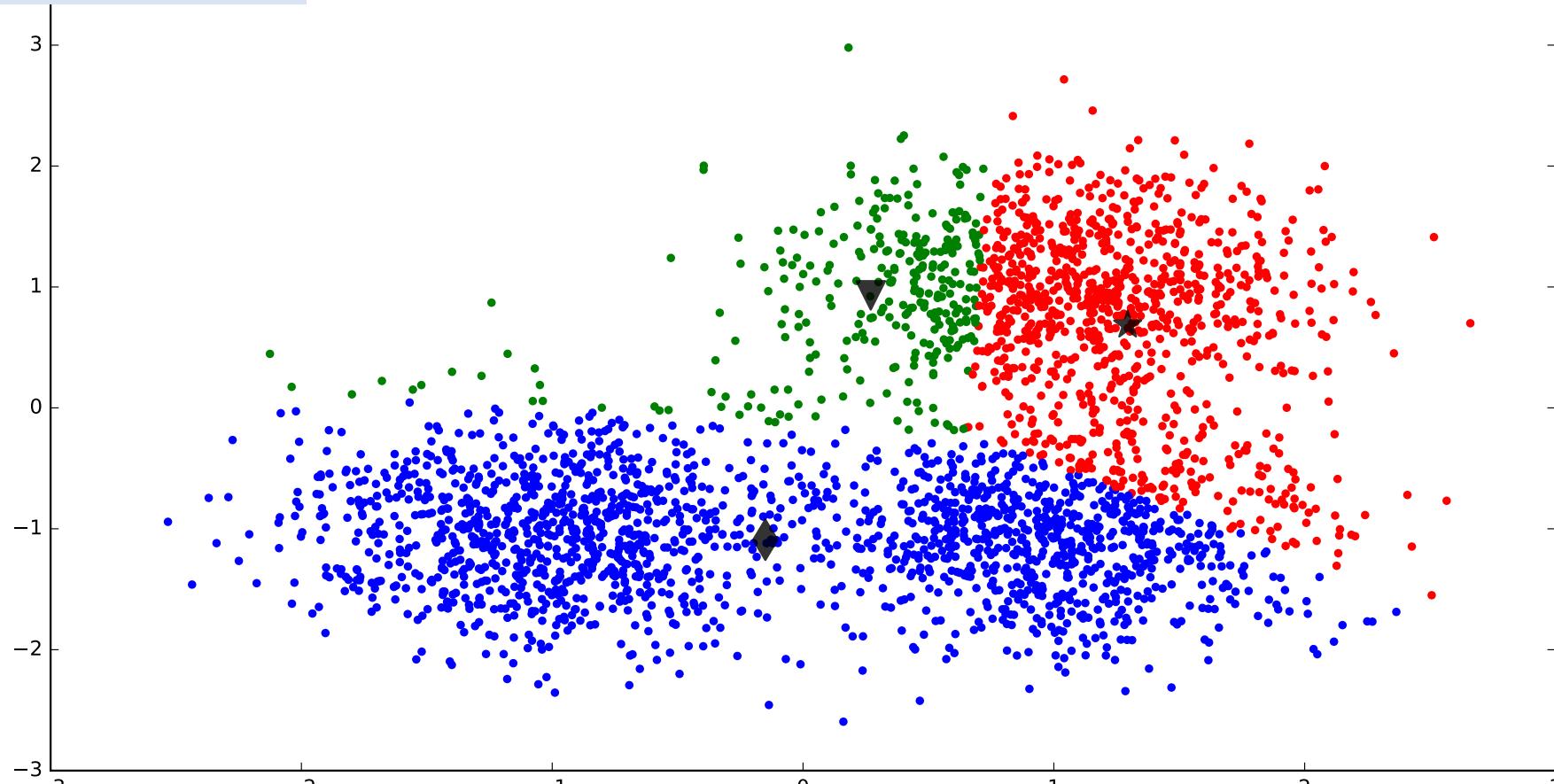
# Lloyd's Algorithm

Assign  $x_j$  to its nearest centroid.



