# CS-E4600
# Algorithmic methods for data mining

Aristides Gionis
dept of Computer Science

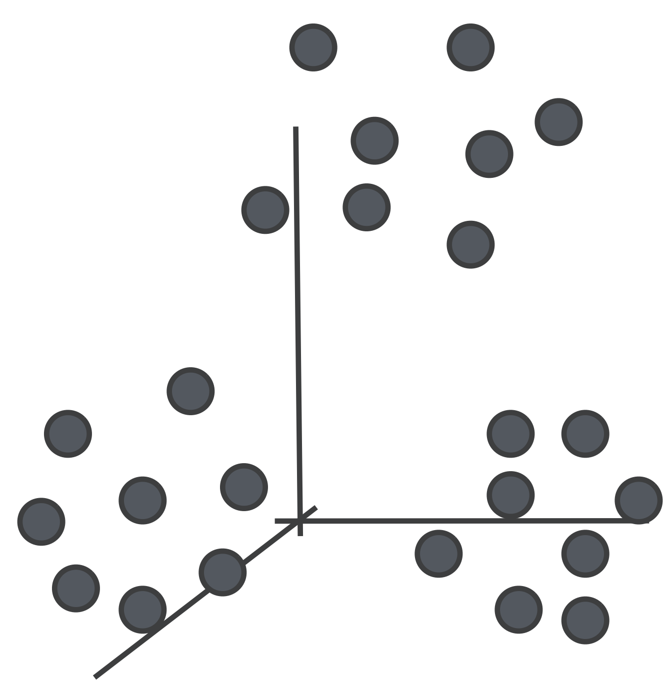
## slide set 9: data clustering

# reading assignment

LRU book : chapter 7

# 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

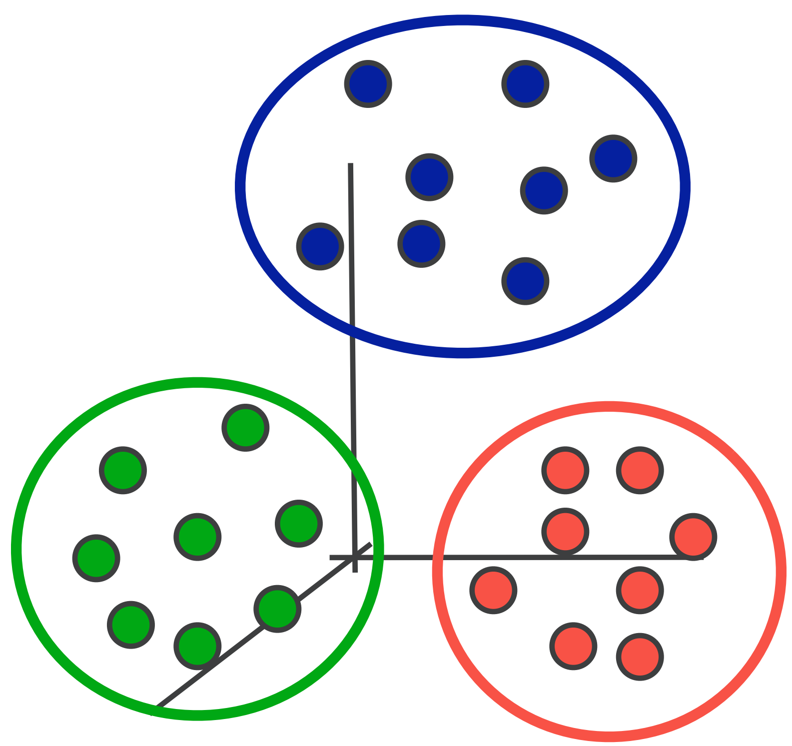




Aalto University

# 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



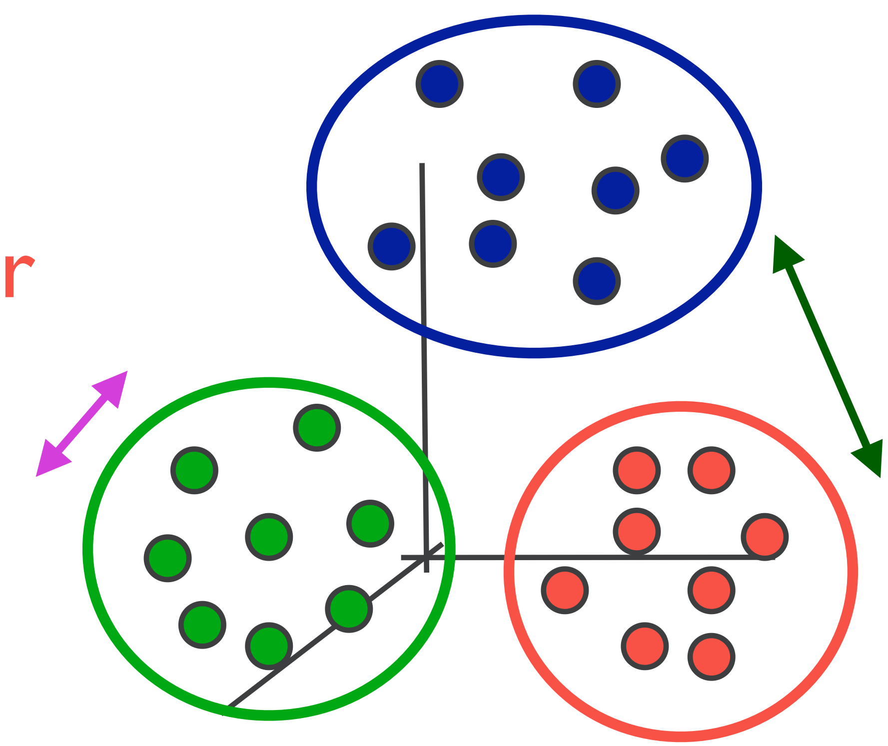


Aalto University

# 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



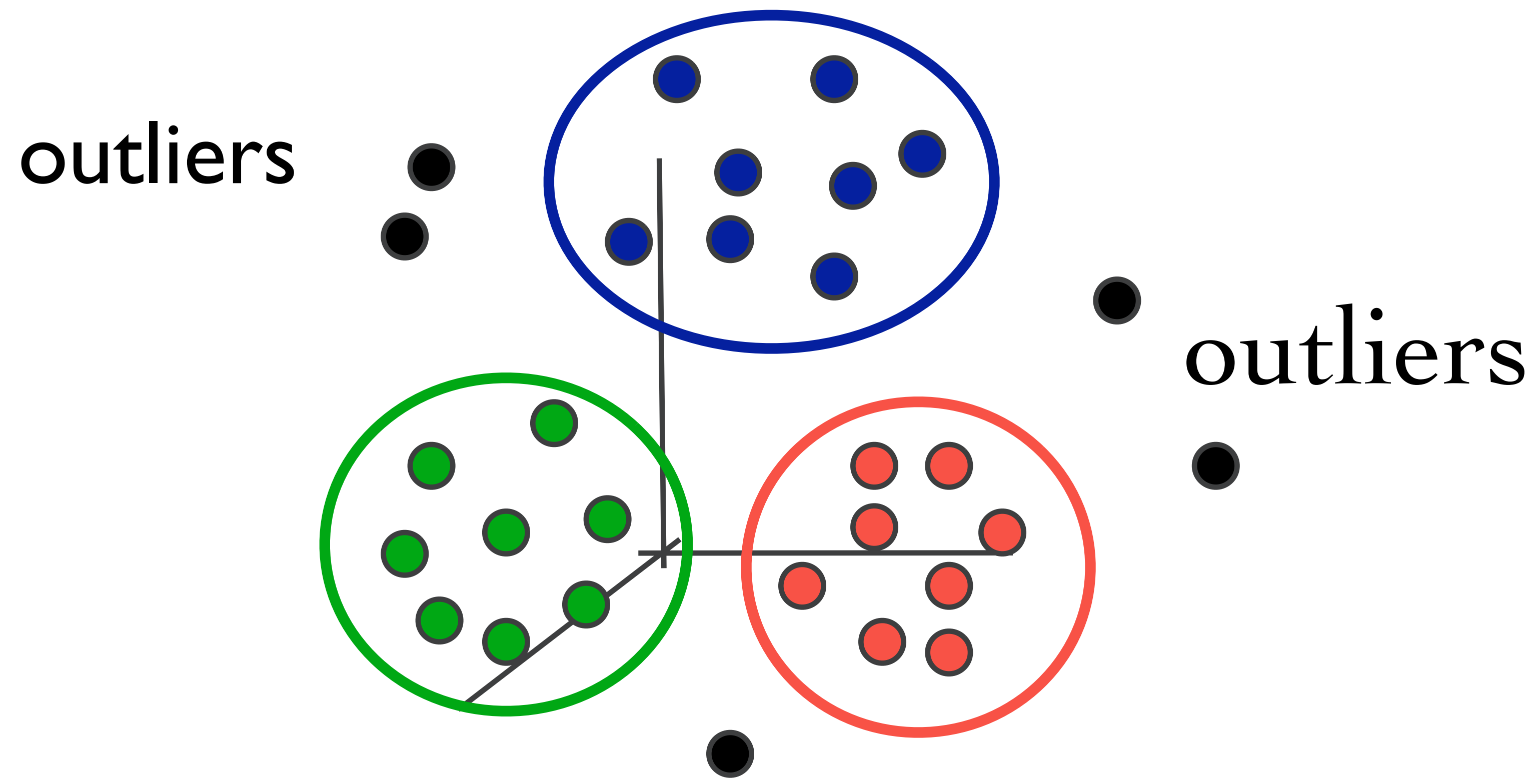
maximize
inter-cluster
distances



Aalto University

# outliers

outliers are objects that do not belong to any cluster, or form very small clusters




sometimes, we are interested in discovering outliers, not clusters (outlier detection)



Aalto University

# clustering — why 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

market segmentation

cluster customers based on their behavior

bioinformatics

cluster similar proteins together (similarity wrt chemical structure and/or functionality etc)

many more...

# clustering — high-level definition

given a collection of data objects

find a grouping so that

- similar objects are in the same cluster
- dissimilar objects are in different clusters



Aalto University

# clustering — basic questions

what does similar mean?

what is a good partition of the objects?

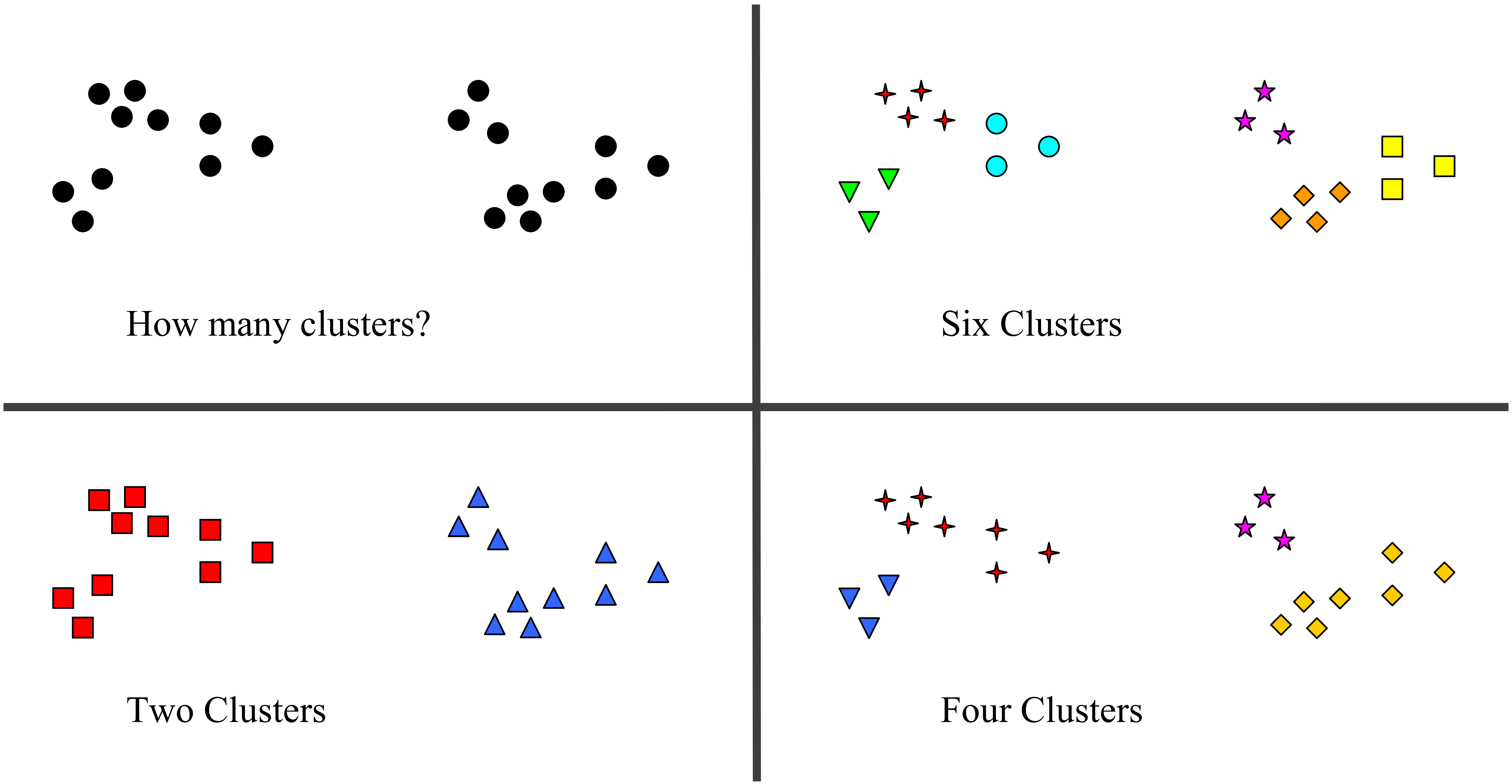
i.e., how is the quality of a solution measured?

how to find a good partition?



