INNOPOLIS UNIVERSITY

Machine Learning

Prof. Adil Khan

Objectives

1. What is clustering? How is it different from classification? Why is it called unsupervised learning?
2. What is k-means? How does it work? What is its objective function? How is it motivated?
3. What is k-means++? Why is it motivated?
4. Limitations of k-means
5. K-means++
6. Hierarchical Clustering

Clustering

Recall: Supervised Learning

- Learning with a teacher
- Teacher provides the supervision
- In the context of machine learning, the labels y_i provides the supervision for x_i

Recall: Supervised Learning (2)

- Easy and well-defined

- Given $Y = f(X) + \epsilon$ and labeled dataset

$$\mathbb{D} = \{(x_i, y_i) | x_i \in \mathbb{R}^p, y_i \in \{-1,1\}\}_{i=1}^{m}$$

- Estimate the function f as $\hat{f}$, such that when a new input data x_0 is given, we can predict its outcome y_0.

Unsupervised Learning

- We have only predictors (a.k.a inputs, or features) but no labels

$$\mathbb{D} = \{(x_i) | x_i \in \mathbb{R}^p \}_{i=1}^{m}$$

- Then what is the goal of learning?

Scenario (1)

- Imagine you run an online store and would like to personalize your customers' shopping experience
- You think you can do this by providing each customer with personalized recommendations
- You do not know each user's personal preferences and tastes but you have lots of data on their purchases
- How can you used this data to create recommendations?

Scenario (2)

- Imagine you are a biologist interested in studying the behavior of bees
- You have collected a lot of videos (or other data on them)
- How can you use this data to study bee behavior?

Unsupervised Learning

- In the discussed scenarios, we are not interested in prediction, because we do not have an associated response variable
- Instead, the unsupervised **learning goal** is to model the hidden patterns or underlying structure in the given input data in order to learn about the data
- This *underlying structure* is what we usually refer to as *groups* of data
- And these groups are what we refer to as ***Clusters***

Example

- Given a set of documents, we used their data to group them into two clusters

SPORTS

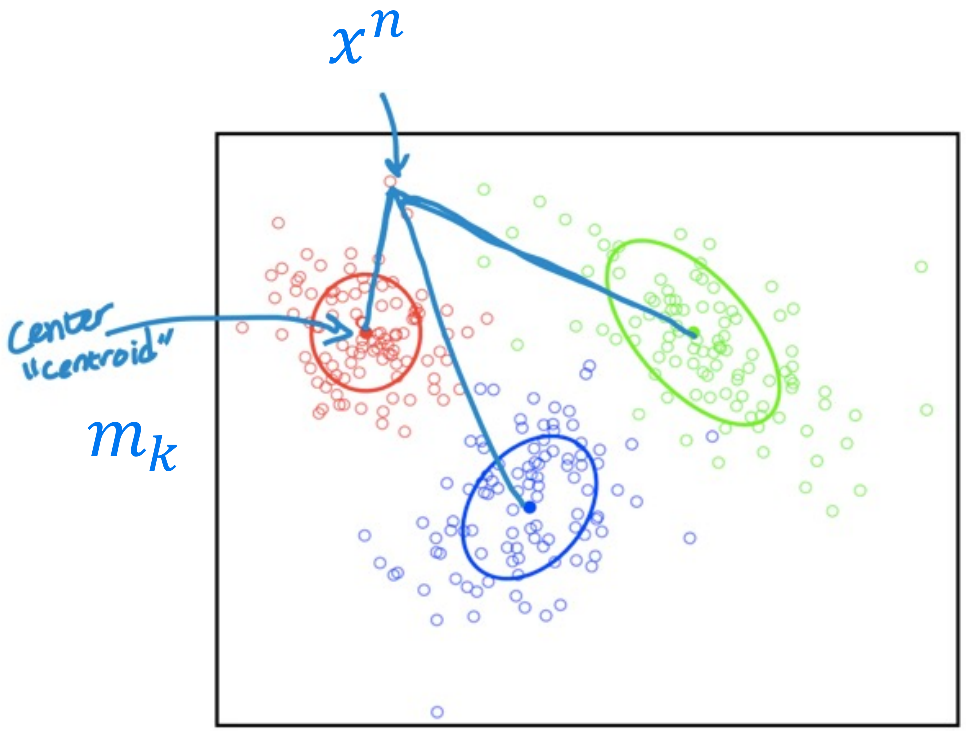
WORLD NEWS

Let’s first learn some basic things about clustering

What defines a cluster?

- Clusters are defined by a center and a spread (which you can also call shape)

- And we assign a given point x^n to a cluster by computing its similarity to the center of the cluster m_k



How can we define *similarity*?

Similarity

- Central to all of the goals of cluster analysis is the notion of the degree of similarity (or dissimilarity) between the individual objects being clustered.