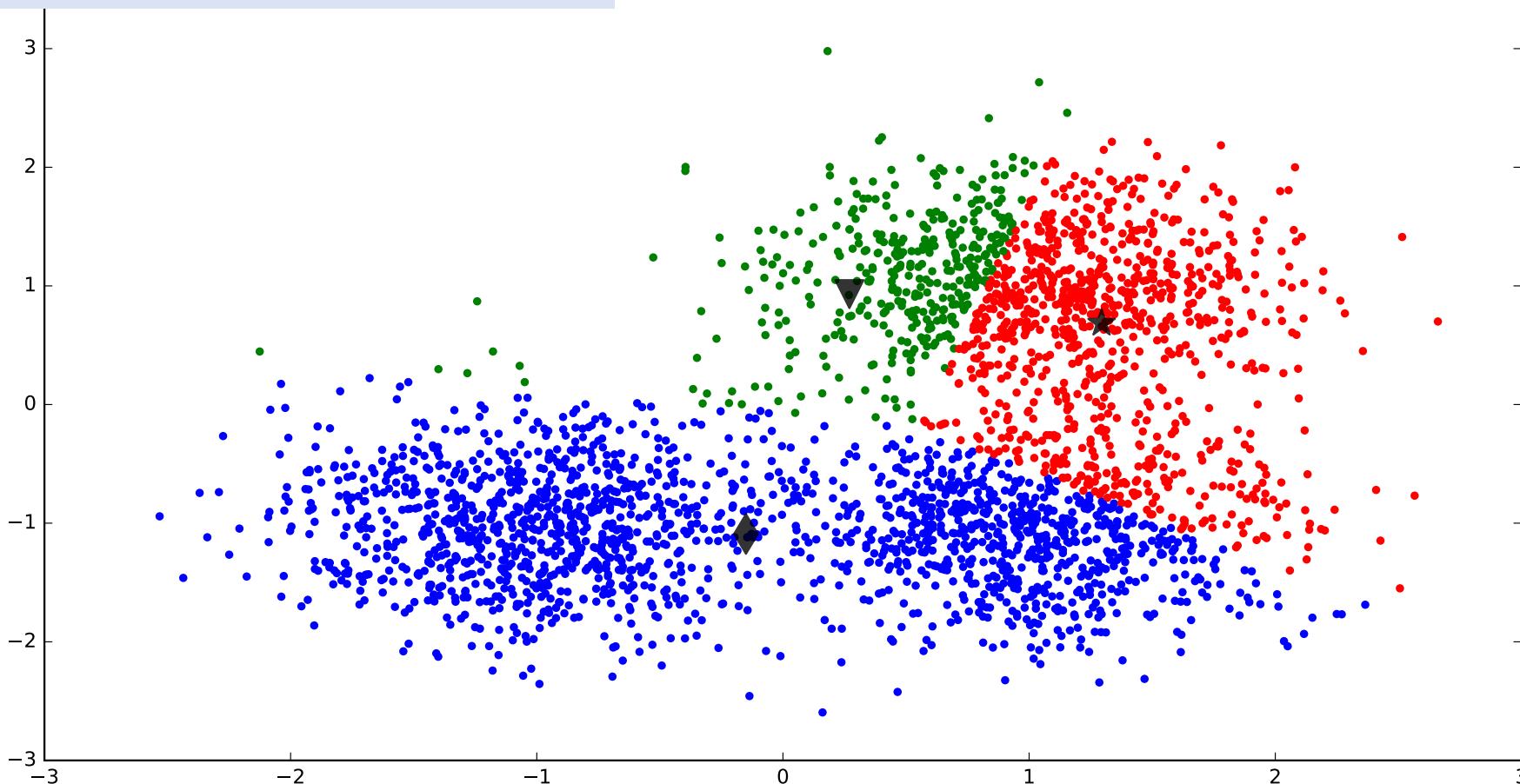
# Lloyd's Algorithm

move centroids



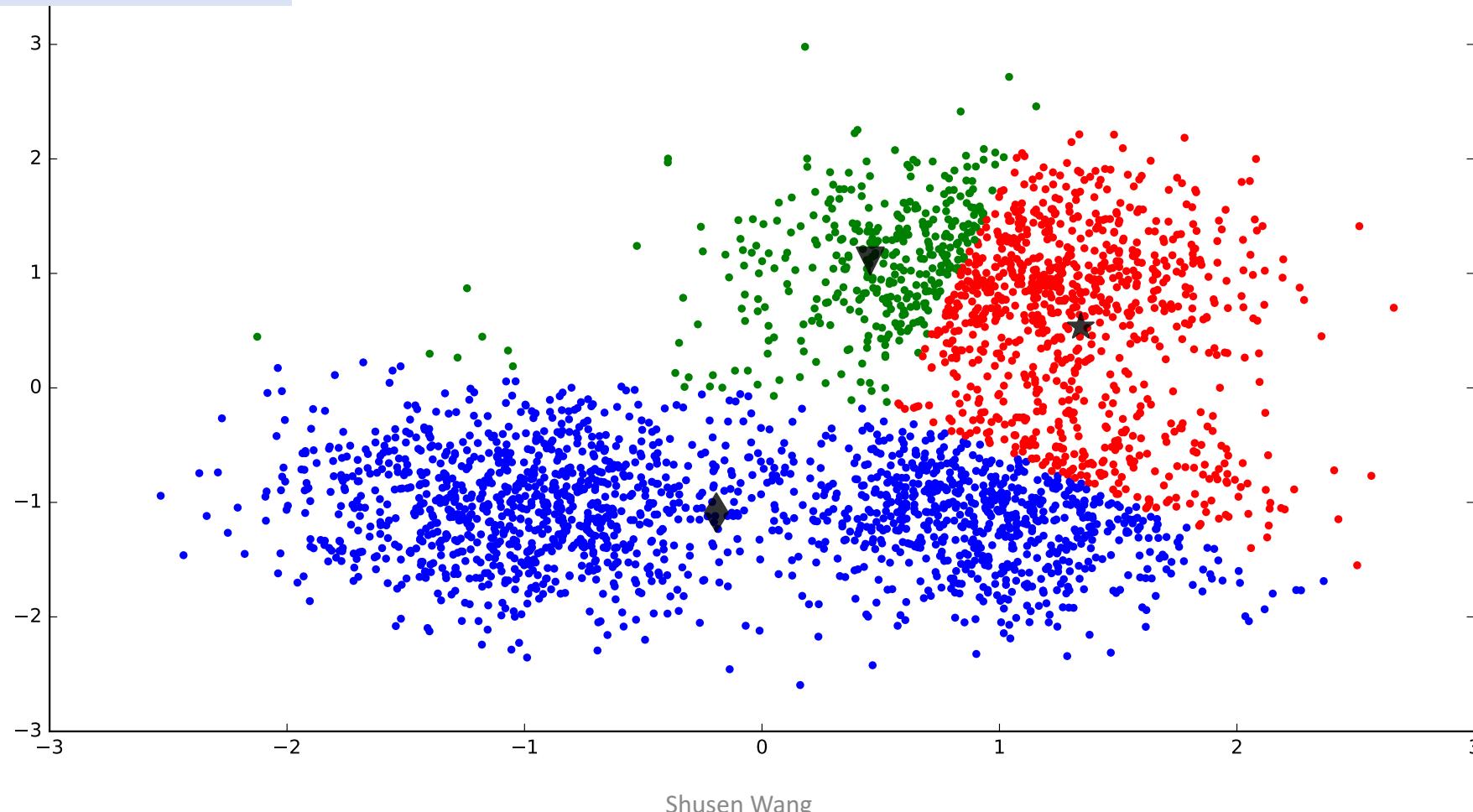
# Lloyd's Algorithm

Assign  $x_j$  to its nearest centroid.



