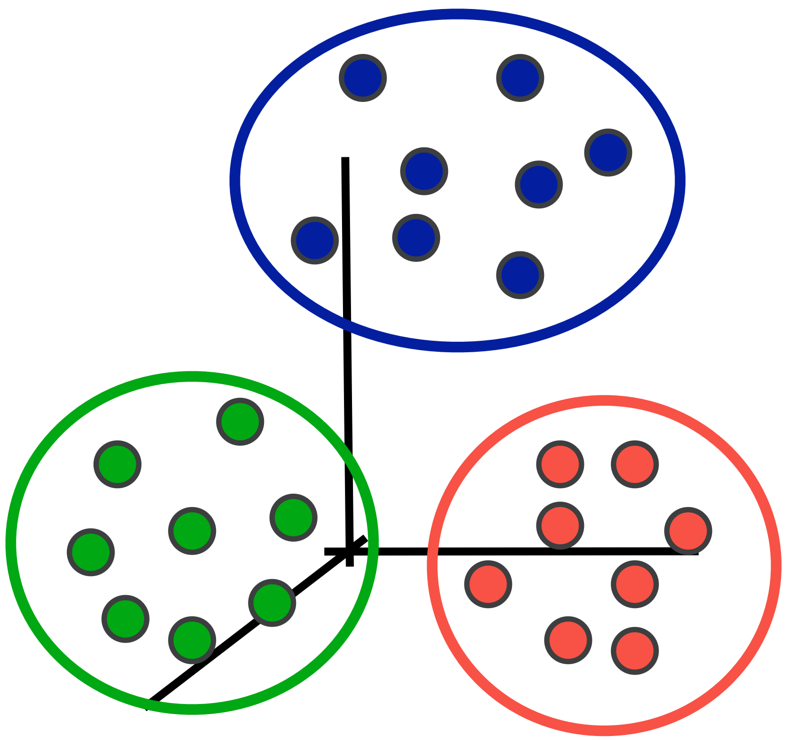
CS 565: Data mining

- Clustering: David Arthur, Sergei Vassilvitskii. *k-means ++: The Advantages of Careful Seeding*. In SODA 2007
- Thanks A. Gionis and S. Vassilvitskii for the slides

What is clustering?

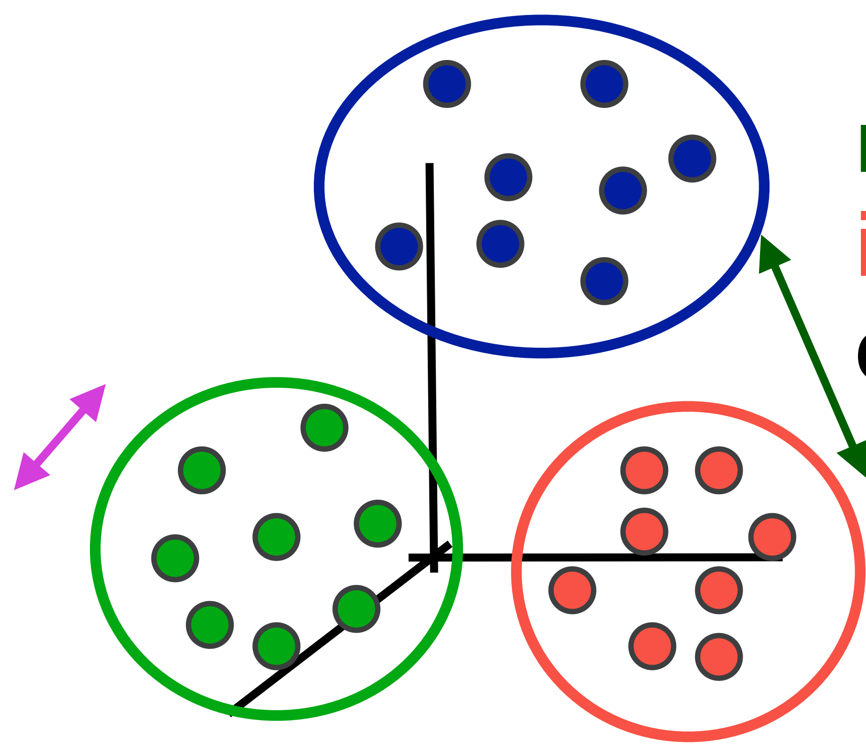
- a grouping of data objects such that the objects within a group are similar (or near) to one another and dissimilar (or far) from the objects in other groups



How to capture this objective?

a grouping of data objects such that the objects within a group are similar (or near) to one another and dissimilar (or far) from the objects in other groups

minimize
intra-cluster
distances



maximize
inter-cluster
distances

The clustering problem

- Given a collection of data objects
- Find a grouping so that
  - similar objects are in the same cluster
  - dissimilar objects are in different clusters

- Why we care ?
- stand-alone tool to gain insight into the data
  - visualization
- preprocessing step for other algorithms
  - indexing or compression often relies on clustering

Applications of clustering

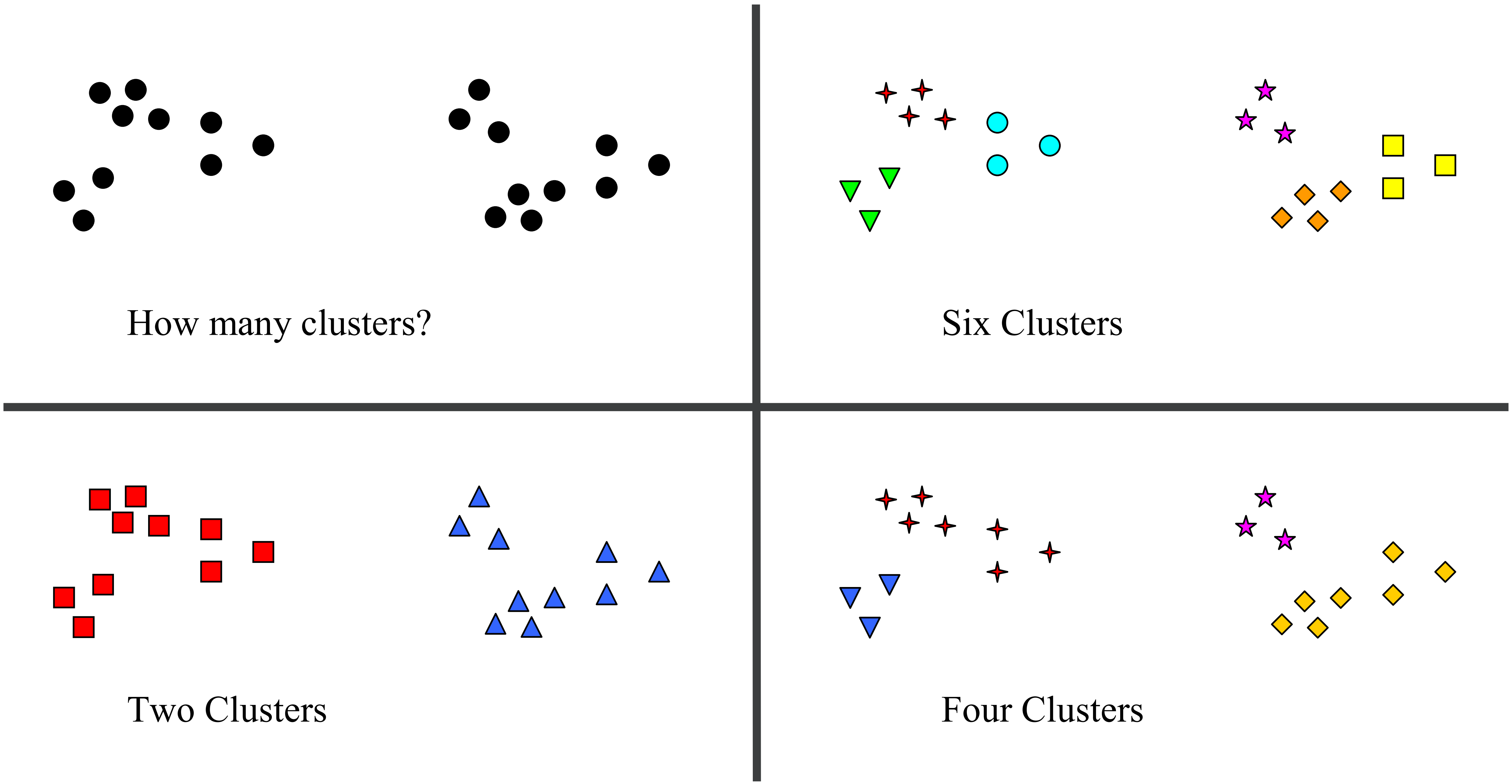
- image processing
  - cluster images based on their visual content
- web mining
  - cluster groups of users based on their access patterns on webpages
  - cluster webpages based on their content
- bioinformatics
  - cluster similar proteins together (similarity wrt chemical structure and/or functionality etc)
- many more...

The clustering problem

- Given a collection of data objects
- Find a grouping so that
  - similar objects are in the same cluster
  - dissimilar objects are in different clusters

- Basic questions:
  - what does similar mean?
  - what is a good partition of the objects?
    i.e., how is the quality of a solution measured?
  - how to find a good partition?

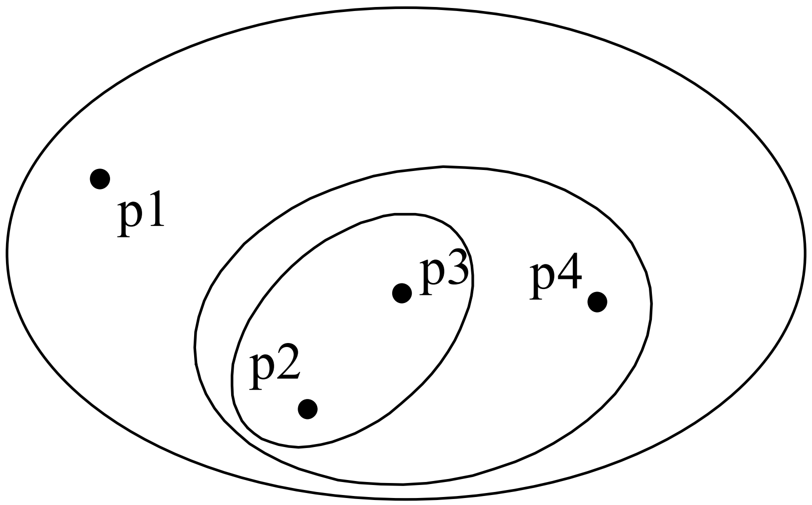
Notion of a cluster can be ambiguous



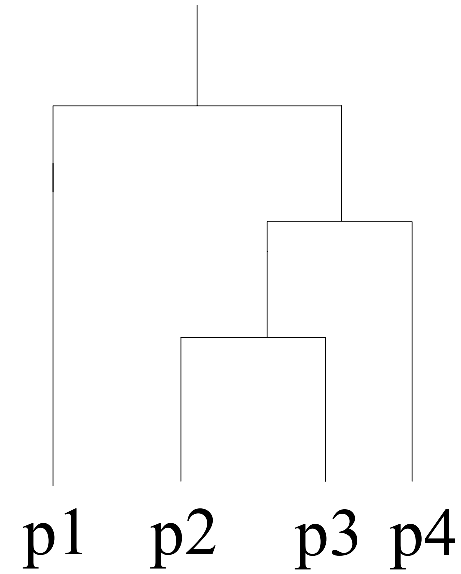
Types of clusterings

- Partitional
  - each object belongs in exactly one cluster
- Hierarchical
  - a set of nested clusters organized in a tree