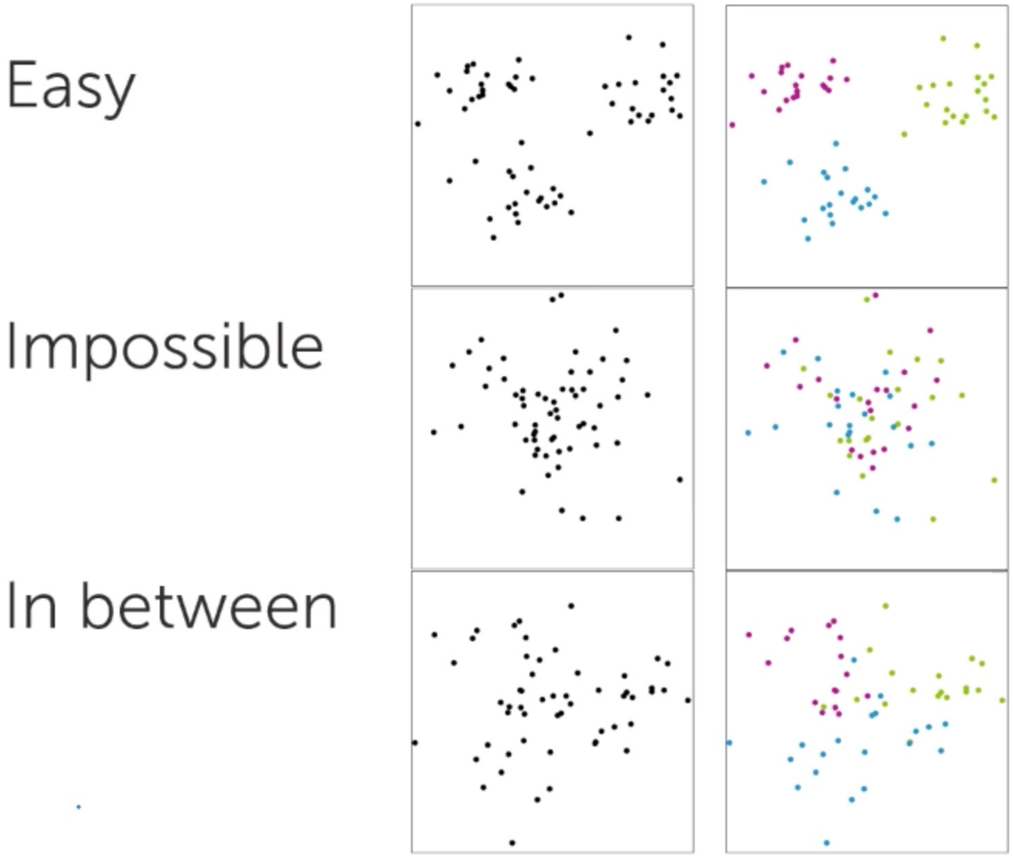
- A clustering method attempts to group the objects based on the definition of similarity (which is usually measured in the form of a distance from the center of a cluster) supplied to it.

Multiple Choices For Measuring Dissimilarity

- *Euclidean distance (the most common)*
- Manhattan distance
- Correlation based distances
  - Pearson correlation distance
  - Eisen cosine correlation distance
  - Spearman correlation distance
  - Kendall correlation distance
- Etc.

Read about them yourself. Materials are easily available online.

Finding Clusters is not Always “Easy” or “Difficult”



Bottom line is: clustering can be easy, hard, or in between depending on the structure of the data