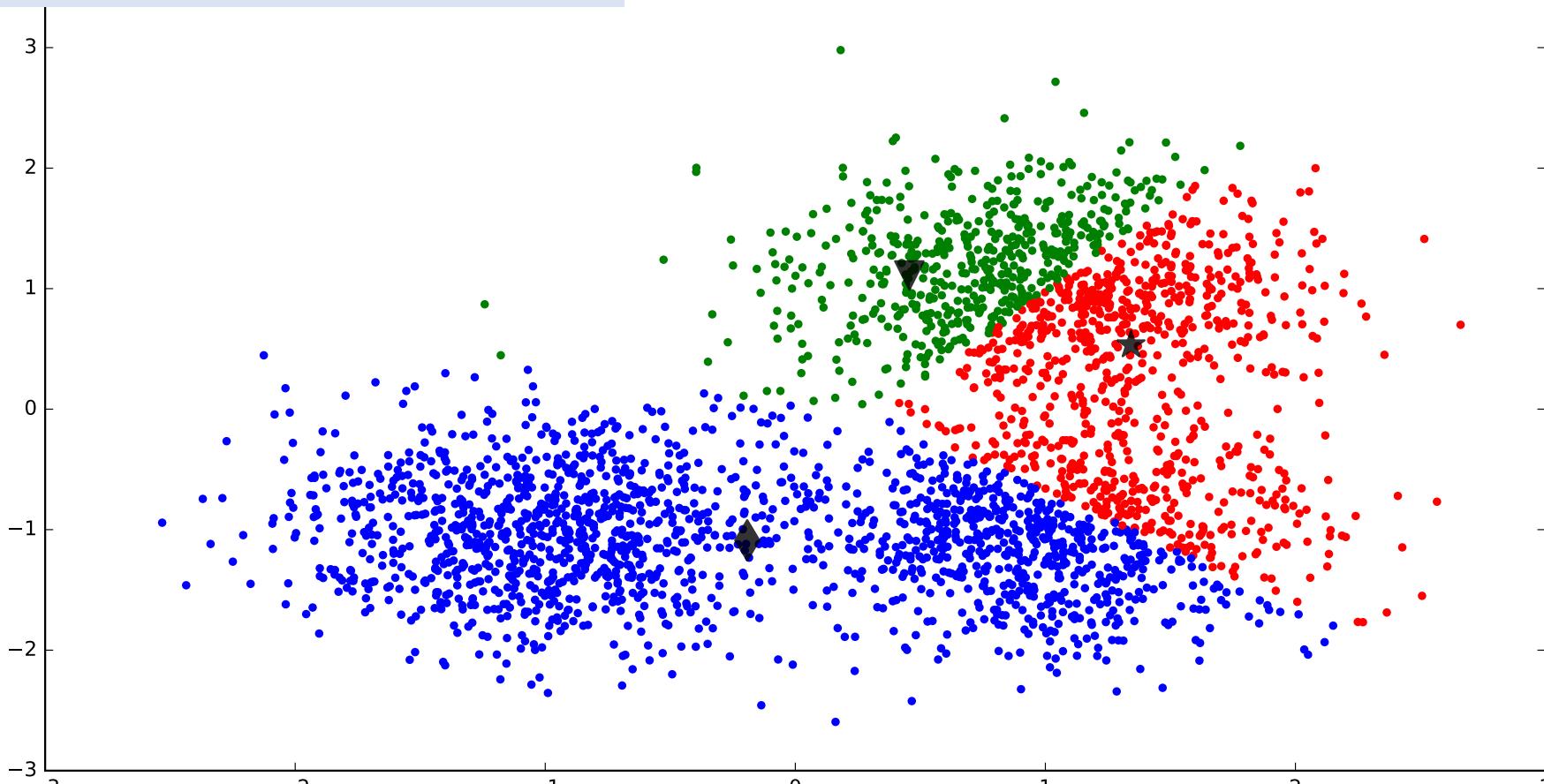
# Lloyd's Algorithm

move centroids



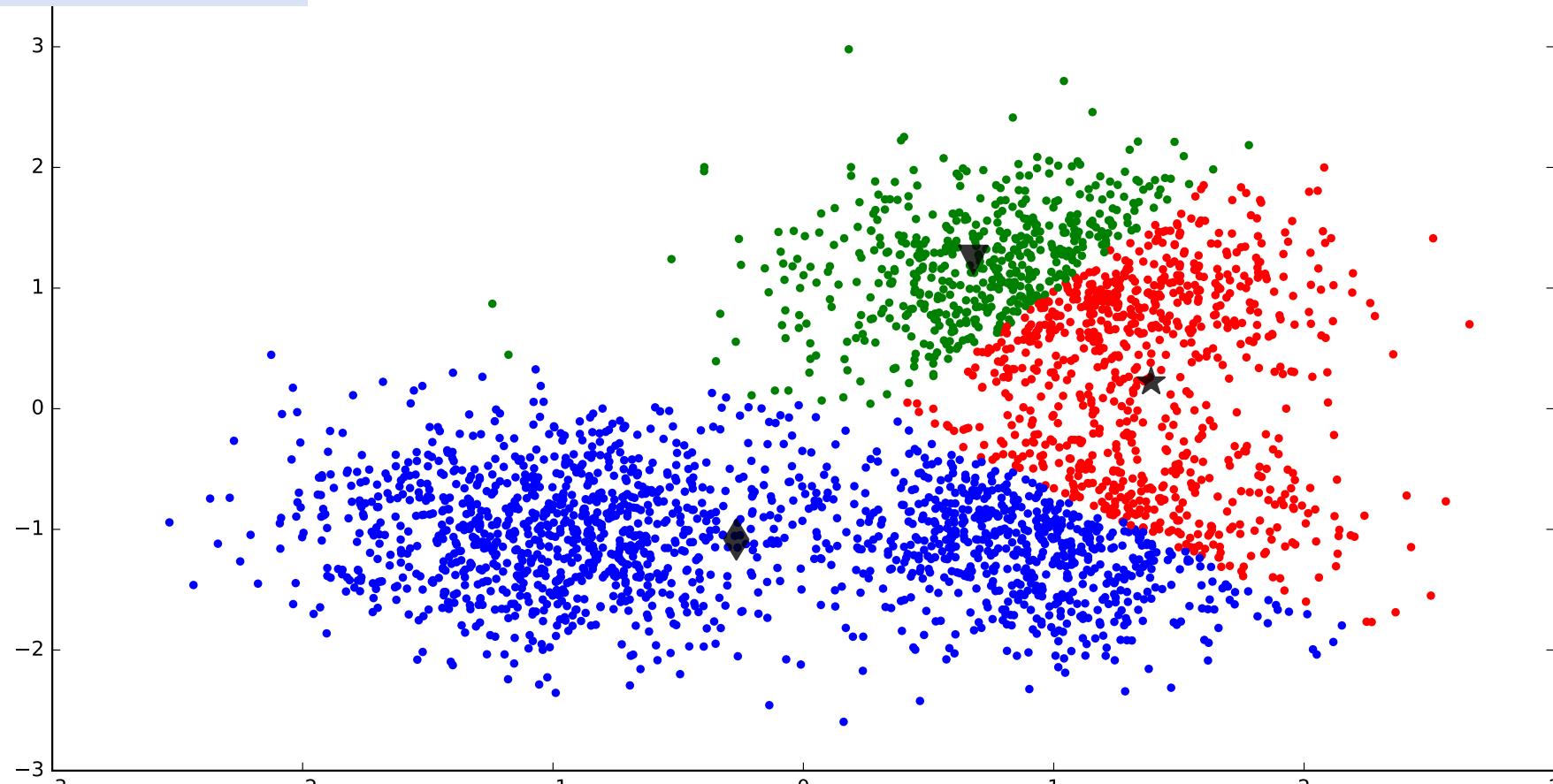
# Lloyd's Algorithm

Assign  $x_j$  to its nearest centroid.