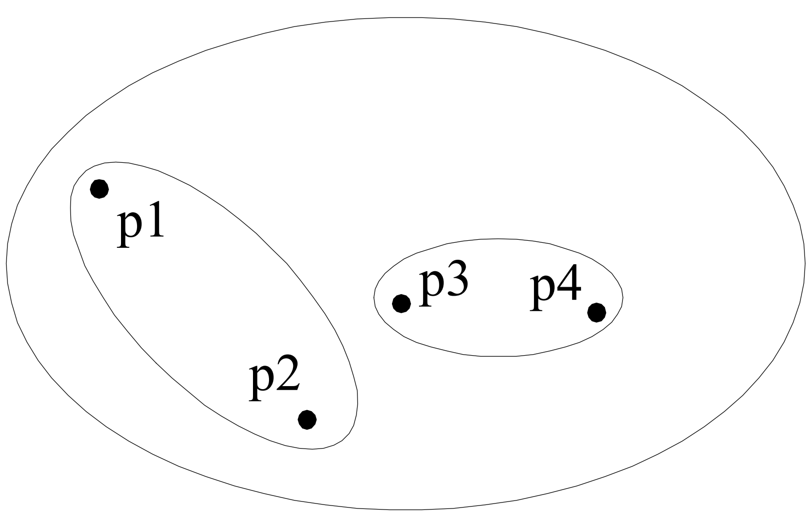
Hierarchical clustering



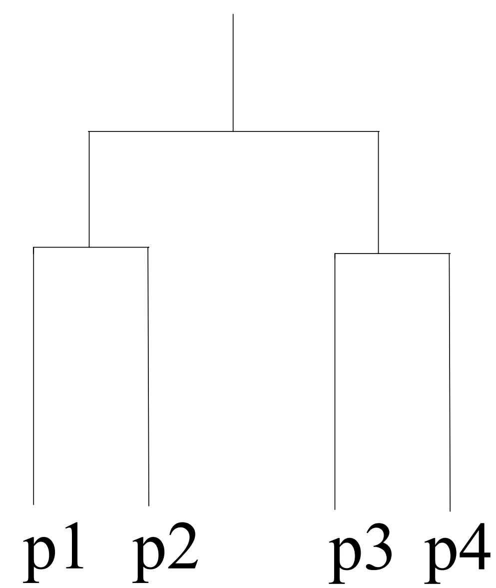
Hierarchical Clustering



Dendrogram

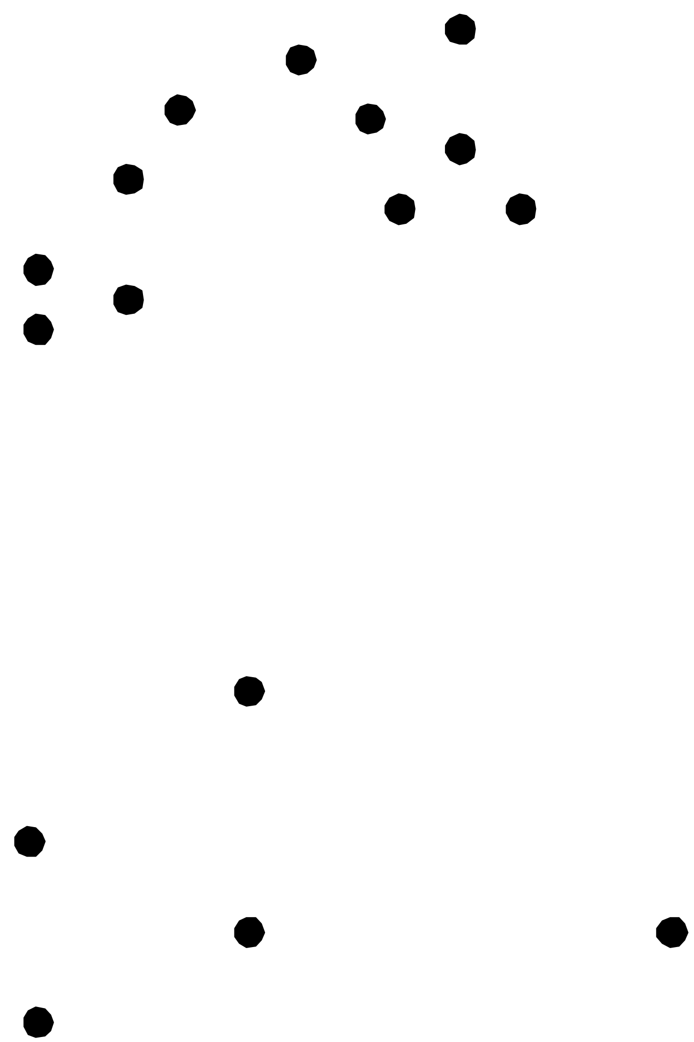


Hierarchical Clustering

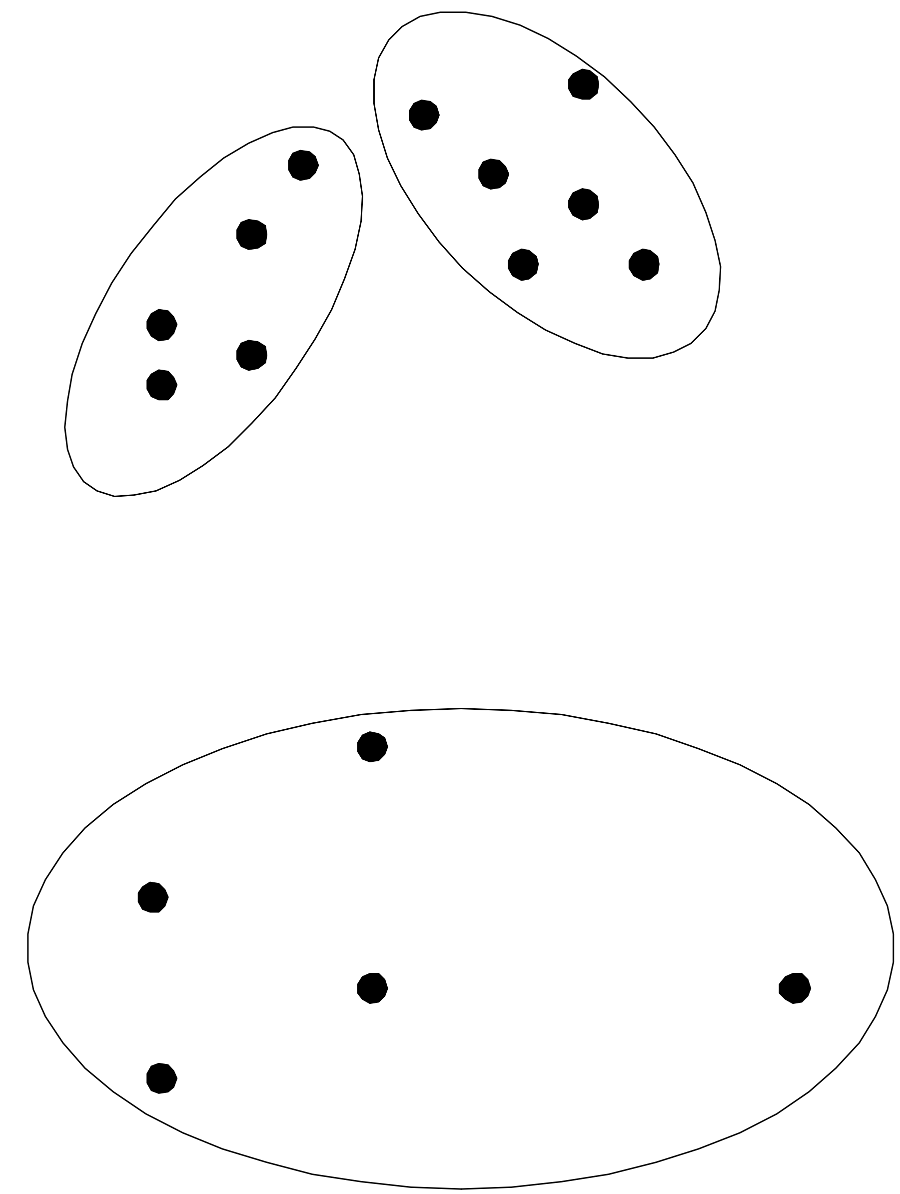


Dendrogram

Partitional clustering



Original Points



A Partitional Clustering

Partitional algorithms

- partition the n objects into k clusters
  - each object belongs to exactly one cluster
  - the number of clusters k is given in advance

The k-means problem

- consider set $X=\{x_1,\ldots,x_n\}$ of n points in R^d
- assume that the number k is given
- problem:
  - find k points $c_1,\ldots,c_k$ (named centers or means)

    so that the cost

$$\sum_{i=1}^{n} \min_j \left\{ L_2^2(x_i, c_j) \right\} = \sum_{i=1}^{n} \min_j ||x_i - c_j||_2^2$$

    is minimized

The k-means problem

- consider set $X=\{x_1,\ldots,x_n\}$ of n points in R^d
- assume that the number k is given
- problem:
  - find k points $c_1,\ldots,c_k$ (named centers or means)
  - and partition X into $\{X_1,\ldots,X_k\}$ by assigning each point x_i in X to its nearest cluster center,
  - so that the cost

$$\sum_{i=1}^{n} \min_j ||x_i - c_j||_2^2 = \sum_{j=1}^{k} \sum_{x \in X_j} ||x - c_j||_2^2$$

is minimized

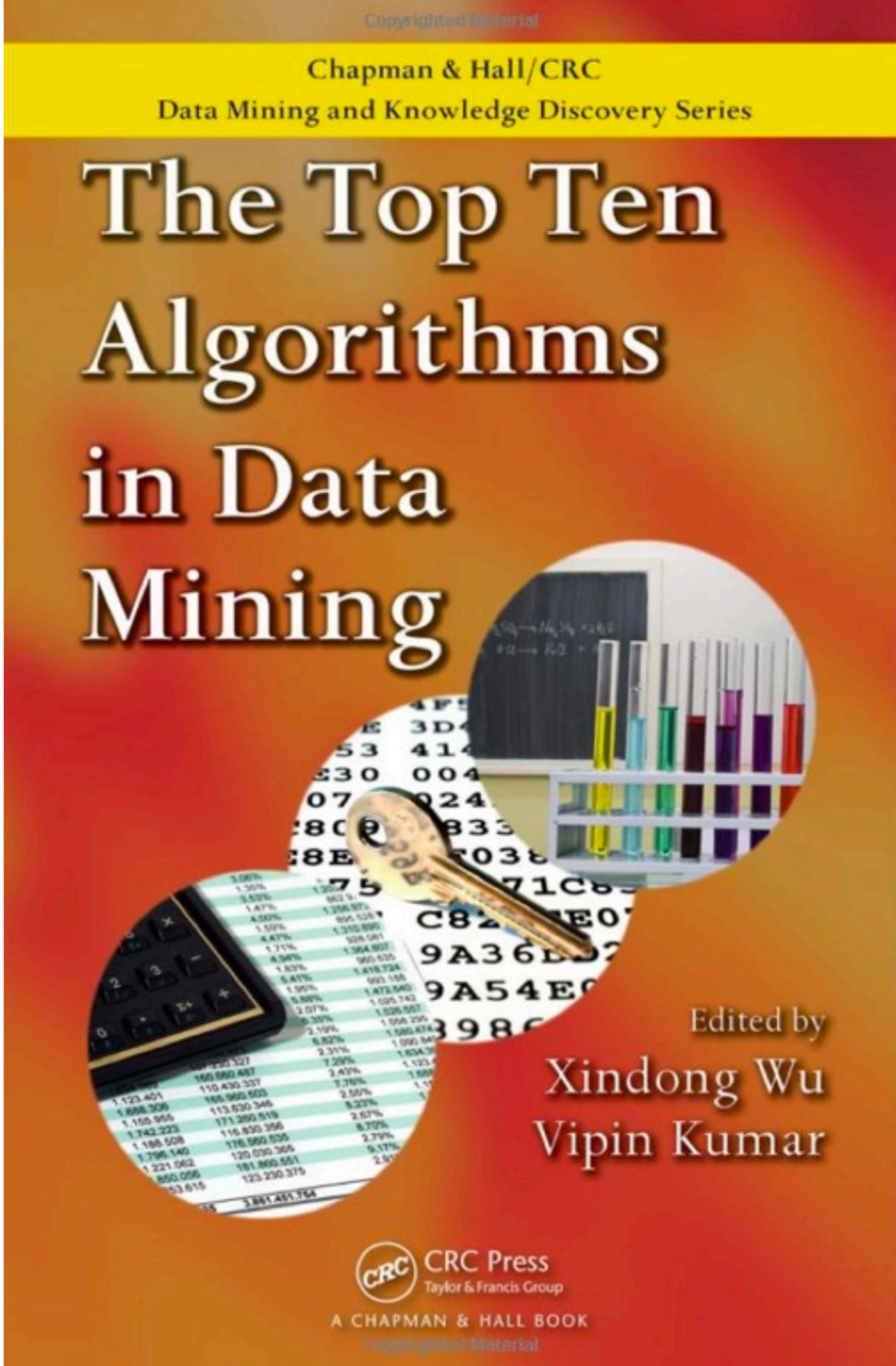
The k-means problem

- k=1 and k=n are easy special cases (why?)
- an NP-hard problem if the dimension of the data is at least 2 (d≥2)
  - for d≥2, finding the optimal solution in polynomial time is infeasible
- for d=1 the problem is solvable in polynomial time
- in practice, a simple iterative algorithm works quite well

The k-means algorithm

- voted among the top-10 algorithms in data mining

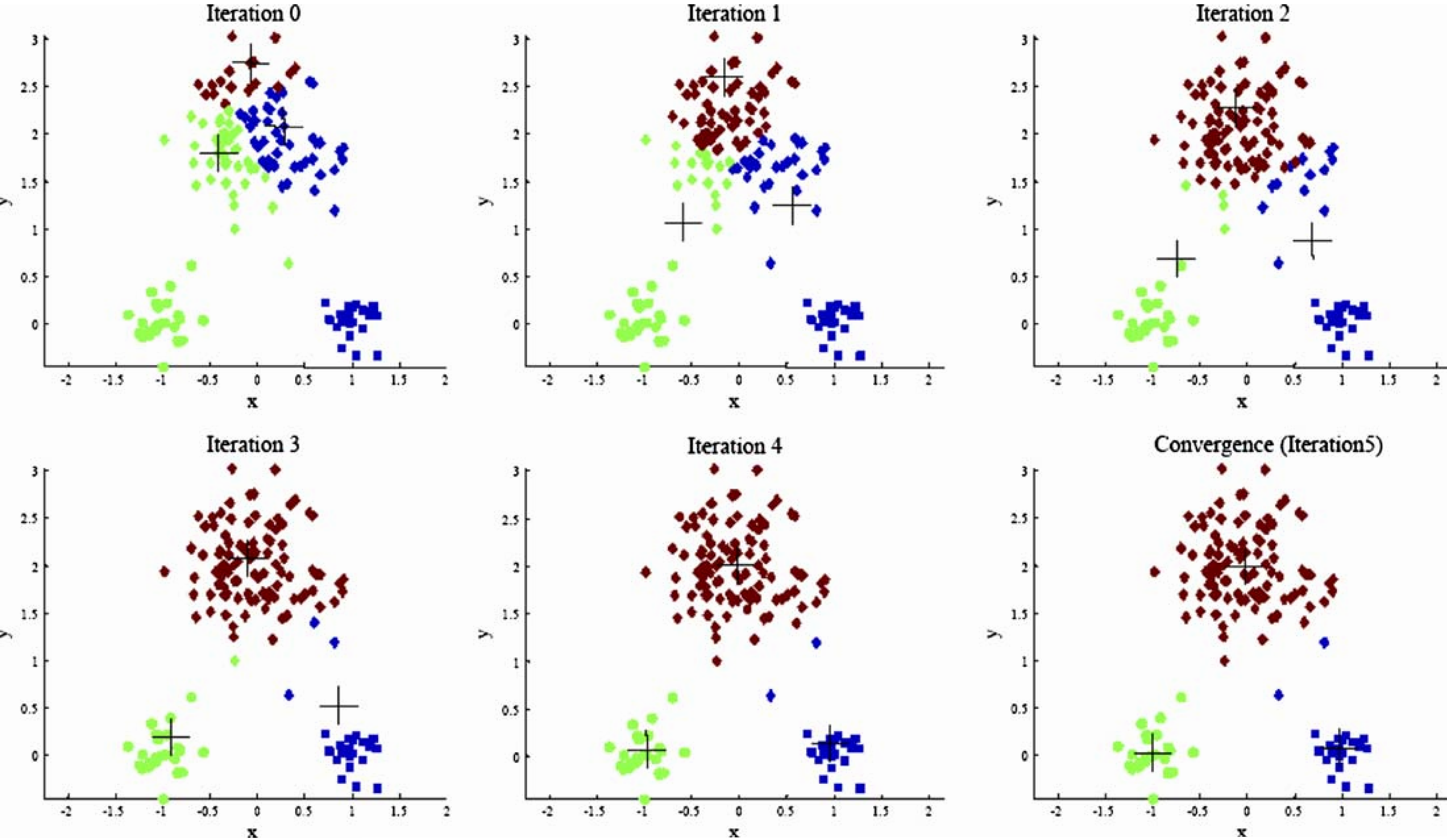
- one way of solving the k-means problem



