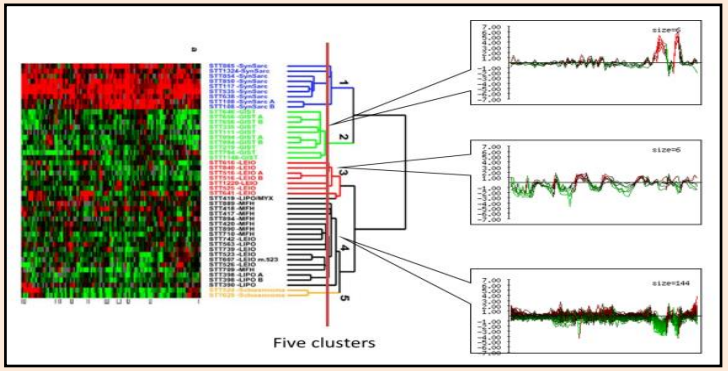
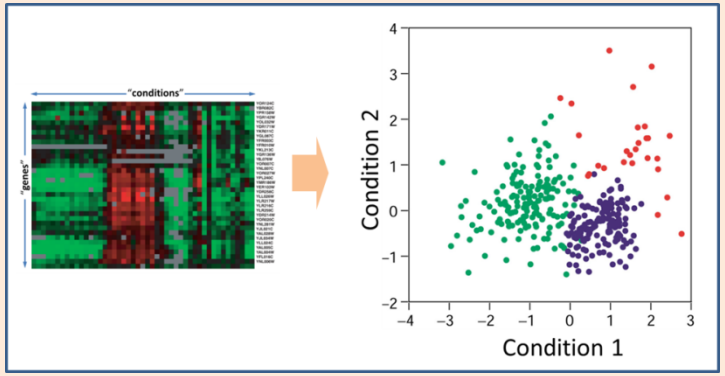
A quick review

The clustering problem:

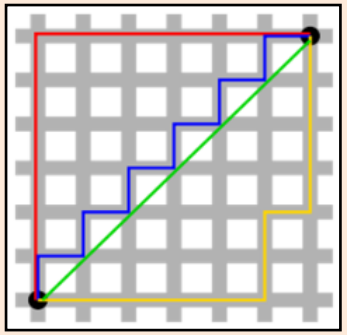
- **Different representations**



- ***homogeneity vs. separation***



- **Many possible distance metrics**
- **Many possible linkage approaches**
- Method matters; metric matters; definitions matter;

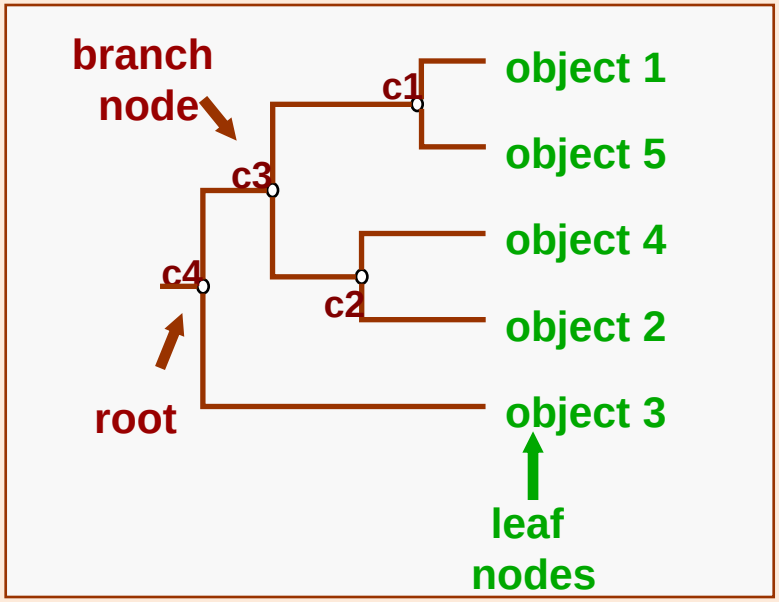


A quick review

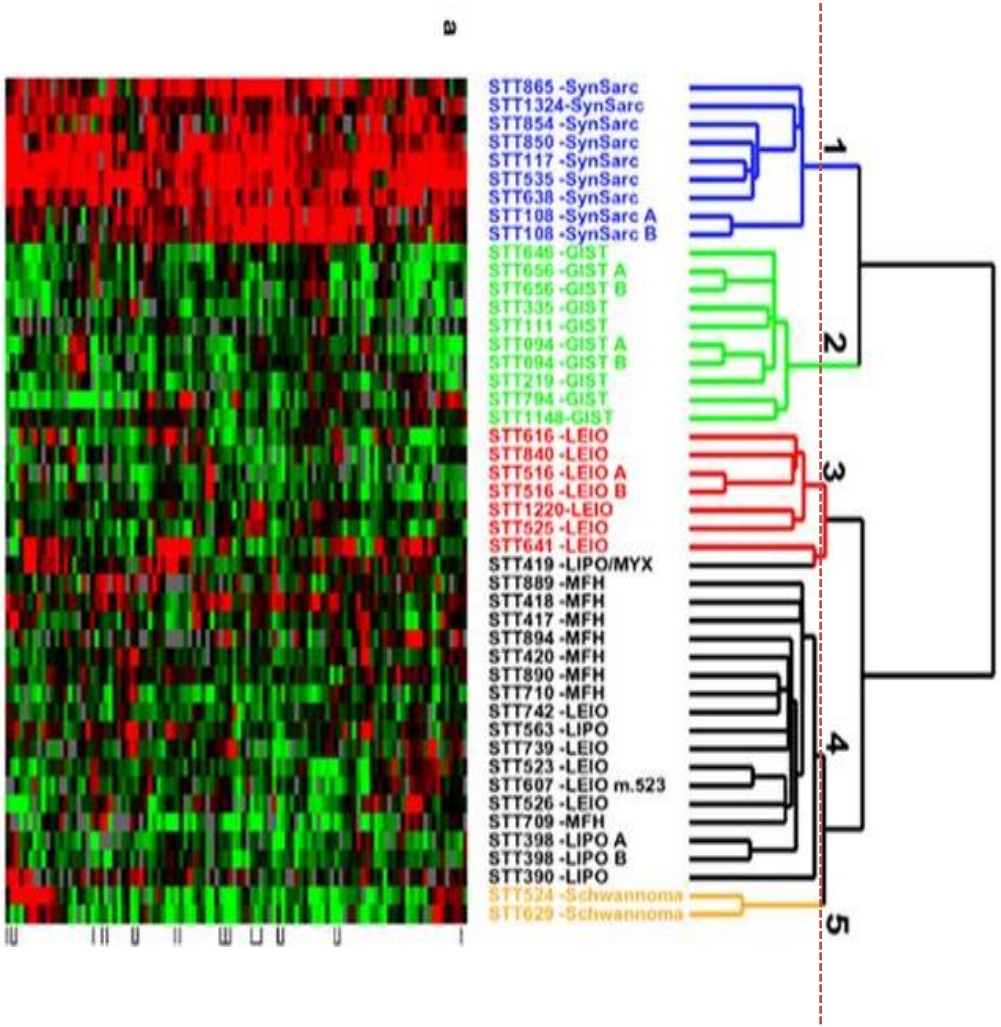
- **Hierarchical clustering:**
  - Takes as input a distance matrix
  - Progressively regroups the closest objects/groups
  - The result is a tree - intermediate nodes represent clusters
  - Branch lengths represent distances between clusters

Distance matrix

	object 1	object 2	object 3	object 4	object 5
object 1	0.00	4.00	6.00	3.50	1.00
object 2	4.00	0.00	6.00	2.00	4.50
object 3	6.00	6.00	0.00	5.50	6.50
object 4	3.50	2.00	5.50	0.00	4.00
object 5	1.00	4.50	6.50	4.00	0.00



Hierarchical clustering result



Five clusters

The “philosophy” of clustering - Summary

- “**Unsupervised learning**” problem
- **No single solution is necessarily the true/correct!**
- There is usually a **tradeoff** between homogeneity and separation:
  - More clusters → increased homogeneity but decreased separation
  - Less clusters → Increased separation but reduced homogeneity
- Method matters; metric matters; definitions matter;
- In most cases, **heuristic methods** or approximations are used.

Clustering

k-mean clustering

Genome 559: Introduction to Statistical and Computational Genomics

Elhanan Borenstein

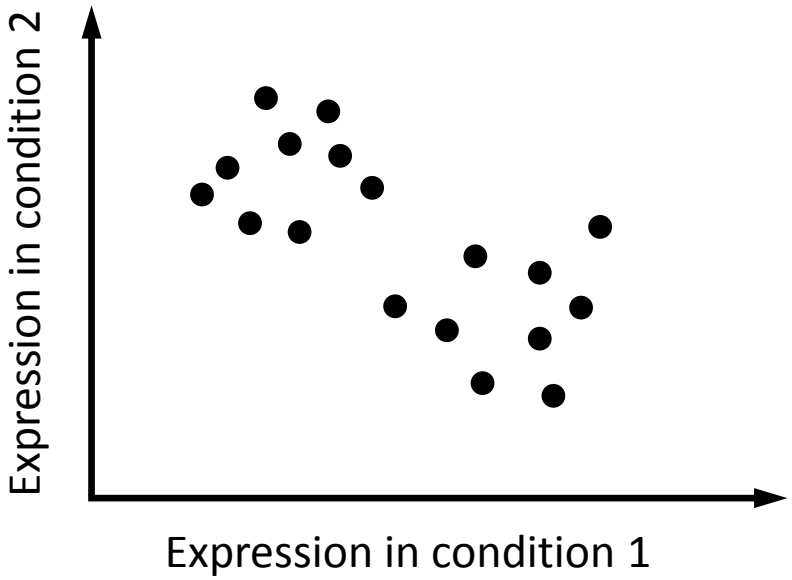
K-mean clustering: A different approach

- **Clear definition of a 'good' clustering solution** (in contrast to hierarchical clustering)
- **Divisive rather than agglomerative** (in contrast to hierarchical clustering)
- **Obtained solution is non-hierarchical** (in contrast to hierarchical clustering)
- **A new algorithmic approach** (unlike any algorithm we learned so far)

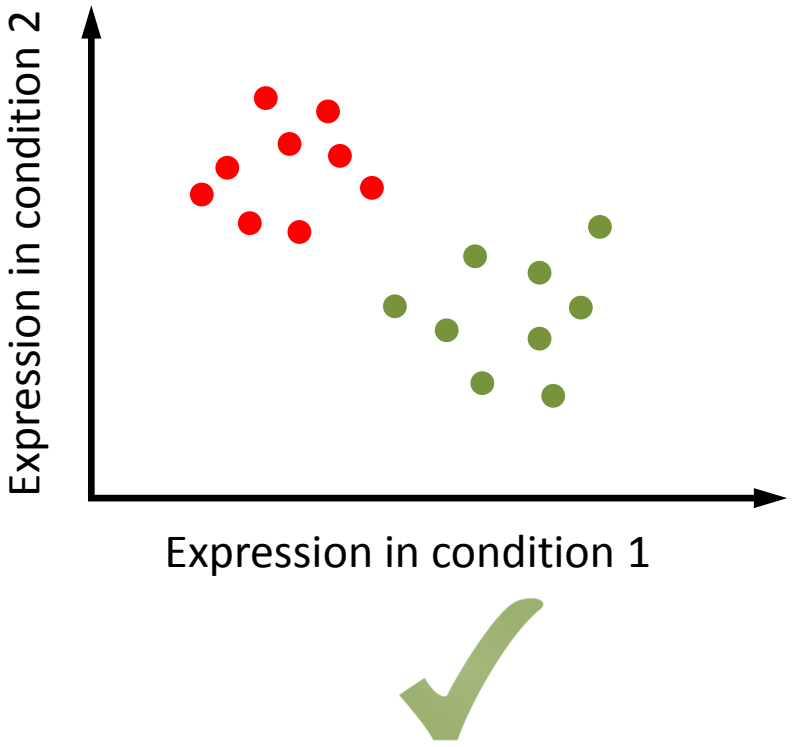
What constitutes a good clustering solution?