Let’s Formulate Clustering as an Optimization Problem

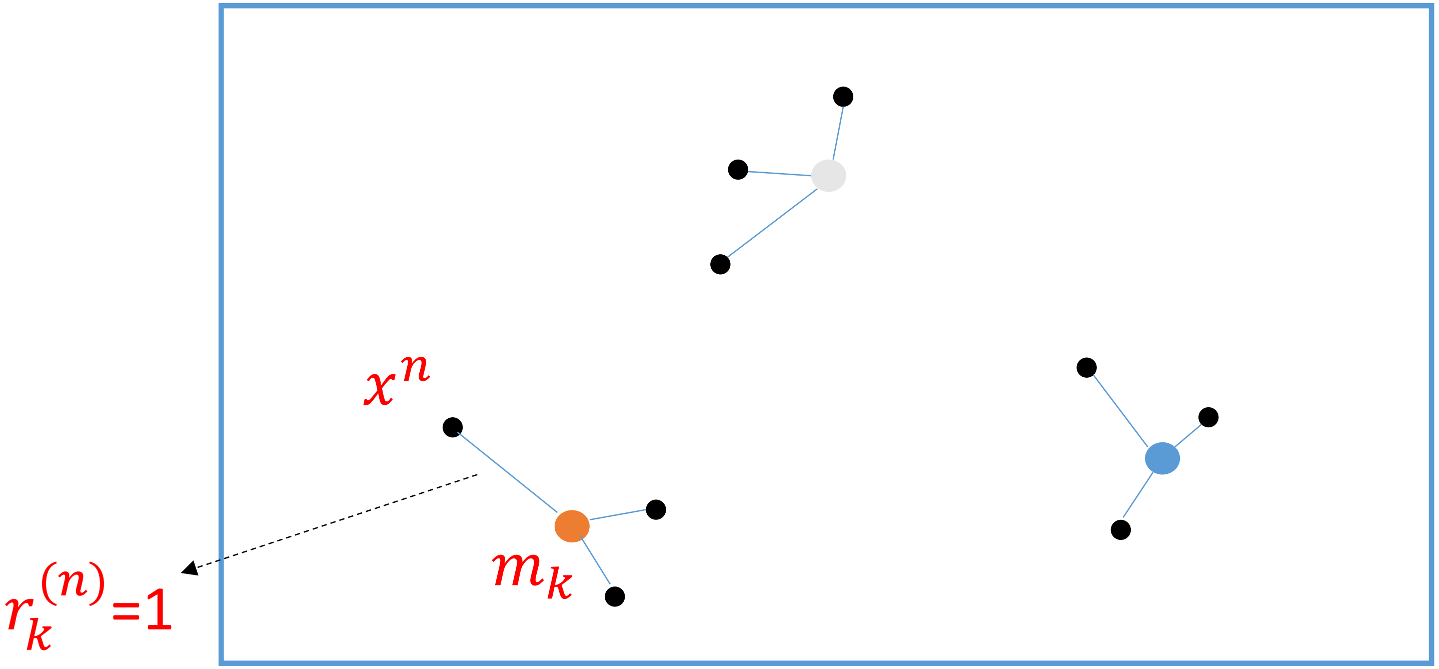
Objective

- Find cluster centers $\{\boldsymbol{m}_k\}_{k=1}^{K}$ and assignments $\{\boldsymbol{r}^{(n)}\}_{n=1}^{N}$ to minimize the sum of squared distances of data points $\{\boldsymbol{x}^{(n)}\}$ to their assigned centers

$\boldsymbol{x}^{(n)} \in \mathbb{R}^D$

$\boldsymbol{m}_k \in \mathbb{R}^D$

$\boldsymbol{r}^{(n)} \in \mathbb{R}^K$



$\boldsymbol{r}^{(n)} = [0, \cdots, 1, \cdots, 0]^T$

Mathematically

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} J(\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}) = \min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$

where $r_k^{(n)} = \mathbb{I}[\mathbf{x}^{(n)} \text{ is assigned to cluster } k]$, i.e., $\mathbf{r}^{(n)} = [0, .., 1, .., 0]^\top$

How to Solve the Optimization Problem?

- Optimization problem:

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$

- Problem is hard when minimizing jointly

- But becomes easy when we fix one and minimize over the other

How to Solve the Optimization Problem? (2)

- That is:

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$

  - If we fix the centers, then we can easily find the optimal cluster assignments by assigning each point to the cluster with the nearest neighbor

$$r_k^{(n)} = \begin{cases} 1 & \text{if } k = \arg\min_j ||\mathbf{x}^{(n)} - \mathbf{m}_j||^2 \\ 0 & \text{otherwise} \end{cases}$$

How to Solve the Optimization Problem? (3)

- Likewise:
  - If we fix the assignments then we can easily find the optimal cluster centers by setting each cluster’s centers to the average of its assigned data points

How to Solve the Optimization Problem? (4)