The k-means algorithm

1. randomly (or with another method) pick k cluster centers $\{c_1,\ldots,c_k\}$
2. for each j, set the cluster X_j to be the set of points in X that are the closest to center c_j
3. for each j let c_j be the center of cluster X_j (mean of the vectors in X_j)
4. repeat (go to step 2) until convergence

Sample execution

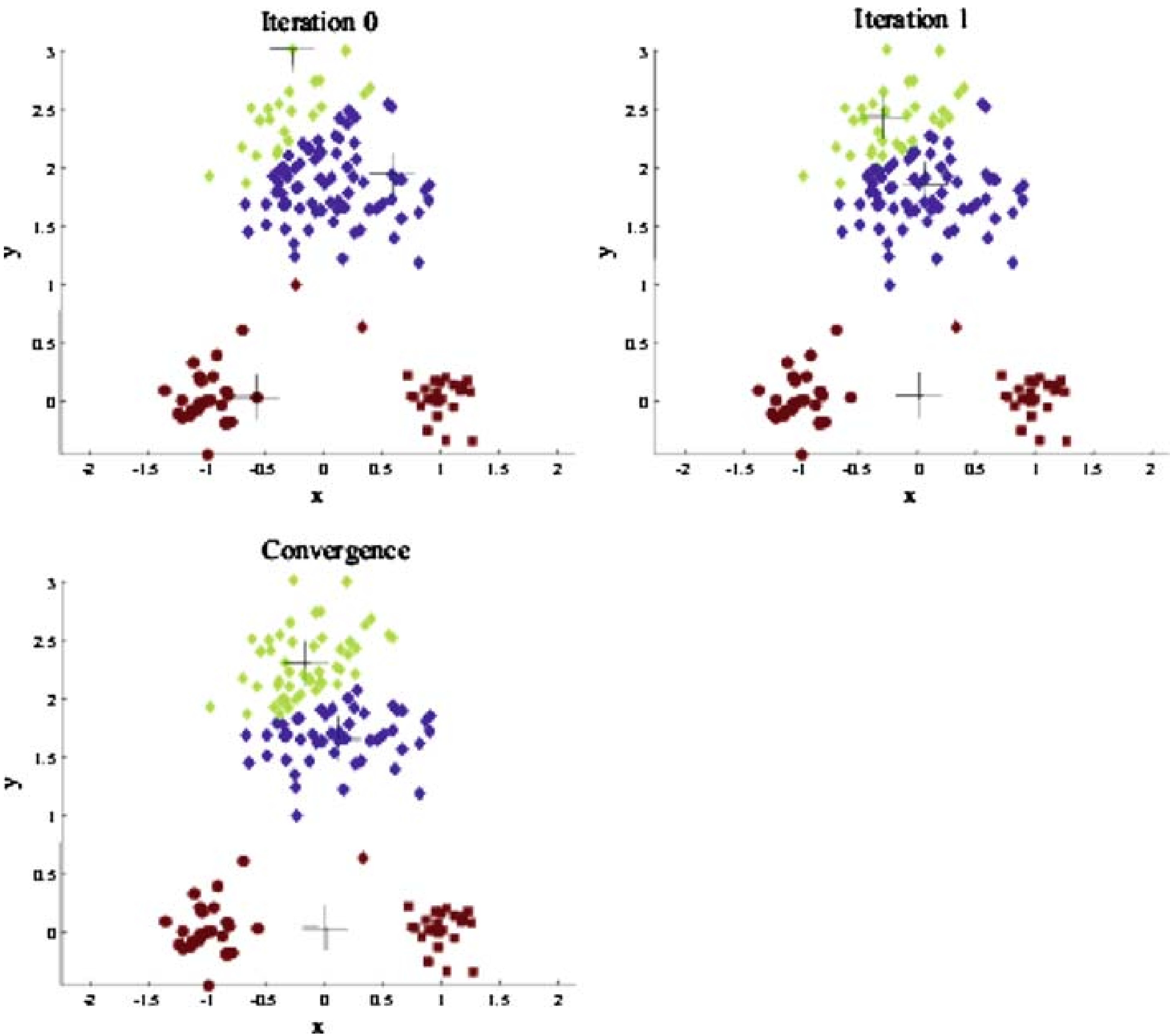


Properties of the k-means algorithm

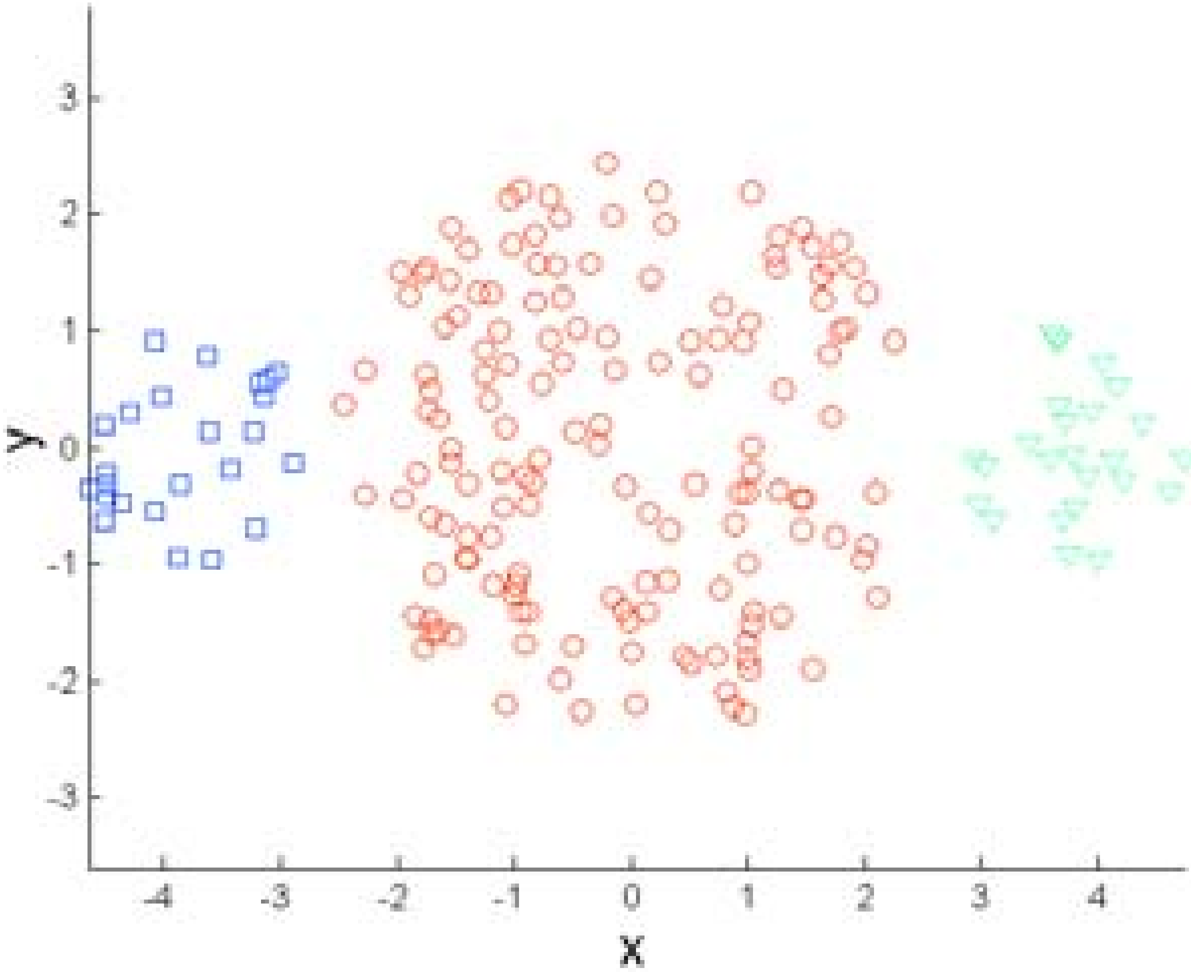
- finds a local optimum
- often converges quickly

  but not always
- the choice of initial points can have large influence in the result