Aalto University

# notion of a cluster can be ambiguous






Aalto University

# types of clusterings

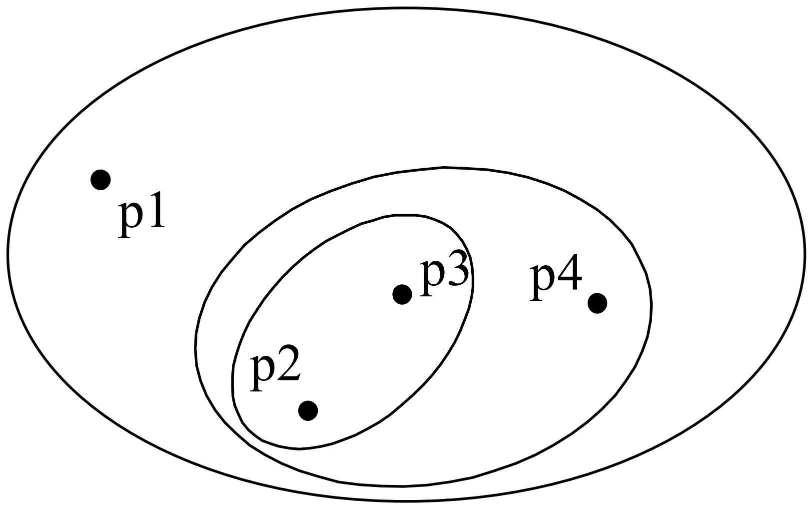
partitional

each object belongs in exactly one cluster

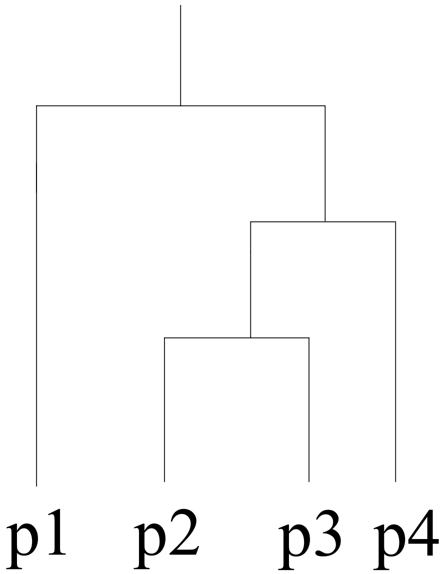
hierarchical

a set of nested clusters organized in a tree

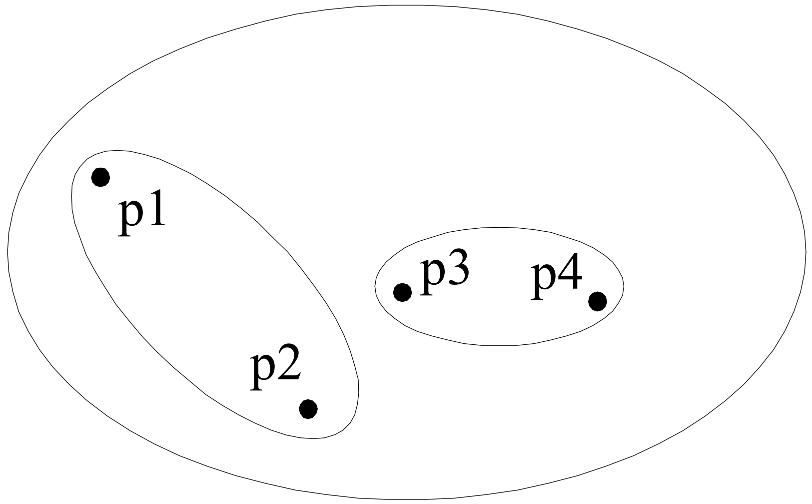
# hierarchical clustering



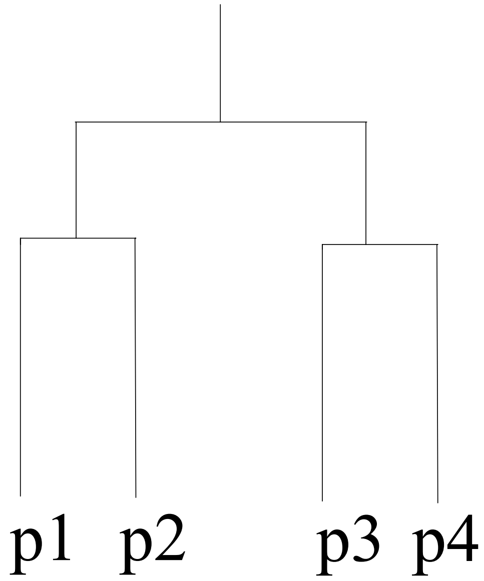

Hierarchical Clustering




Dendrogram




Hierarchical Clustering

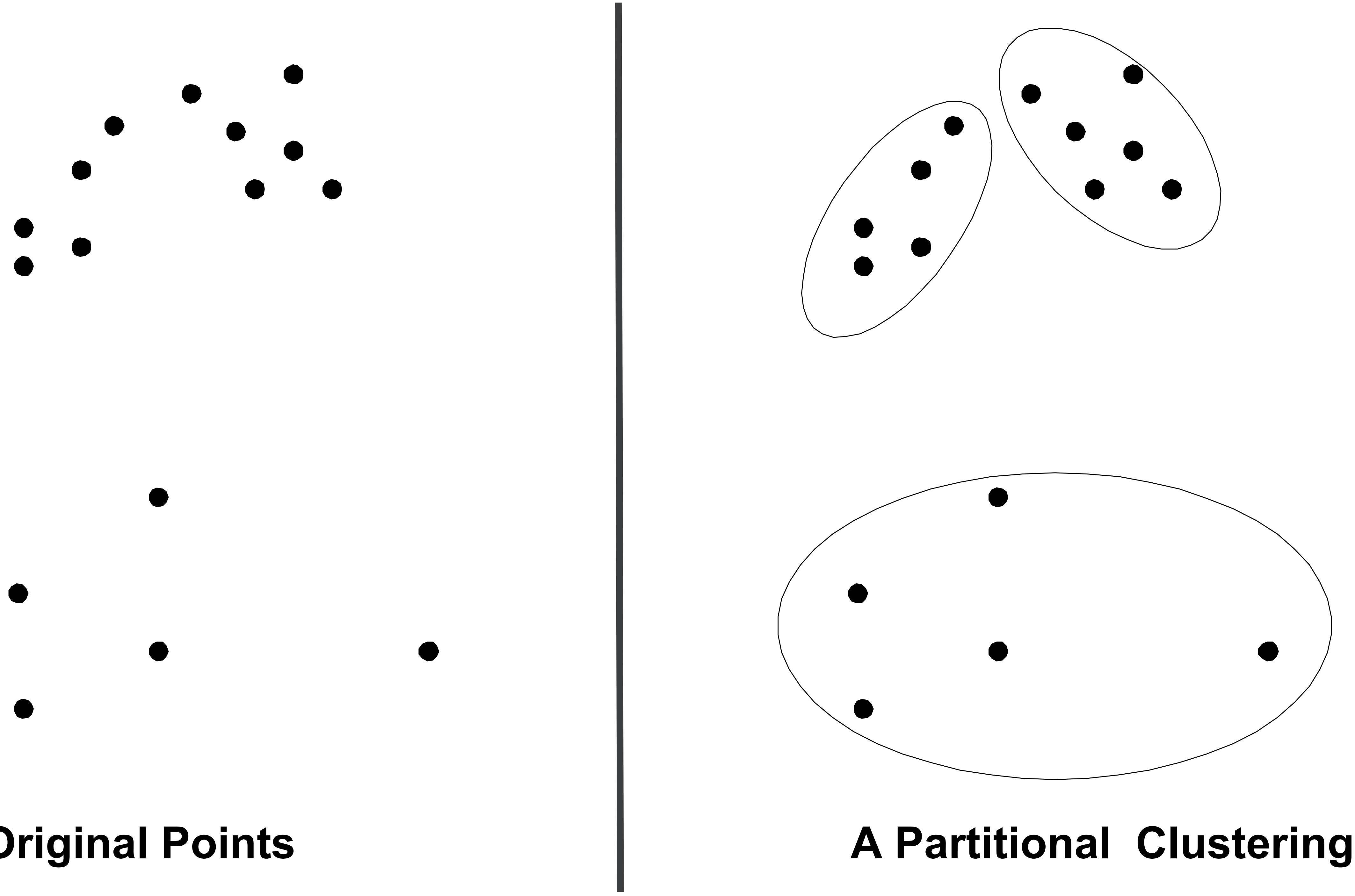



Dendrogram



Aalto University

# partitional clustering



**Original Points** | **A Partitional Clustering**



Aalto University

# 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



Aalto University

# 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



Aalto University

# 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

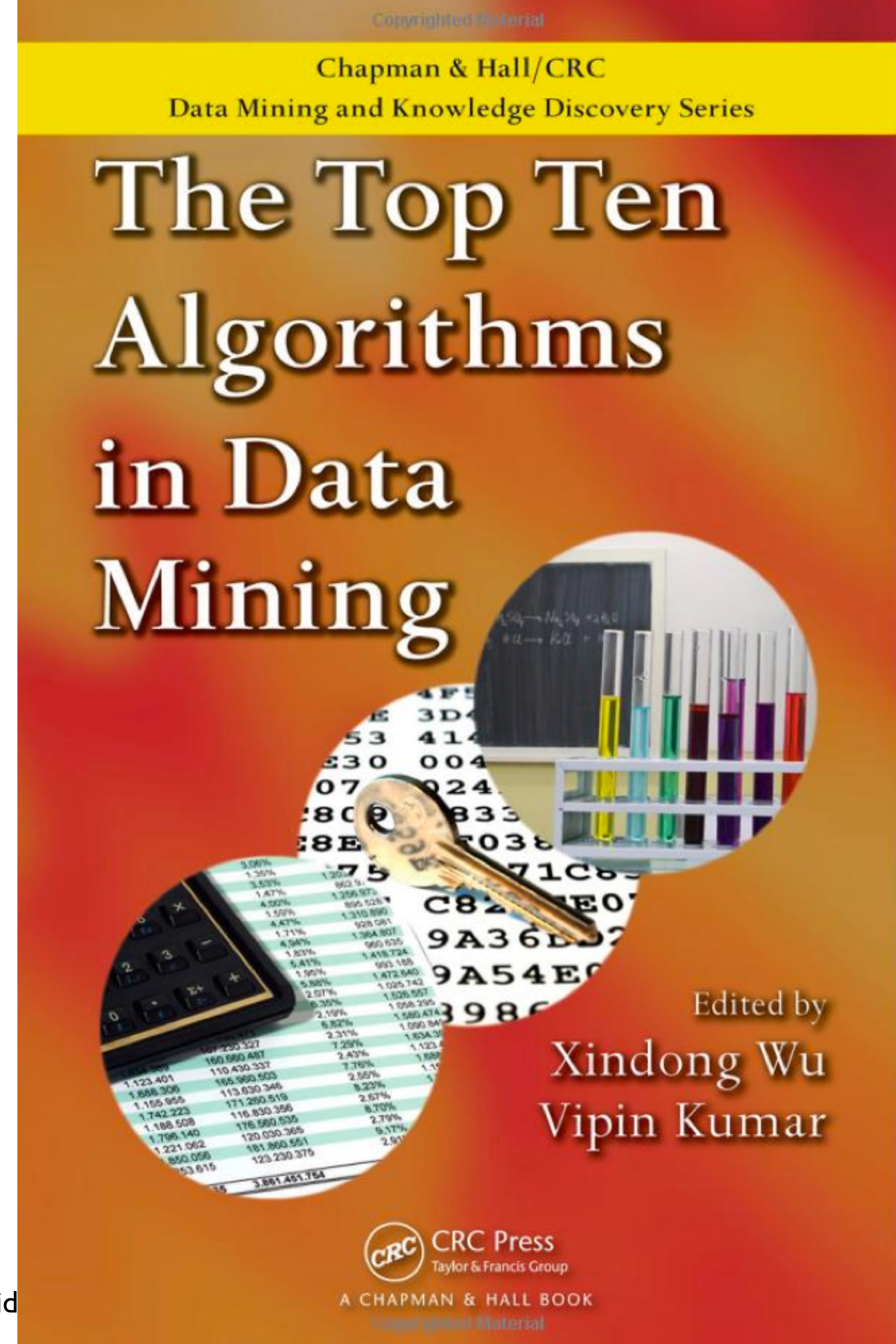
$$\sum_{i=1}^{n} \min_j ||x_i - c_j||_2^2 = \sum_{j=1}^{k} \sum_{x \in X_j} ||x - c_j||_2^2$$