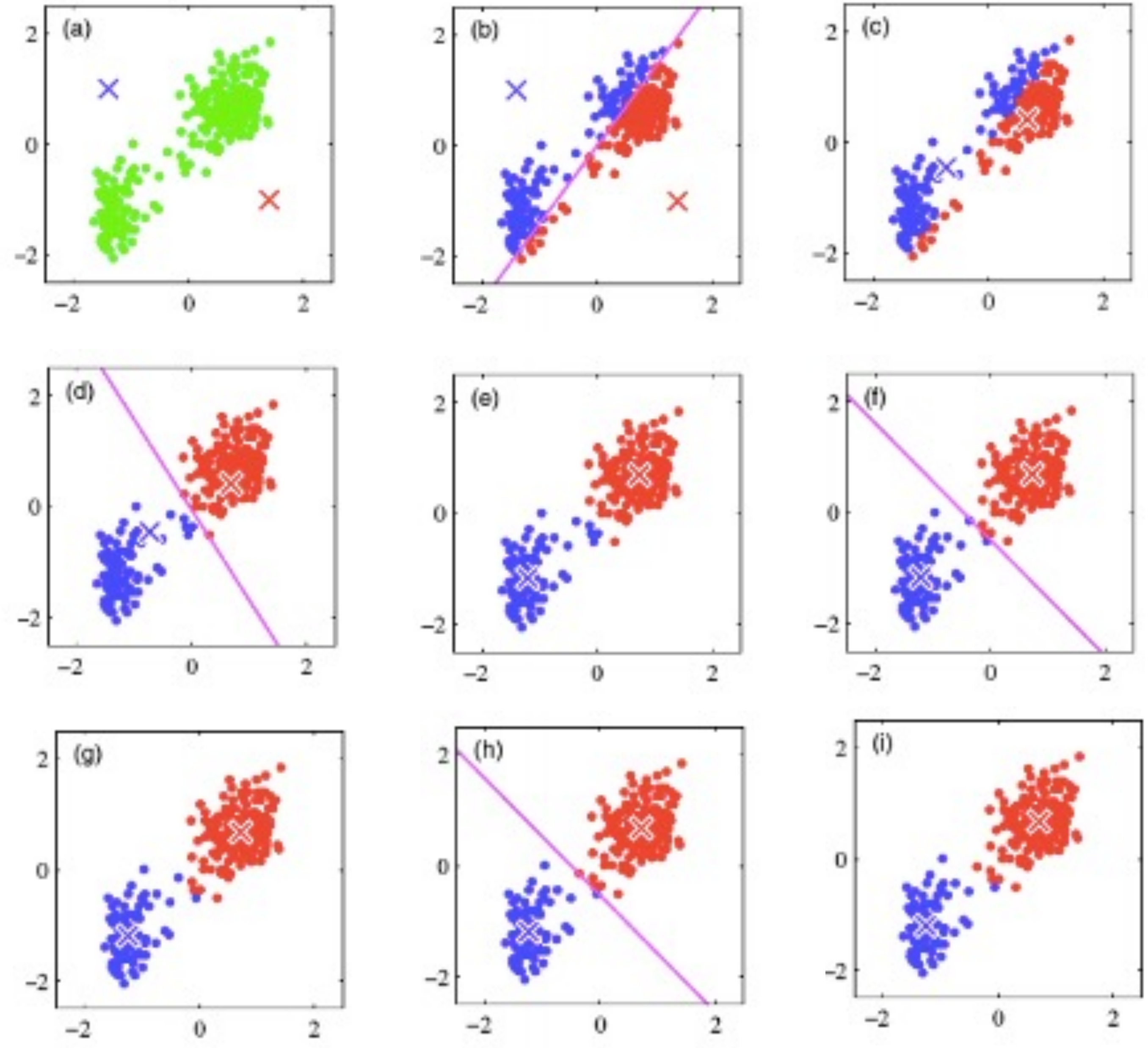
- Thus

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$

  - We solve this optimization problem by alternating between minimizing $J(\{\boldsymbol{m}_k\},\{\boldsymbol{r}^{(n)}\})$ with respected to $\{\boldsymbol{m}_k\}$ and $\{\boldsymbol{r}^{(n)}\}$
  - This is called alternating minimization

This is called k-means Clustering

Example



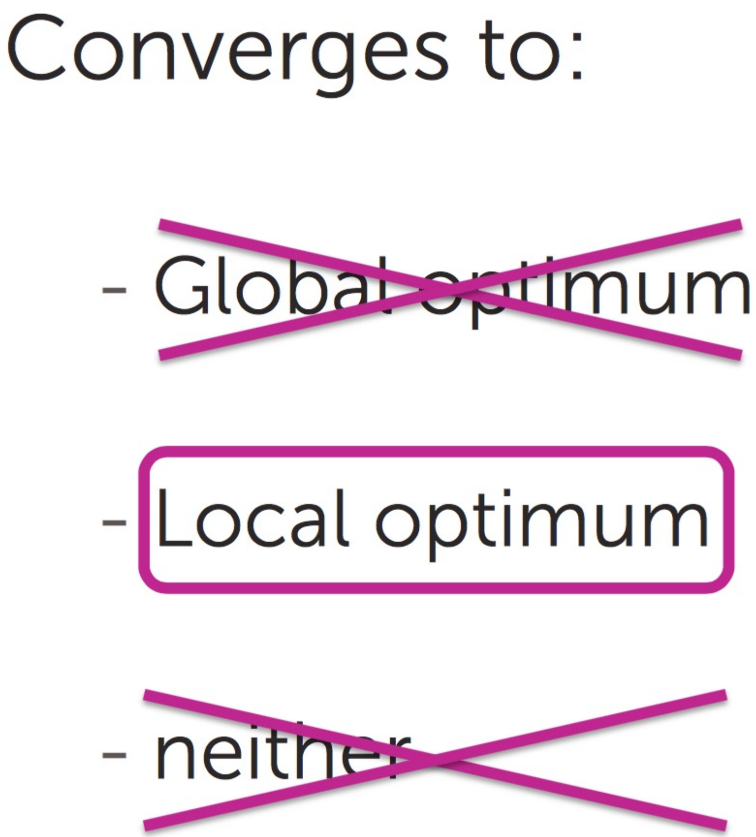
K-means Clustering

- Simply works as follows

  - Initialization: randomly initialize cluster centers
  - The algorithm then iteratively alternates between the following two steps

    - Assignment Step: assign each data point to the closest center
    - Refitting Step: move each cluster center to the mean of the data points assigned to it