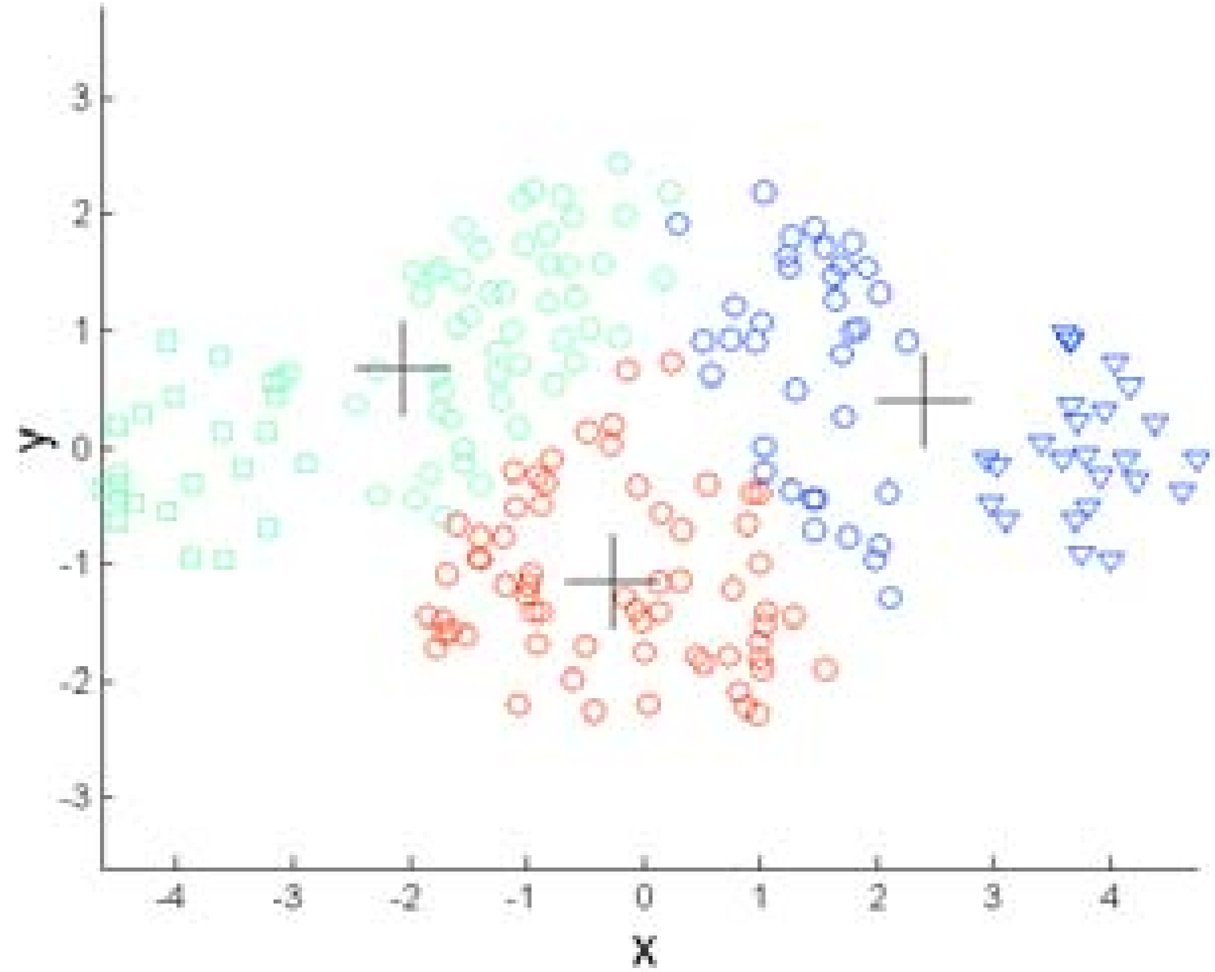
Effects of bad initialization



Limitations of k-means: different sizes

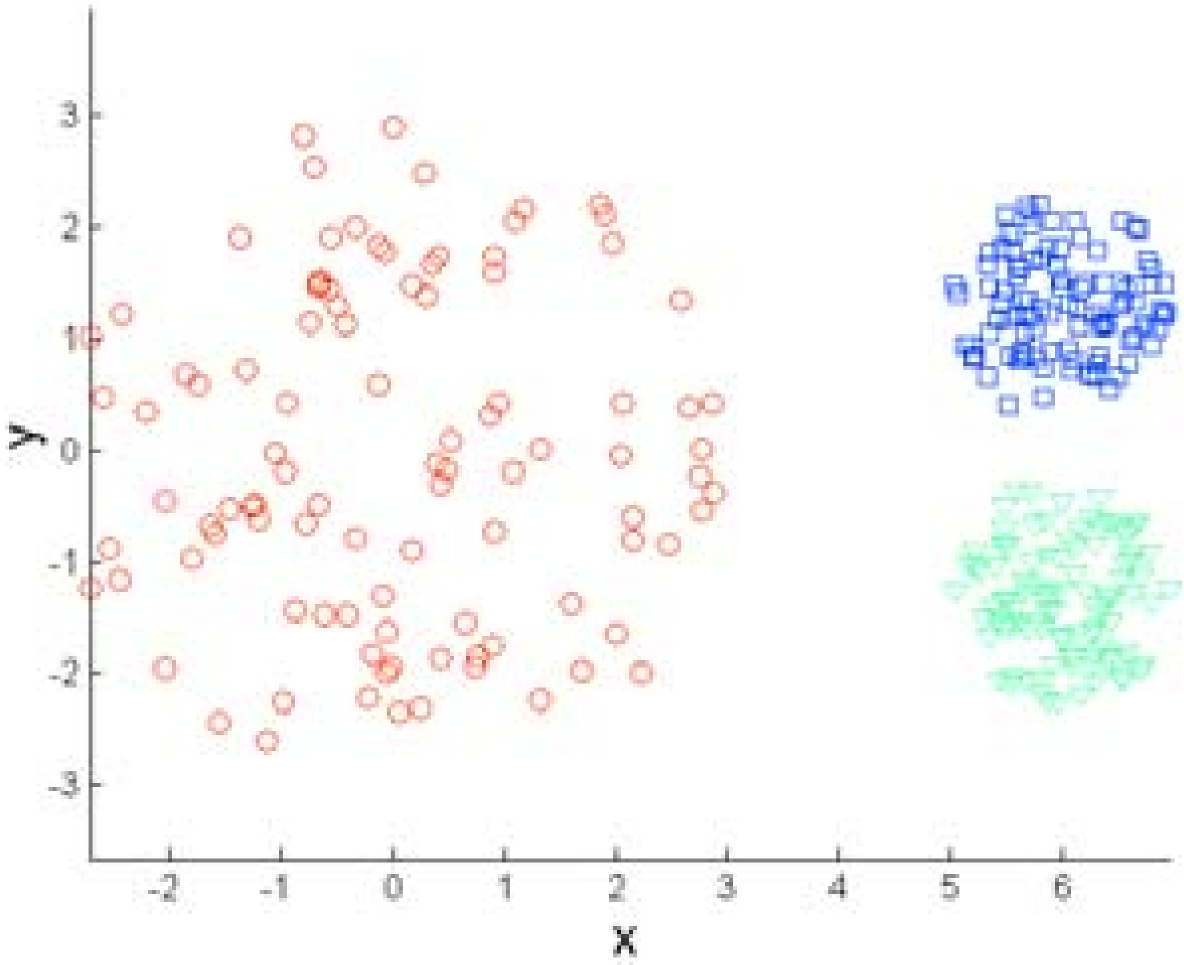


Original Points

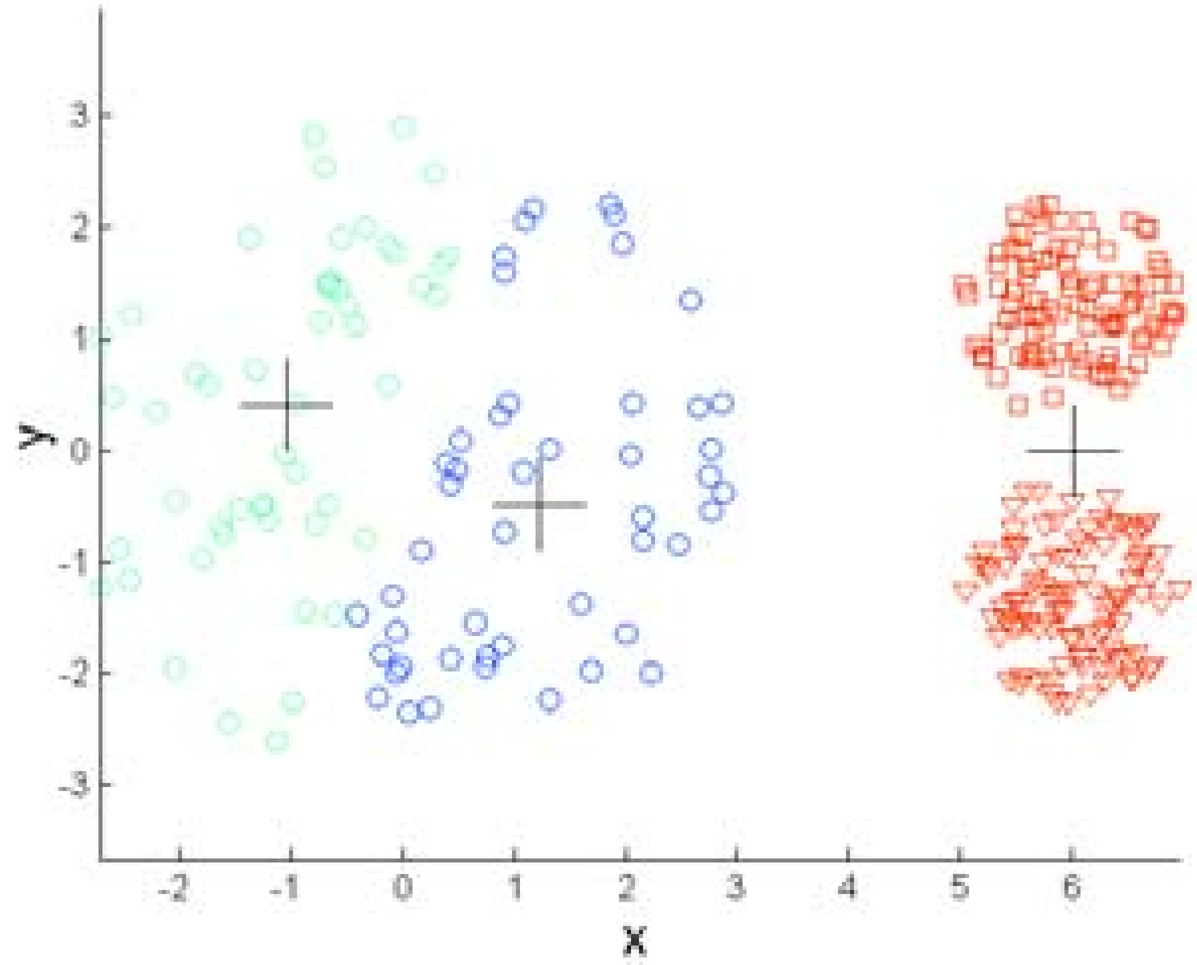


K-means (3 Clusters)

Limitations of k-means: different density

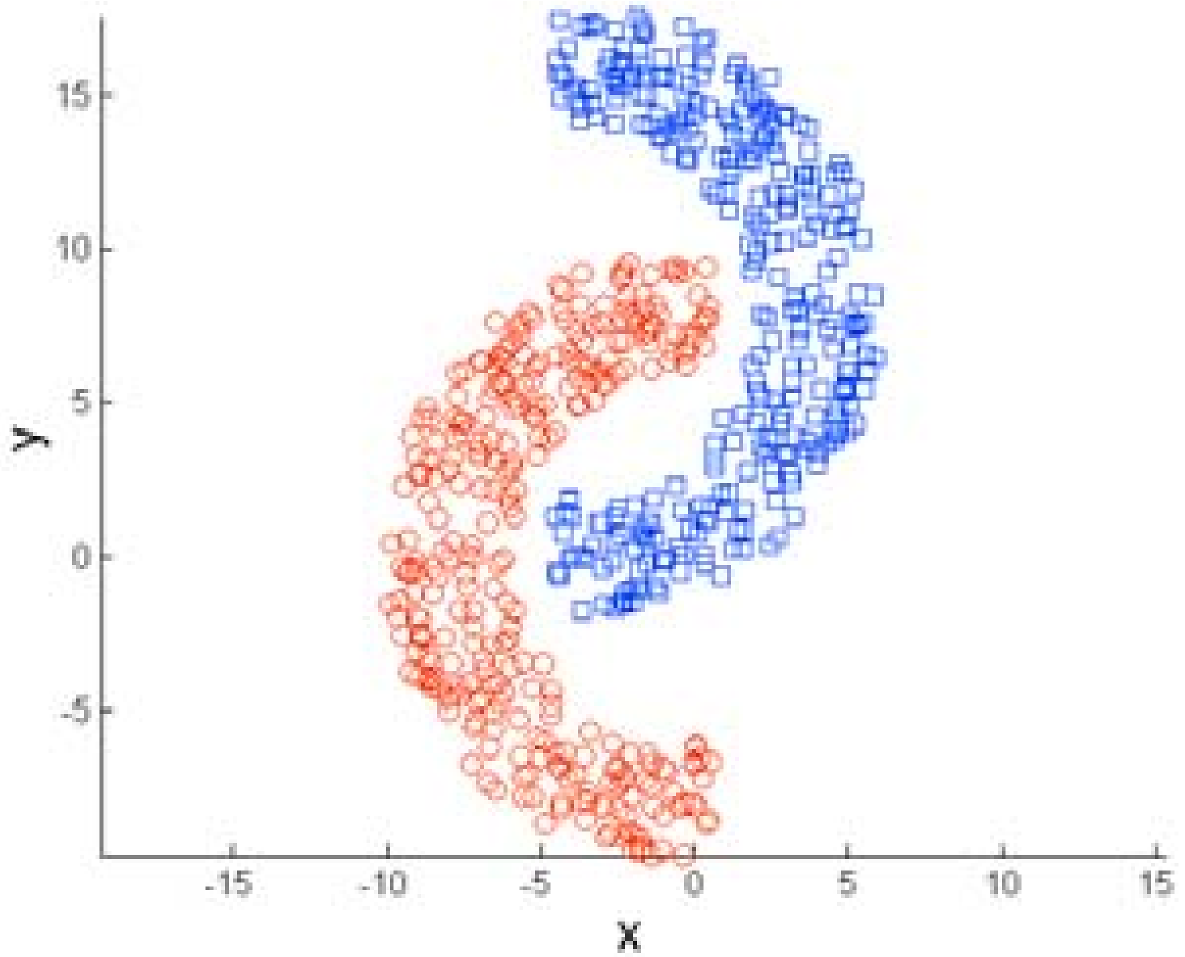


Original Points

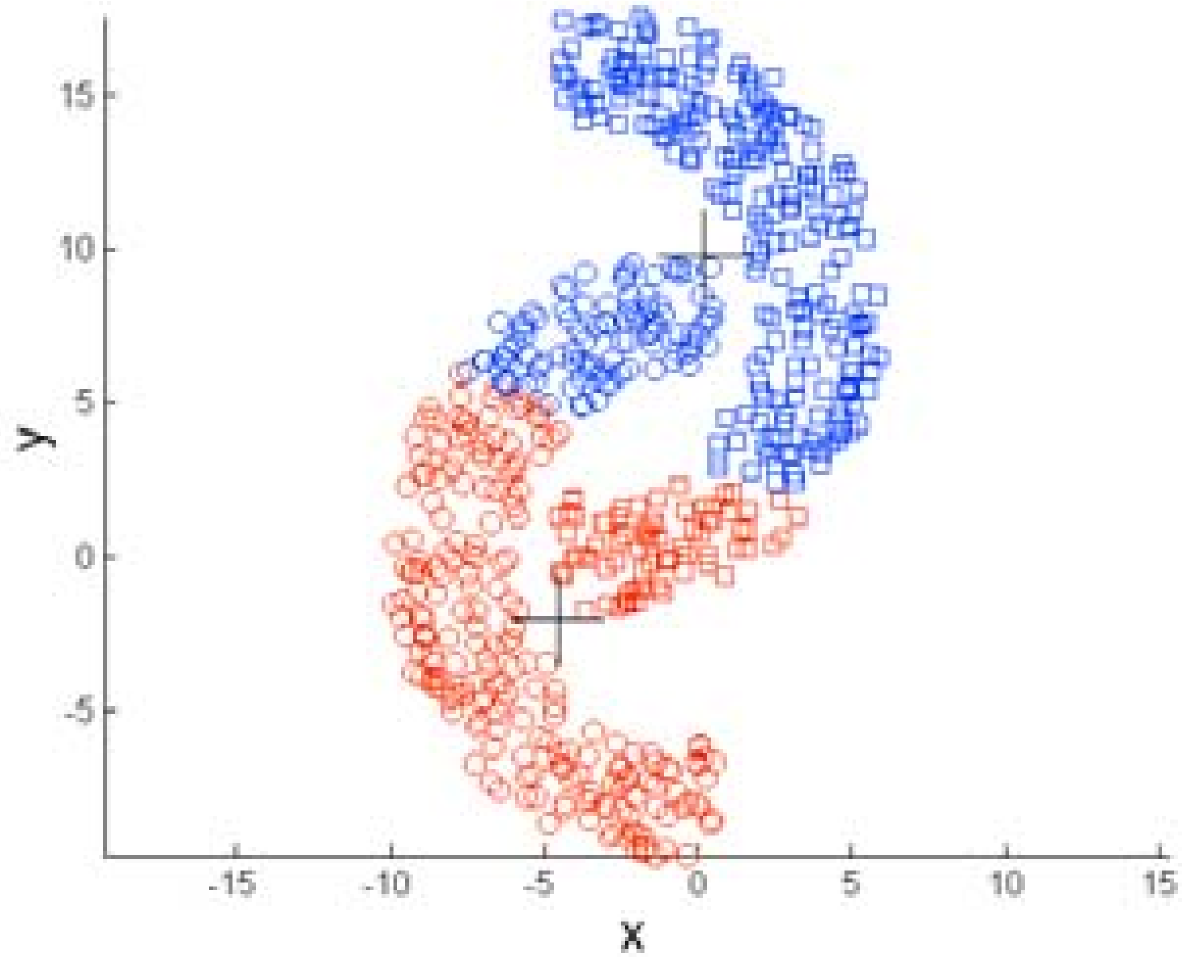


K-means (3 Clusters)

Limitations of k-means: non-spherical shapes



Original Points



K-means (2 Clusters)