is minimized

# the k-means algorithm

voted among the top-10 algorithms in data mining

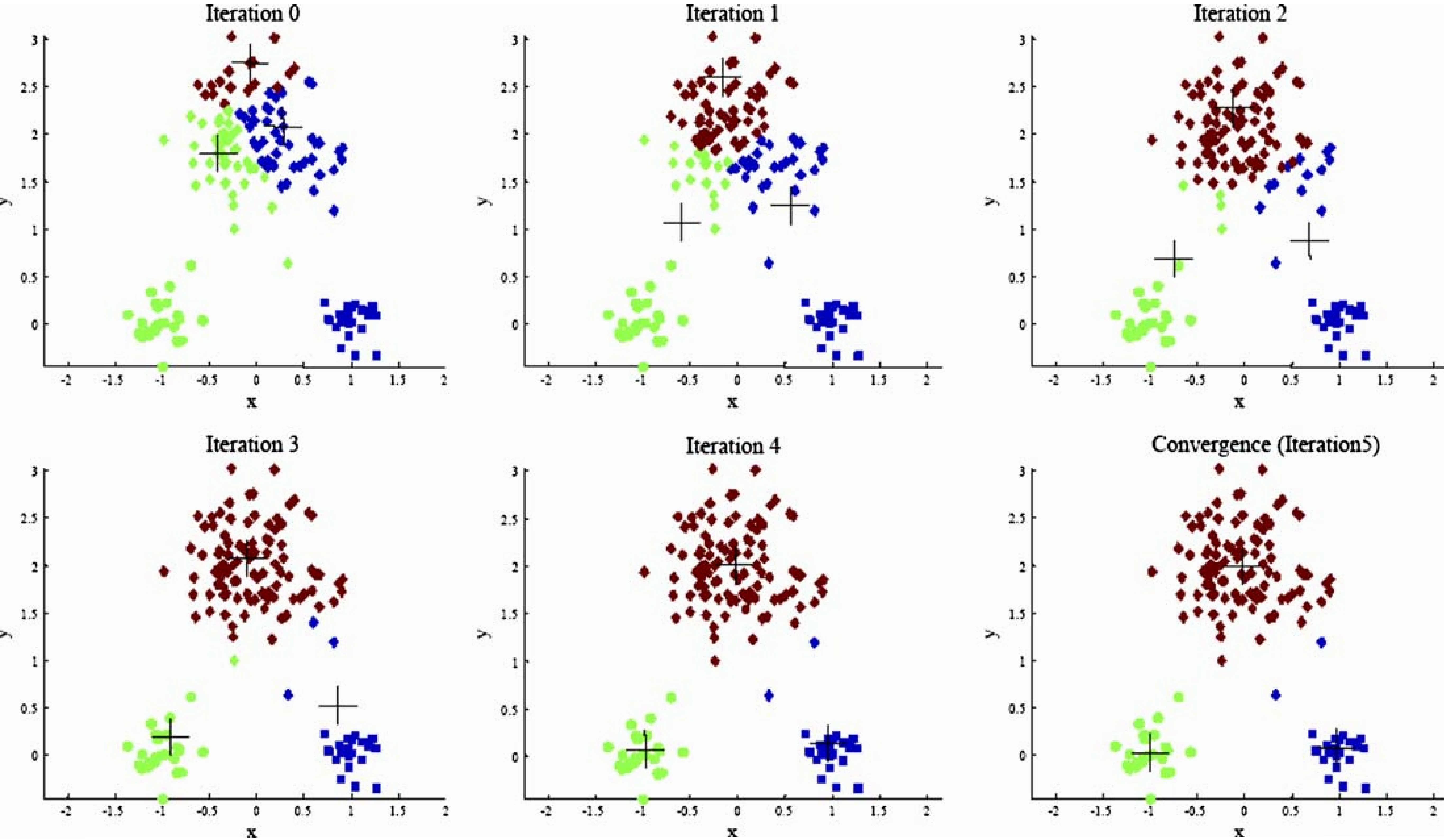
one way of solving the k-means problem



# 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

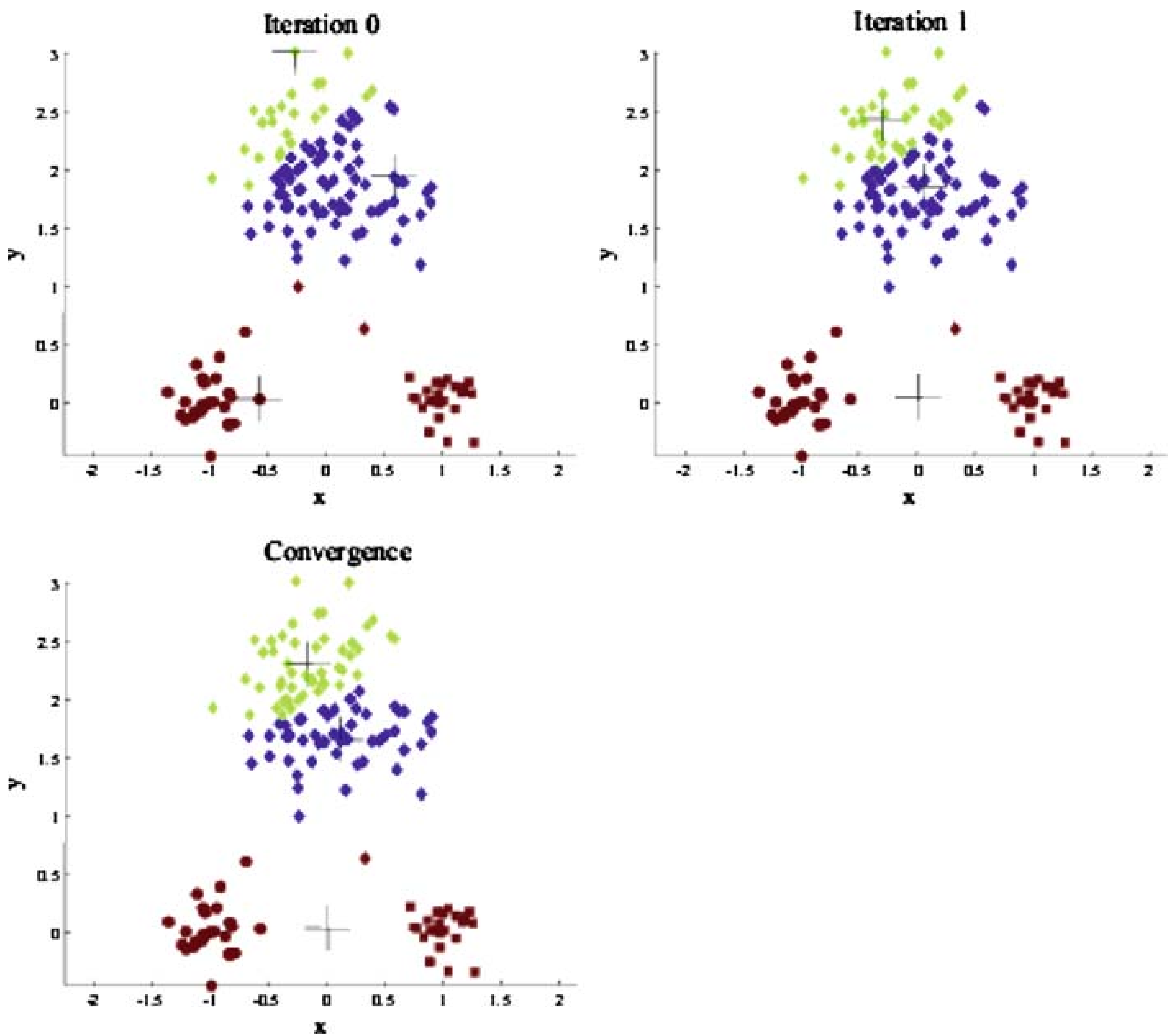
finds a local optimum

often converges quickly

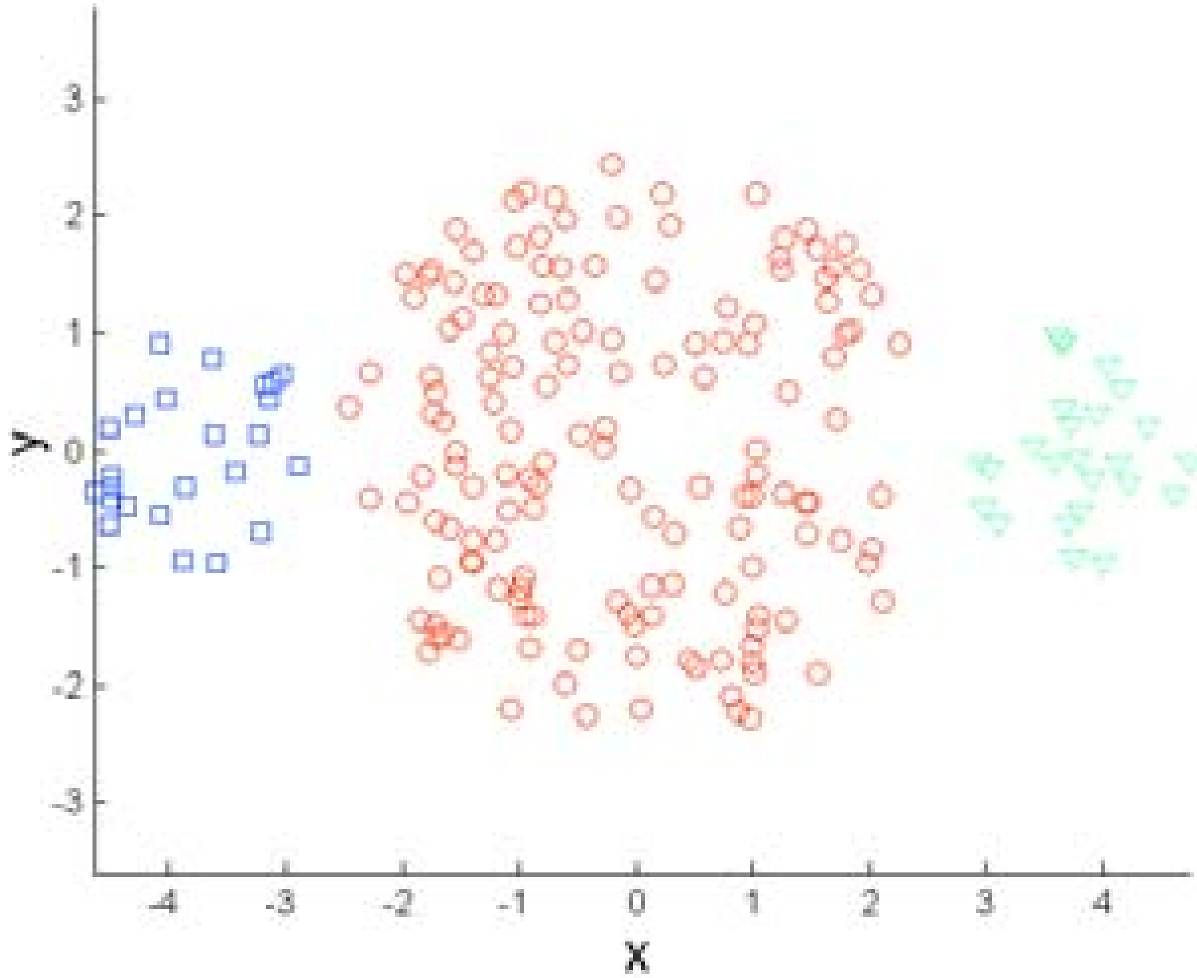
but not always

the choice of initial points can have large influence in the result

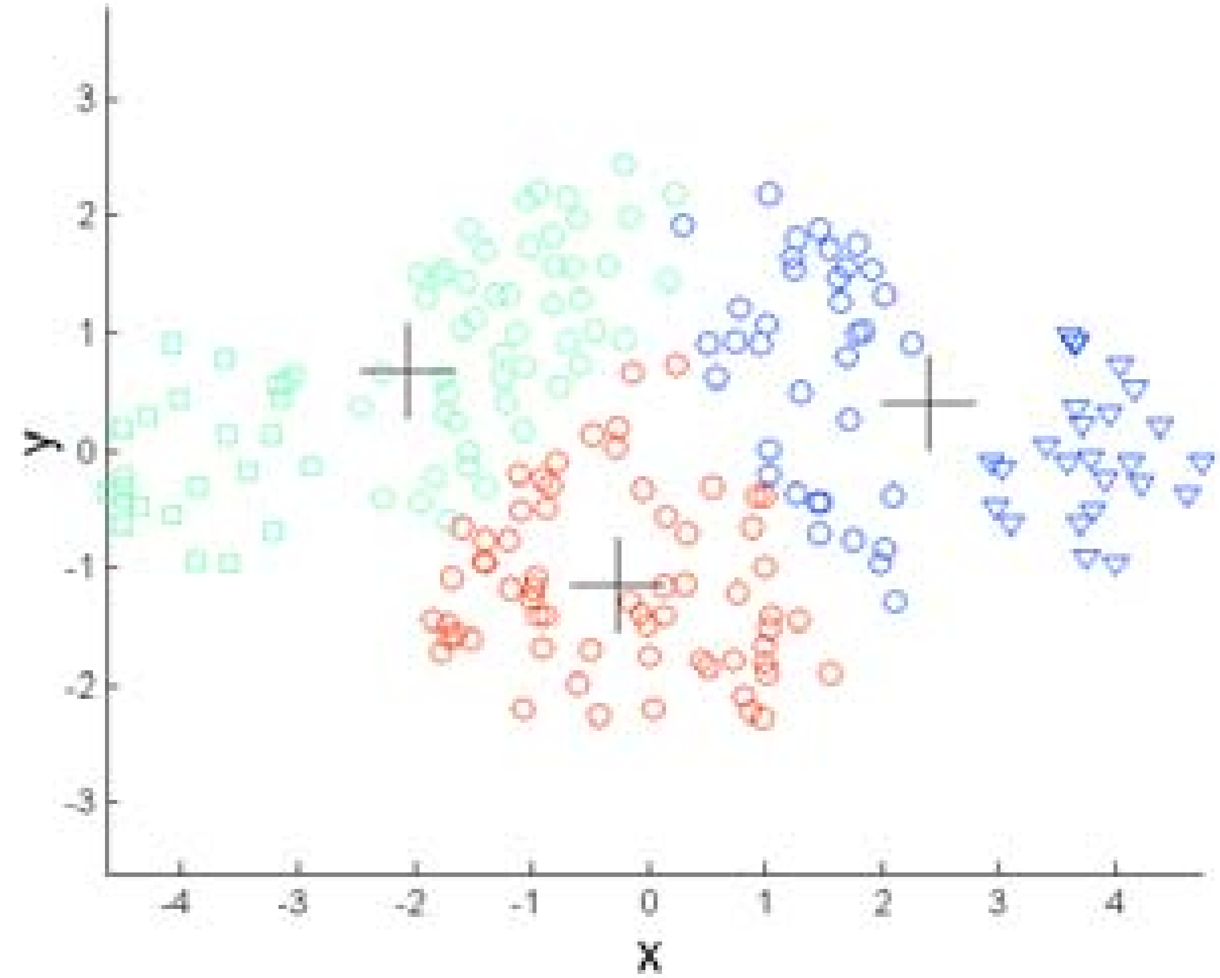
# effects of bad initialization



# limitations of k-means: different sizes

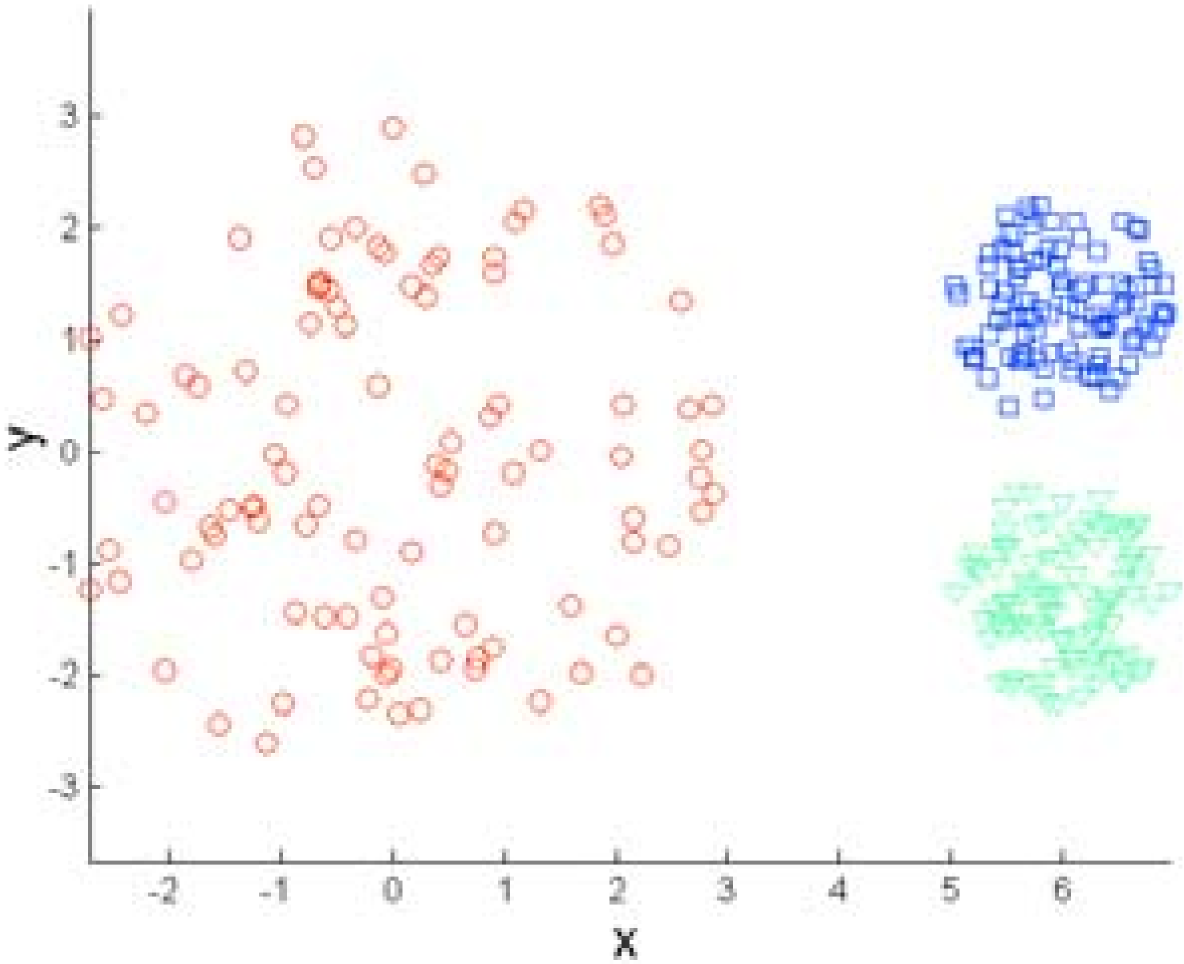


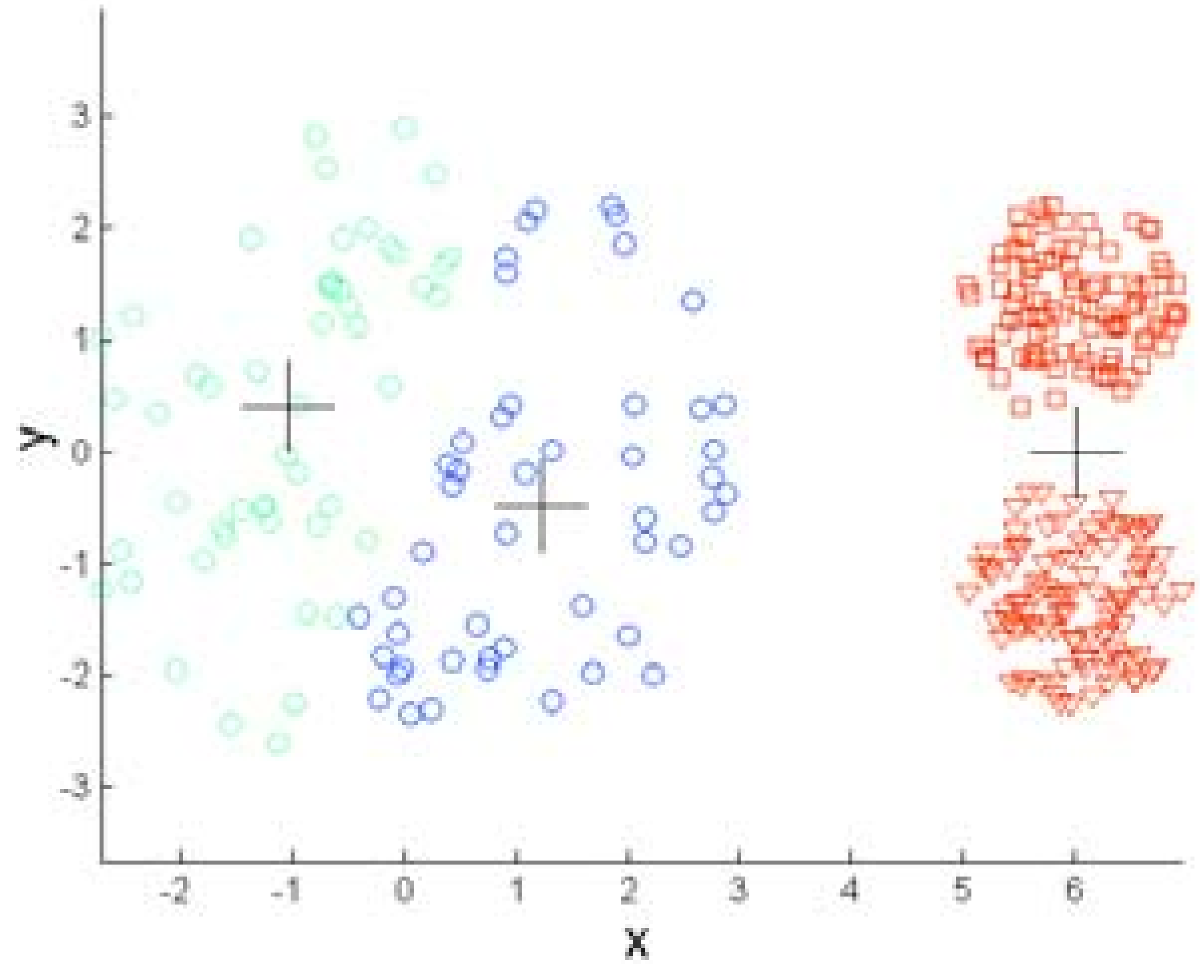

**Original Points**




**K-means (3 Clusters)**

# limitations of k-means: different density



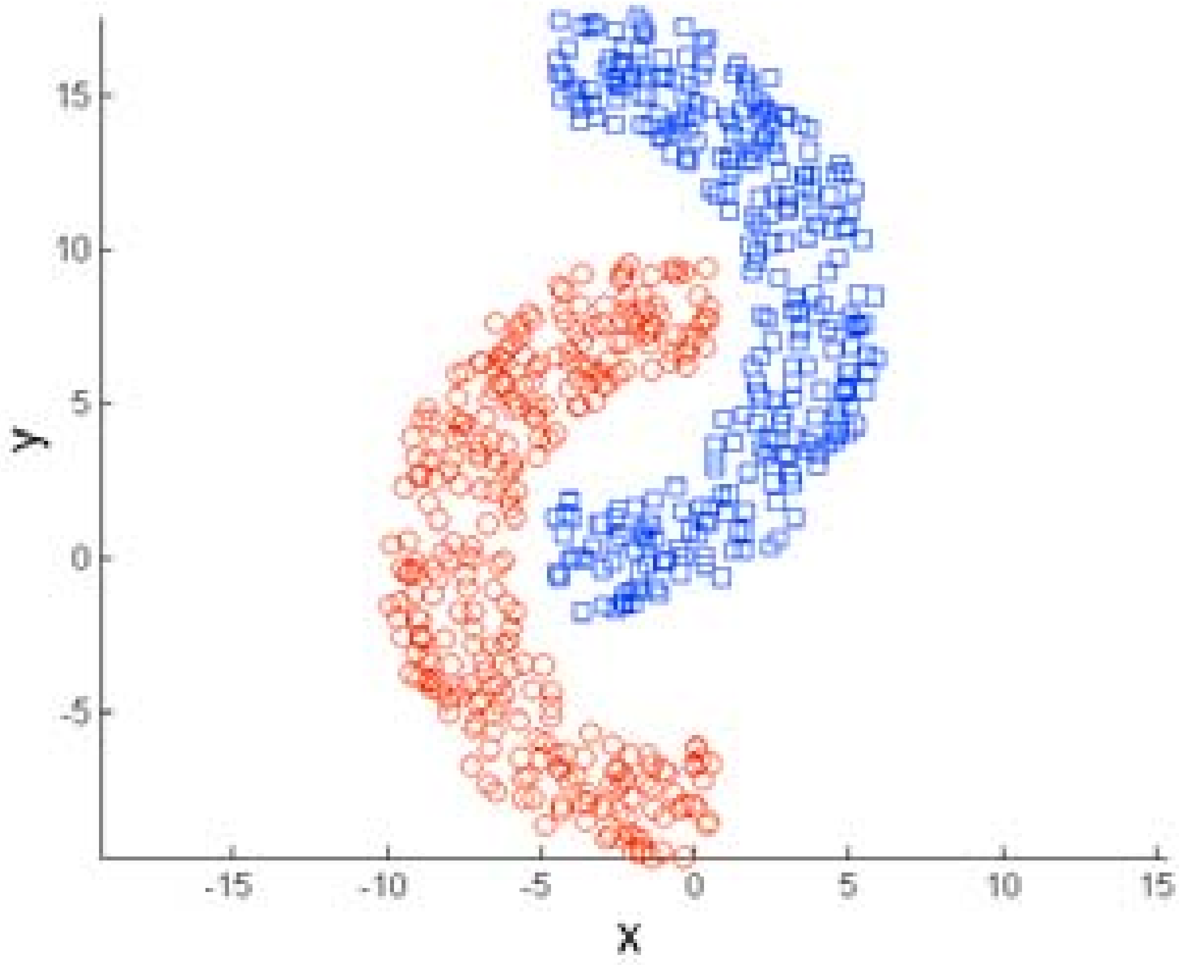

**Original Points**

**K-means (3 Clusters)**

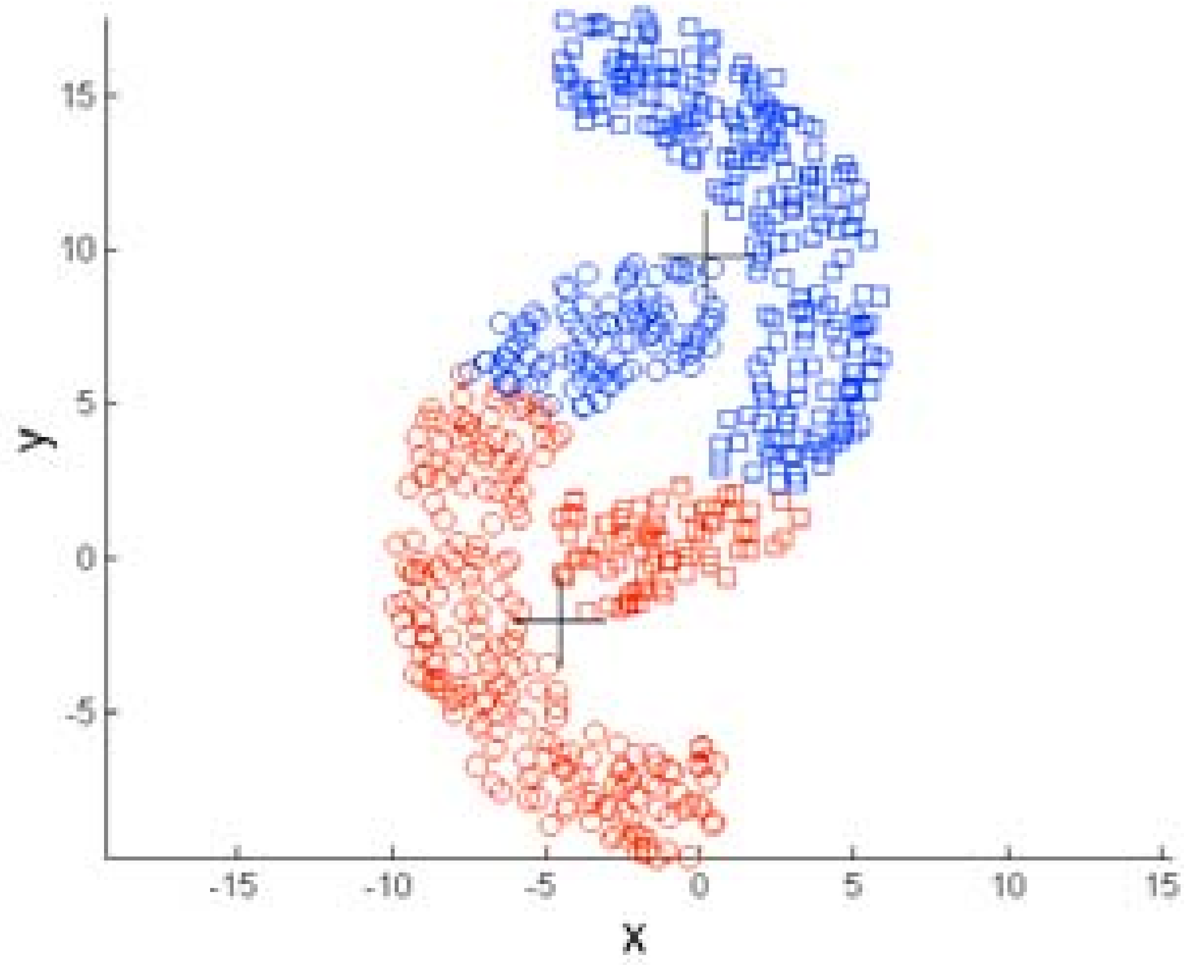


Aalto University

# 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

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

# generalizations, variants

can we generalize to non Euclidean points ?

yes, as long as we can compute means of clusters...

other problem formulations obtained by modifying the objective function

# variant : 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



Aalto University

# the k-median problem

what about the 1-median problem for Euclidean points?

also known as Fermat's problem

solution to the 1-median problem (Torricelli point) can be approximated to a given precision by an iterative algorithm

the general k-median problem is NP-hard

there exist polynomial time approximation algorithms, assuming that the underlying distance is a metric