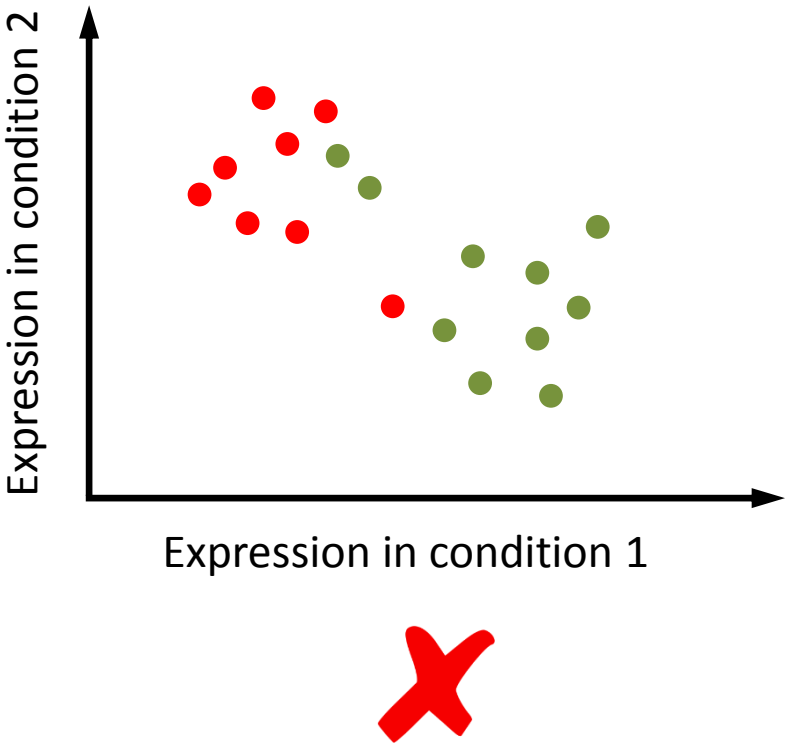
(What exactly are we trying to find?)

Defining a good clustering solution

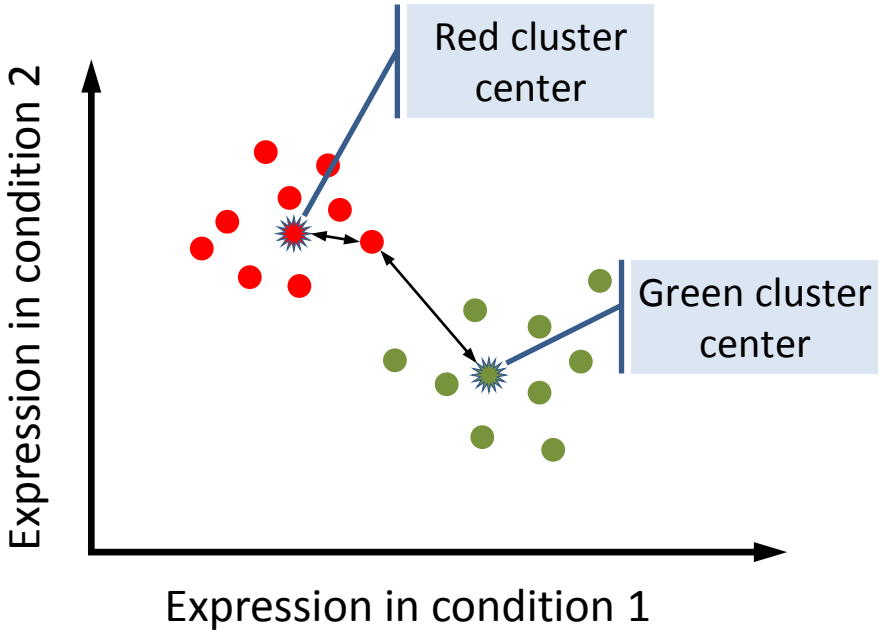


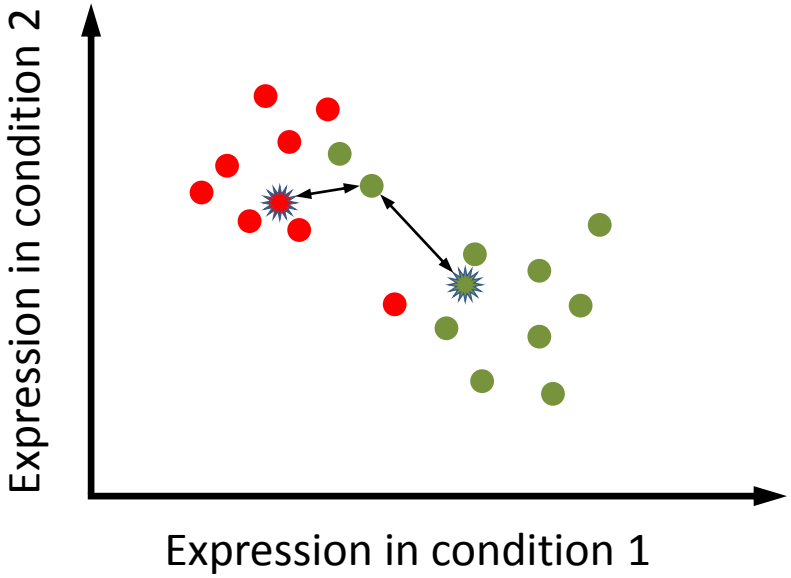
Defining a good clustering solution





Defining a good clustering solution

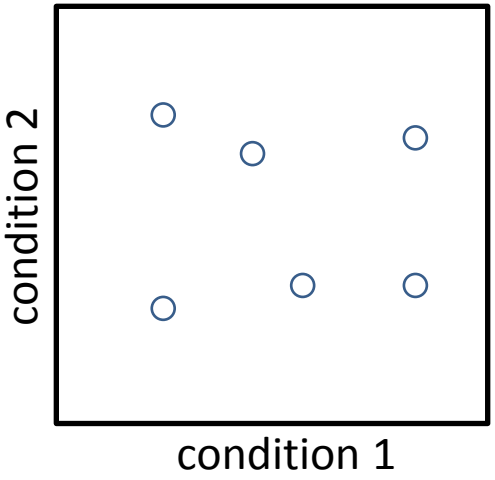




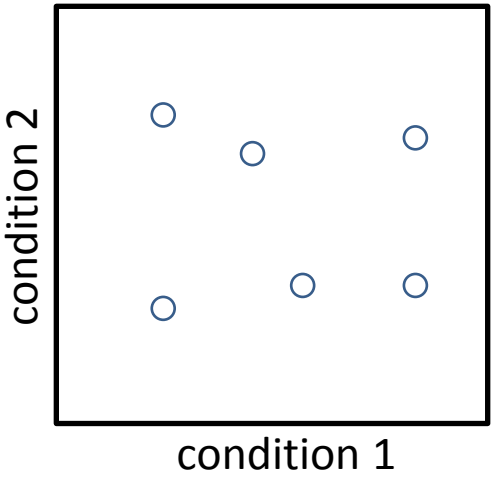
<u>The K-mean approach</u>

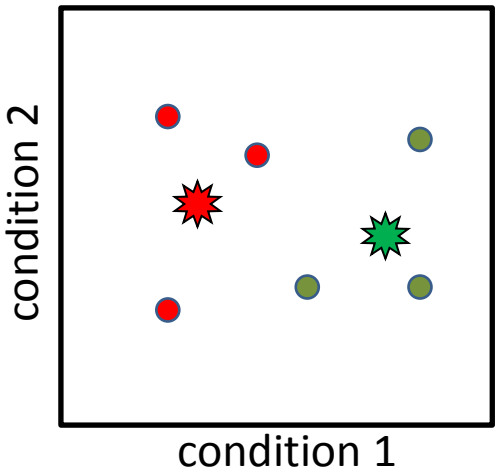
Clustering of *n* observations/points into *k* clusters is 'good'
if each observation is assigned to the cluster with the nearest mean/center

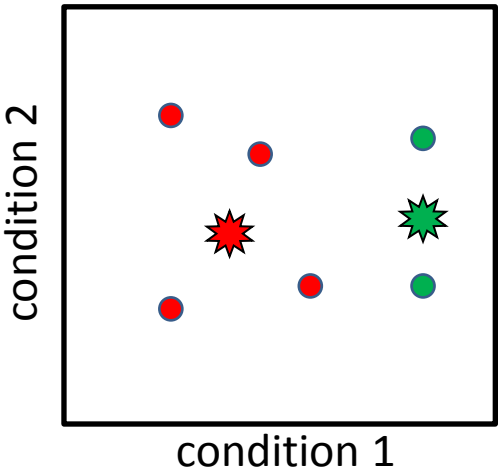
Defining a good clustering solution

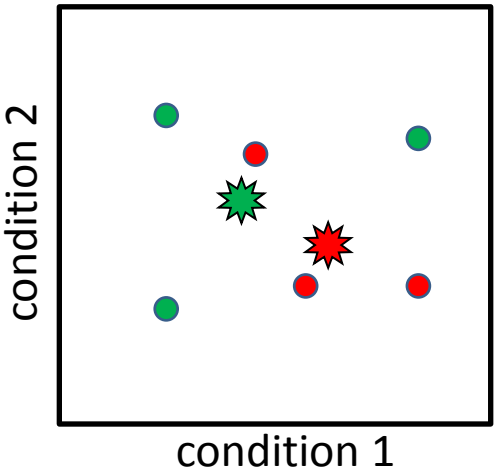


Defining a good clustering solution



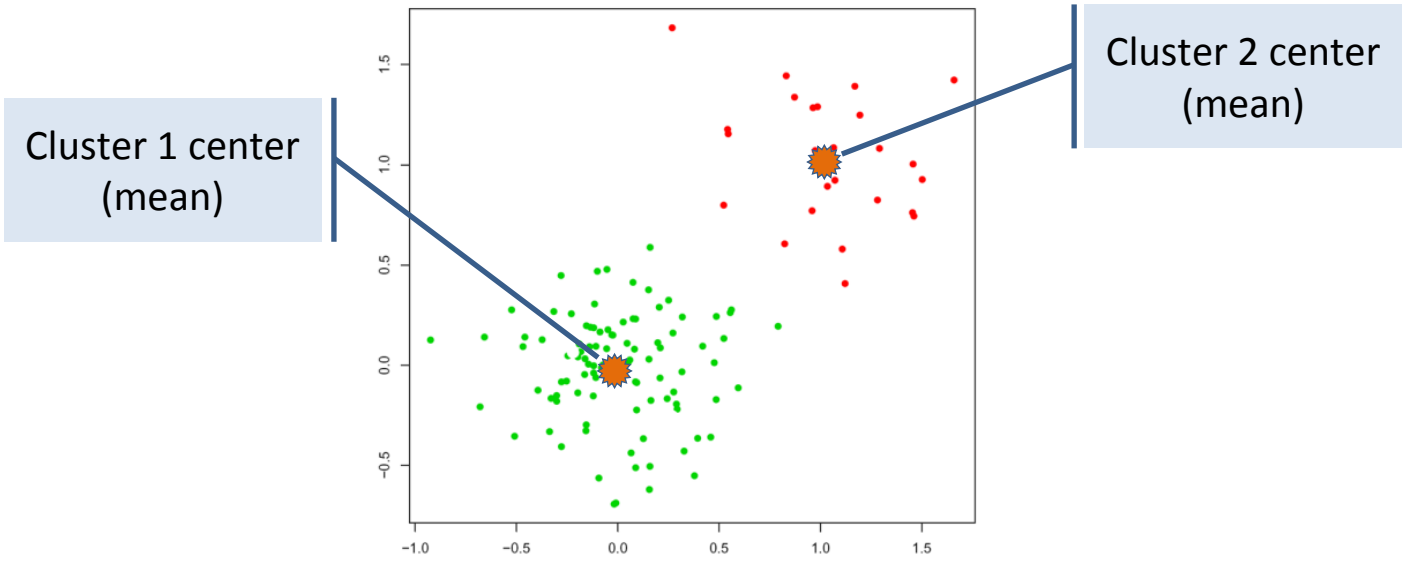






K-mean clustering

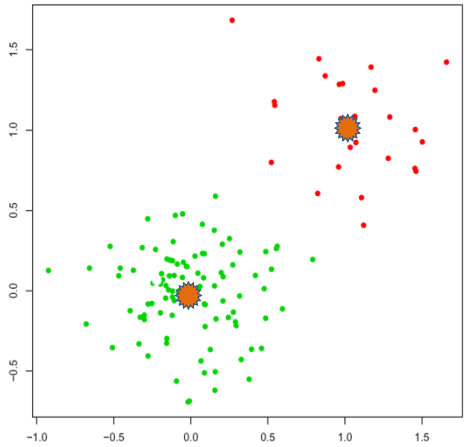
- **An algorithm for partitioning *n* observations/points into *k* clusters such that each observation belongs to the cluster with the nearest mean/center**



But how do we find a clustering solution with this property?

K-mean clustering: Chicken and egg?