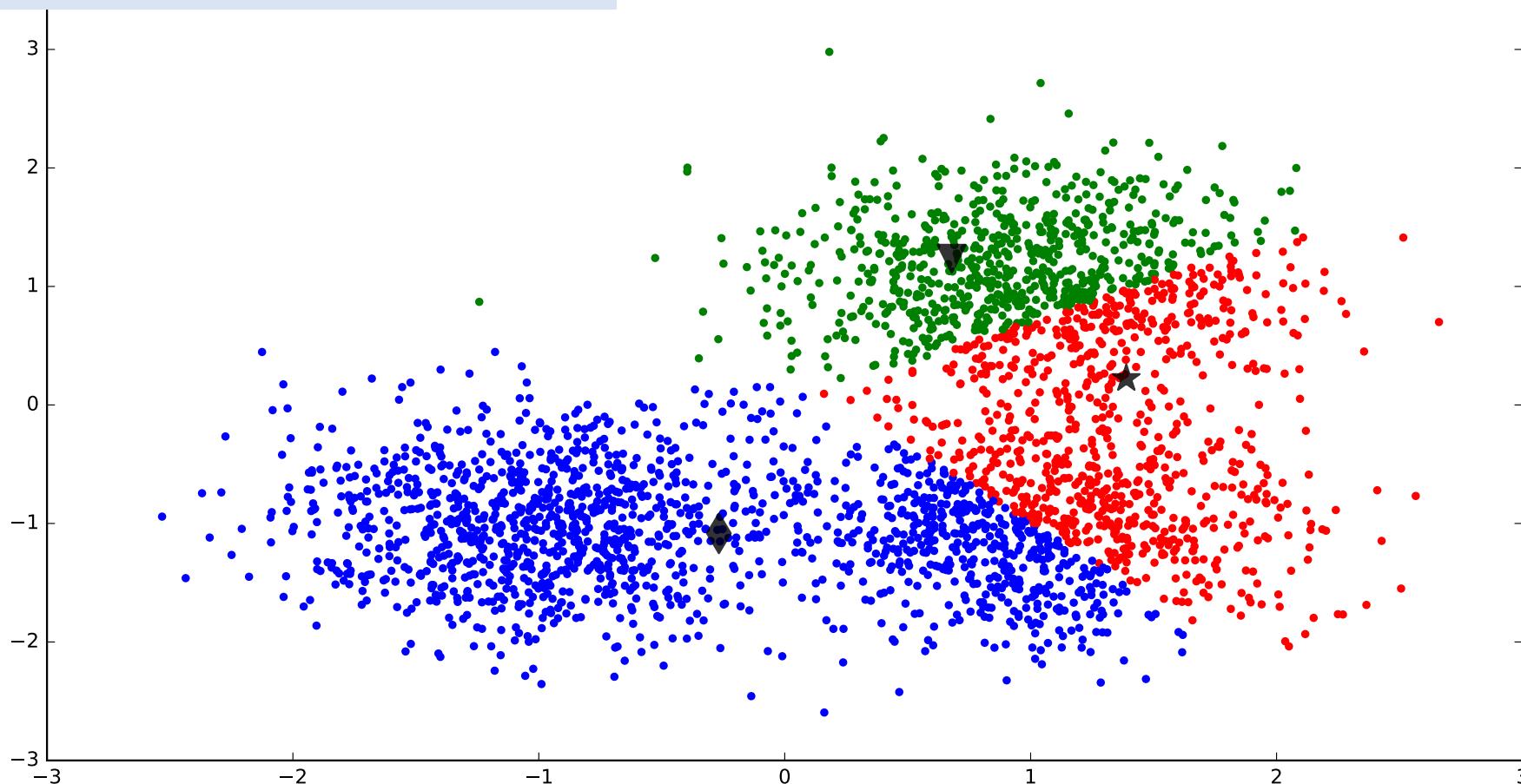
# Lloyd's Algorithm

move centroids



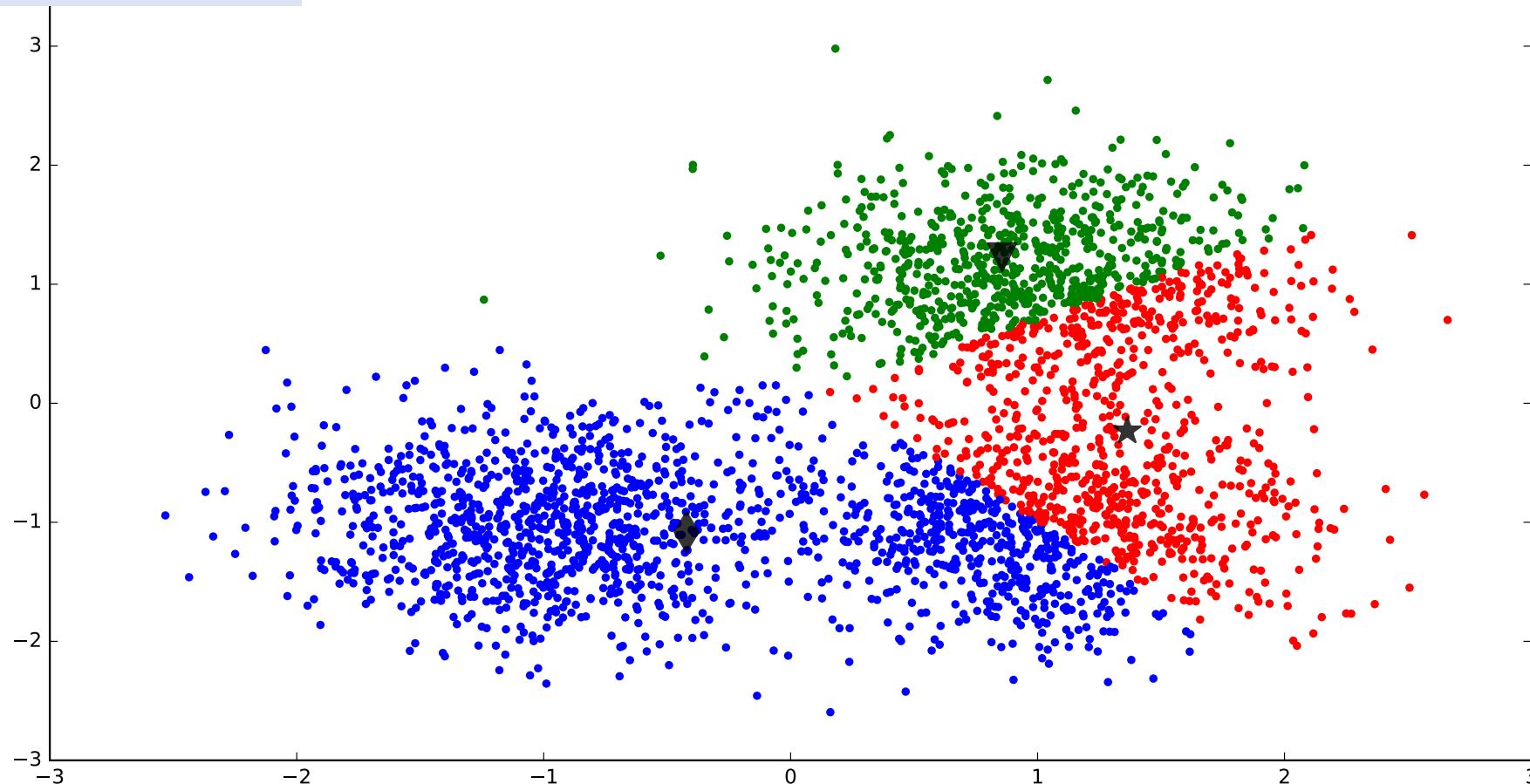
# Lloyd's Algorithm

Assign  $x_j$  to its nearest centroid.