# 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



Aalto University

# discussion on the k-medoids algorithm

k-medoids is a practical algorithm (heuristic) for solving the k-median problem

no approximation guarantee

very similar to the k-means algorithm

same advantages and disadvantages

how about efficiency?

it depends on how efficiently we can solve the 1-median problem for each cluster



Aalto University

# yet another variant : the k-center problem

consider set $X=\{x_1,\dots,x_n\}$ of $n$ points in $R^d$

assume that the number $k$ is given

problem:

find $k$ points $c_1,\dots,c_k$ (named center)

and partition $X$ into $\{X_1,\dots,X_k\}$ by assigning each point $x_i$ in $X$ to its nearest cluster center,

so that the cost

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

is minimized



Aalto University

# 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



Aalto University

# parenthesis… approximation algorithms

problem $P$:

given input $I$ find solution $S^*$ such that $P(I,S^*)$ is optimized (say, minimized)

assume finding $S^*$ is NP-hard

approximation algorithm $A$:

given input $I$ of size $n$, finds $A(I)$ in polynomial time, s.t.

$$P(I,A(I)) \leq f(n)\, P(I,S^*)$$

for all inputs $I$



Aalto University

# approximation algorithms

given input I of size n, find A(I) s.t.

$$P(I,A(I)) \leq f(n)\ P(I,S^*)$$

P(I,S*): value of the objective function for solution S on input I

f(n) : approximation factor (or approximation guarantee)

- approximation scheme (arbitrarily close to 1) $1+\varepsilon$
- constant (independent of n) 1.5, 2, 3, ...
- logarithmic, $f(n)=\log n$, $\log^2 n$,...
- other, $f(n)=\mathrm{sqrt}(n)$,…