- An algorithm for partitioning *n* observations/points into *k* clusters such that each observation belongs to the cluster with the nearest mean/center



- **Note the two components of this definition:**
  - Partitioning of *n* points into clusters
  - Clusters' means
- **A chicken and egg problem:**

  I do not know the means before I determine the partitioning

  I do not know the partitioning before I determine the means

The K-mean clustering algorithm

An iterative approach

- **Key principle - cluster around mobile centers:**
  Start with some **random** locations of means/centers,
  partition into clusters according to these centers,
  then correct the centers according to the clusters, and repeat

  [similar to EM (expectation-maximization) algorithms]

K-mean clustering algorithm

- The number of centers, ***k***, has to be specified a-priori
- **Algorithm:**
  1. Arbitrarily select *k* initial centers
  2. Assign each element to the closest center
  3. Re-calculate centers (mean position of the assigned elements)
  4. Repeat 2 and 3 until …

K-mean clustering algorithm

- The number of centers, ***k***, has to be specified a-priori
- **Algorithm:**
  1. Arbitrarily select *k* initial centers
  2. Assign each element to the closest center
  3. Re-calculate centers (mean position of the assigned elements)
  4. Repeat 2 and 3 until one of the following termination conditions is reached:
     i. The clusters are the same as in the previous iteration (stable solution)
     ii. The clusters are as in **some** previous iteration (cycle)
     iii. The difference between two iterations is small??
     iv. The maximum number of iterations has been reached

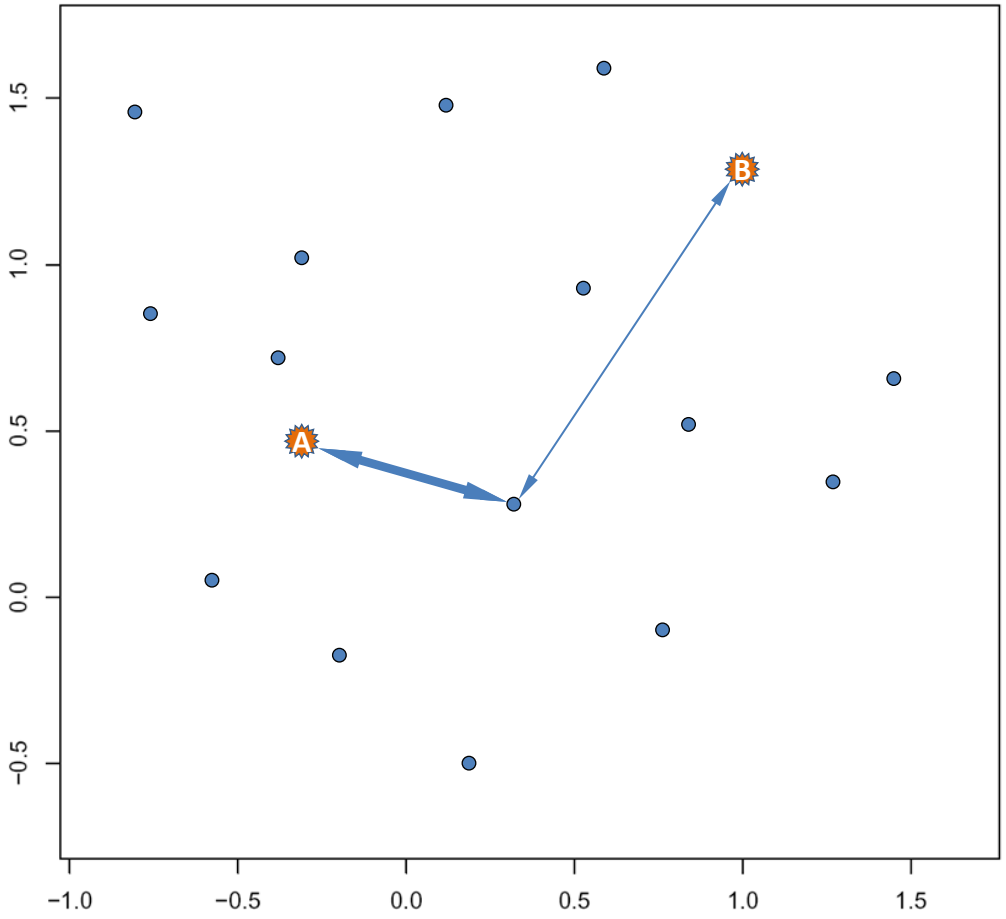
K-mean clustering algorithm

- The number of centers, ***k***, has to be specified a-priori
- **Algorithm:**
  1. Arbitrarily select *k* initial centers
  2. **Assign each element to the closest center**
  3. Re-calculate centers (mean position of the assigned elements)
  4. Repeat 2 and 3 until one of the following termination conditions is reached:
     i. The clusters are the same as in the previous iteration (stable solution)
     ii. The clusters are as in **some** previous iteration (cycle)
     iii. The difference between two iterations is small??
     iv. The maximum number of iterations has been reached

How can we do this efficiently?

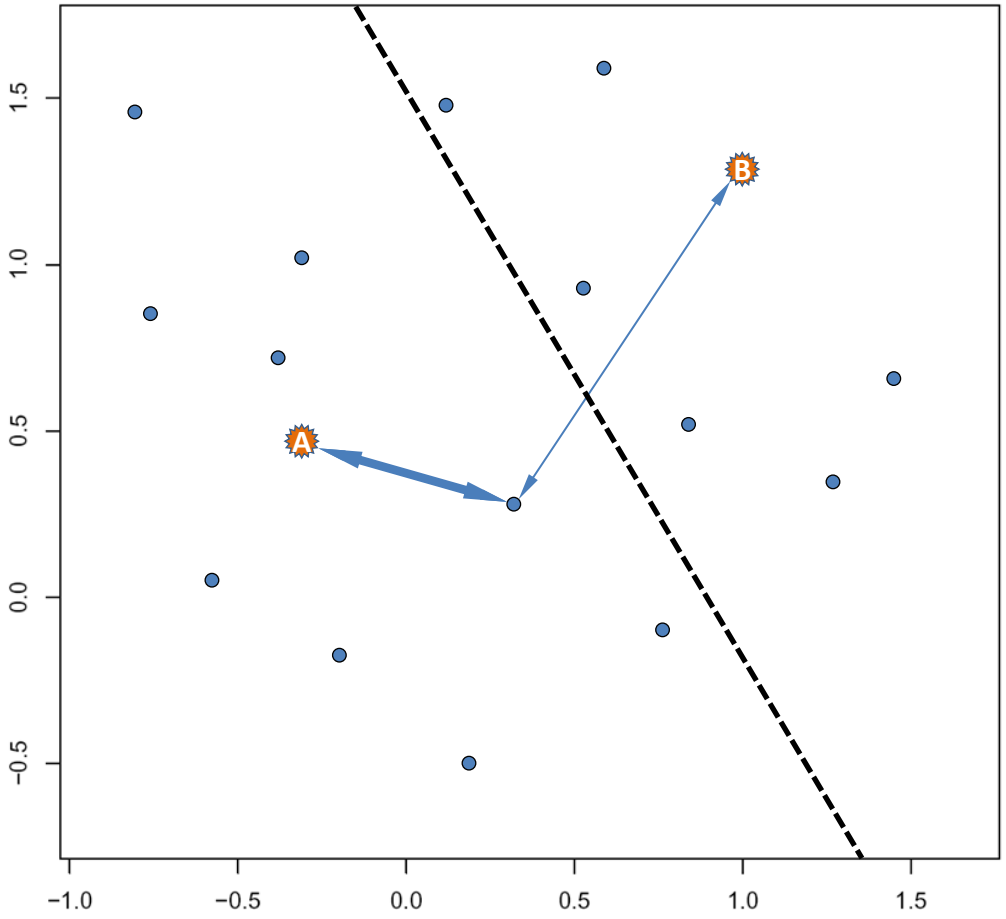
Assigning elements to the closest center

- Could be computationally intensive ….



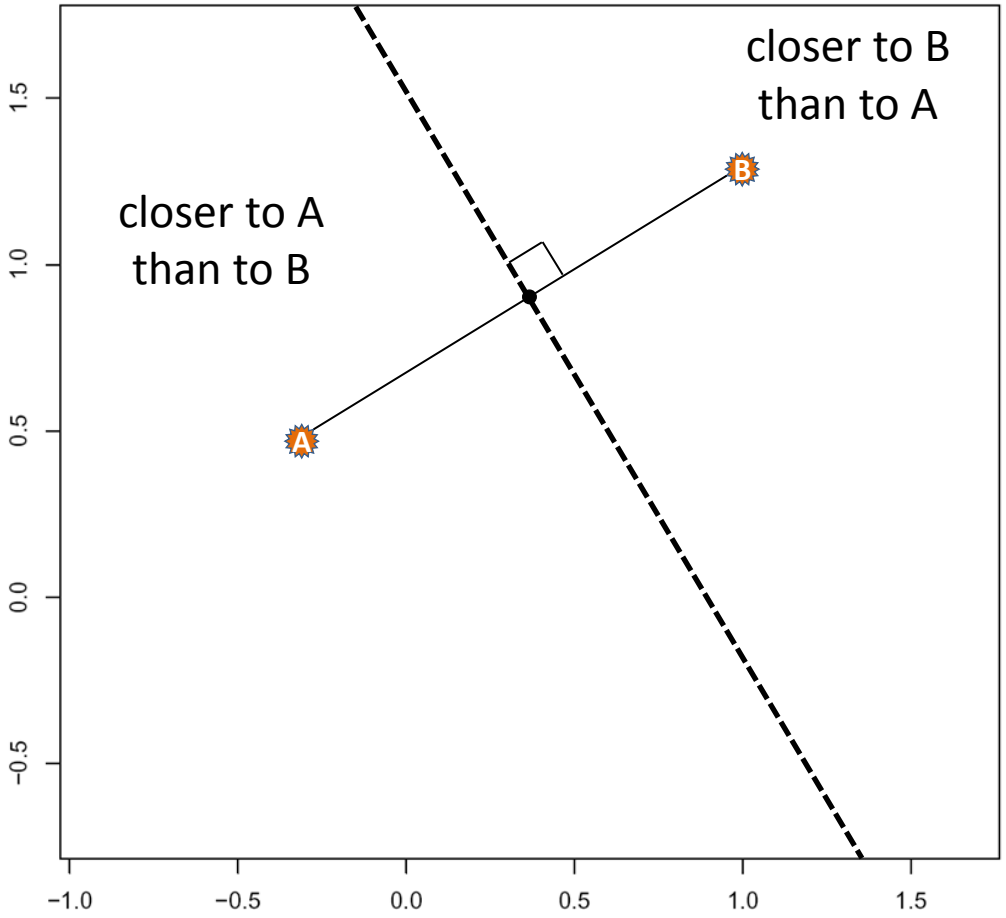
Assigning elements to the closest center

- Could be computationally intensive ….
- Preprocessing (by partitioning the space) can help



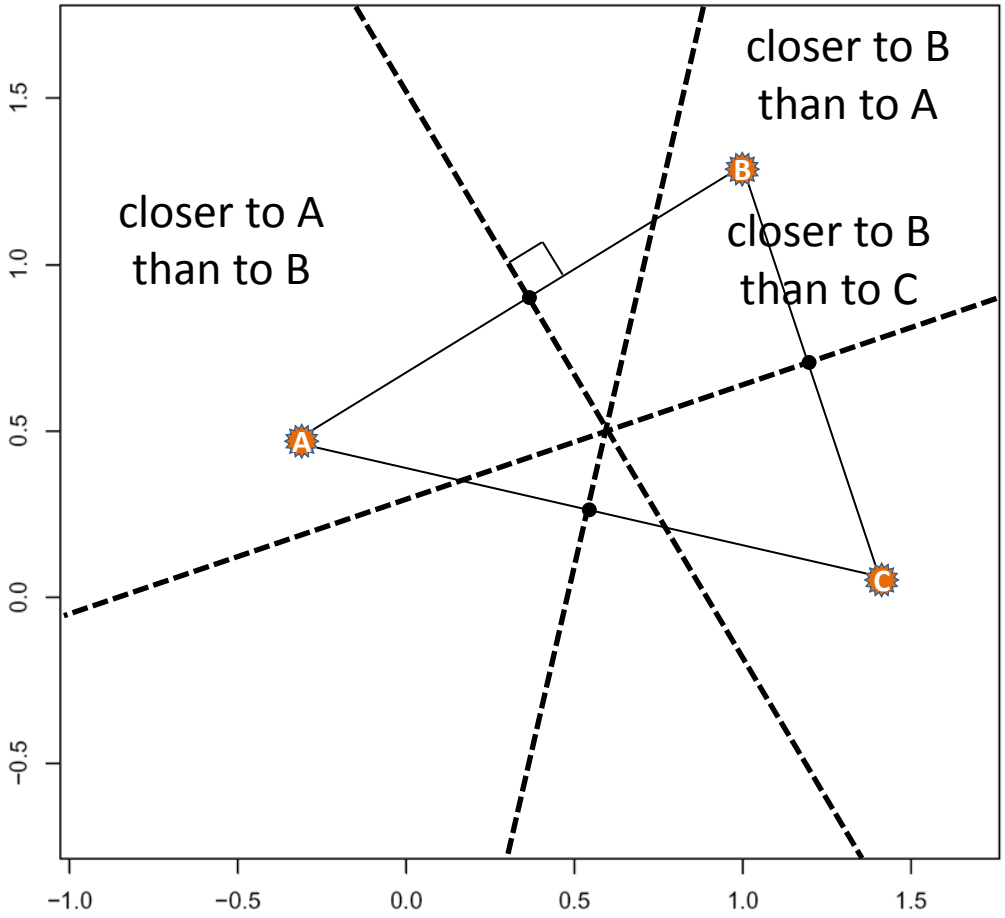
Assigning elements to the closest center