Discussion on the k-means algorithm

- finds a local optimum
- often converges quickly
  - but not always
- the choice of initial points can have large influence in the result
- tends to find spherical clusters
- outliers can cause a problem
- different densities may cause a problem

Initialization

- random initialization
- random, but repeat many times and take the best solution
  - helps, but solution can still be bad
- pick points that are distant to each other
  - k-means++
  - provable guarantees

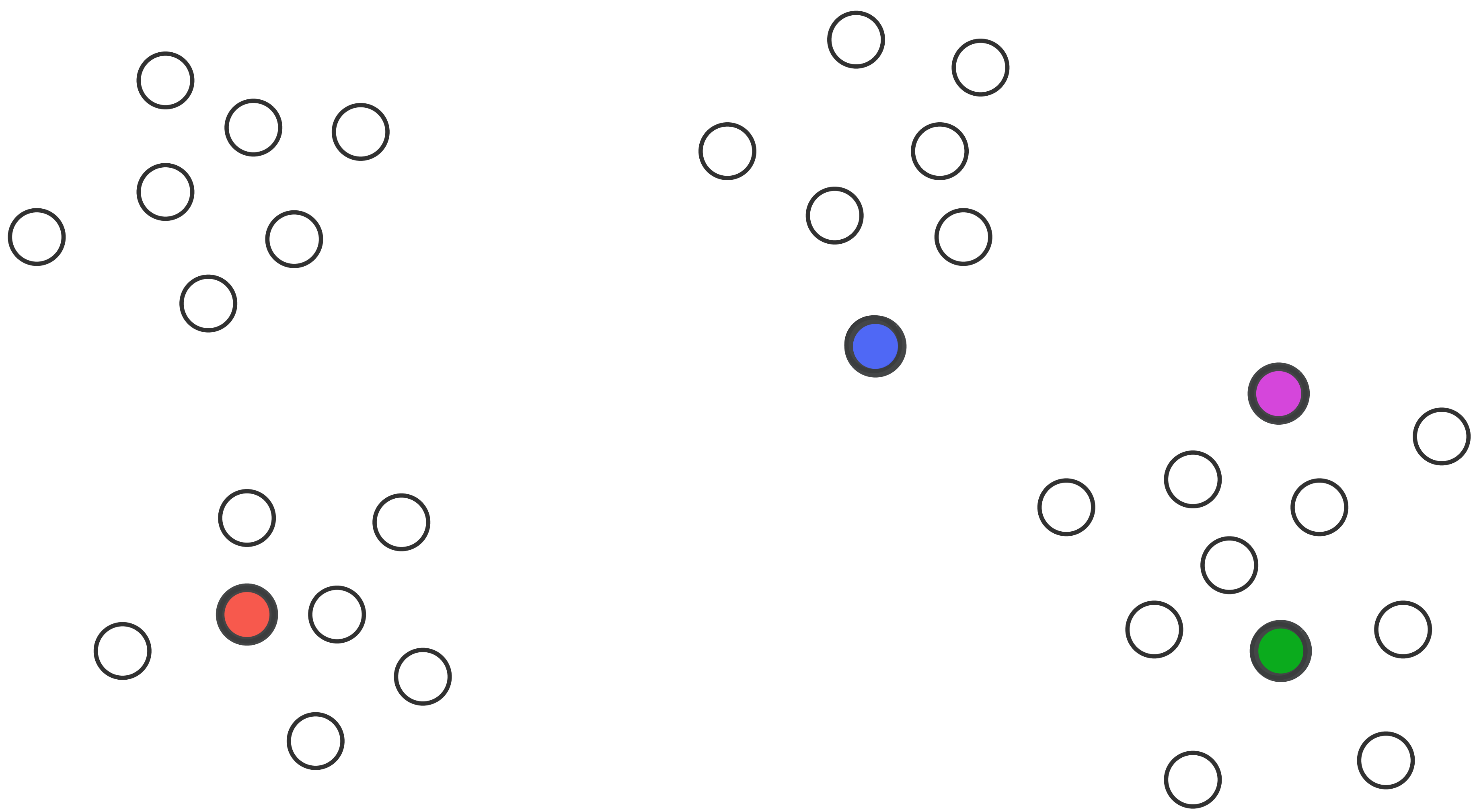
k-means++

David Arthur and Sergei Vassilvitskii

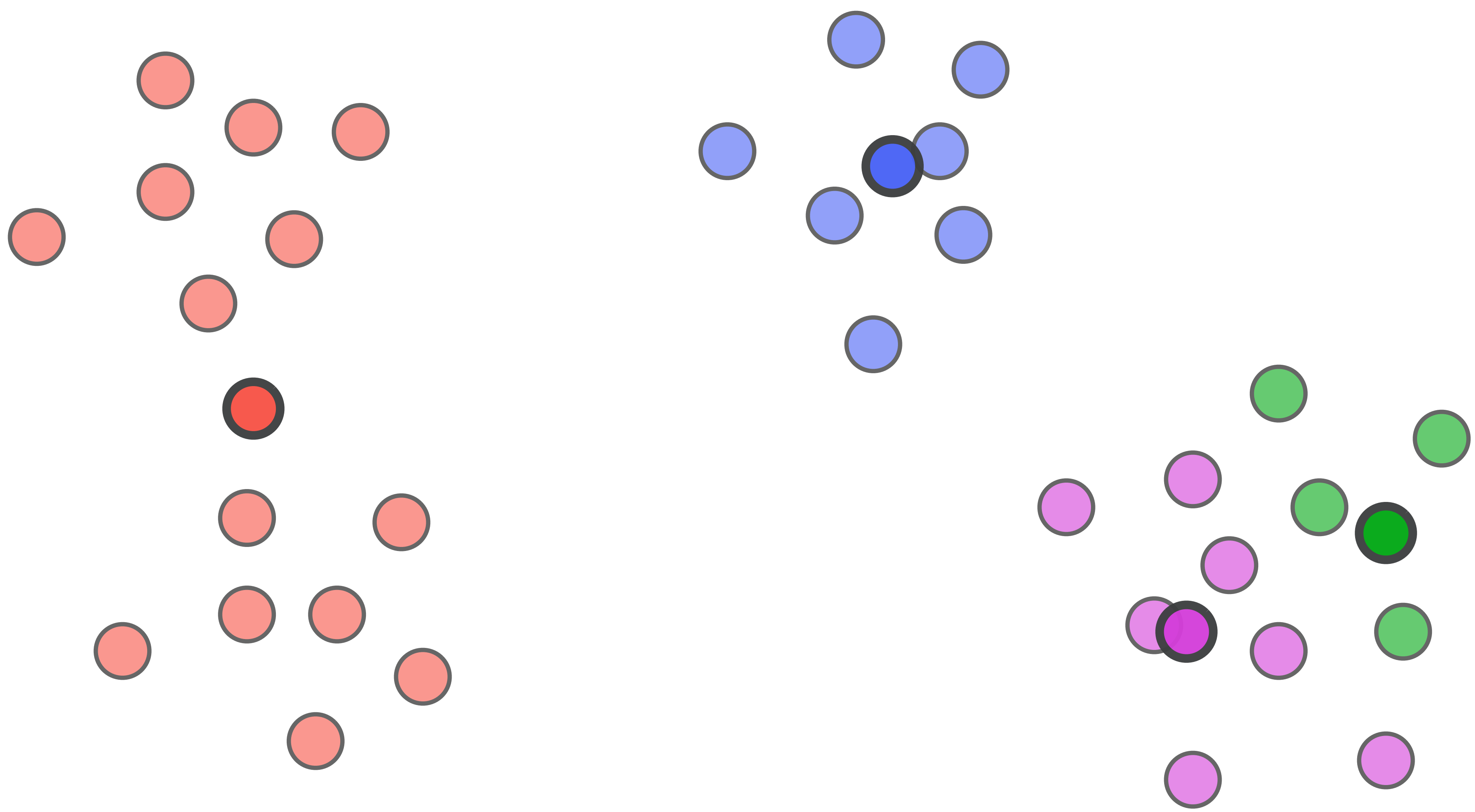
k-means++: The advantages of careful seeding

SODA 2007

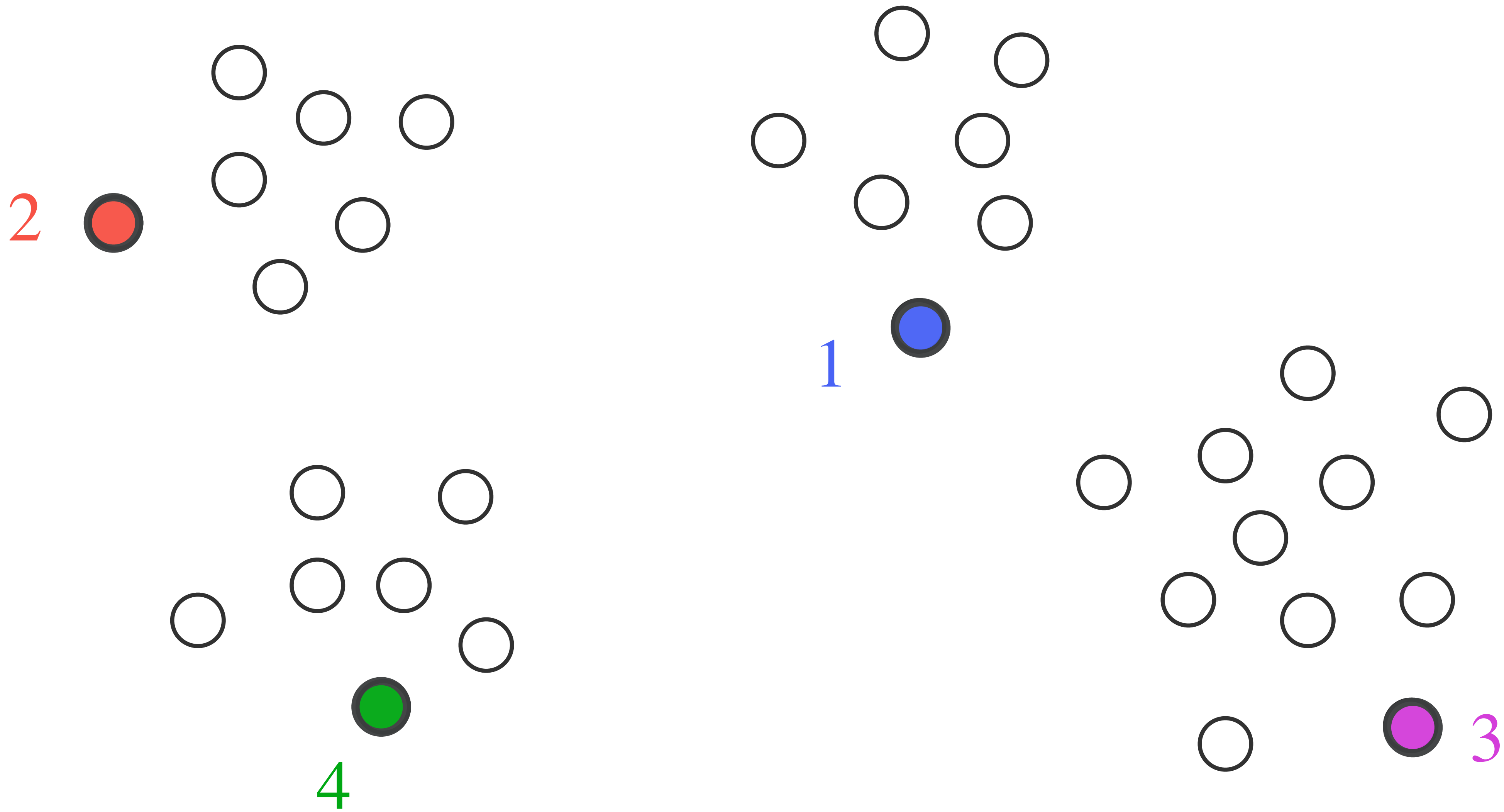
k-means algorithm: random initialization



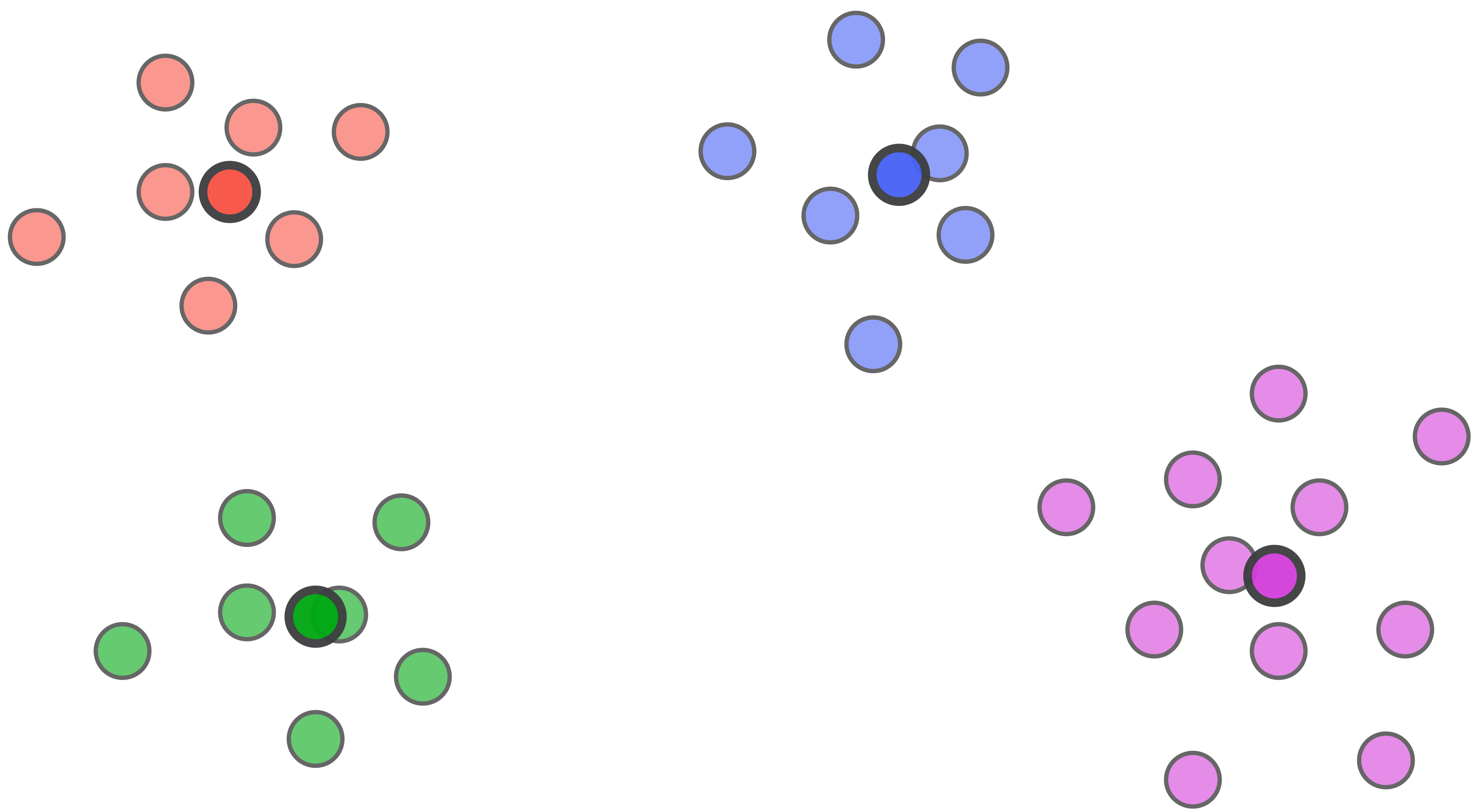
k-means algorithm: random initialization