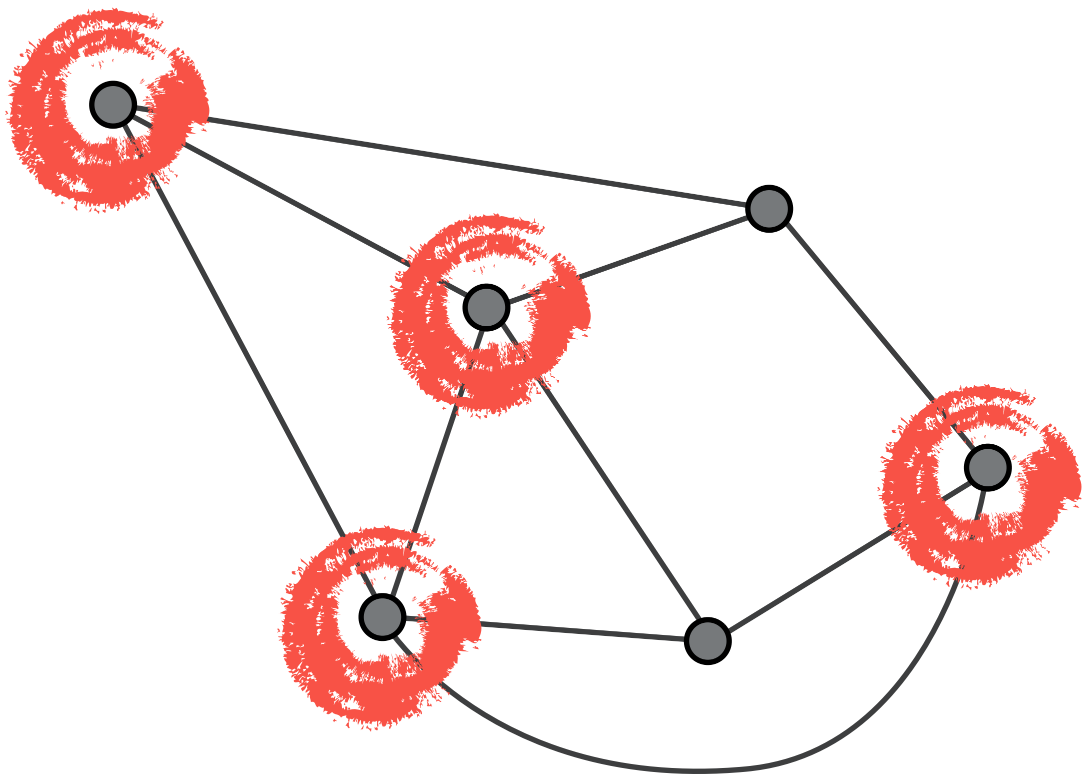
Aalto University

# example

vertex cover problem

given a graph

find the smallest set of vertices that cover all the edges



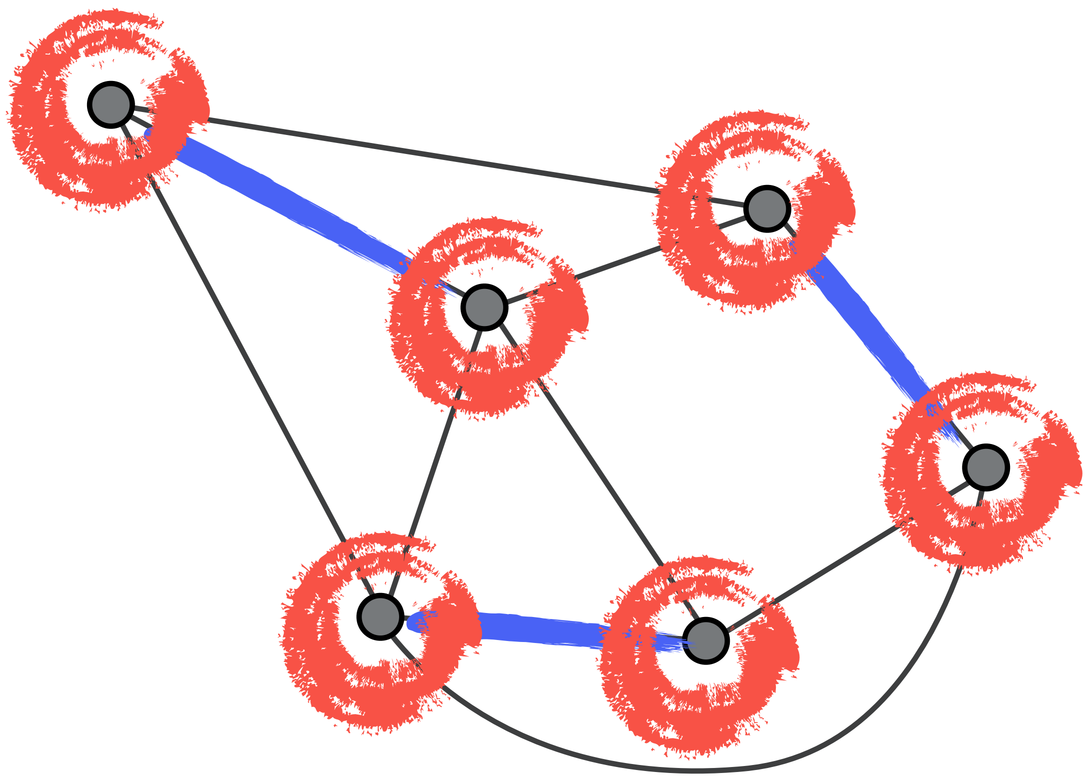


Aalto University

# vertex cover algorithm

find a maximal matching

take all the vertices of the matching



2-approximation !

what about greedy?



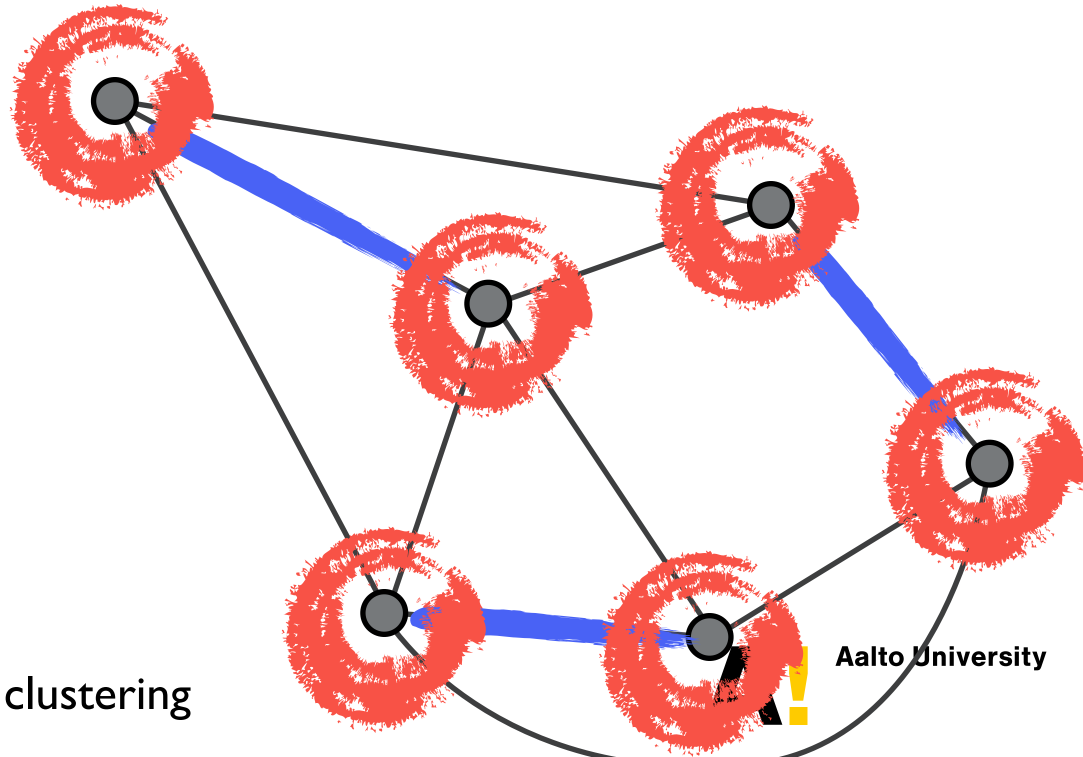
Aalto University

# analysis

optimal has to cover the edges of the matching

(optimal) $\geq$ (matching size) = (1/2)(solution of algo)

(solution of algo) $\leq$ 2 (optimal)





Aalto University

# ...parenthesis

# recall : 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 center)

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

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

is minimized



Aalto University

# 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 $d(x,S) = \min_{y \in S} d(x,y)$

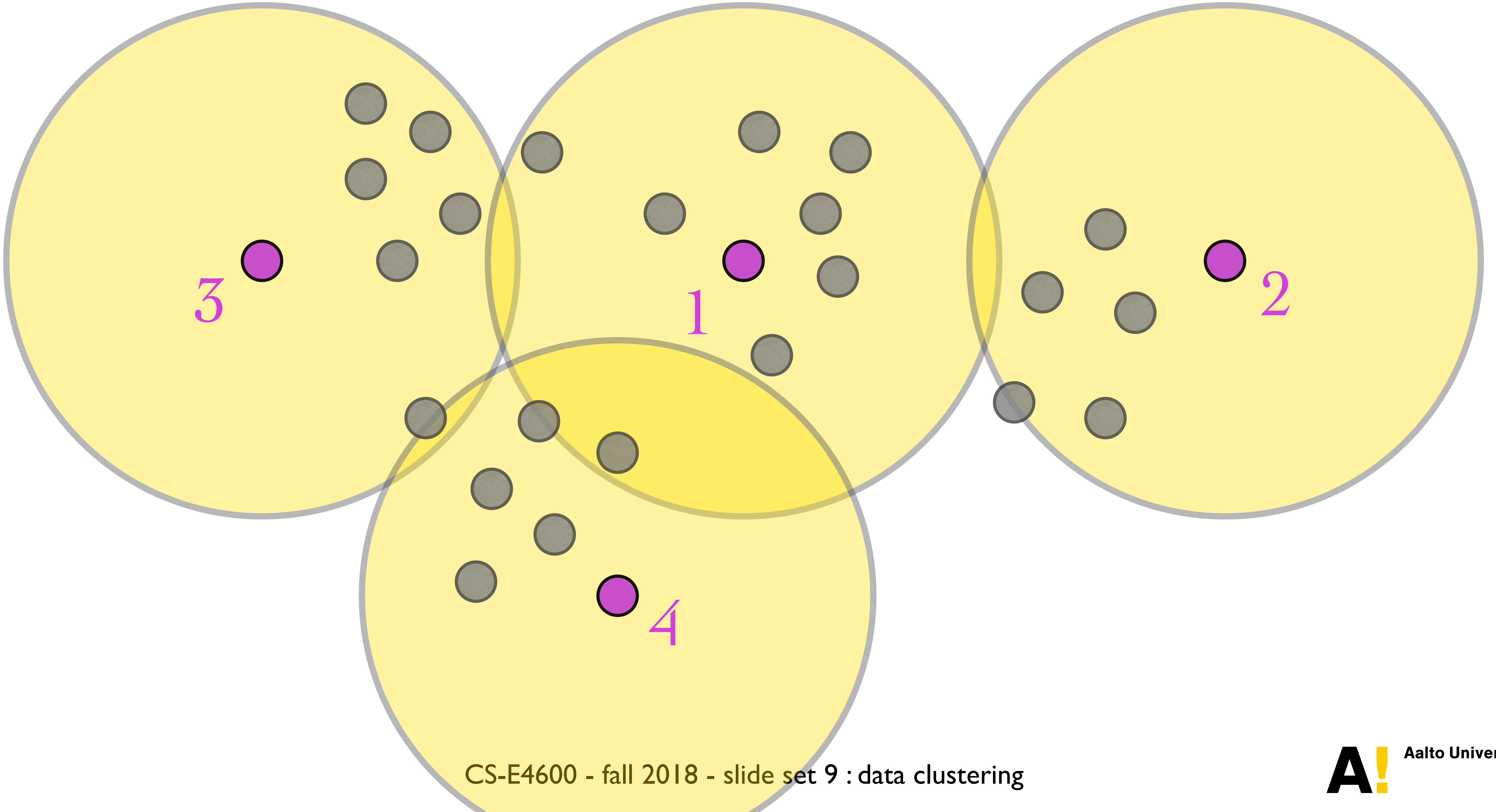
label that point $i$

assign the remaining unlabeled data points to the closest labeled data point



Aalto University

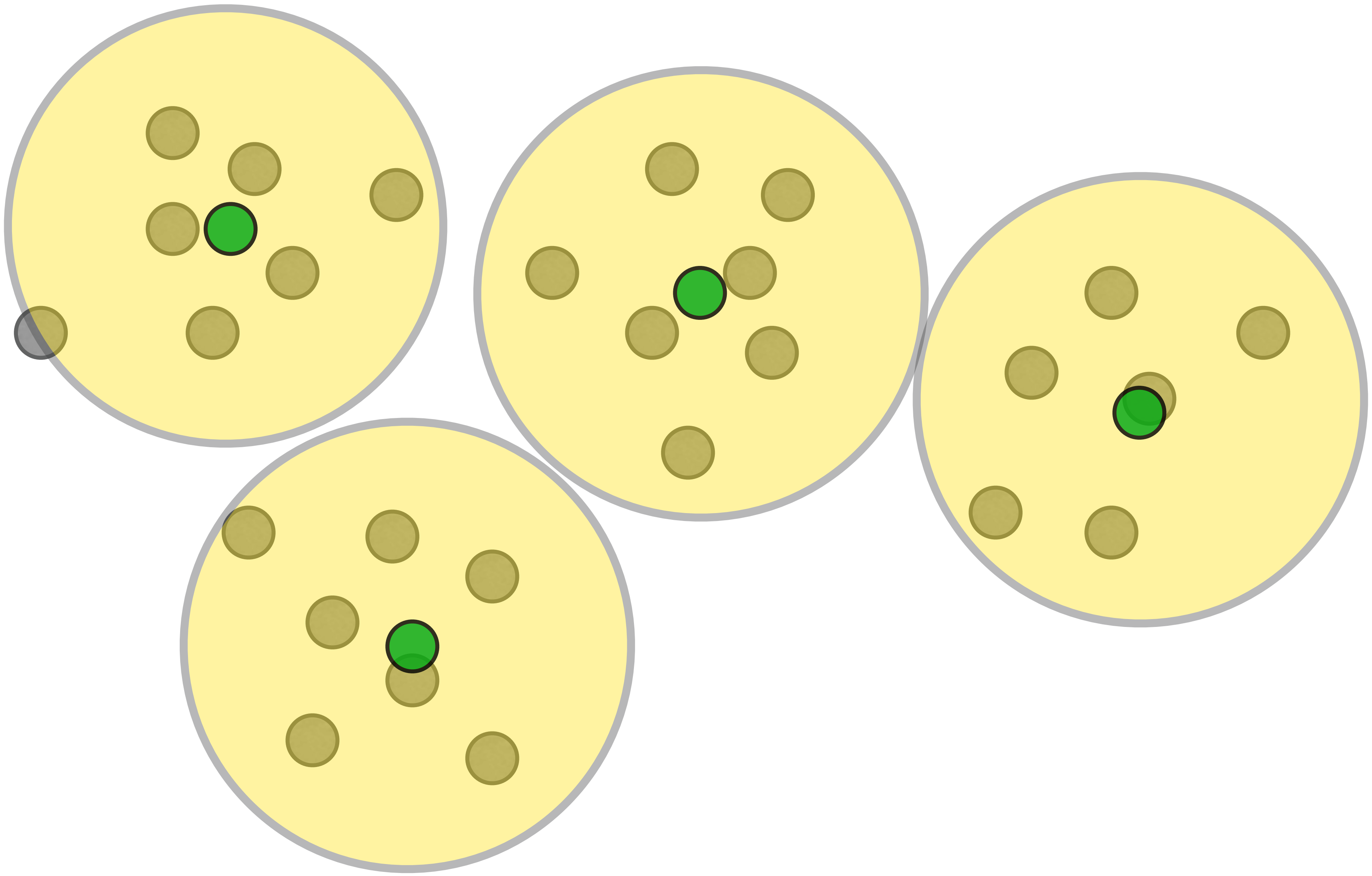
# furthest-first traversal algorithm: example