- Could be computationally intensive ….
- Preprocessing (by partitioning the space) can help



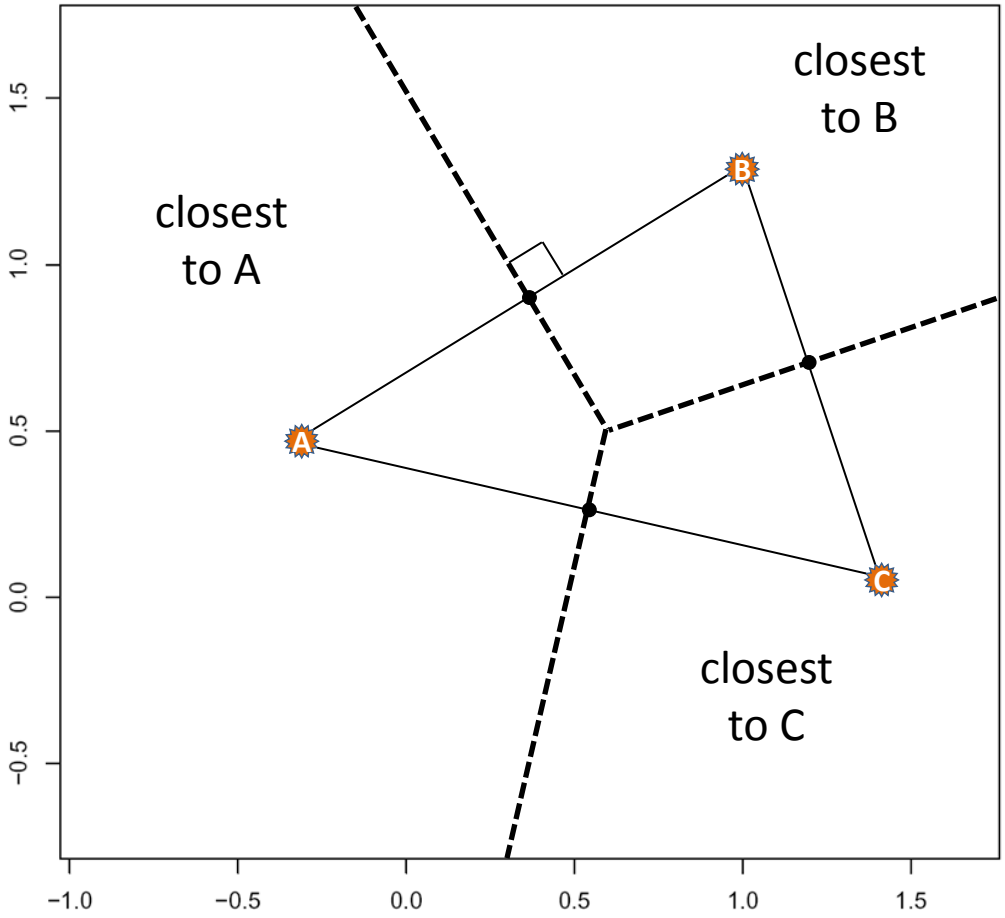
Assigning elements to the closest center

- Could be computationally intensive ….
- Preprocessing (by partitioning the space) can help



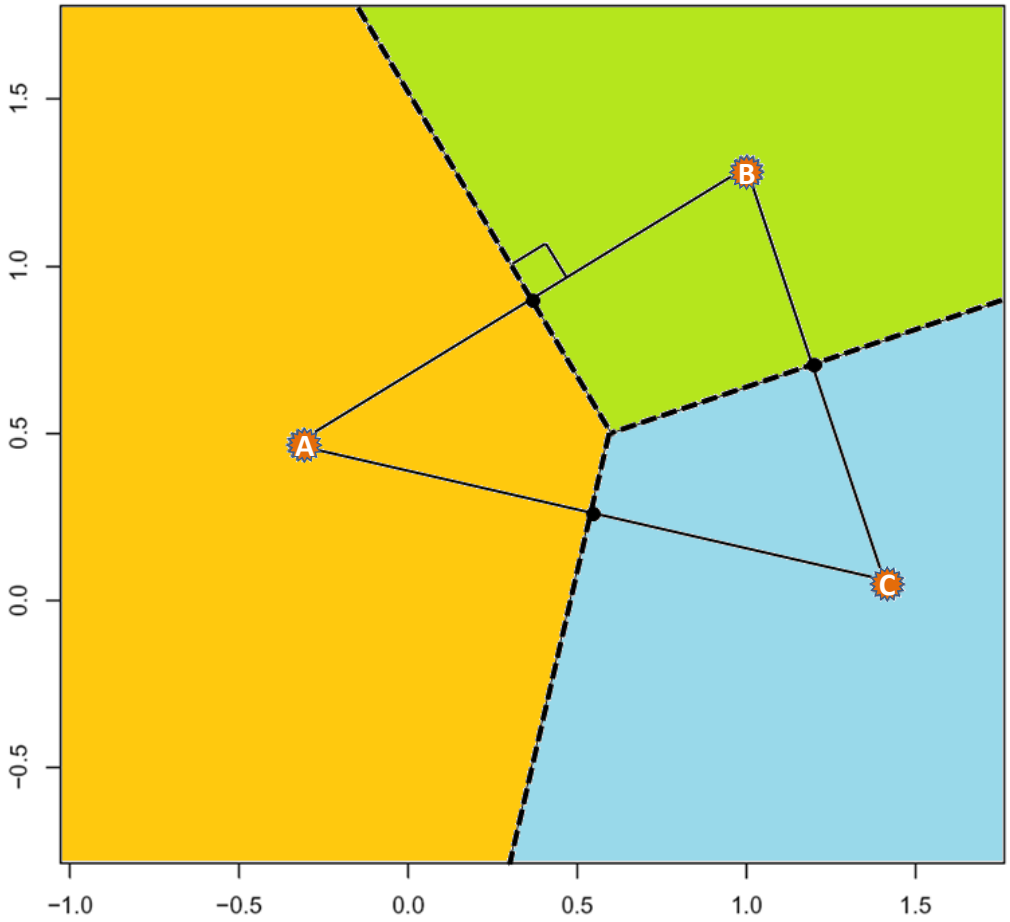
Assigning elements to the closest center

- Could be computationally intensive ….
- Preprocessing (by partitioning the space) can help



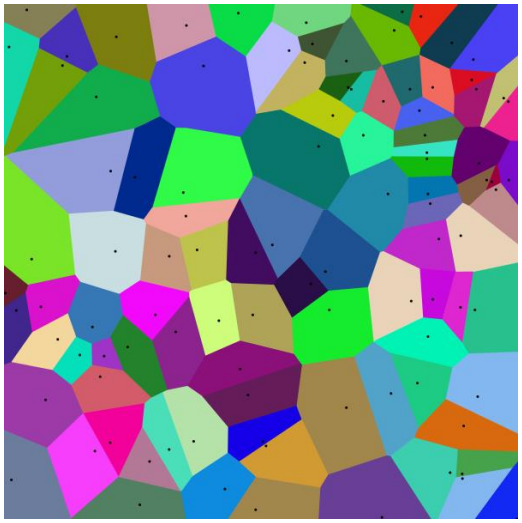
Assigning elements to the closest center

- Could be computationally intensive ….
- Preprocessing (by partitioning the space) can help



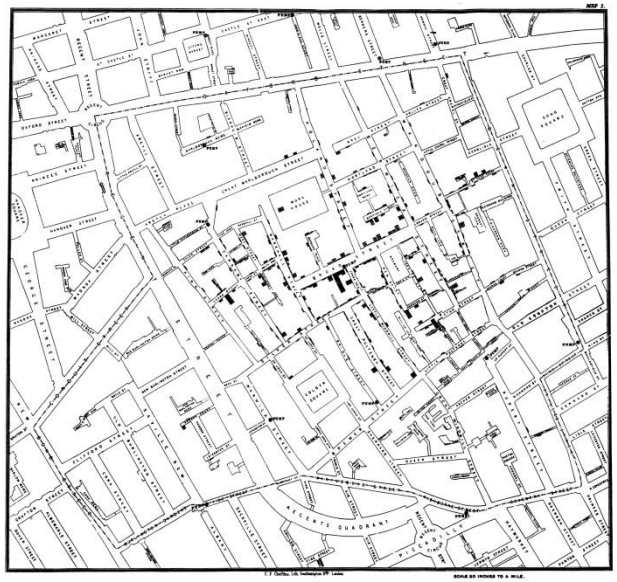
Voronoi diagram

- Decomposition of a metric space determined by distances to a specified discrete set of “centers” in the space
  (each colored cell represents the collection of all points in this space that are closer to a specific center than to any other)



- Several algorithms exist to find the Voronoi diagram

- **Numerous** applications
  (e.g., the 1854 Broad Street cholera outbreak in Soho England, Aviation, and many others)

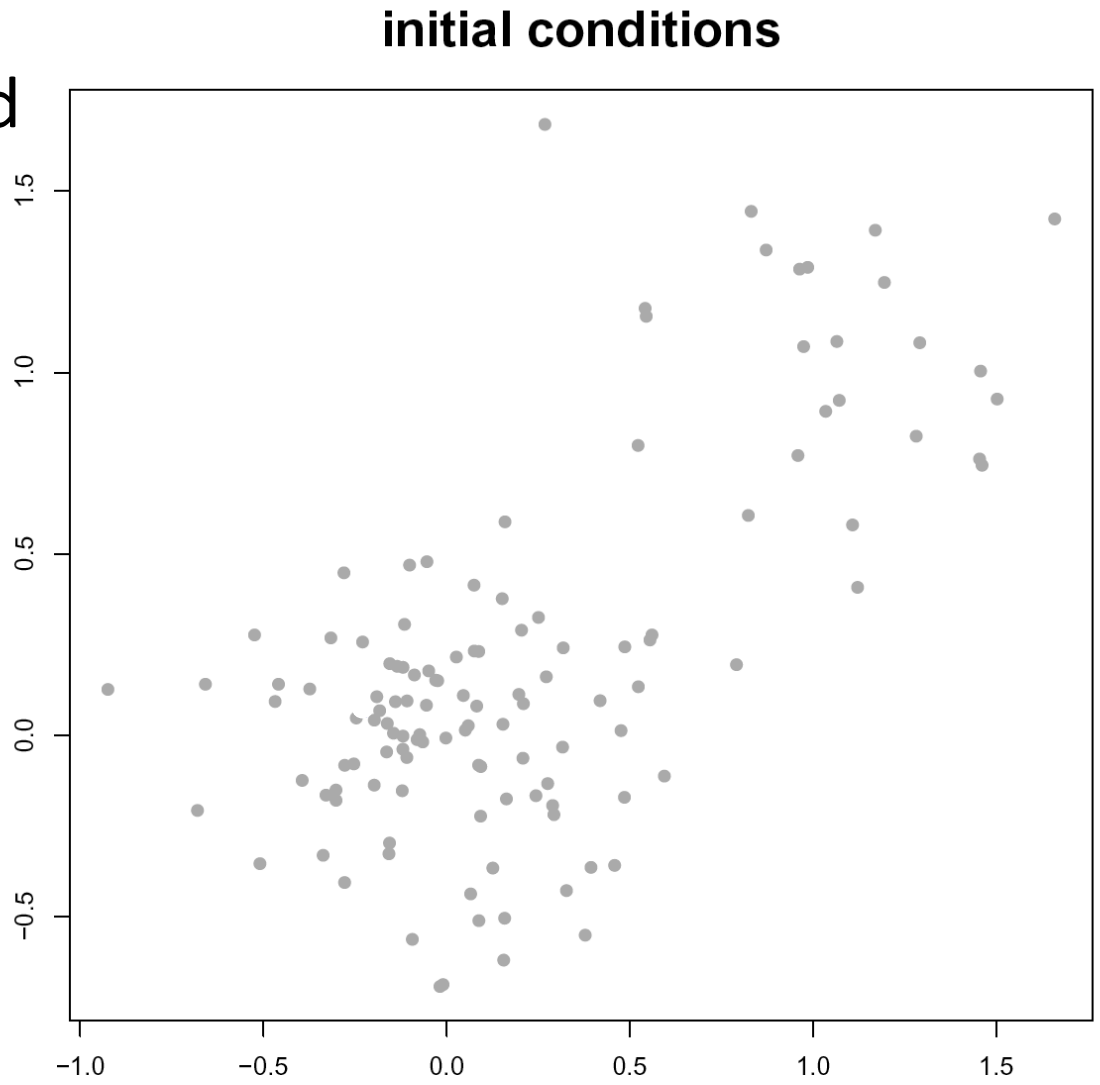


K-mean clustering algorithm

- The number of centers, ***k***, has to be specified a-priori
- **Algorithm:**
  1. Arbitrarily select *k* initial centers
  2. Assign each element to the closest center **(Voronoi)**
  3. Re-calculate centers (mean position of the assigned elements)
  4. Repeat 2 and 3 until one of the following termination conditions is reached:
     i. The clusters are the same as in the previous iteration (stable solution)
     ii. The clusters are as in **some** previous iteration (cycle)
     iii. The difference between two iterations is small??
     iv. The maximum number of iterations has been reached