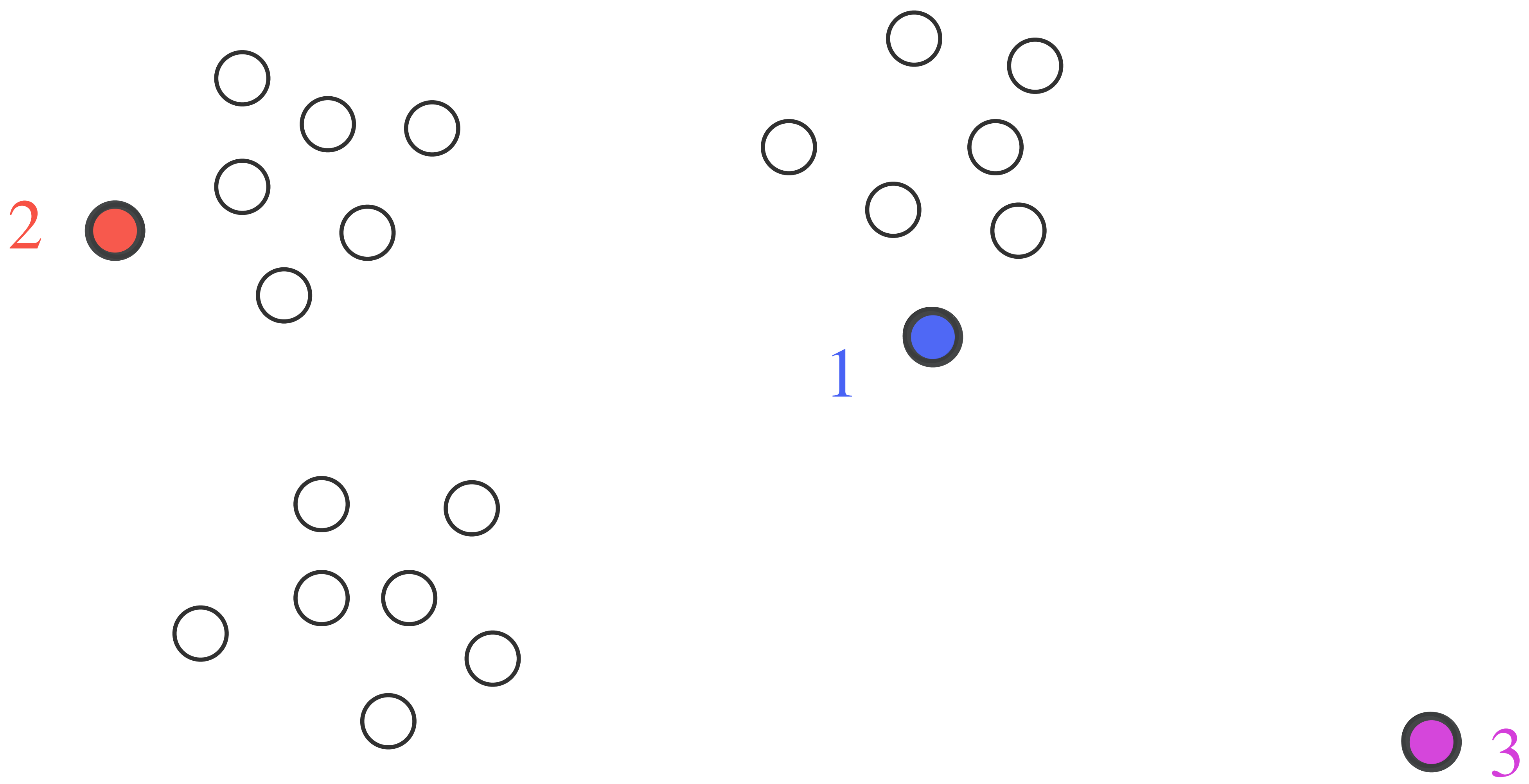
k-means algorithm: initialization with further-first traversal



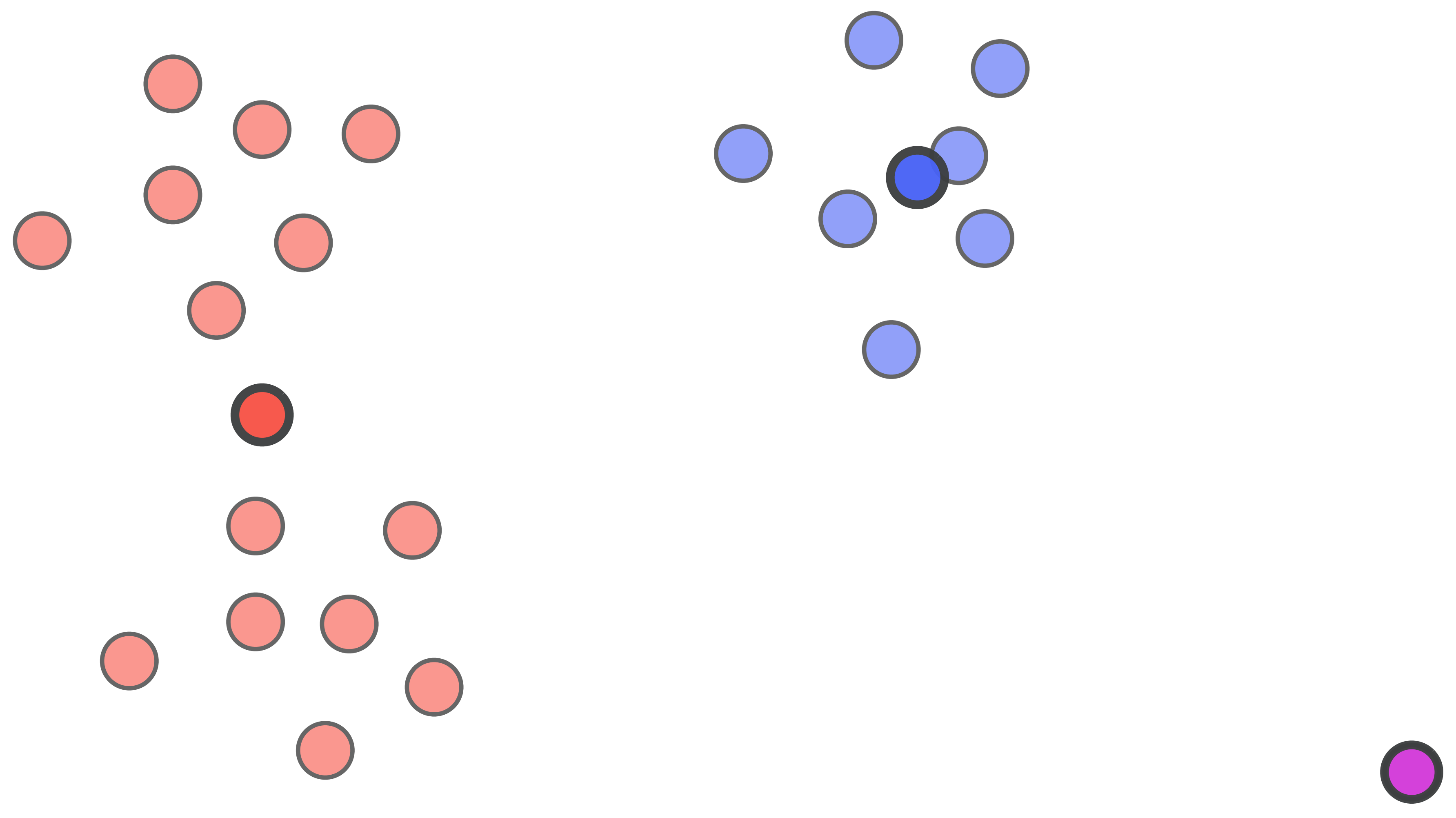
k-means algorithm: initialization with further-first traversal



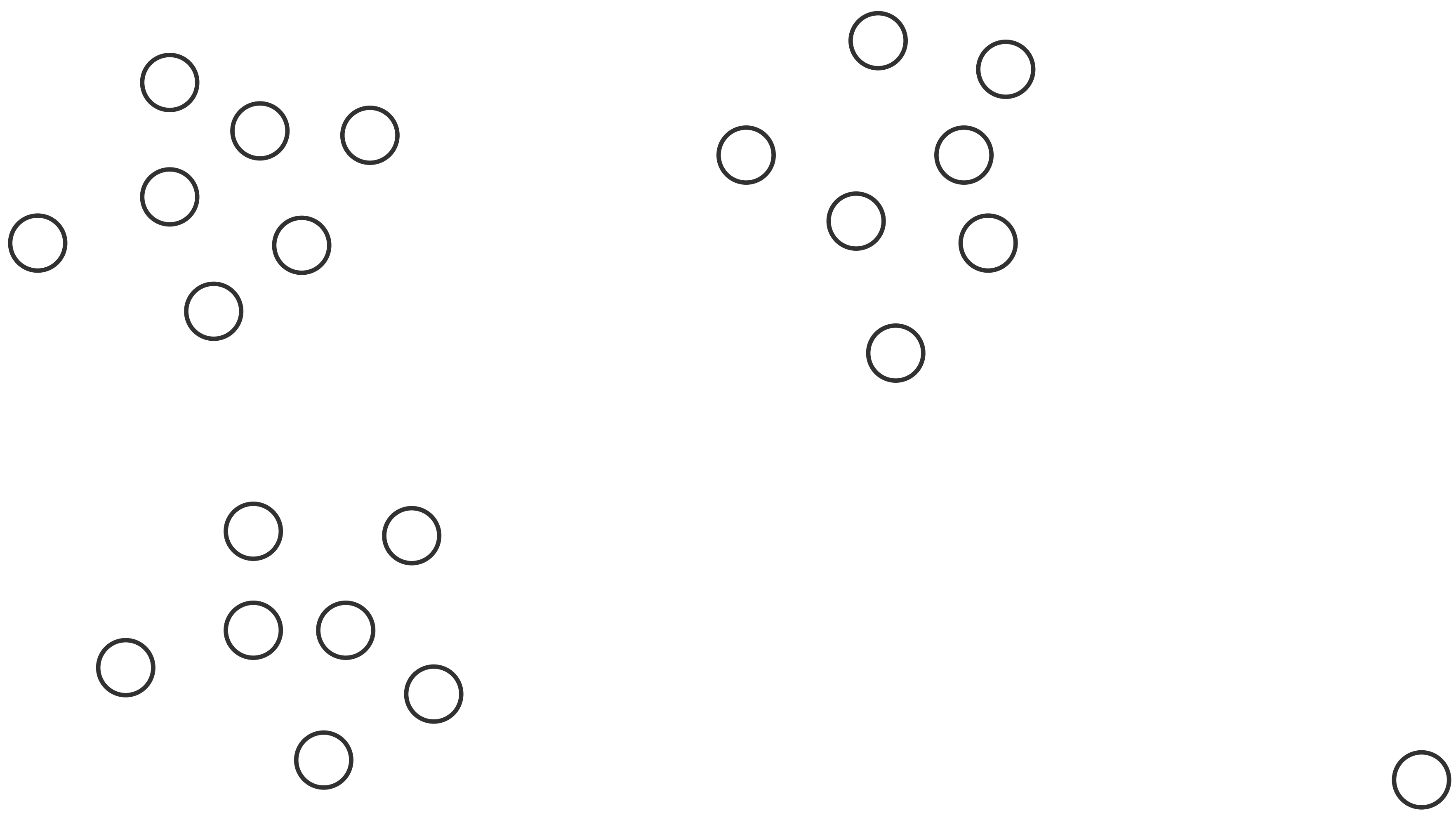
but... sensitive to outliers



but... sensitive to outliers



Here random may work well



k-means++ algorithm

- interpolate between the two methods
- let $D(x)$ be the distance between x and the nearest center selected so far
- choose next center with probability proportional to