Convergence of K-means

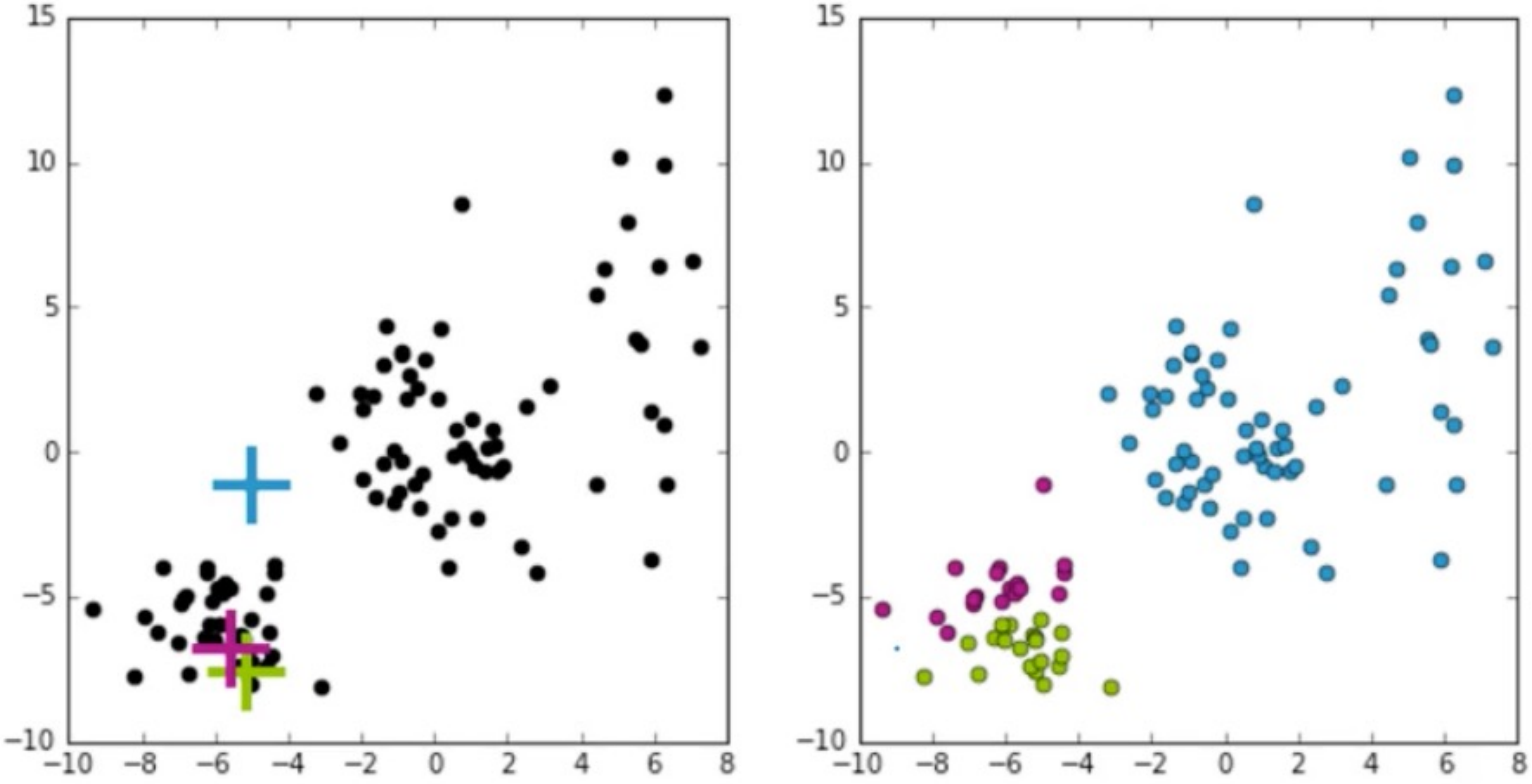


Converges to:

- ~~Global optimum~~
- Local optimum
- ~~neither~~

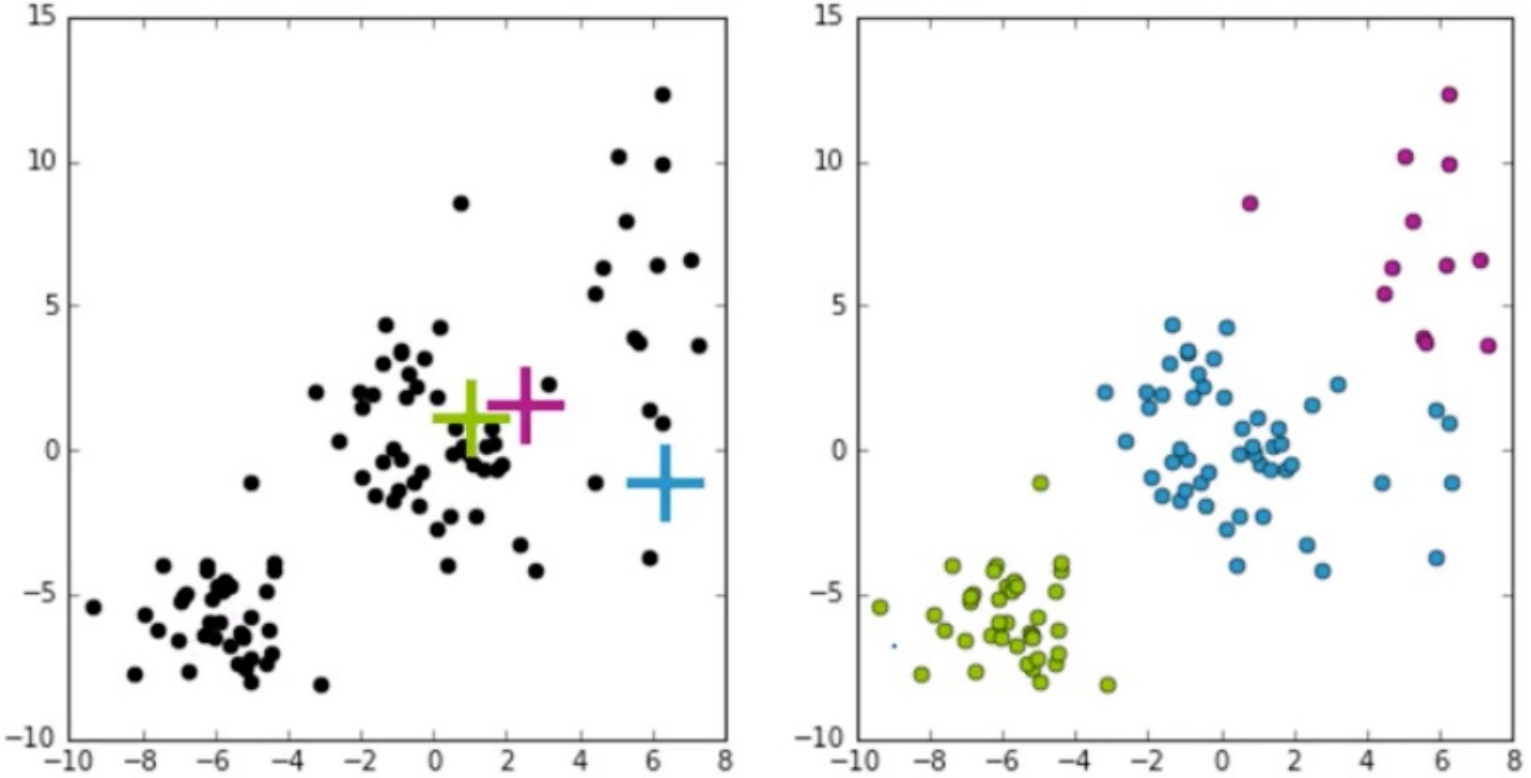
The first one does not mean that it can never converge to global optimum, it is just that there is no guarantee that it will.