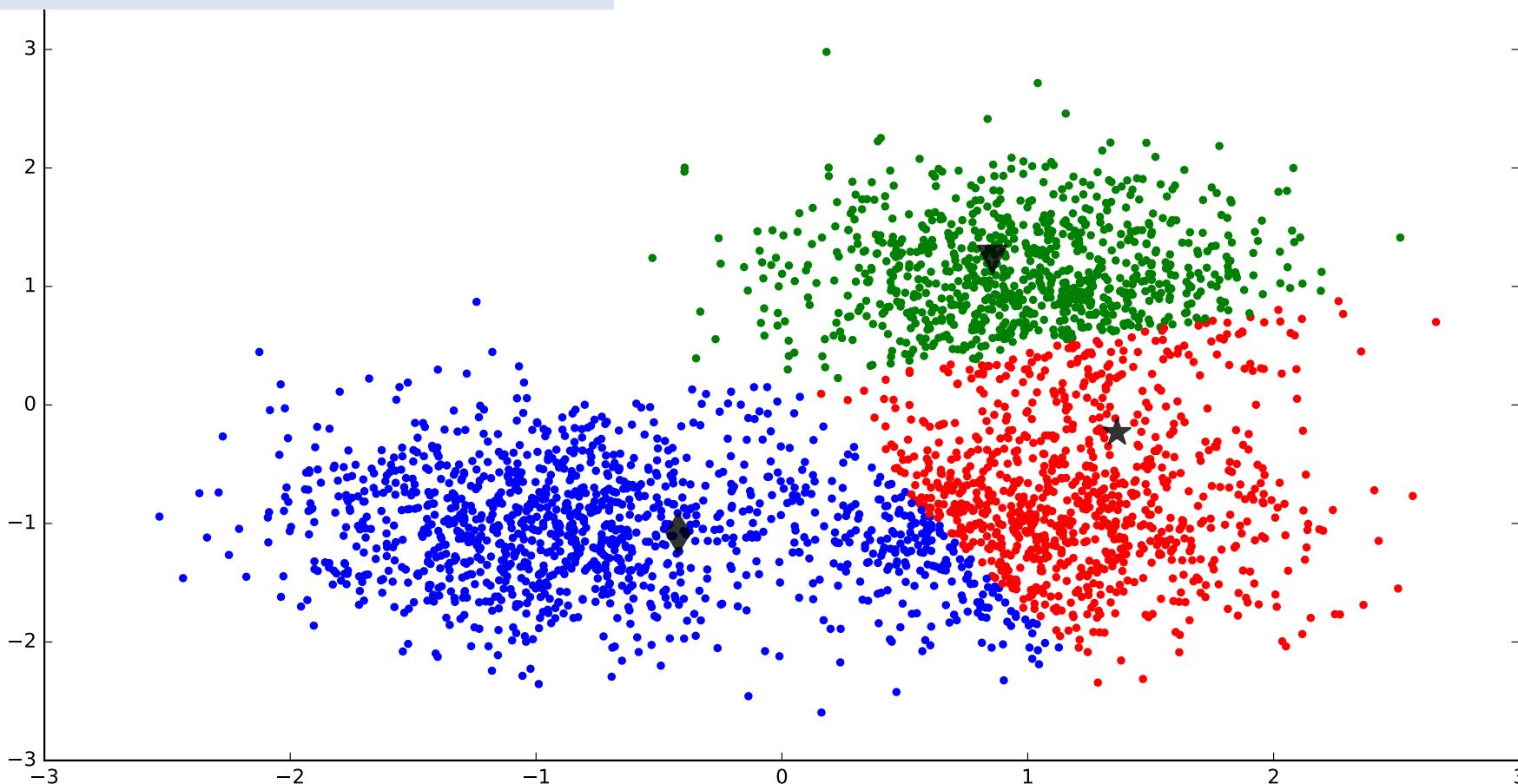
# Lloyd's Algorithm

move centroids



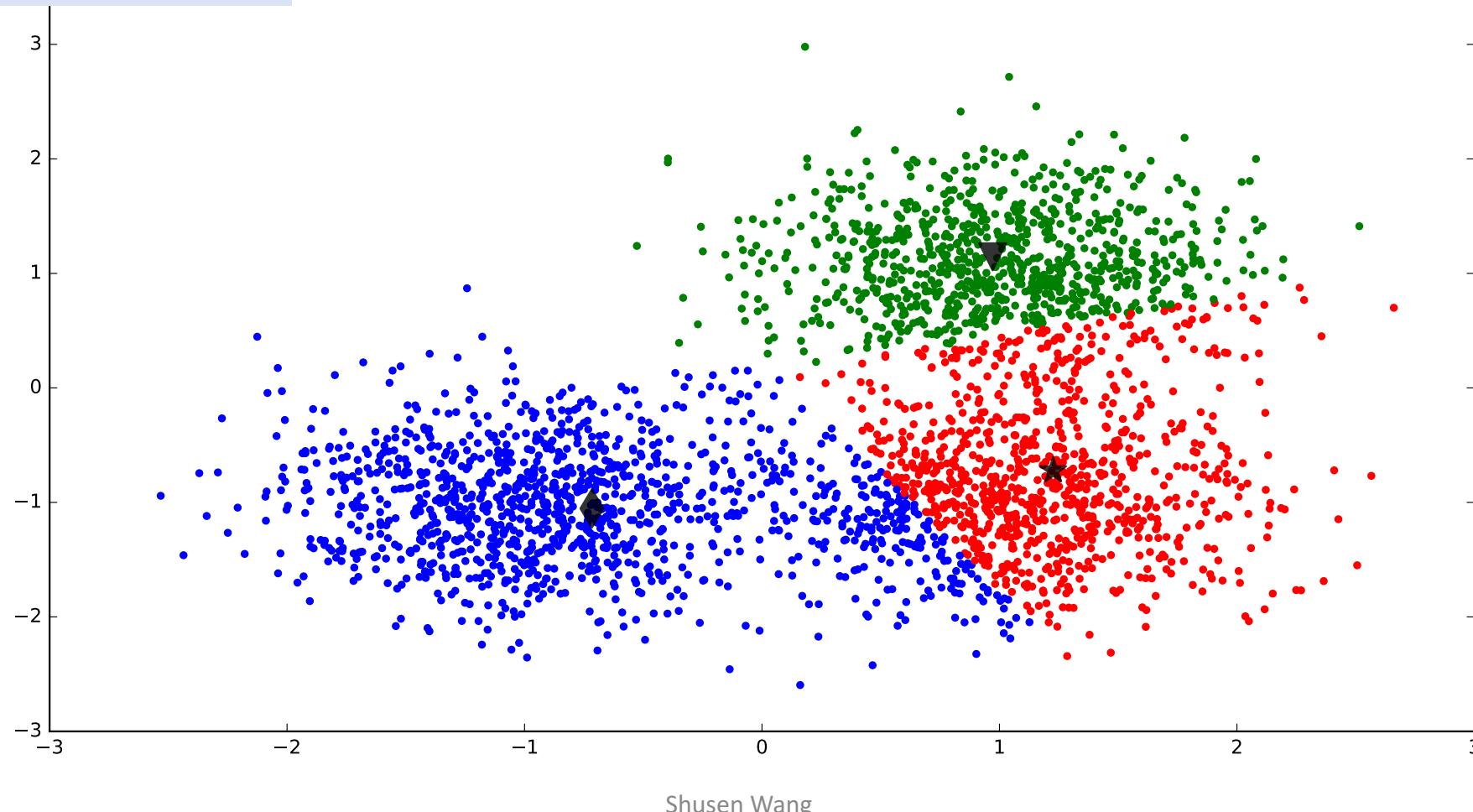
# Lloyd's Algorithm

Assign  $x_j$  to its nearest centroid.