Aalto University

# furthest-first traversal algorithm: example





Aalto University

# 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: $R_2 \geq R_3 \geq \ldots \geq R_k$

proof :

consider indices $i$ and $j$ with $j > i$

$R_j = d(j, p(j))$

$= d(j, \{1, 2, \ldots\ldots\ldots\ldots, j-1\})$

$= d(j, \{1, 2, \ldots, i-1, \ldots, j-1\})$

$\leq d(j, \{1, 2, \ldots, i-1\})$ // $j > i$

$\leq d(i, \{1, 2, \ldots, i-1\})$ // $j$ was present when $i$ was selected

$= R_i$

# 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 :

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



Aalto University

# analysis

theorem

let $C$ be the clustering produced by the FFT algorithm

let $C^*$ be the optimal clustering

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

proof:

let $C^*_1, \ldots, C^*_k$ be the clusters of the optimal $k$-clustering

if these clusters contain points $\{1,\ldots,k\}$ then

$$R(C) \leq 2R(C^*) \quad ✪$$

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

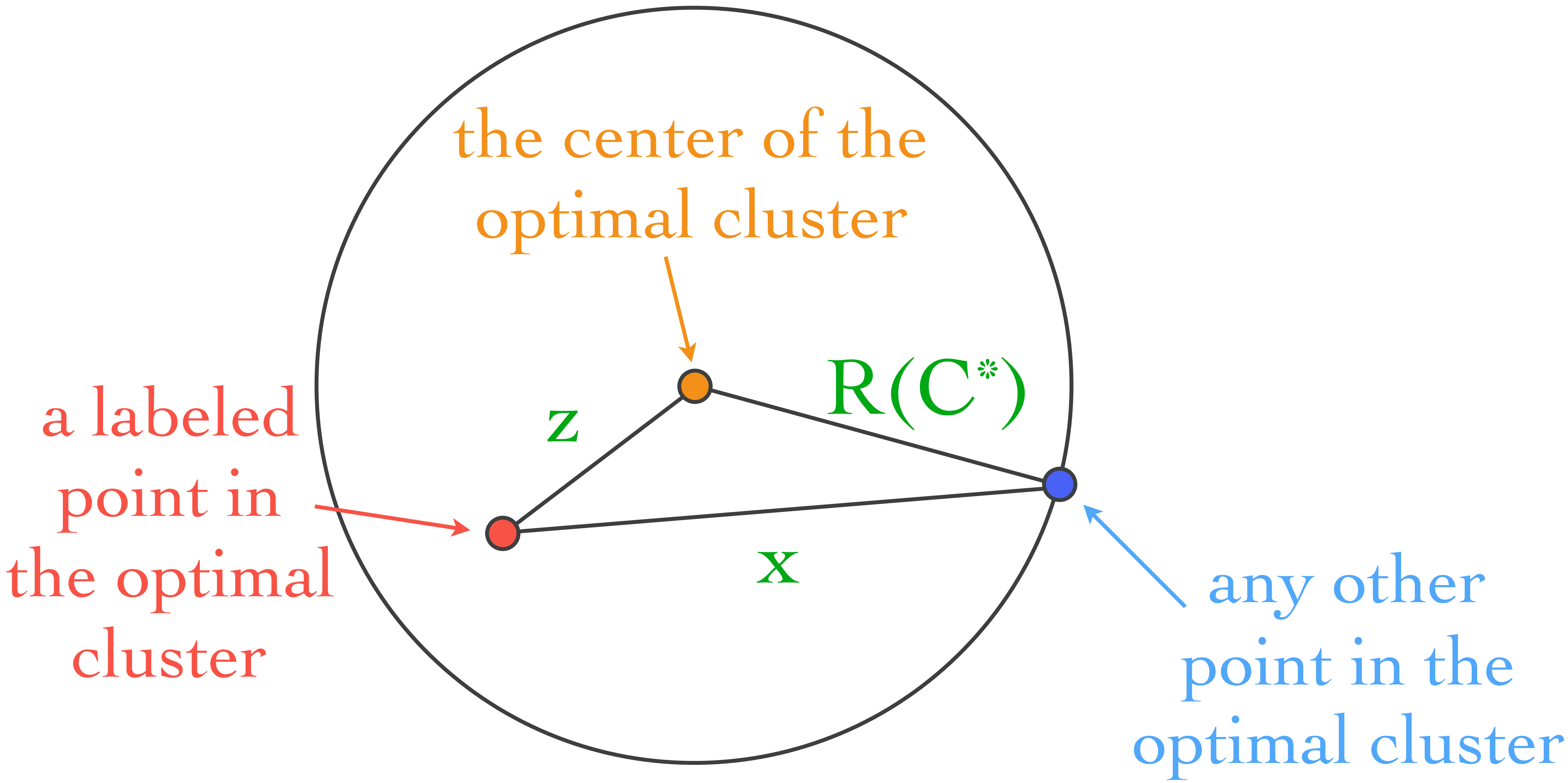
this (optimal) cluster must have radius at least

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



Aalto University

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




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

# k-means++

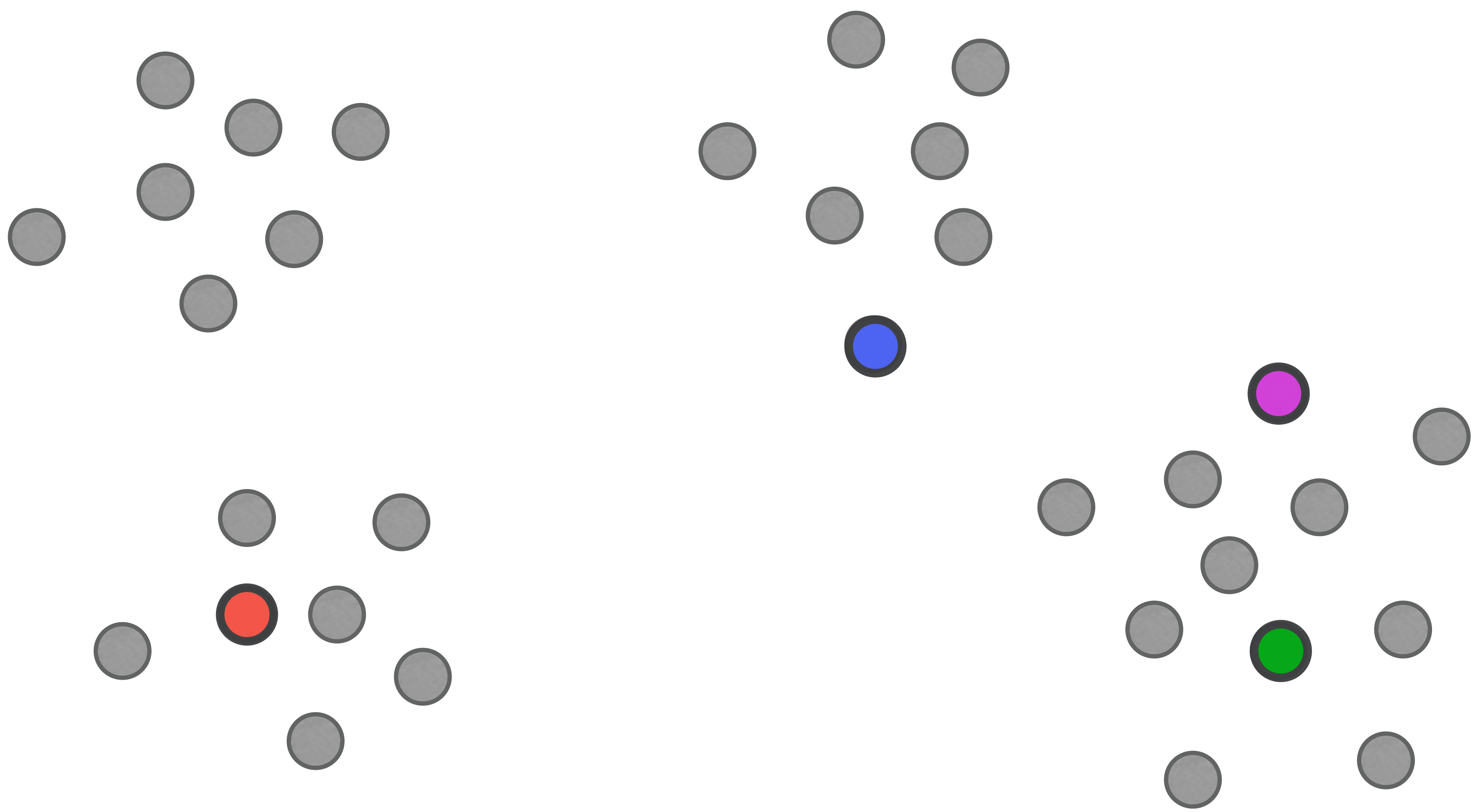
# optional reading assignment

David Arthur and Sergei Vassilvitskii

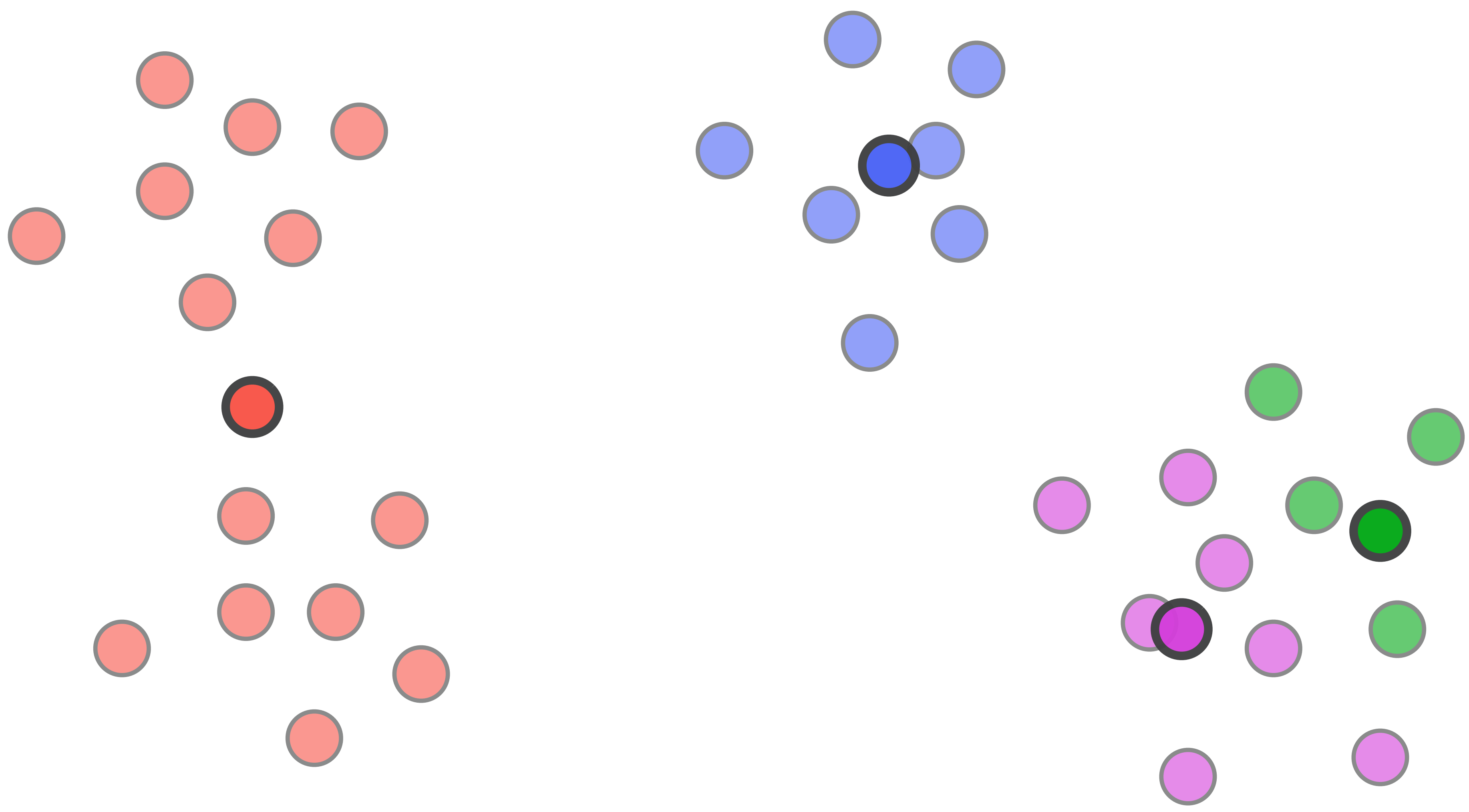
k-means++: The advantages of careful seeding

SODA 2007

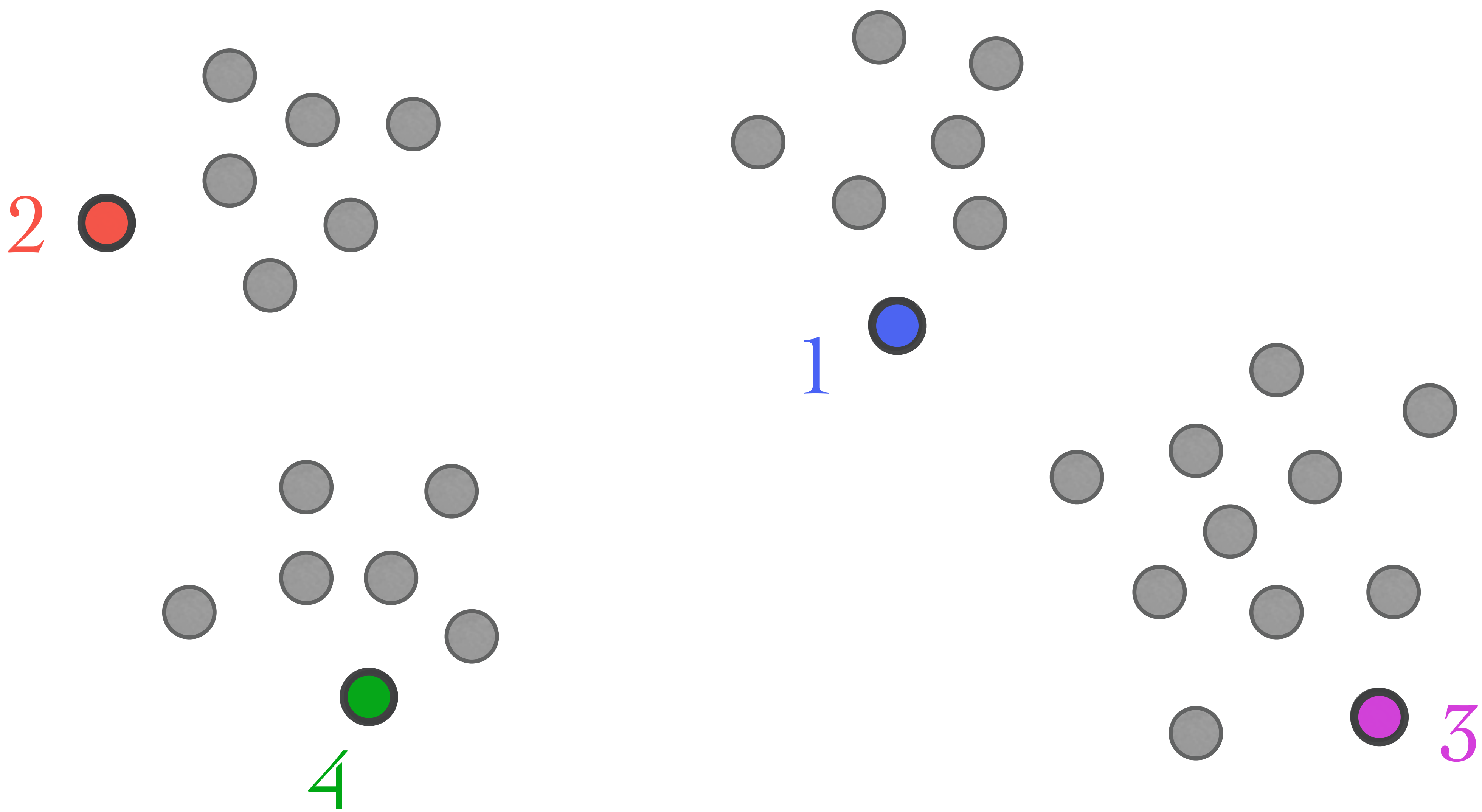
# k-means algorithm: random initialization