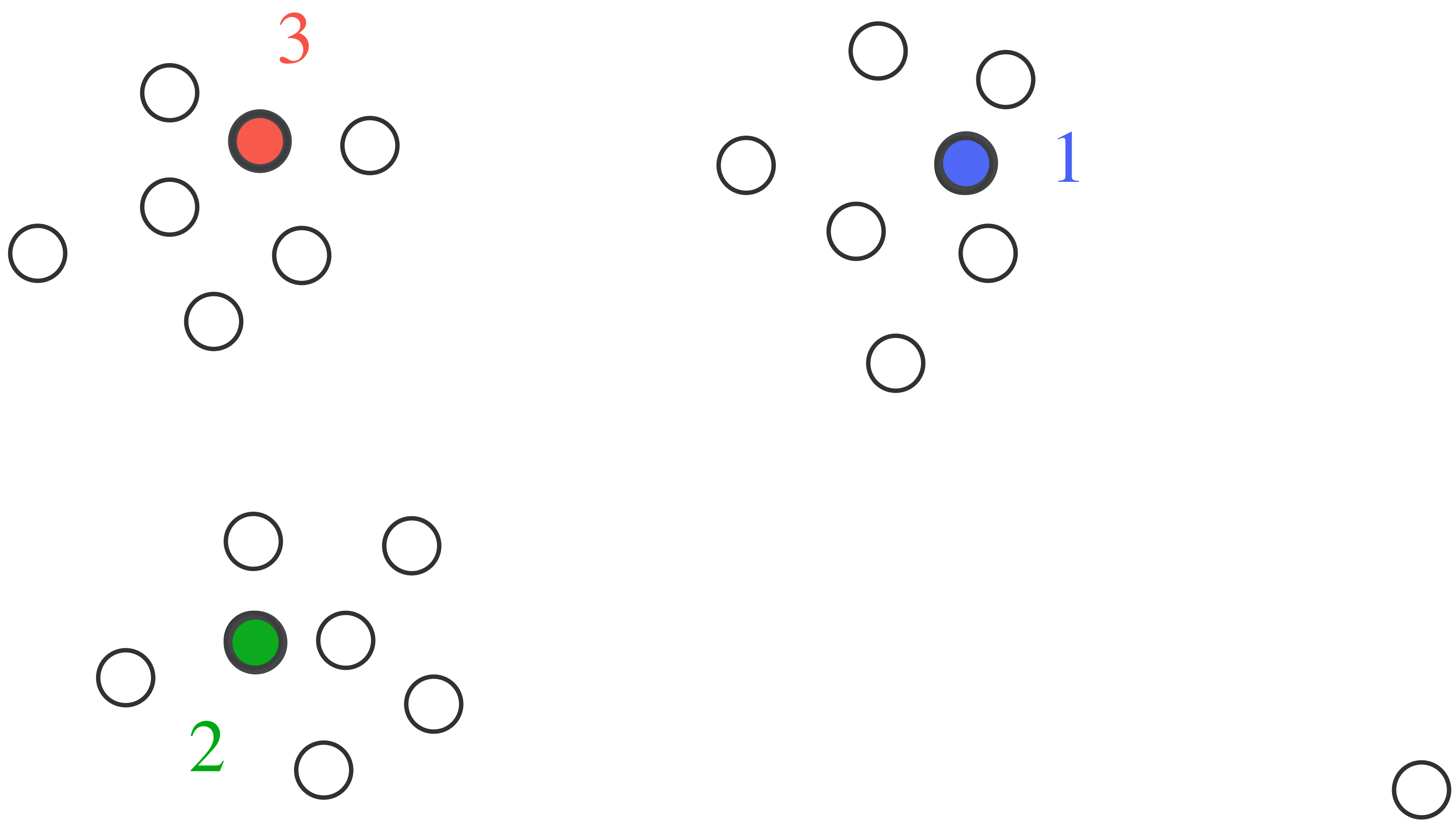
$$(D(x))^a = D^a(x)$$

- $a = 0$ random initialization
- $a = \infty$ furthest-first traversal
- $a = 2$ k-means++

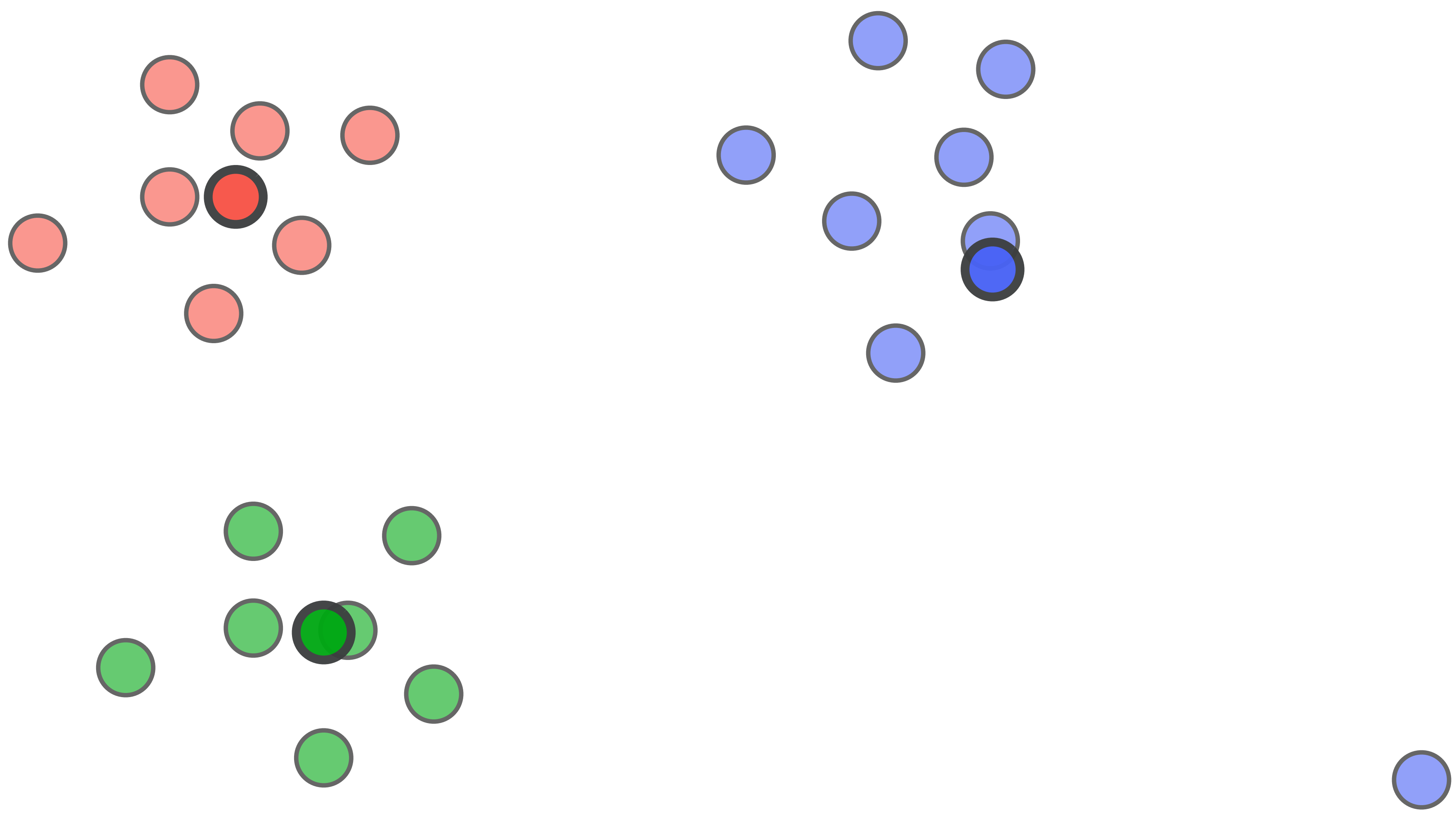
k-means++ algorithm

- initialization phase:
  - choose the first center uniformly at random
  - choose next center with probability proportional to $D^2(x)$
- iteration phase:
  - iterate as in the k-means algorithm until convergence

k-means++ initialization



k-means++ result



k-means++ provable guarantee

Theorem:

k-means++ is O(logk) approximate in expectation

k-means++ provable guarantee

- approximation guarantee comes just from the first iteration (initialization)
- subsequent iterations can only improve cost

k-means++ analysis

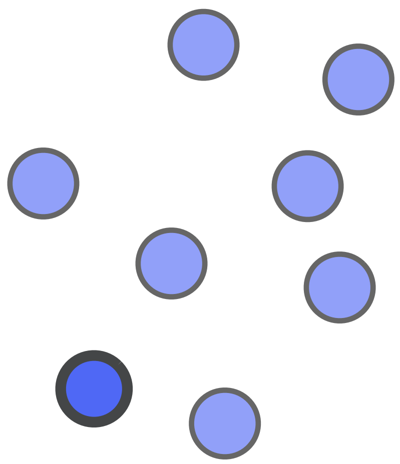
- consider optimal clustering C^*
- assume that k-means++ selects a center from a new optimal cluster
- then
  - k-means++ is 8-approximate in expectation
- intuition: if no points from a cluster are picked, then it probably does not contribute much to the overall error
- an inductive proof shows that the algorithm is O(logk) approximate

k-means++ proof : first cluster

- fix an optimal clustering C^*
- first center is selected uniformly at random
- bound the total error of the points in the optimal cluster of the first center

k-means++ proof : first cluster

- let A be the first cluster
- each point $a_0 \in A$ is equally likely to be selected as center

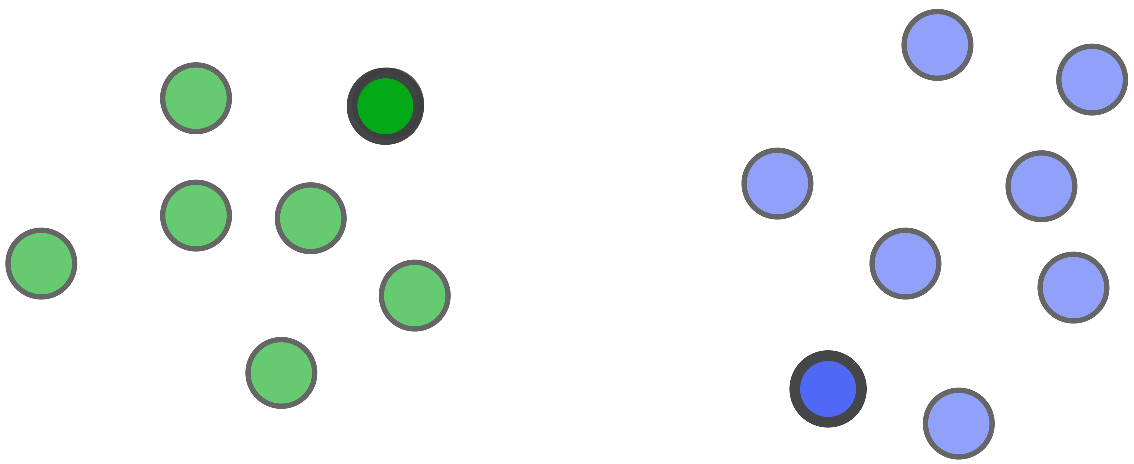


✦ expected error:

$$E[\phi(A)] = \sum_{a_0 \in A} \frac{1}{|A|} \sum_{a \in A} ||a - a_0||^2$$

$$= 2 \sum_{a \in A} ||a - \bar{A}||^2 = 2\phi^*(A)$$

k-means++ proof : other clusters



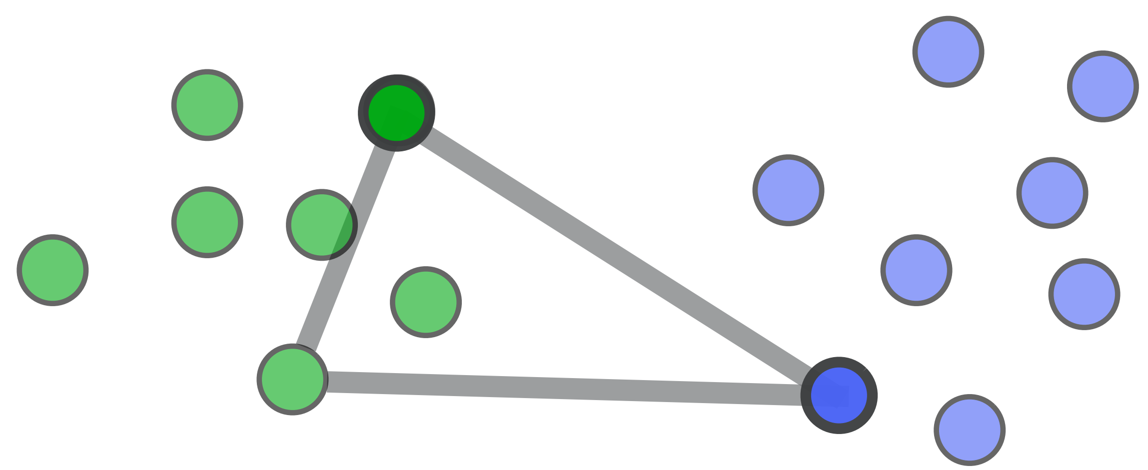
- suppose next center is selected from a new cluster in the optimal clustering C^*
- bound the total error of that cluster

k-means++ proof : other clusters

- let B be the second cluster and b_0 the center selected

$$E[\phi(B)] = \sum_{b_0 \in B} \frac{D^2(b_0)}{\sum_{b \in B} D^2(b)} \sum_{b \in B} \min\{D(b), ||b - b_0||^2\}$$

triangle inequality:



$$D(b_0) \leq D(b) + ||b - b_0||$$

$$D^2(b_0) \leq 2D^2(b) + 2||b - b_0||^2$$

k-means++ proof : other clusters

$$D^2(b_0) \leq 2D^2(b) + 2||b - b_0||^2$$

- average over all points b in B

$$D^2(b_0) \leq \frac{2}{|B|} \sum_{b \in B} D^2(b) + \frac{2}{|B|} \sum_{b \in B} ||b - b_0||^2$$

- recall

$$E[\phi(B)] = \sum_{b_0 \in B} \frac{D^2(b_0)}{\sum_{b \in B} D^2(b)} \sum_{b \in B} \min\{D(b), ||b - b_0||^2\}$$

$$\leq 4 \sum_{b \in B} \frac{1}{|B|} \sum_{b_0 \in B} ||b - b_0||^2 = 4 \sum_{b \in B} 2||b - \bar{B}||^2 = 8\phi^*(B)$$

k-means++ analysis

- if that k-means++ selects a center from a new optimal cluster
- then
  - k-means++ is 8-approximate in expectation
- an inductive proof shows that the algorithm is O(logk) approximate

Lesson learned

- no reason to use k-means and not k-means++

- k-means++ :
  - easy to implement
  - provable guarantee
  - works well in practice

The k-median problem

- consider set $X=\{x_1,\ldots,x_n\}$ of n points in R^d
- assume that the number k is given
- problem:
  - find k points $c_1,\ldots,c_k$ (named medians)
  - and partition X into $\{X_1,\ldots,X_k\}$ by assigning each point x_i in X to its nearest cluster median,
  - so that the cost

$$\sum_{i=1}^{n} \min_j ||x_i - c_j||_2 = \sum_{j=1}^{k} \sum_{x \in X_j} ||x - c_j||_2$$

is minimized

the k-medoids algorithm

or PAM (partitioning around medoids)

1. randomly (or with another method) choose k medoids $\{c_1,\ldots,c_k\}$ from the original dataset X
2. assign the remaining $n-k$ points in X to their closest medoid c_j
3. for each cluster, replace each medoid by a point in the cluster that improves the cost
4. repeat (go to step 2) until convergence

Discussion on the k-medoids algorithm

- very similar to the k-means algorithm
- same advantages and disadvantages
- how about efficiency?