Initialization Matters



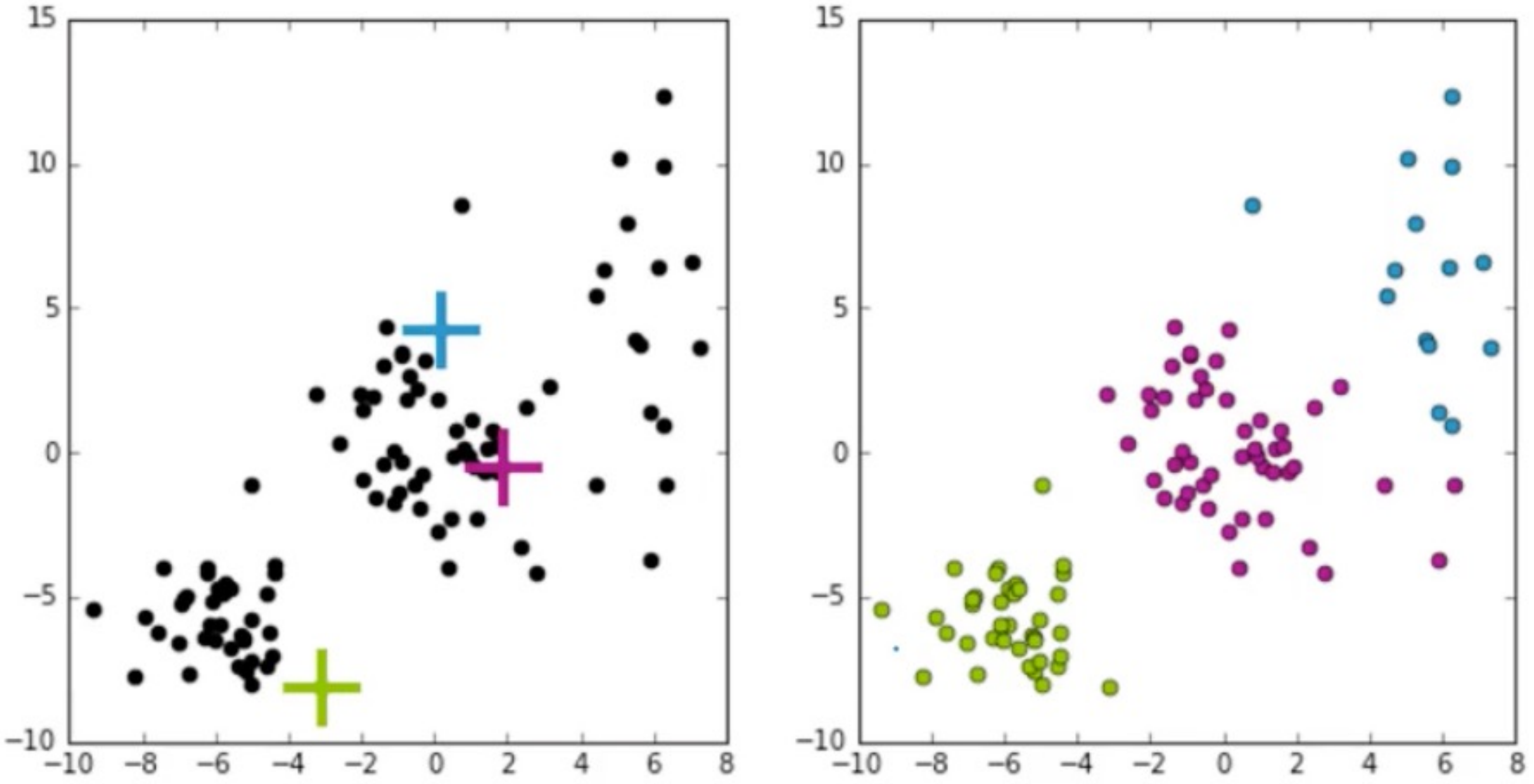
Centroids are chosen as shown in the first figure

Initialization Matters (2)