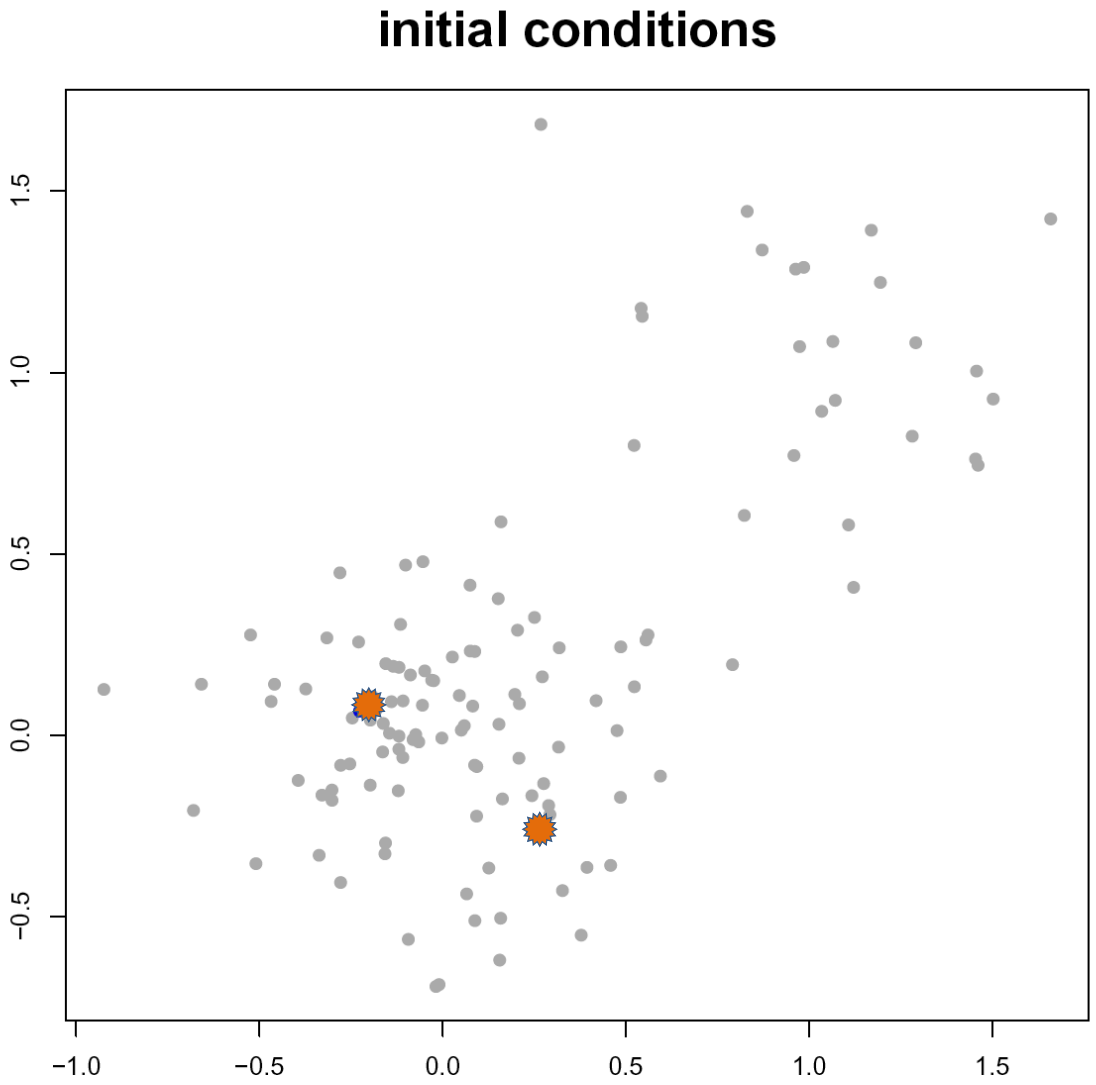
K-mean clustering example

- Two sets of points randomly generated
  - 200 centered on (0,0)
  - 50 centered on (1,1)



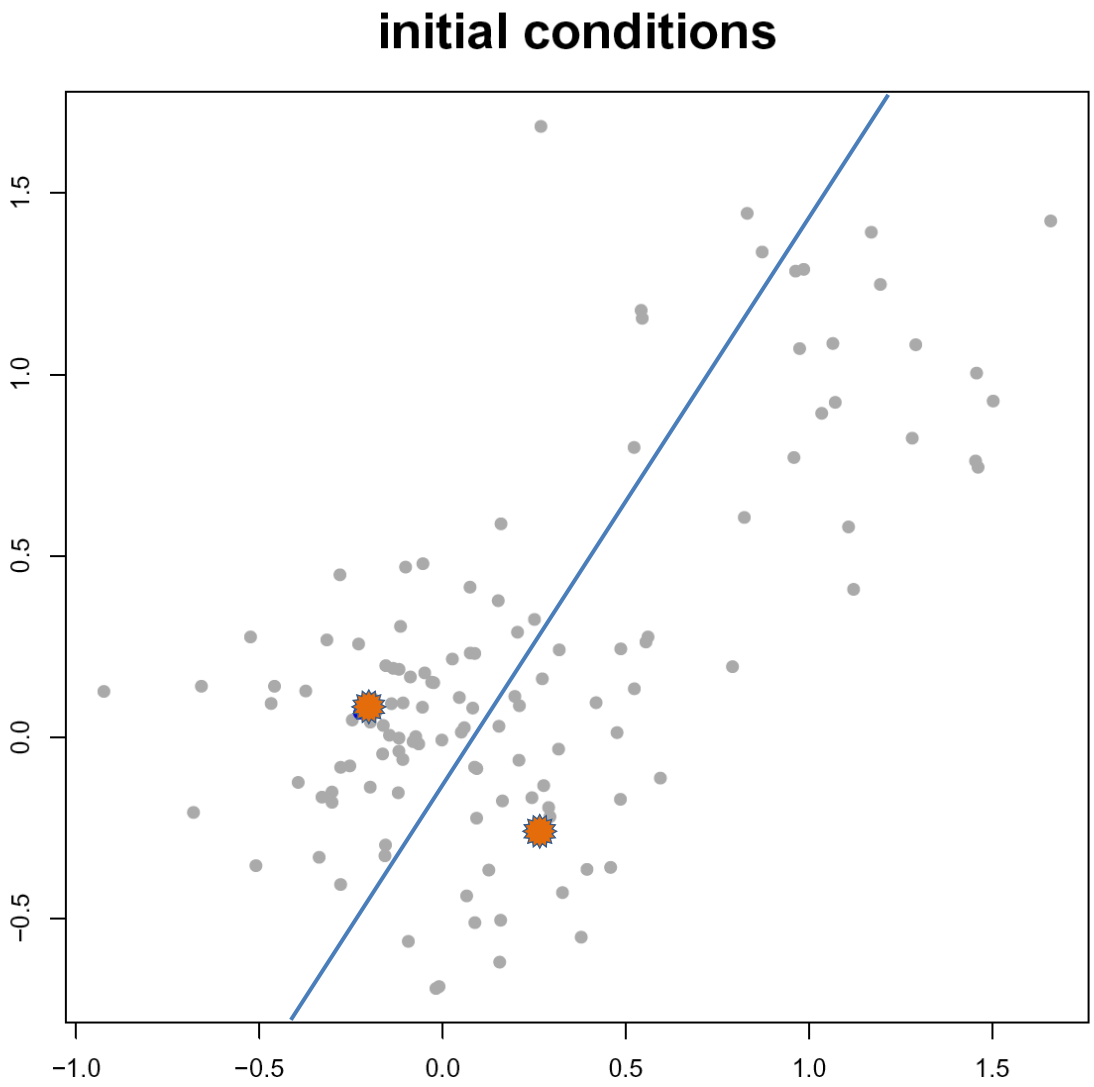
K-mean clustering example

- Two points are randomly chosen as centers (stars)