# k-means algorithm: random initialization



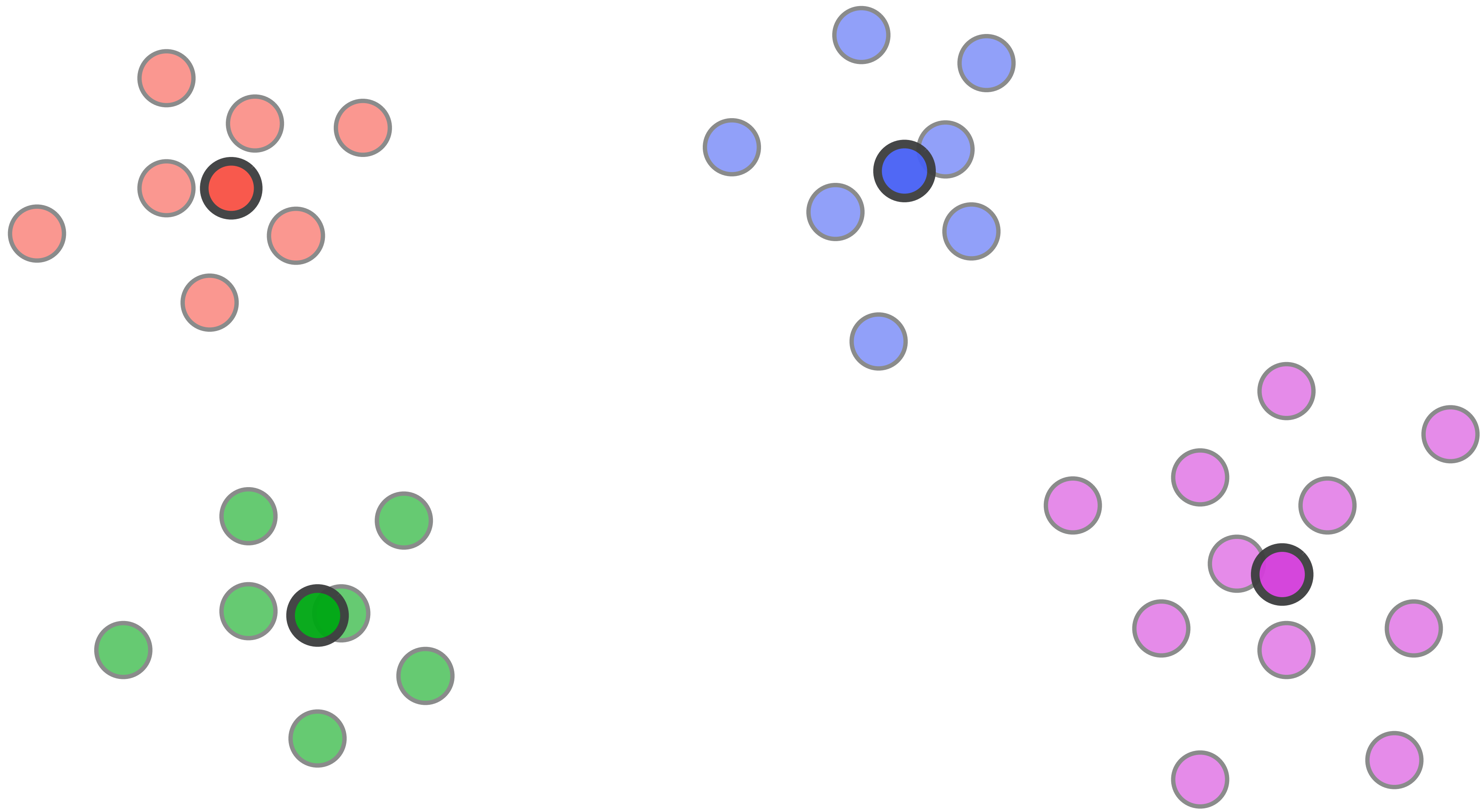
# k-means algorithm: initialization with further-first traversal