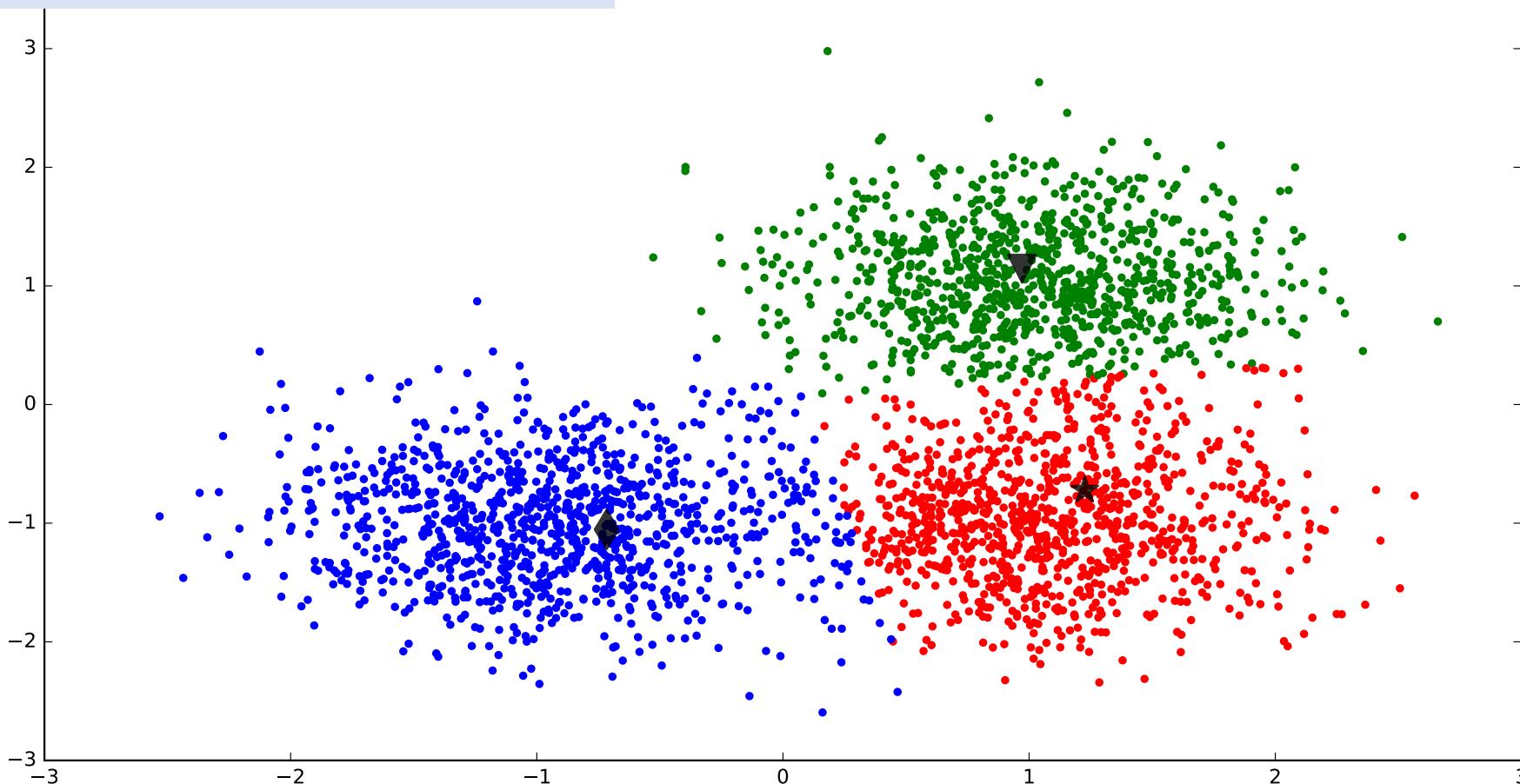
# Lloyd's Algorithm

move centroids



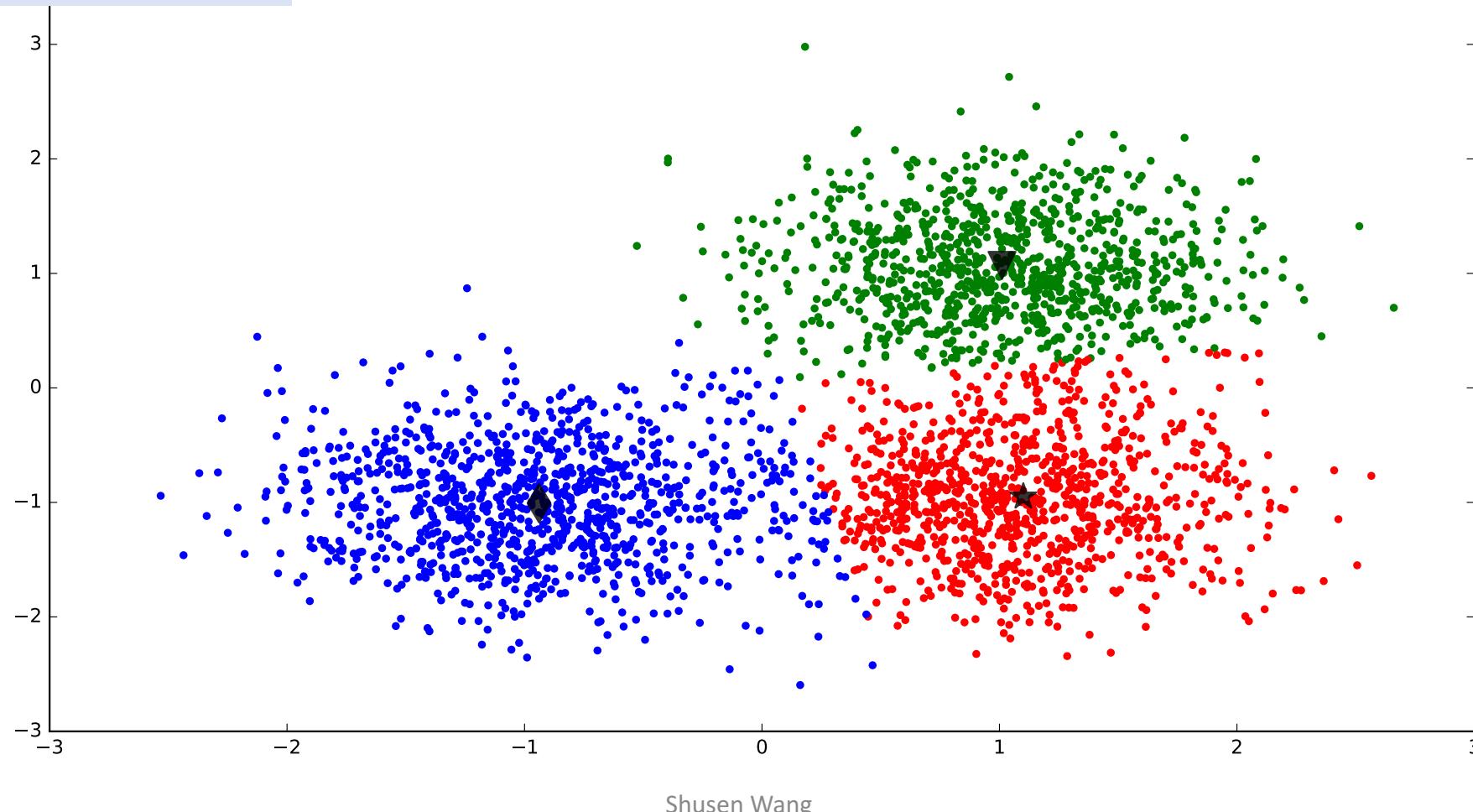
# Lloyd's Algorithm

Assign  $x_j$  to its nearest centroid.