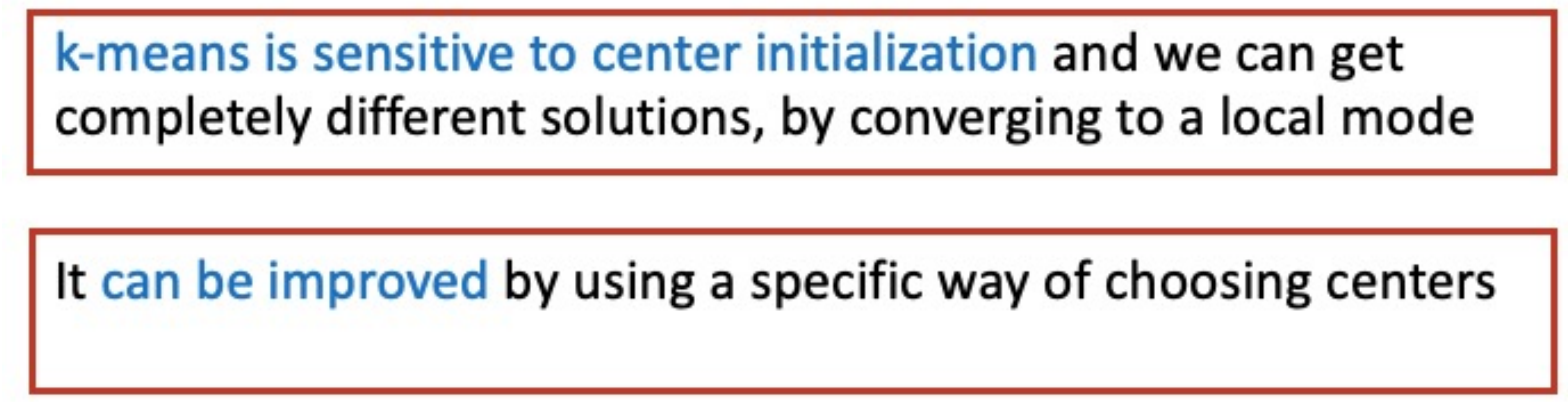
Different Initialization

Initialization Matters (3)



An even different initialization

Bottom-line



k-means is sensitive to center initialization and we can get completely different solutions, by converging to a local mode

It can be improved by using a specific way of choosing centers

K-means++

K-means++

- The intuition behind this approach is that spreading out the *k* initial cluster centers is a good thing

- The first cluster center is chosen uniformly at random from the data points that are being clustered

- After which each subsequent cluster center is chosen from the remaining data points with probability proportional to its squared distance from the existing cluster center.

How does K-means++ work?

- Let $D(x)$ denote the shortest distance from a data point to the closest center we have already chosen

- Then k-means ++ works as follows

1. Take the first center $\boldsymbol{m}_1$, chosen uniformly at random from $\{\boldsymbol{x}^{(n)}\}$
2. Take the next center $\boldsymbol{m}_k$, choosing $\boldsymbol{x}^{(n)}$ with probability $\frac{D(x)^2}{\sum_{x^{(n)} \in X} D(x)^2}$
3. Repeat step 2 until we have chosen K cluster centers

4. Proceed with the standard k-means algorithm

K-means++ Summary

- Smart initialization is computationally costly relative to random initialization

- However, due to smart initialization, the subsequent k-means often converges more rapidly

- Tends to improve the quality of the local optimum

How to choose K?

K-means Objective

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \sum_{n=1}^{N} \sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$

- Think about what will happen if we choose K to be some large value, like $K = N$

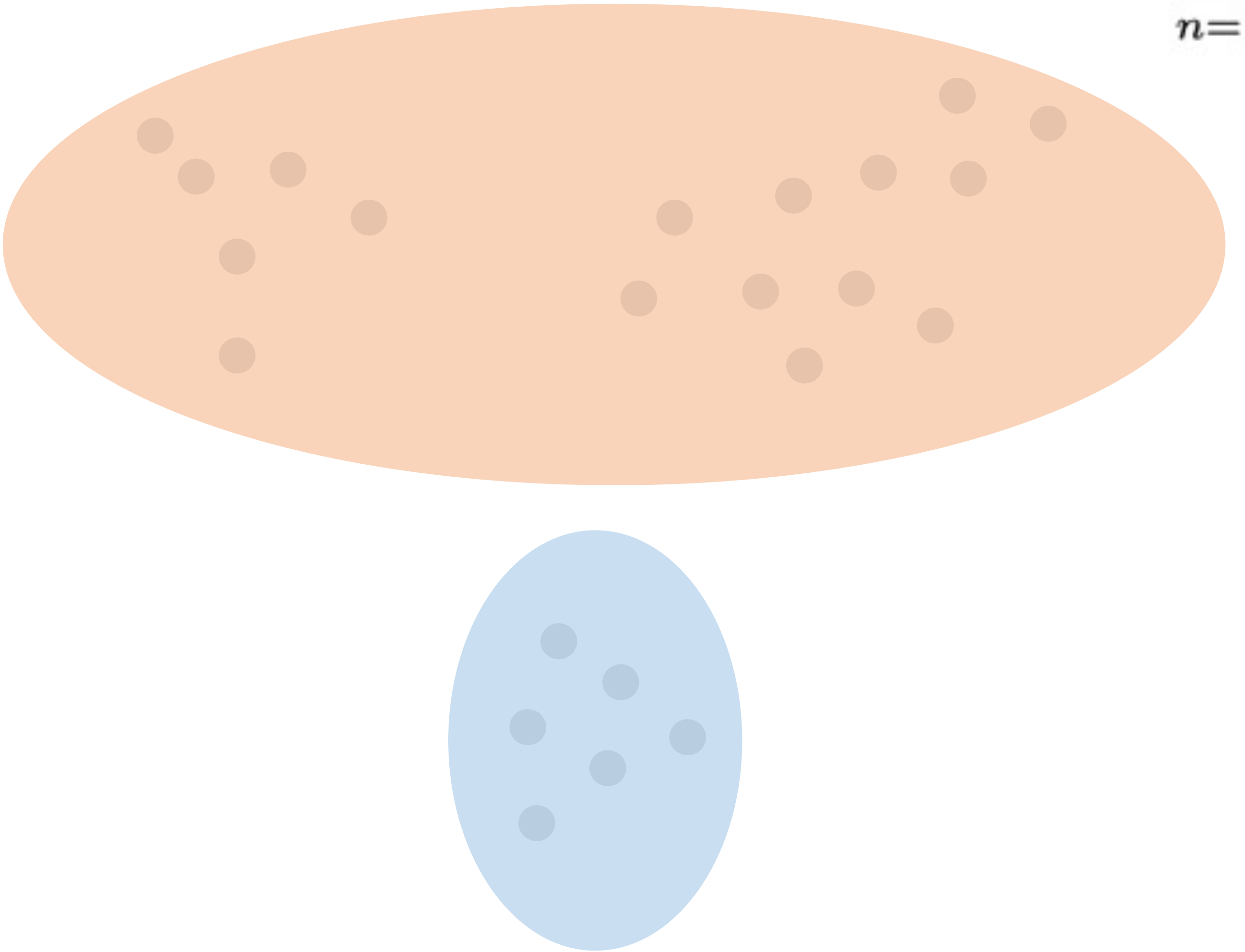
- The above value will converge to ZERO – But would it be an optimum solution?