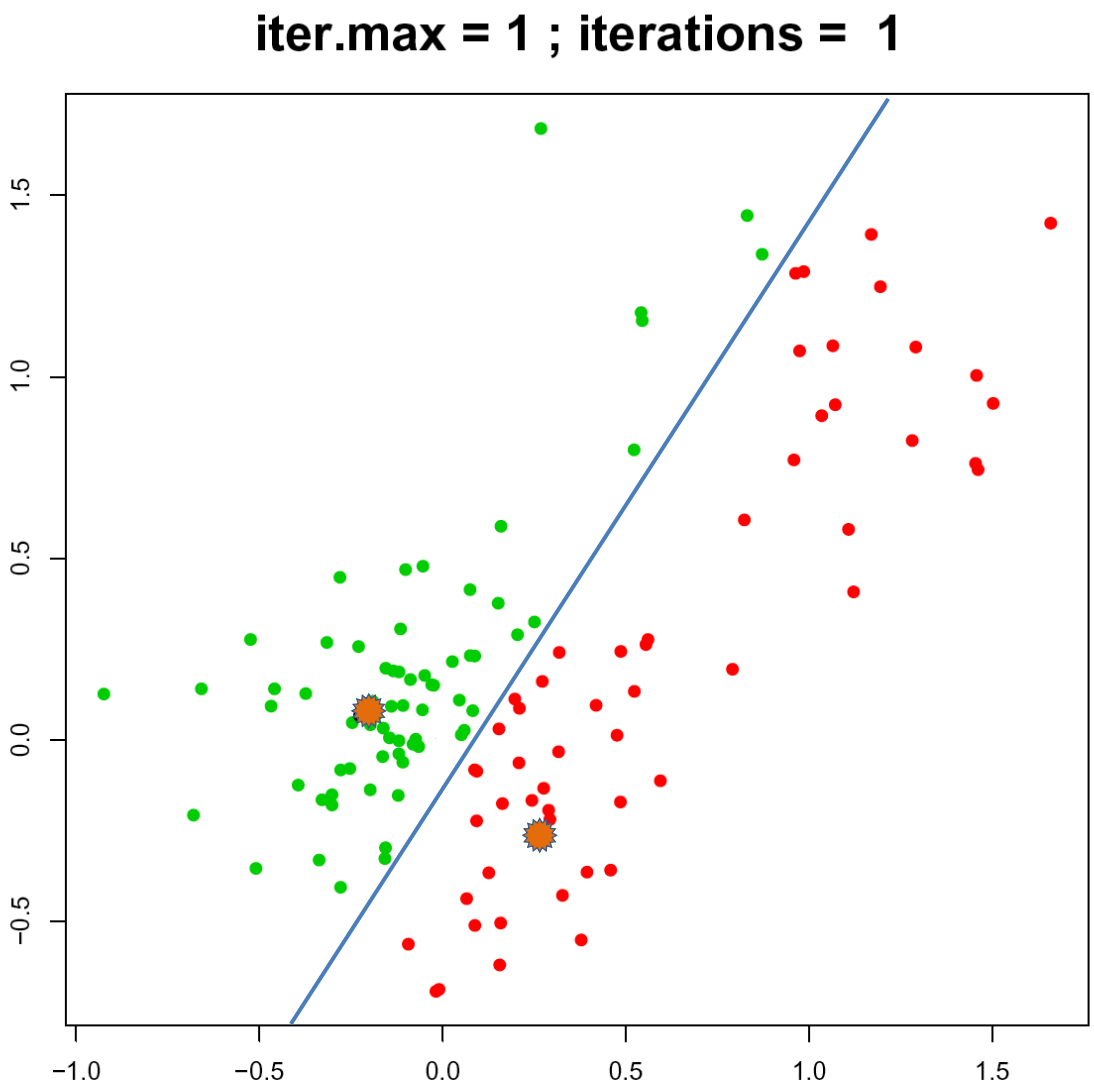
K-mean clustering example

- Each dot can now be assigned to the cluster with the closest center



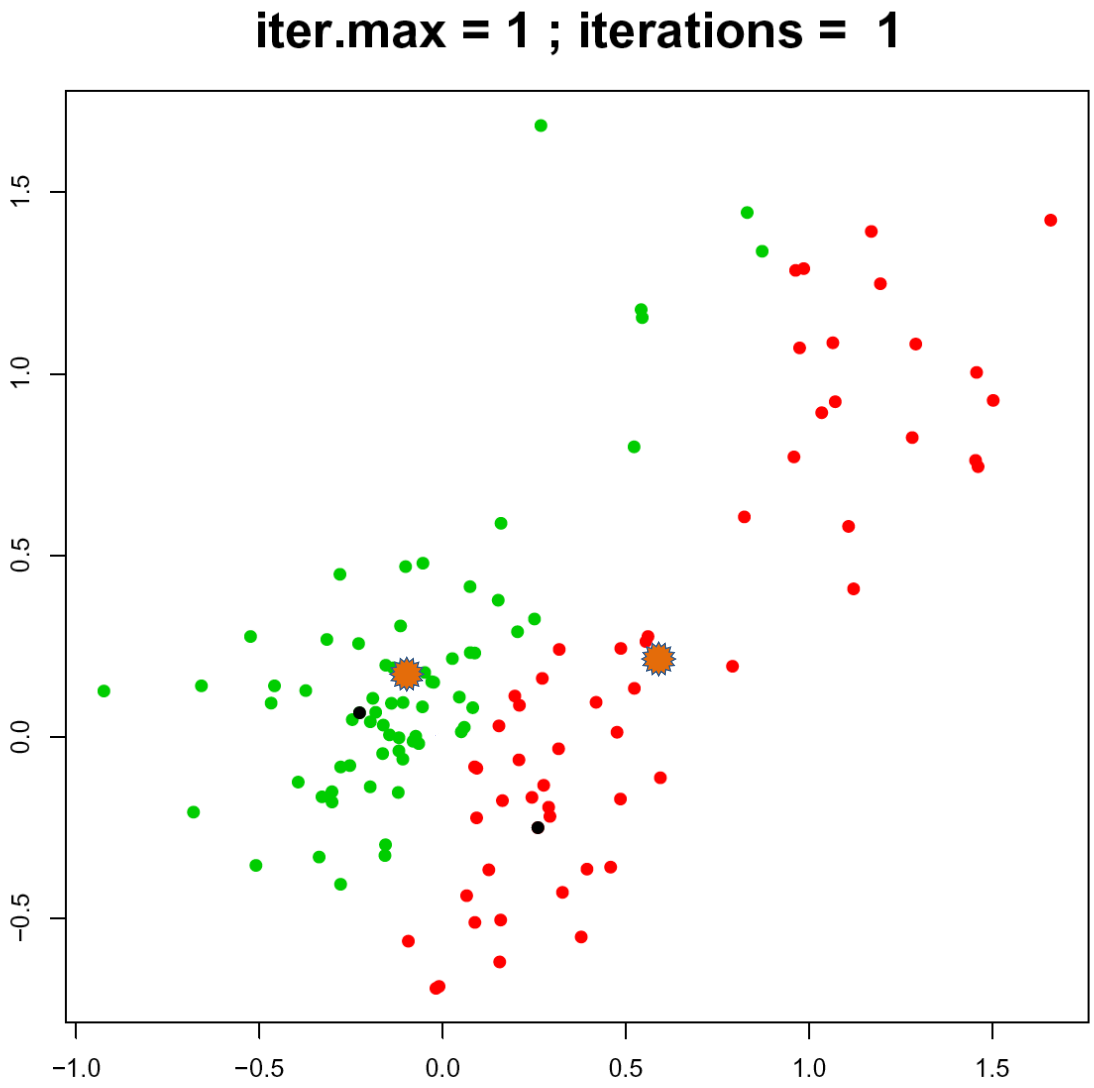
K-mean clustering example

- First partition into clusters



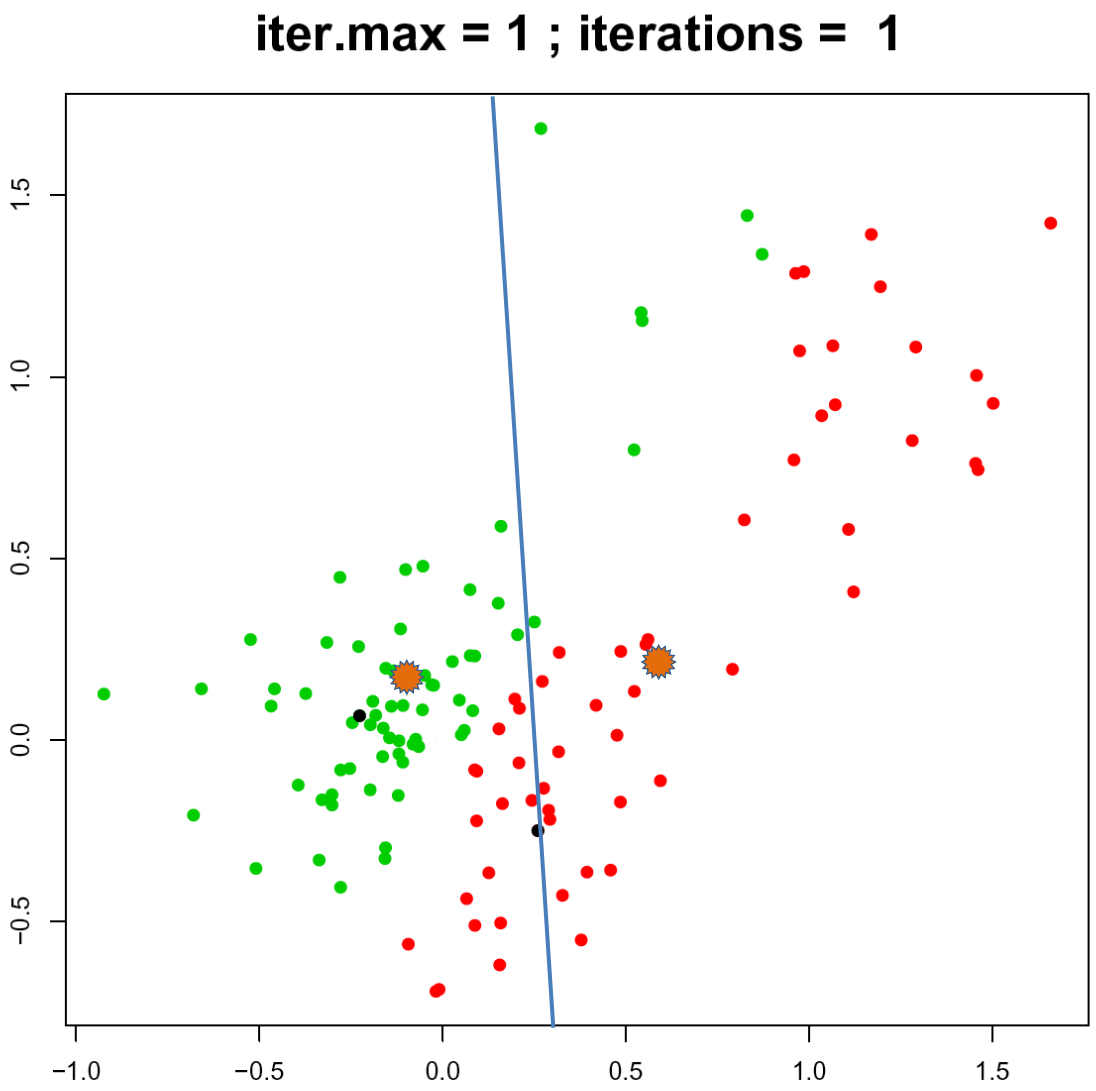
K-mean clustering example

- Centers are re-calculated



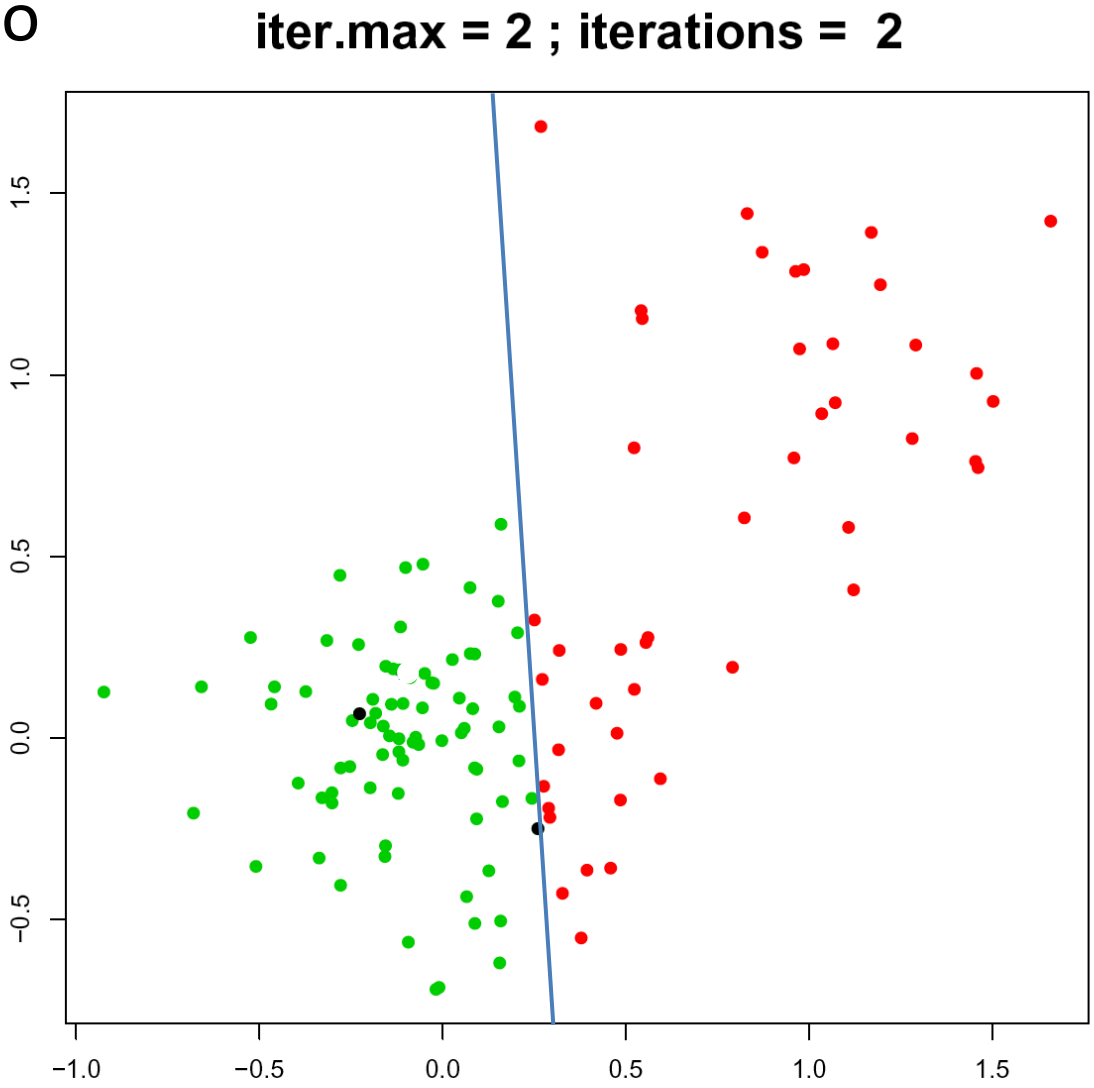
K-mean clustering example

- And are again used to partition the points



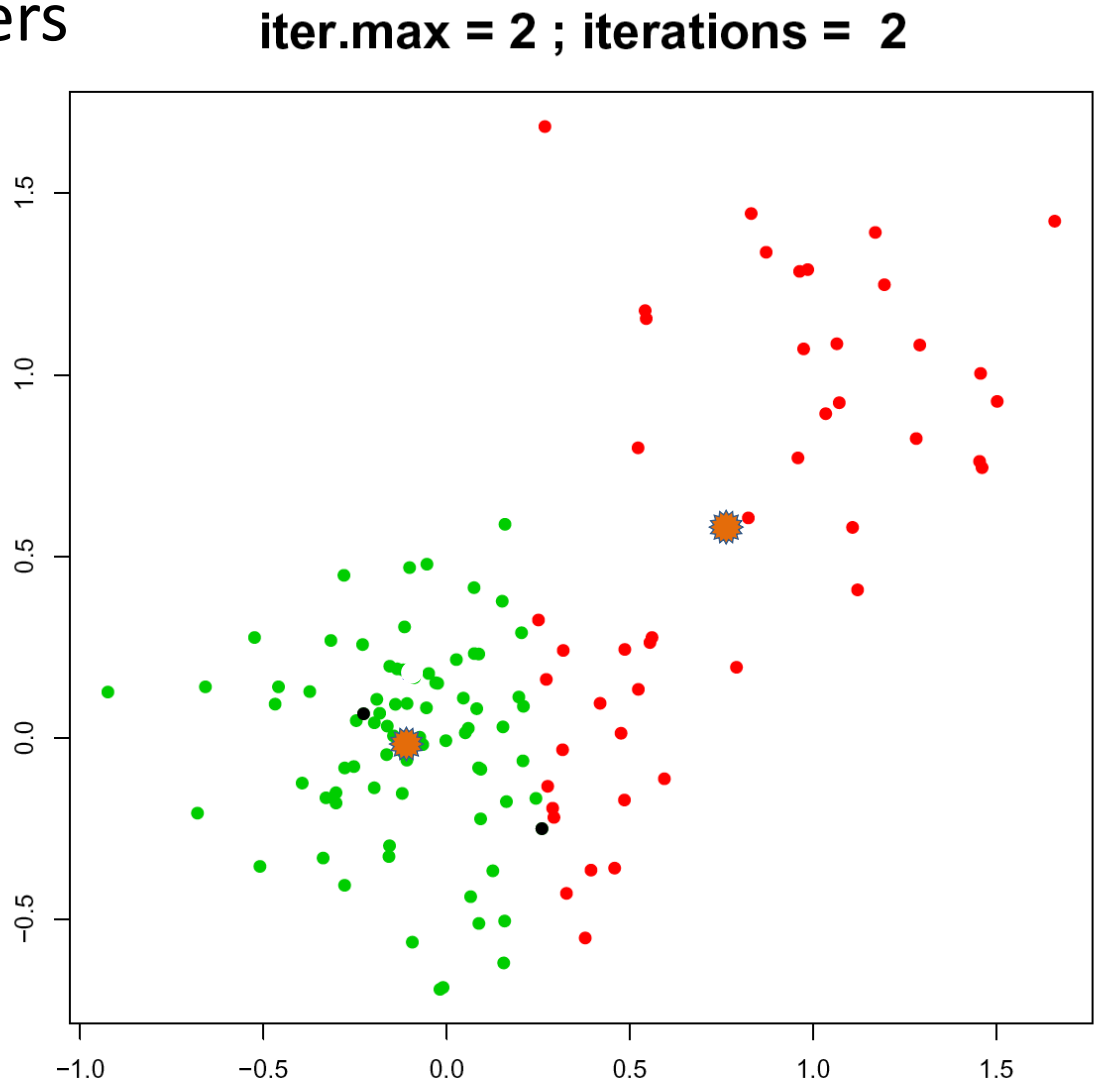
K-mean clustering example

- Second partition into clusters



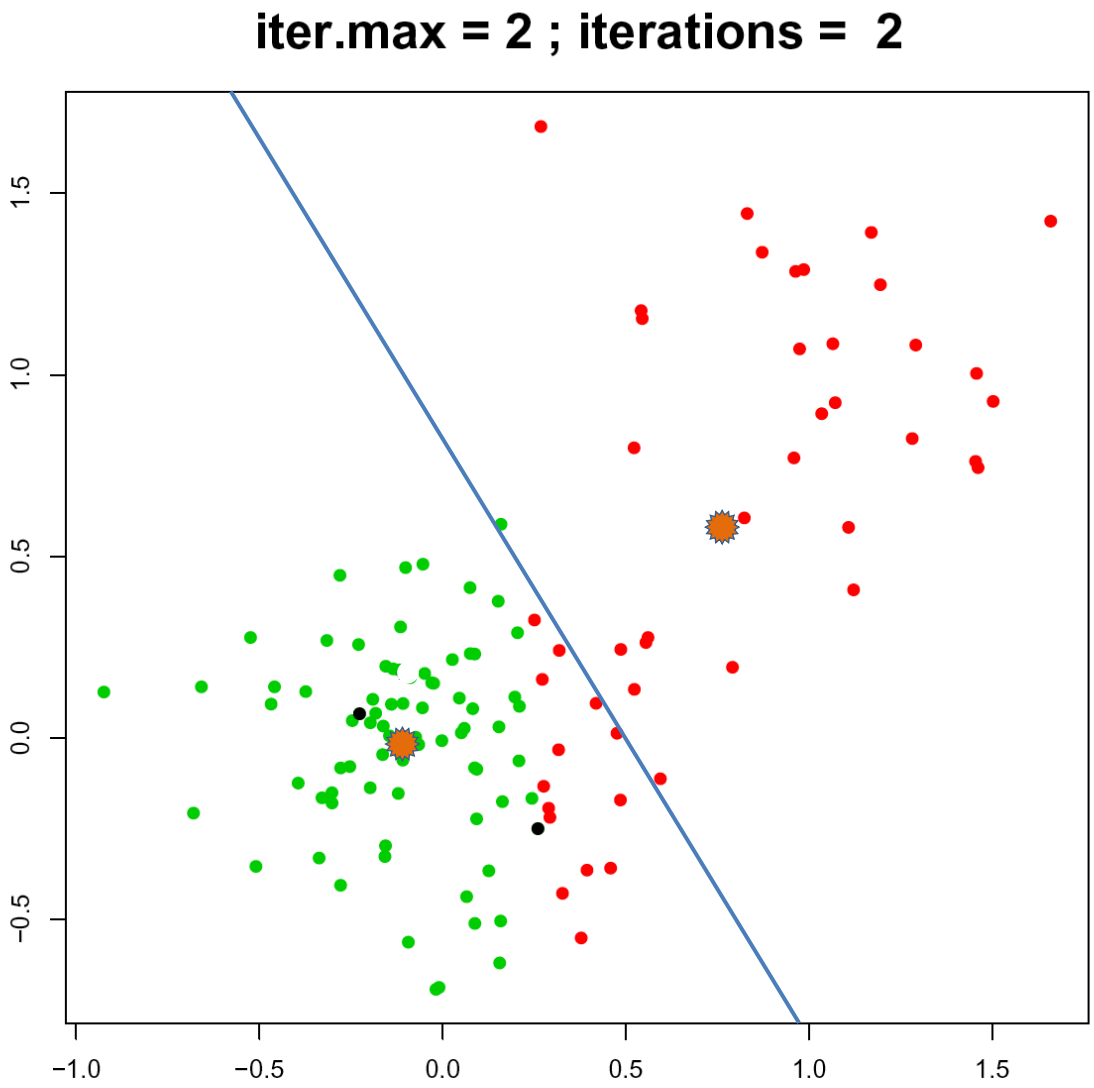
K-mean clustering example

- Re-calculating centers again



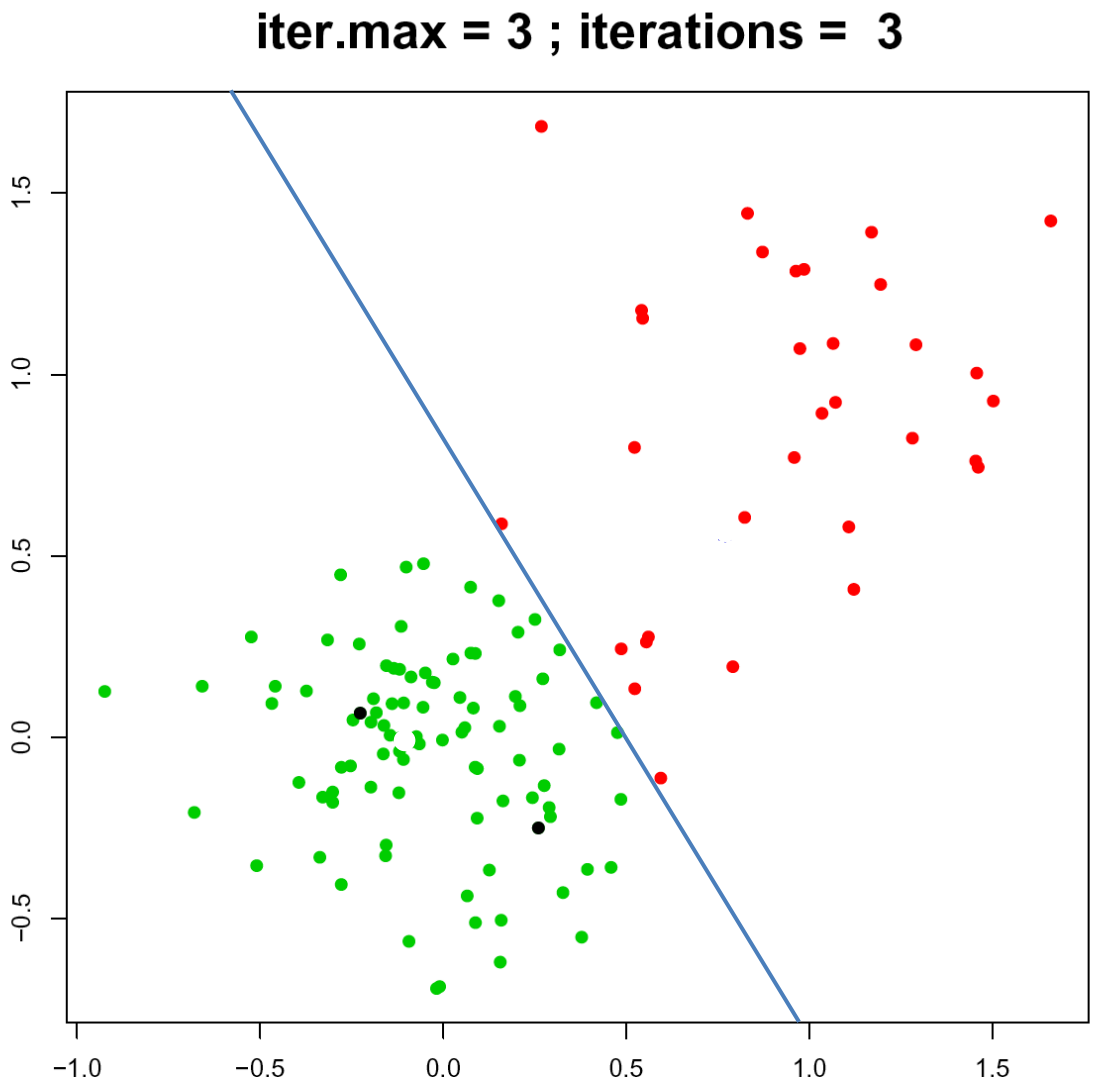
K-mean clustering example

- And we can again partition the points