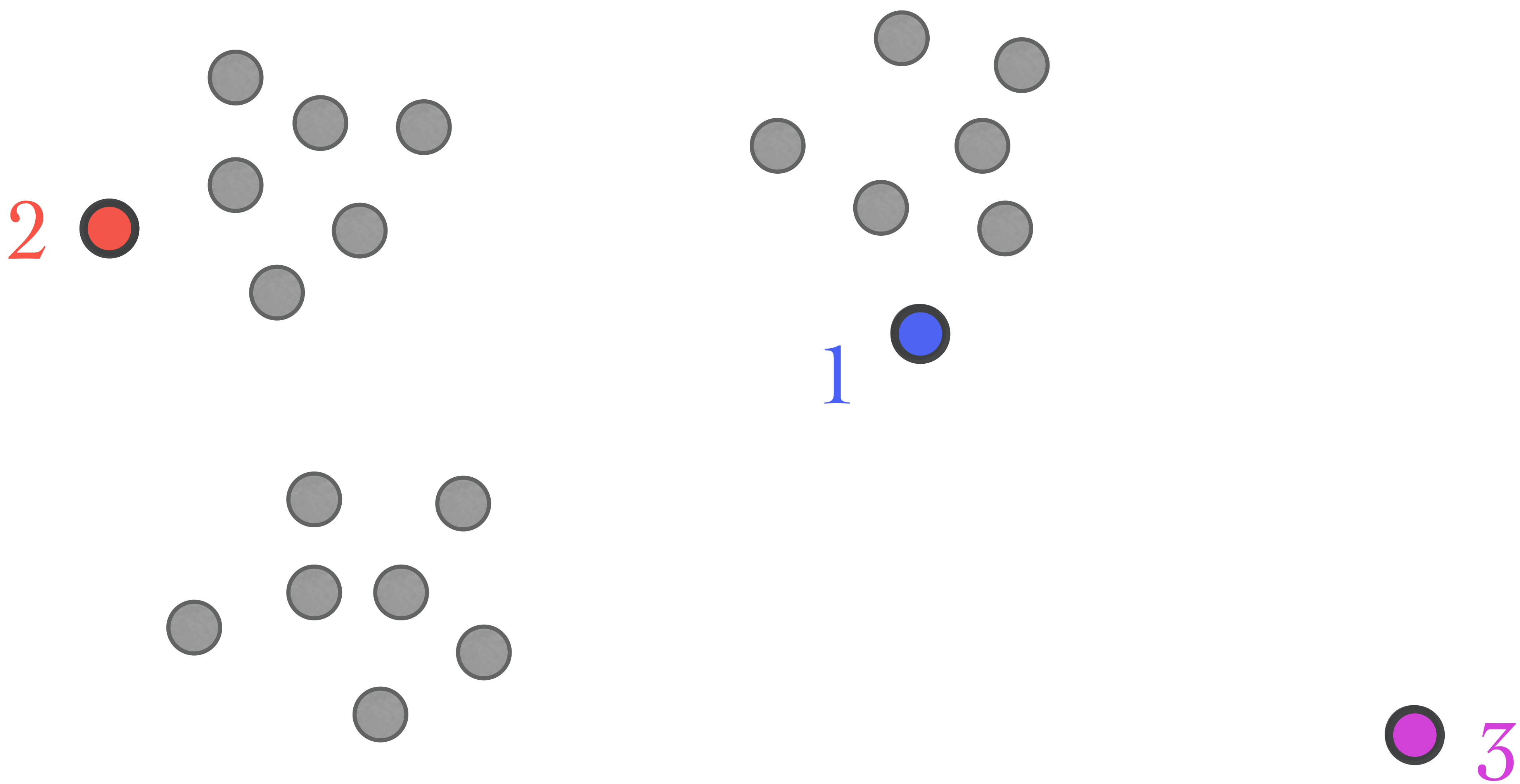



Aalto University

# k-means algorithm: initialization with further-first traversal



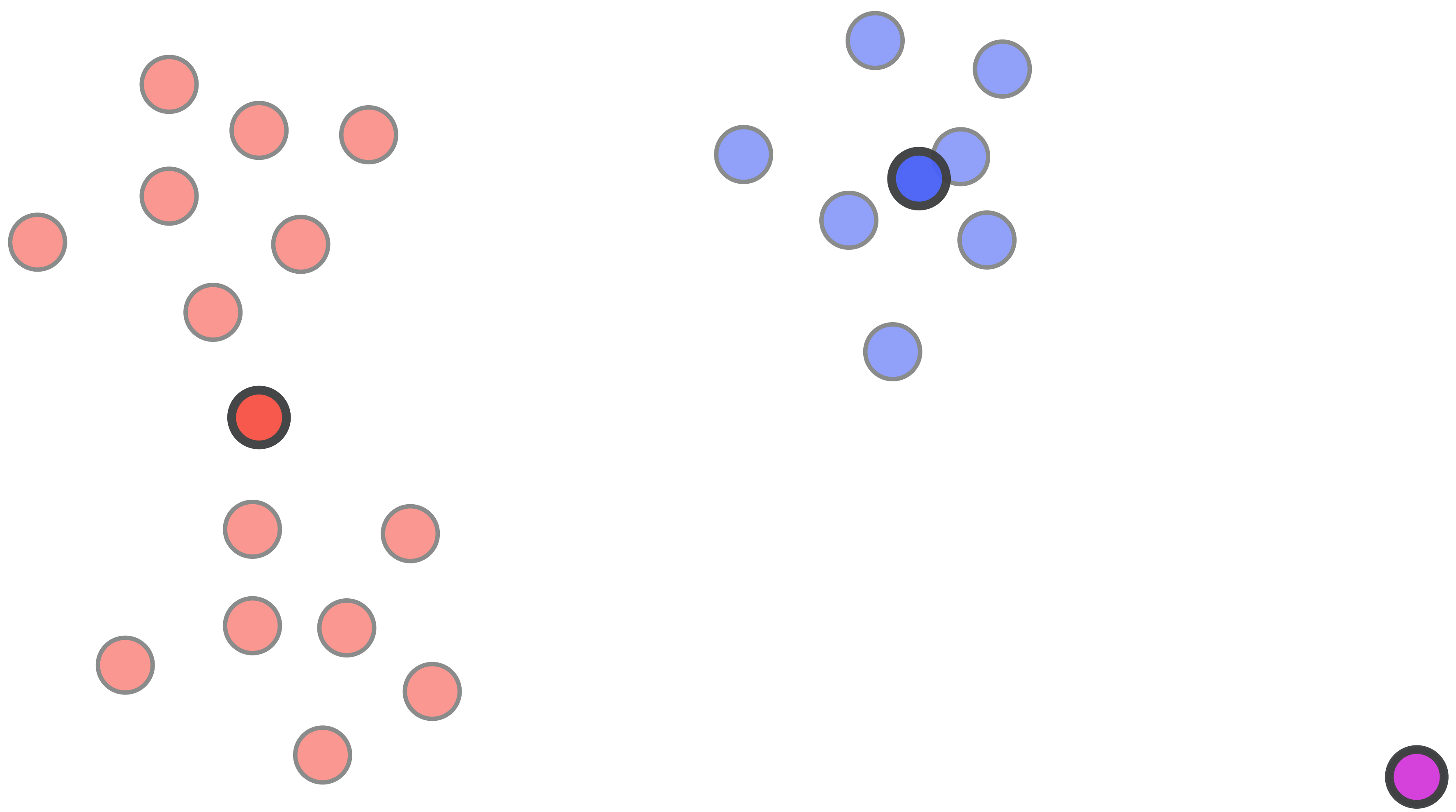
# but sensitive to outliers



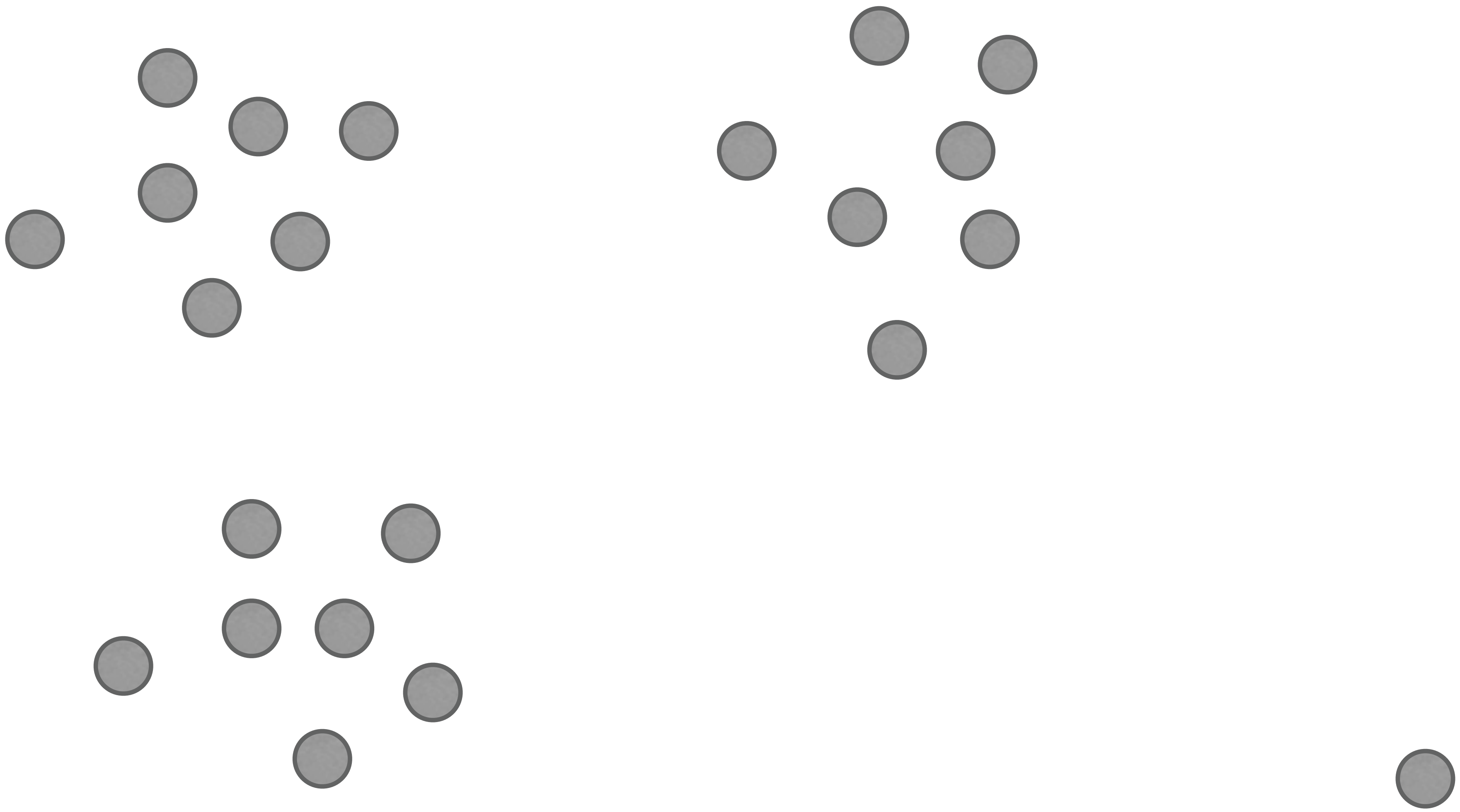



Aalto University

# but sensitive to outliers



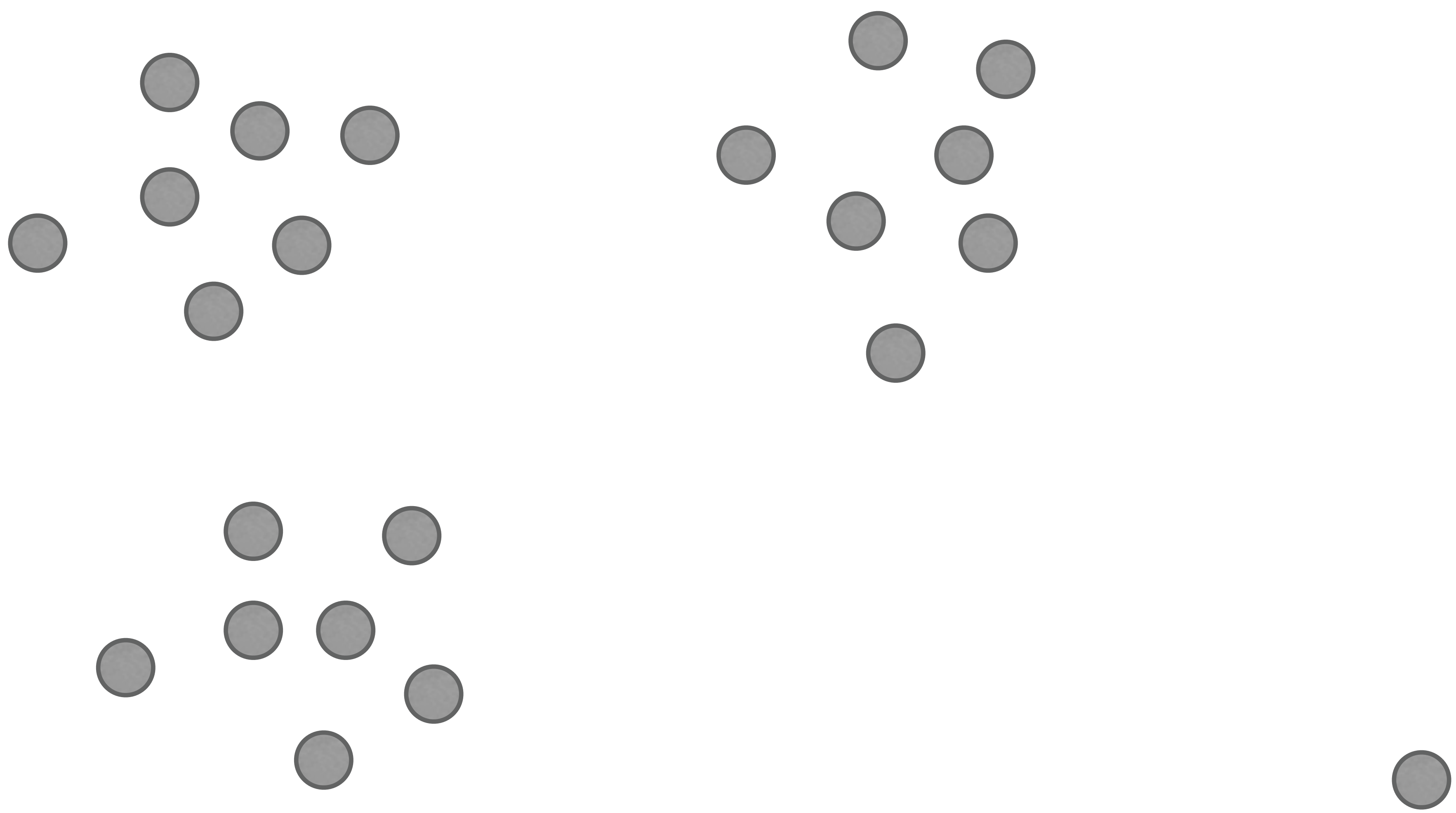
# here random may work well



we want to select seeds that that are :

1. far from existing seeds (explore a new area of the data)
2. have many near-by points (potentially discover a new cluster)

how do we accomplish both objectives?





Aalto University

# k-means++ algorithm

interpolate between the two methods (furtherst and random)

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 :

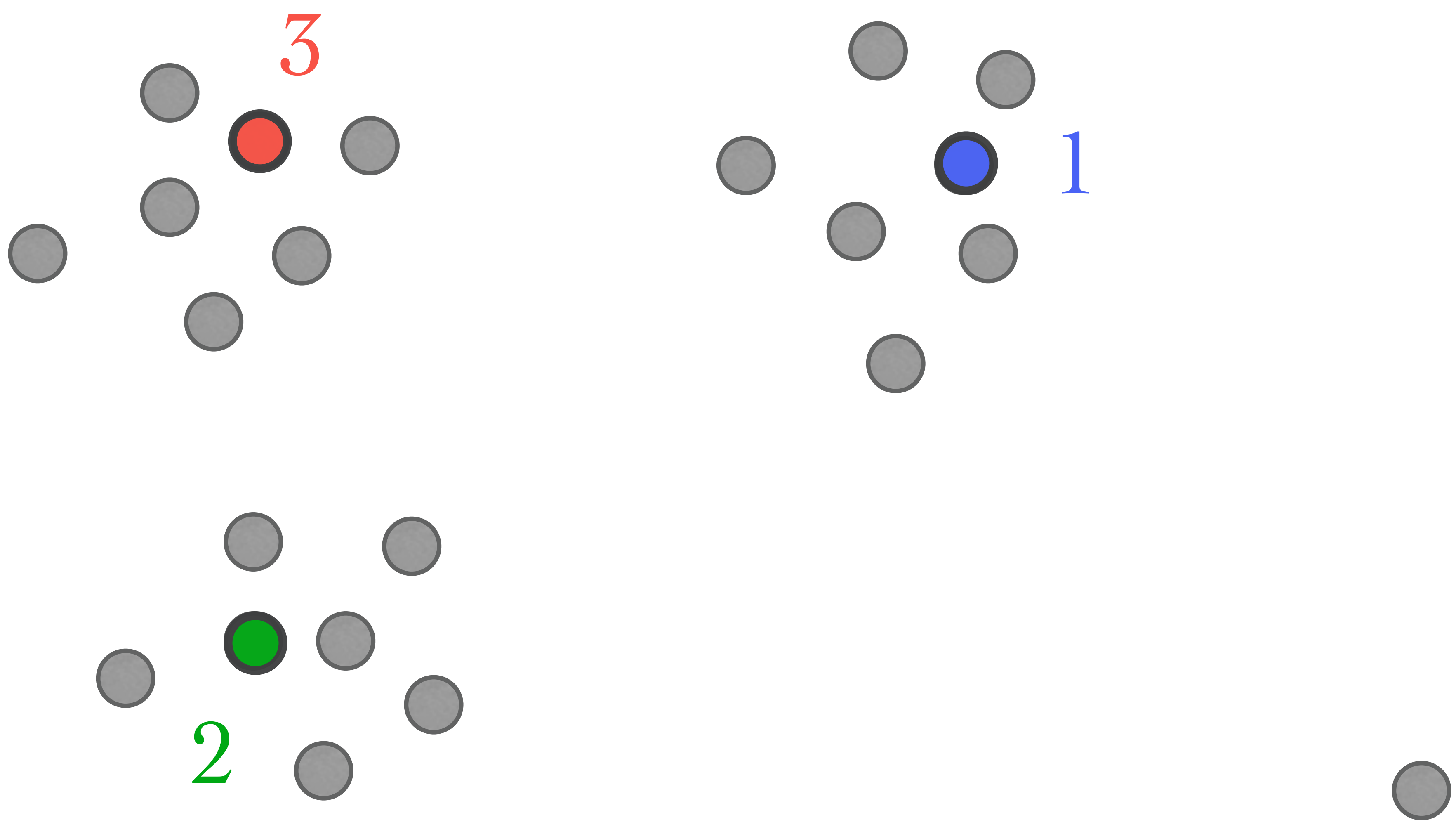
choose the first center uniformly at random

choose next center with probability proportional to $D^2(x)$

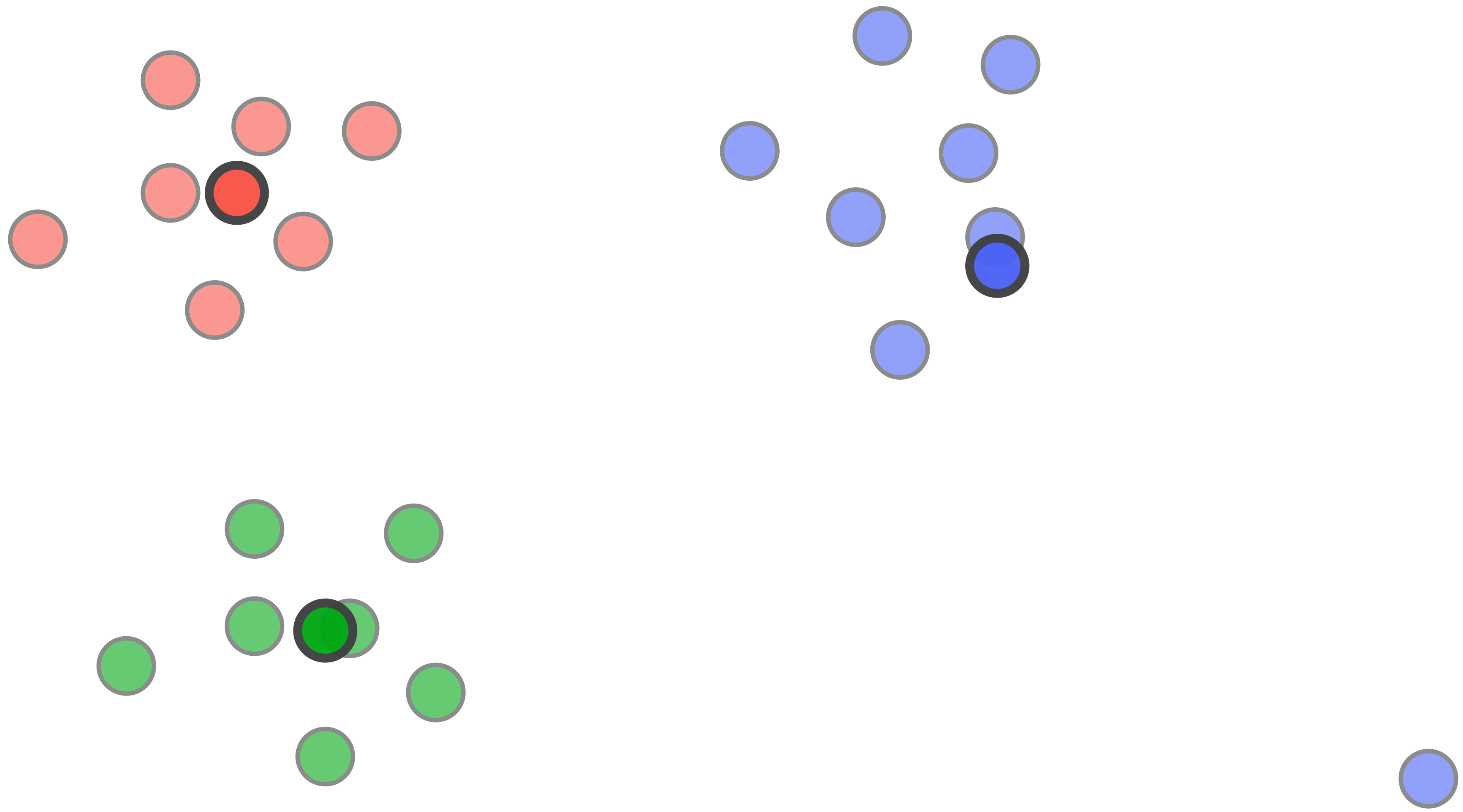
iteration phase :

iterate as in the k-means algorithm until convergence

# k-means++ initialization



# k-means++ initialization





Aalto University

# k-means++ provable guarantee

theorem:

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



Aalto University

# 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 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 $O(\log k)$ 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

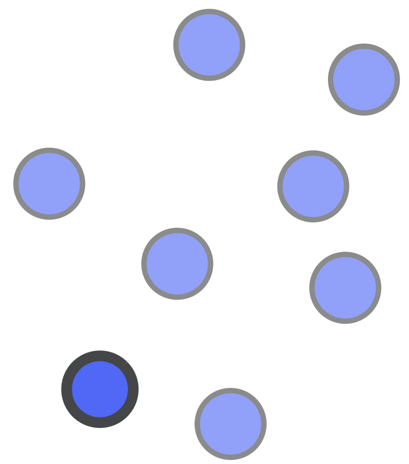


Aalto University

# 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:

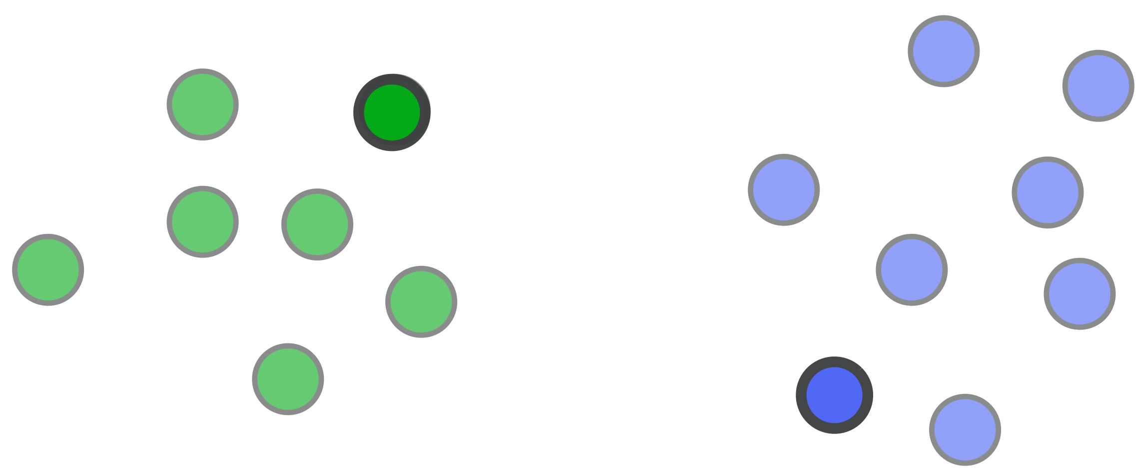
$$E[\phi(A)] = \sum_{a_0 \in A} \frac{1}{|A|} \sum_{a \in A} ||a - a_0||^2$$

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



Aalto University

# 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



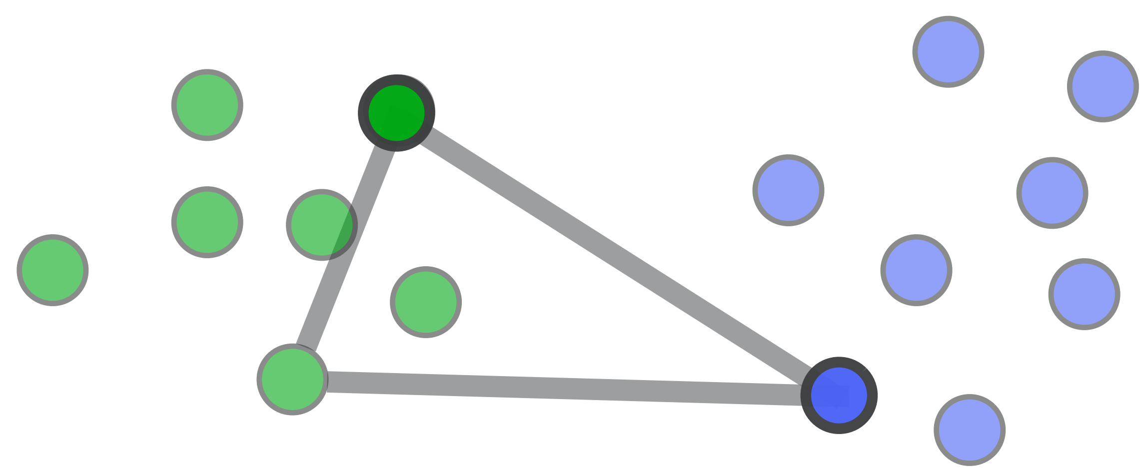
Aalto University

# 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:



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

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



Aalto University

# 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

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

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



Aalto University

# 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



Aalto University