The Local-kMedian algorithm

- Pick a random set of k cluster centers $S = \{c_1, \ldots, c_k\}$
- $\mathbb{P}_S$: partition induced by assigning each data point to its closest point in S
- Repeat
  - Find S' ``**similar**" to S
  - If $\text{kMedian-Cost}(\mathbb{P}_{S'}) < \text{kMedian-Cost}(\mathbb{P}_S)$ then $S = S'$
- Until convergence
- Similar S' is find via **swaps** or **p-swaps.**

The Local-kMedian algorithm

- Proposition: If $\mathbb{P}_S$ is the partition output by the Local-kMedian with single swaps and $\mathbb{P}_{S^*}$ is the optimal partition for the k-Median problem, then

$$\text{kMedian-Cost}(\mathbb{P}_S) \leq 5 \times \text{kMedian-Cost}(\mathbb{P}_{S^*})$$

The Local-kMedian algorithm

- Proposition: If $\mathbb{P}_S$ is the partition output by the Local-kMedian with p-swaps and $\mathbb{P}_{S*}$ is the optimal partition for the k-Median problem, then

$$\text{kMedian-Cost}(\mathbb{P}_S) \leq (3 + 2/p) \times \text{kMedian-Cost}(\mathbb{P}_{S*})$$

The k-center problem

- consider set $X=\{x_1,...,x_n\}$ of n points in R^d
- assume that the number k is given
- problem:
  - find k points $c_1,...,c_k$ (named centers)
  - and partition X into $\{X_1,...,X_k\}$ by assigning each point x_i in X to its nearest cluster center,
  - so that the cost

is minimized $$\max_{i=1}^{n} \min_{j=1}^{k} ||x_i - c_j||_2$$

Properties of the k-center problem

- NP-hard for dimension $d \geq 2$
- for $d=1$ the problem is solvable in polynomial time (how?)
- a simple combinatorial algorithm works well

The k-center problem

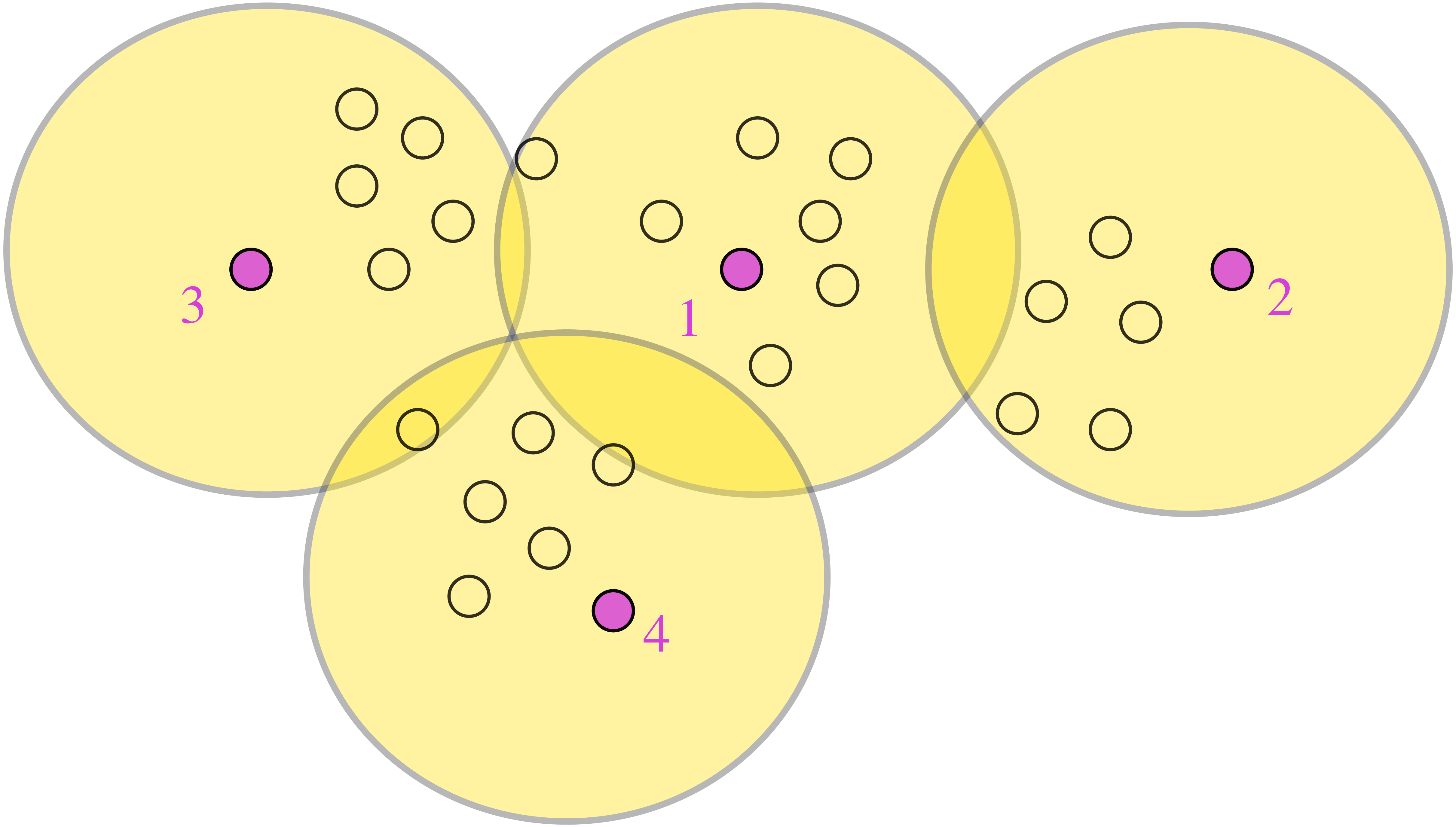
- consider set $X=\{x_1,\ldots,x_n\}$ of n points in R^d
- assume that the number k is given
- problem:
  - find k points $c_1,\ldots,c_k$ (named centers)
  - and partition X into $\{X_1,\ldots,X_k\}$ by assigning each point x_i in X to its nearest cluster center,
  - so that the cost

is minimized $$\max_{i=1}^{n} \min_{j=1}^{k} ||x_i - c_j||_2$$

Furthest-first traversal algorithm

- pick any data point and label it 1
- for $i=2,...,k$
  - find the unlabeled point that is furthest from $\{1,2,...,i-1\}$
  - // use $d(x,S) = \min_{y \in S} d(x,y)$
  - label that point i
- assign the remaining unlabeled data points to the closest labeled data point

Furthest-first traversal algorithm: example



Furthest-first traversal algorithm

- furthest-first traversal algorithm gives a factor 2 approximation

Furthest-first traversal algorithm

- pick any data point and label it 1
- for $i=2,\ldots,k$
  - find the unlabeled point that is furthest from $\{1,2,\ldots,i-1\}$
  - // use $d(x,S) = \min_{y \in S} d(x,y)$
  - label that point i
  - $p(i) = \text{argmin}_{j<i}\, d(i,j)$
  - $R_i = d(i,p(i))$
- assign the remaining unlabeled data points to the closest labeled data point

Analysis

- Claim 1: $R_1 \geq R_2 \geq \ldots \geq R_k$
- proof:
  - $R_j = d(j,p(j))$
    $= d(j,\{1,2,\ldots,j-1\})$
    $\leq d(j,\{1,2,\ldots,i-1\})$ // j > i
    $\leq d(i,\{1,2,\ldots,i-1\}) = R_i$

Analysis

- Claim 2:
  - let C be the clustering produced by the FFT algorithm
  - let $R(C)$ be the cost of that clustering
  - then $R(C) = R_{k+1}$

- proof:
  - for any $i>k$ we have :

$$d(i,\{1,2,\ldots,k\}) \leq d(k+1,\{1,2,\ldots,k\}) = R_{k+1}$$

Analysis

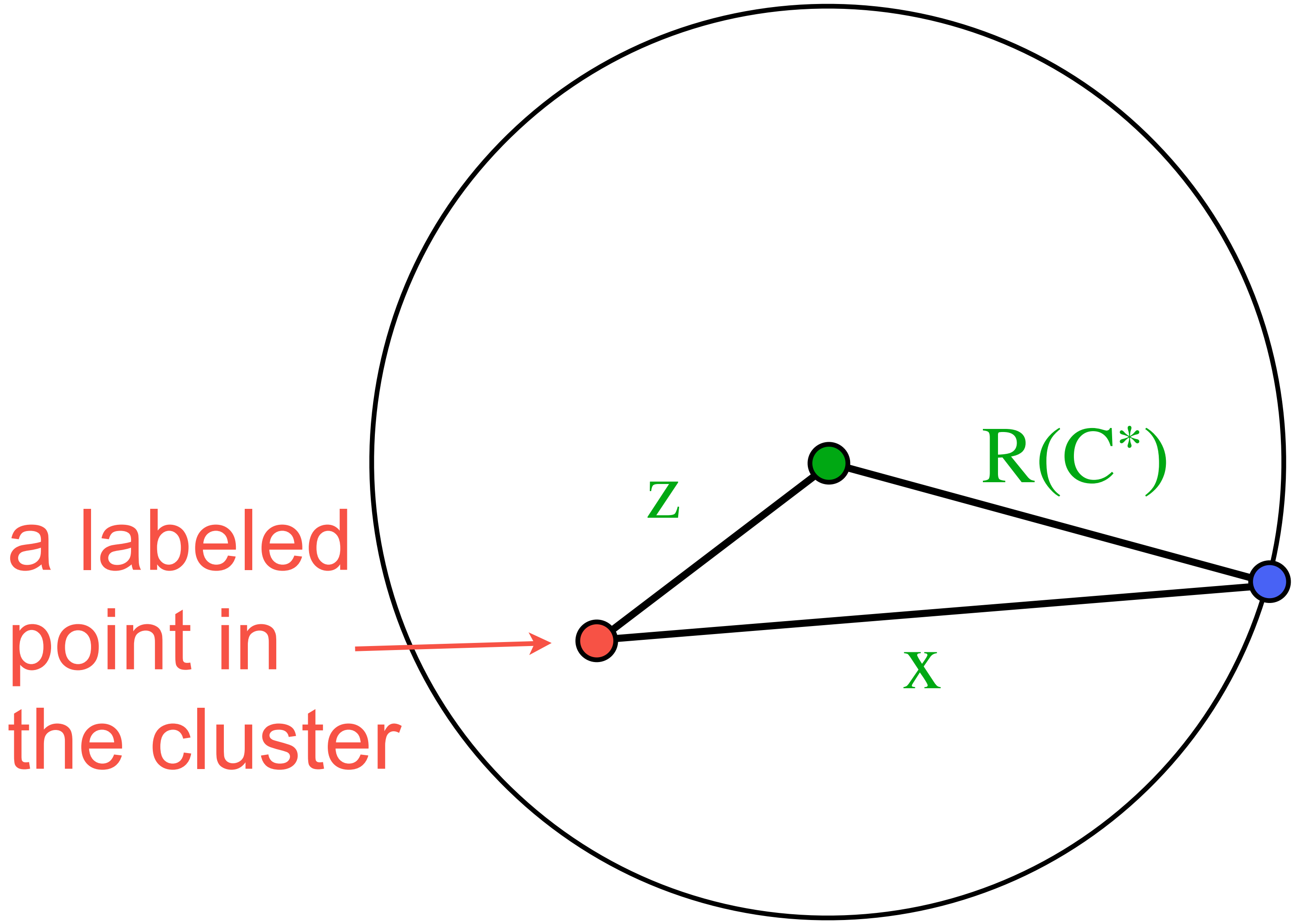
- Theorem
  - let C be the clustering produced by the FFT algorithm
  - let C^* be the optimal clustering
  - then $R(C) \le 2R(C^*)$
- proof:
  - let $C^*_1, \ldots, C^*_k$ be the clusters of the optimal k-clustering
  - if these clusters contain points $\{1,\ldots,k\}$ then

$$R(C) \le 2R(C^*)$$

  - otherwise suppose that one of these clusters contains two or more of the points in $\{1,\ldots,k\}$
  - these points are at distance at least R_k from each other
  - this (optimal) cluster must have radius

$$\tfrac{1}{2} R_k \ge \tfrac{1}{2} R_{k+1} = \tfrac{1}{2} R(C)$$

$R(C) \leq 2R(C^*)$



$R(C) \leq x \leq z + R(C^*) \leq 2R(C^*)$