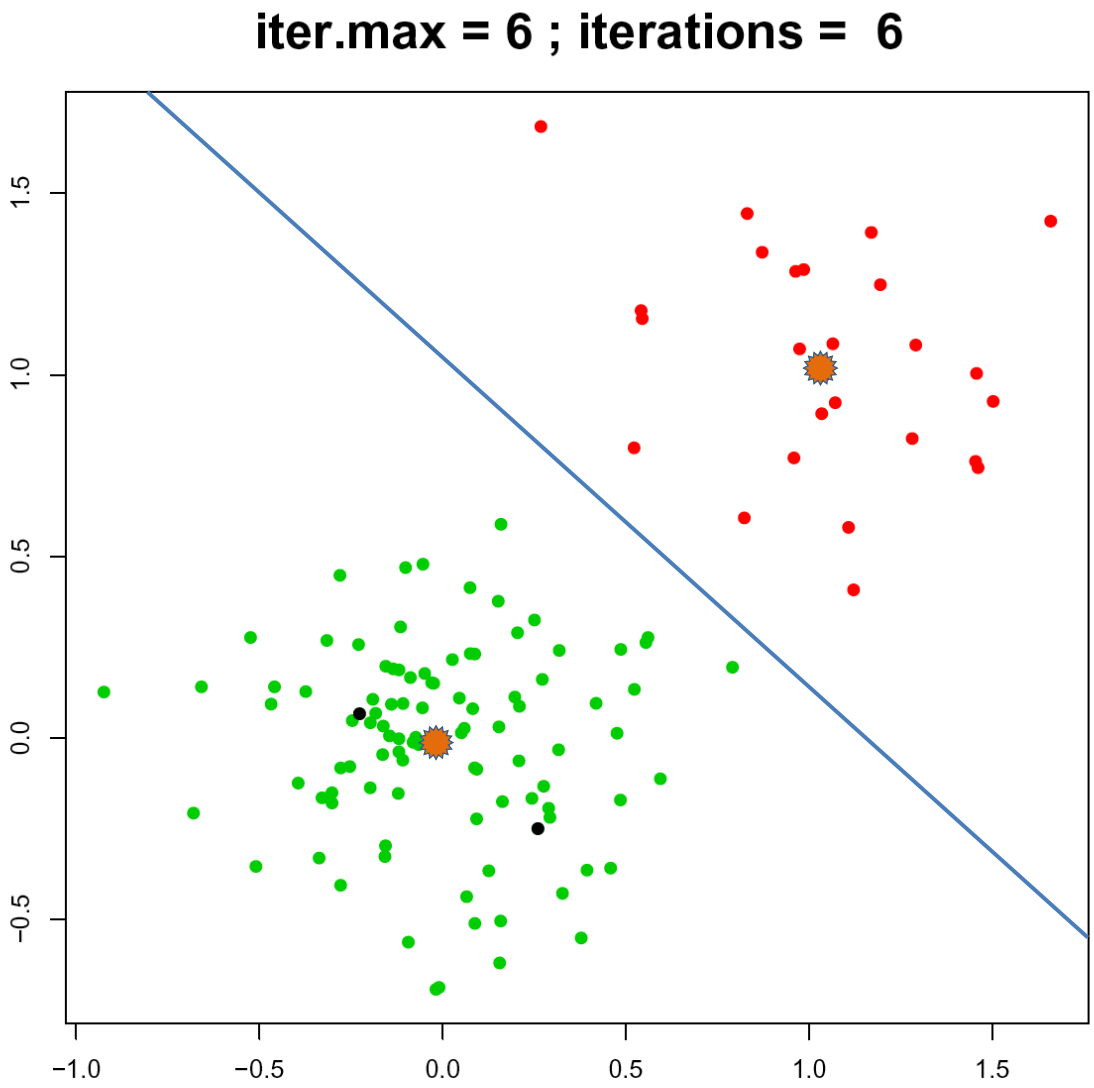
K-mean clustering example

- Third partition into clusters



K-mean clustering example

- After 6 iterations:
- The calculated centers remains stable



K-mean clustering: Summary

- The convergence of k-mean is usually quite fast (sometimes 1 iteration results in a stable solution)
- K-means is time- and memory-efficient
- **Strengths:**
  - Simple to use
  - Fast
  - Can be used with very large data sets
- **Weaknesses:**
  - **The number of clusters has to be predetermined**
  - The results may vary depending on the initial choice of centers

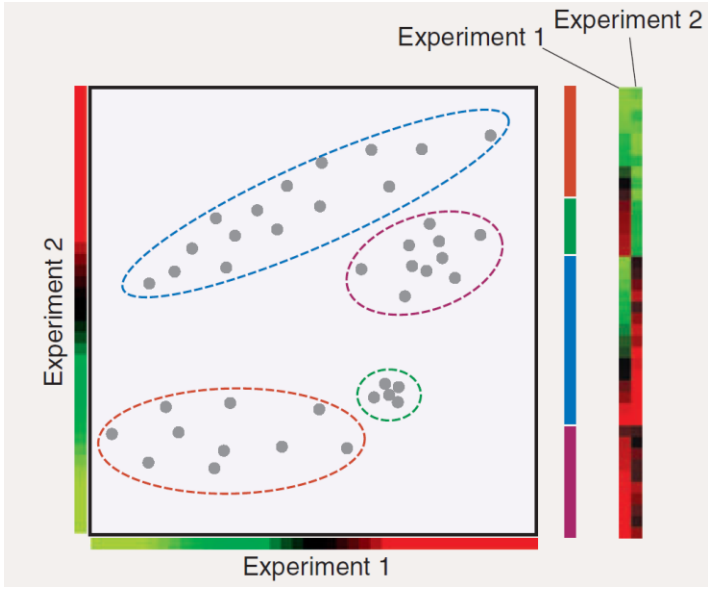
K-mean clustering: Variations

- Expectation-maximization (**EM**):
  maintains **probabilistic** assignments to clusters,