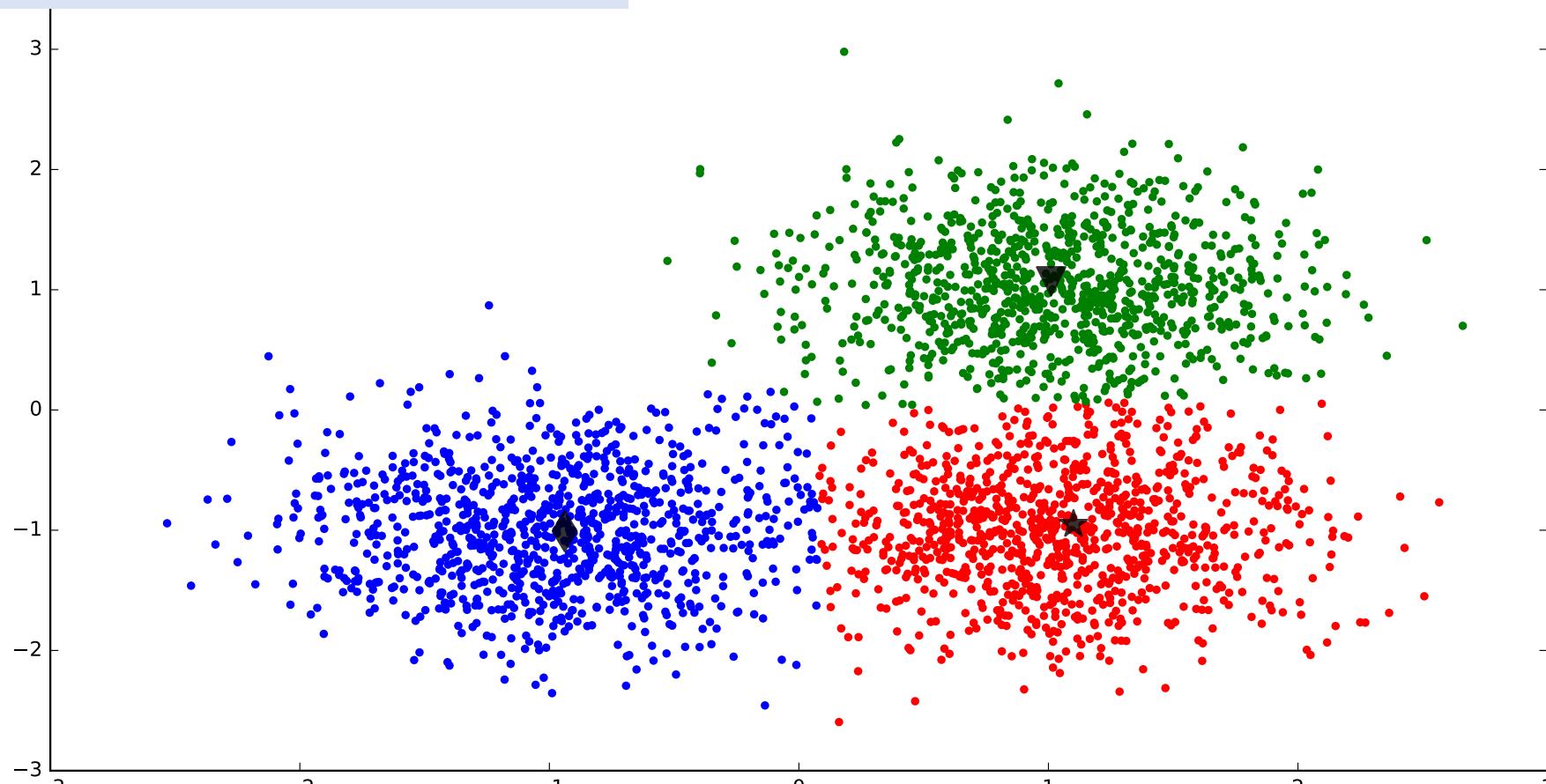
# Lloyd's Algorithm

move centroids



# Lloyd's Algorithm

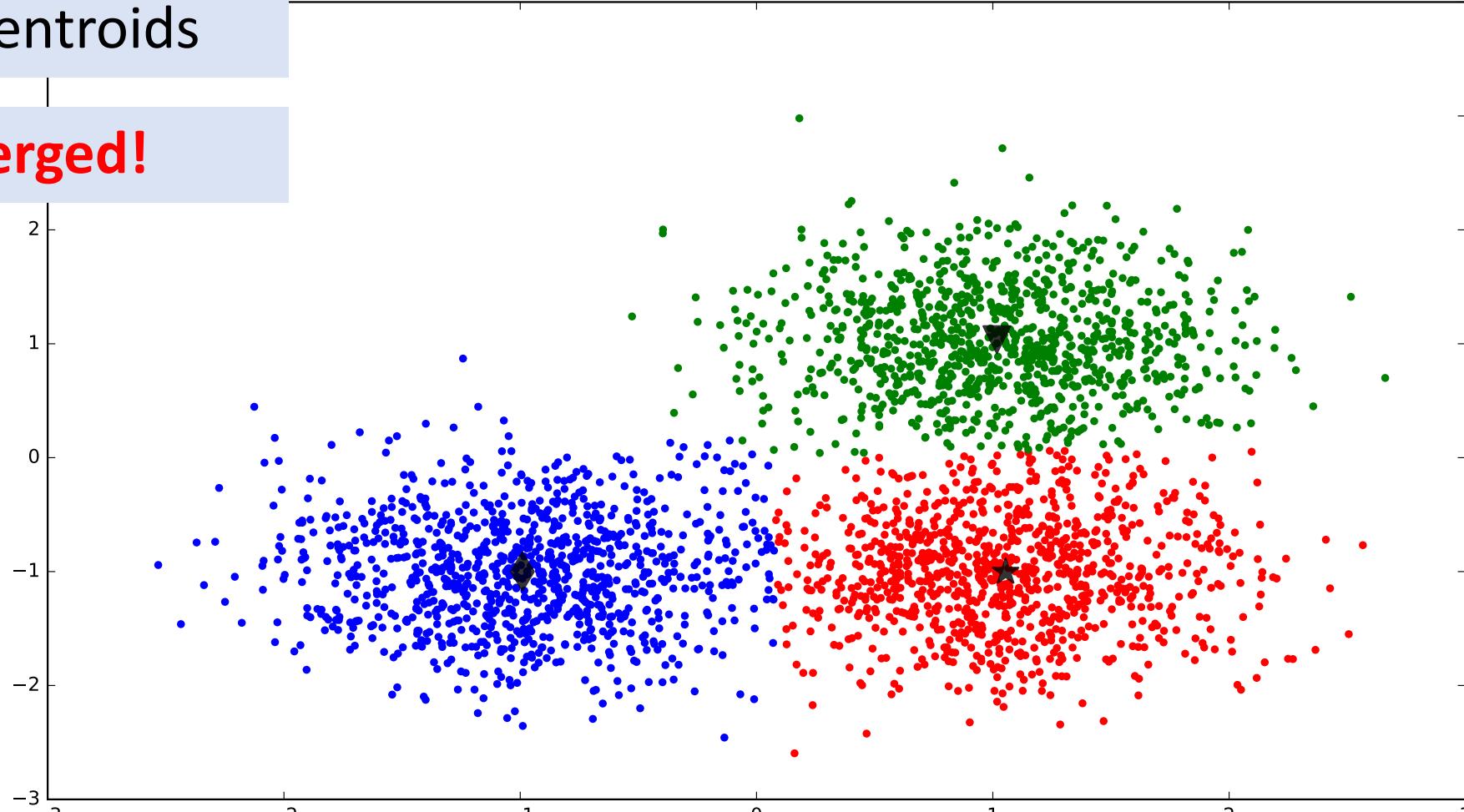
Assign  $x_j$  to its nearest centroid.