  instead of deterministic assignments, and multivariate
  Gaussian distributions instead of means.

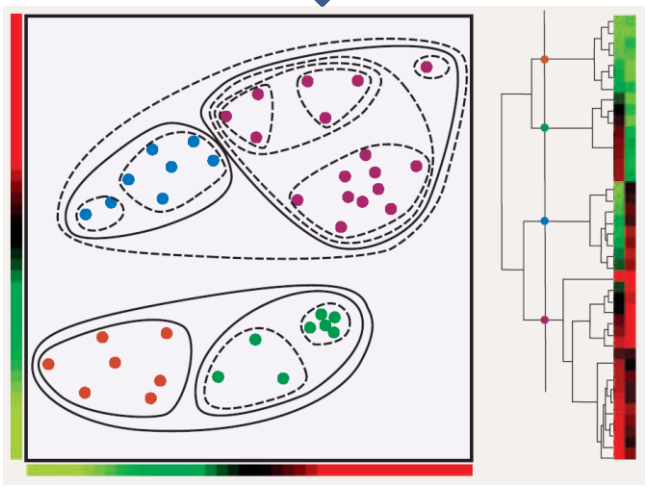
- k-means++: attempts to choose better starting points.

- Some variations attempt to escape local optima by
  swapping points between clusters

An important take-home message

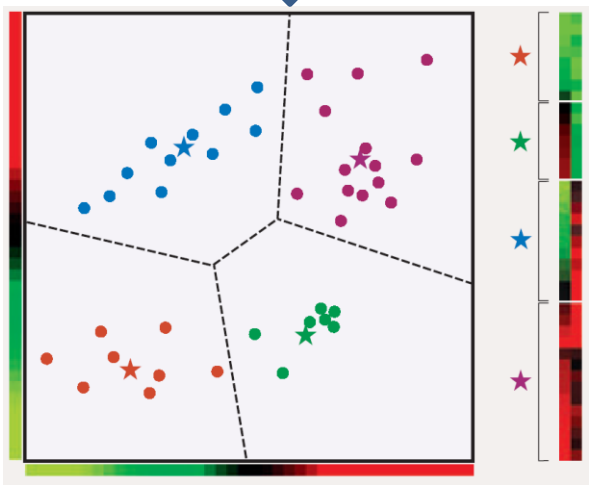


Hierarchical clustering



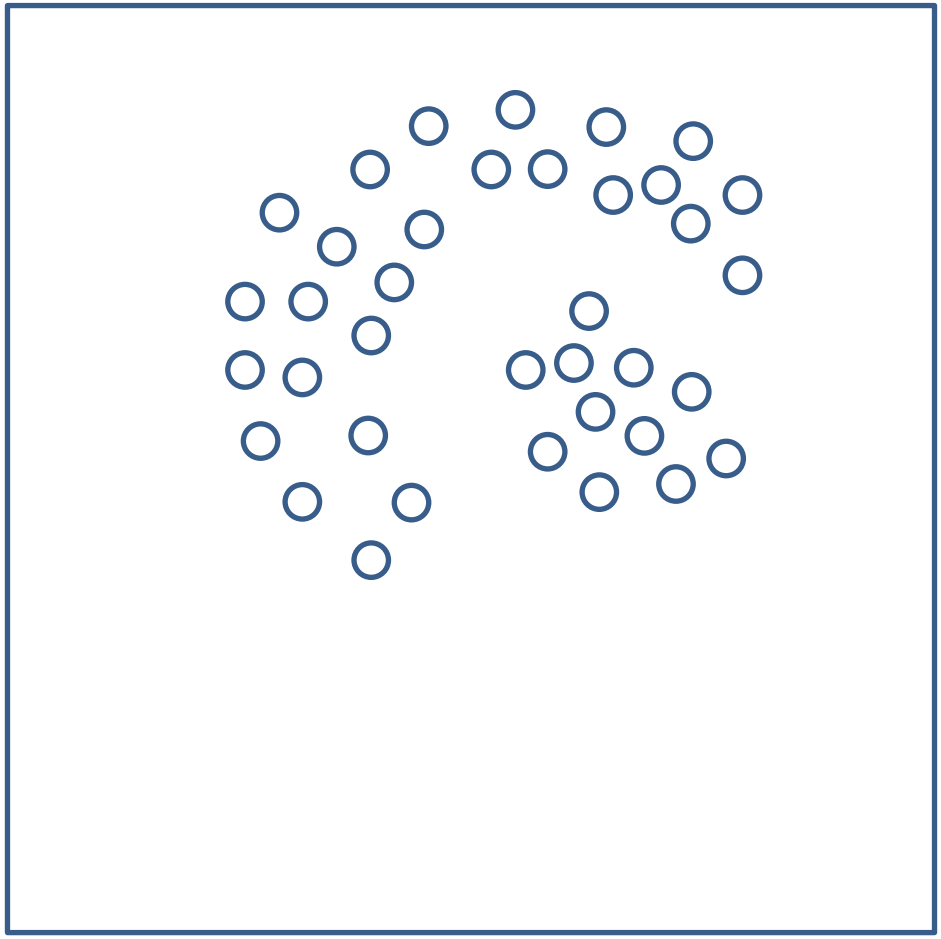
?

K-mean clustering



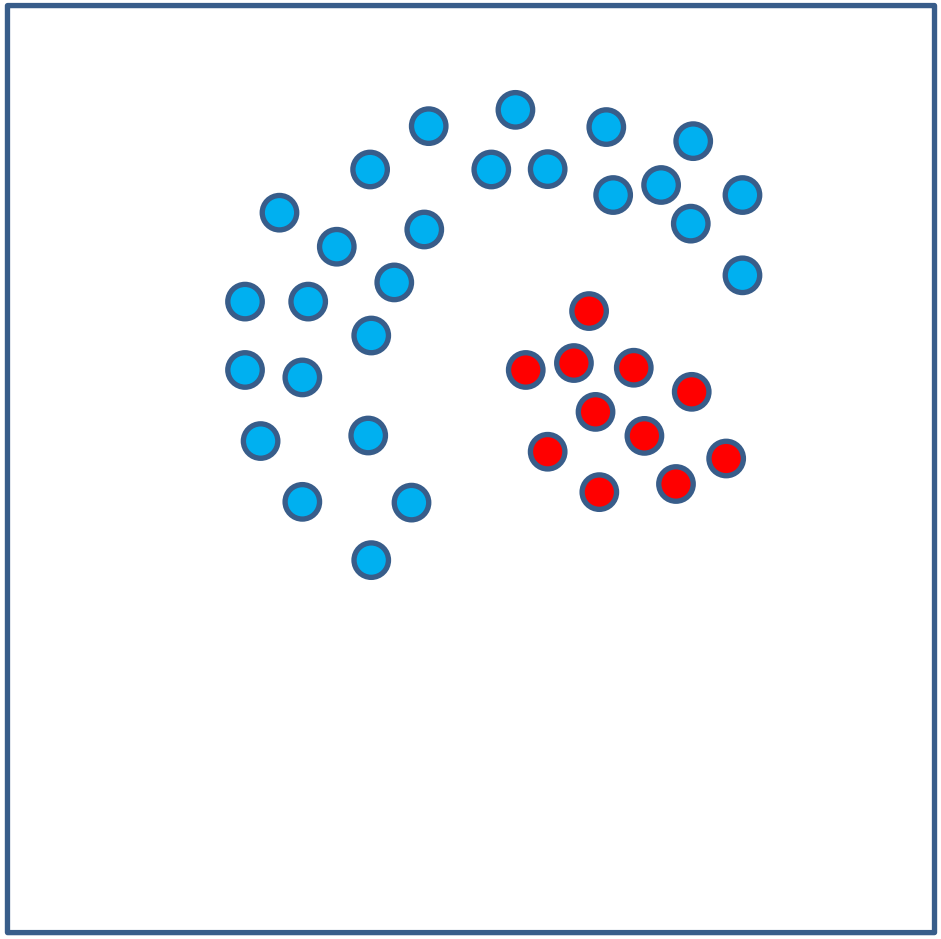
D'haeseleer, 2005

What else are we missing?



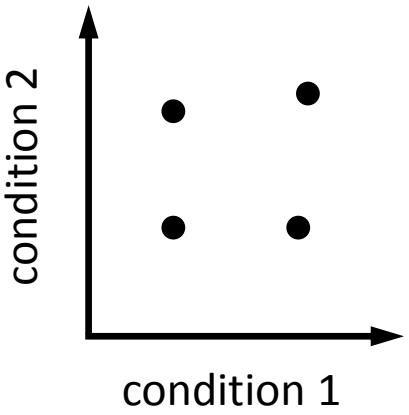
What else are we missing?

- What if the clusters are not “linearly separable”?



Defining a good clustering solution

The K-mean approach



Defining a good clustering solution

The K-mean approach