# Lloyd's Algorithm

move centroids

converged!



# Summary

# Summary

- Clustering task: partition  $[n]$  into  $k$  subsets according to the feature vectors.
- K-means model:  $\min_{S_1, \dots, S_k} \sum_{i=1}^k \sum_{j \in S_i} \left\| \mathbf{x}_j - \frac{1}{|S_i|} \sum_{l \in S_i} \mathbf{x}_l \right\|_2^2$ .
- Lloyd's algorithm for solving the model (approximately).
- Other algorithms: Forgy, MacQueen, and Hartigan.

Tasks

clustering

Methods

K-means

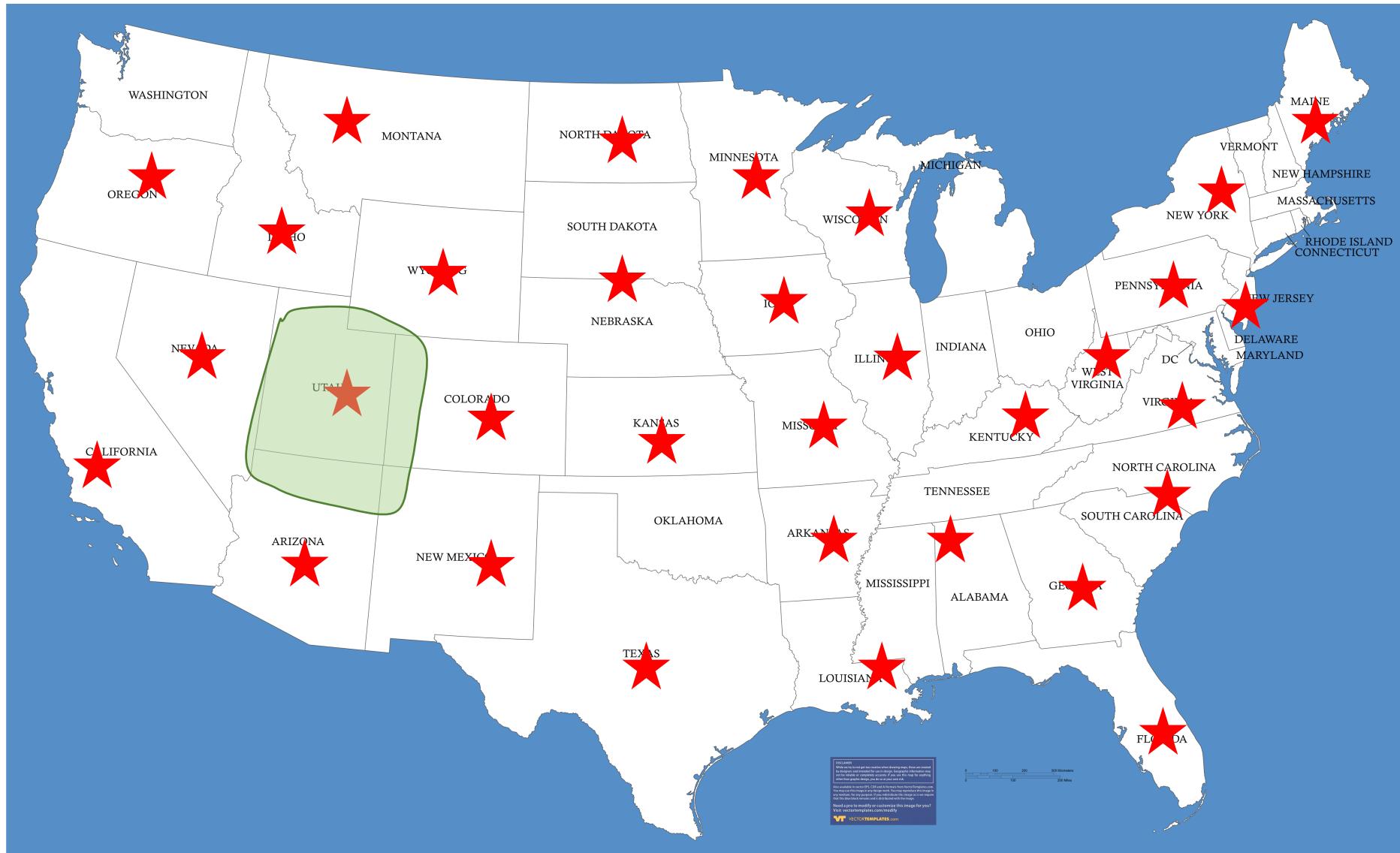
Algorithms

Lloyd's algorithm

# Summary

- Clustering task: partition  $[n]$  into  $k$  subsets according to the feature vectors.
- K-means model:  $\min_{S_1, \dots, S_k} \sum_{i=1}^k \sum_{j \in S_i} \left\| \mathbf{x}_j - \frac{1}{|S_i|} \sum_{l \in S_i} \mathbf{x}_l \right\|_2^2$ .
- Lloyd's algorithm for solving the model (approximately).
- Other algorithms: Forgy, MacQueen, and Hartigan.
- There is not such a thing called “**k-means algorithm**”!

# Application: Space Partitioning for KNN



# Evaluation Metrics

评价指标

- Objective function of k-means. k-均值的目标函数
- Accuracy. 准确度
- Normalized mutual information (NMI). 归一化互信息 ( NMI )。

See scikit-learn: <http://scikit-learn.org/stable/modules/classes.html#clustering-metrics>