Overfitting in Clustering

- Remember, we are in unsupervised learning.
  - In this case, overfitting would mean that we did not learn the true structures in the data, instead we ended up memorizing the position of the sample in the space.

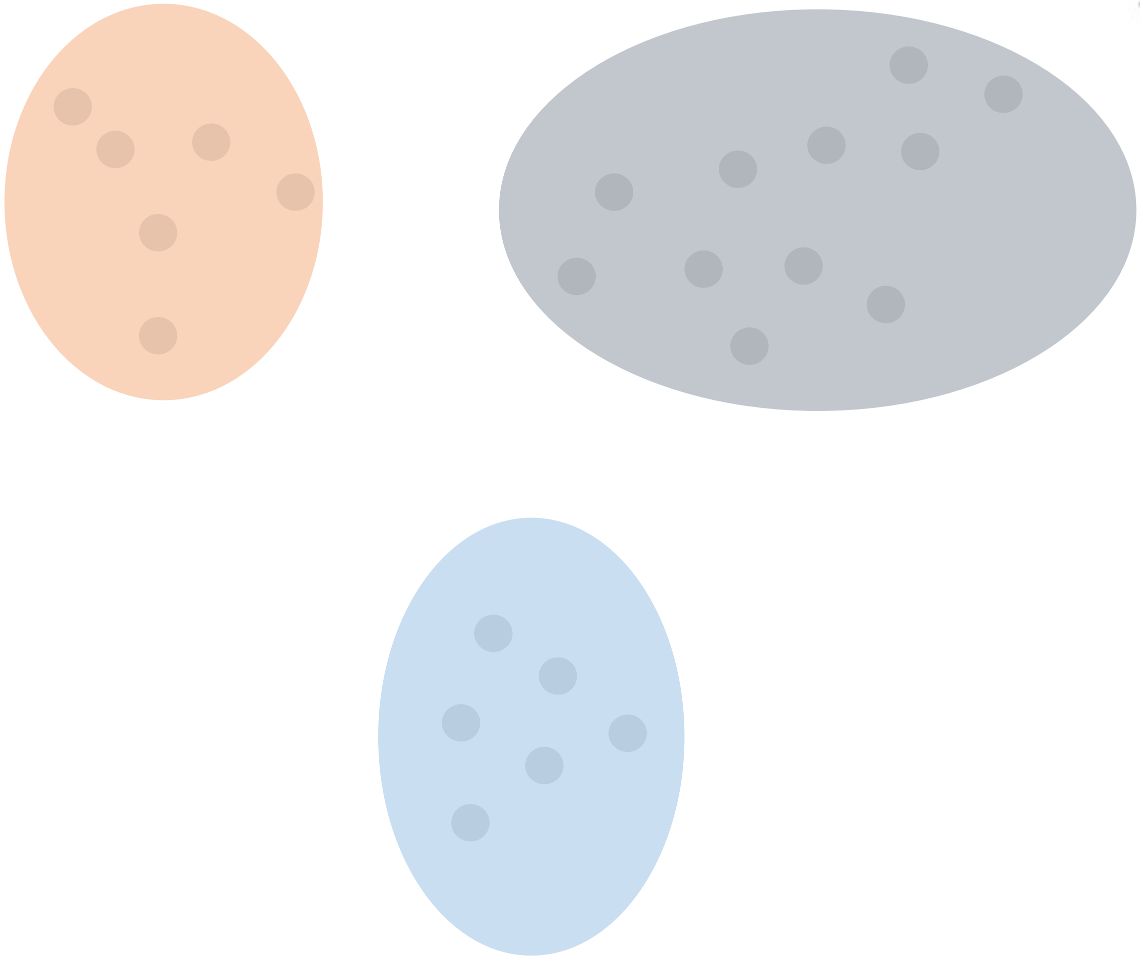
Example (K=2)

$$\sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$



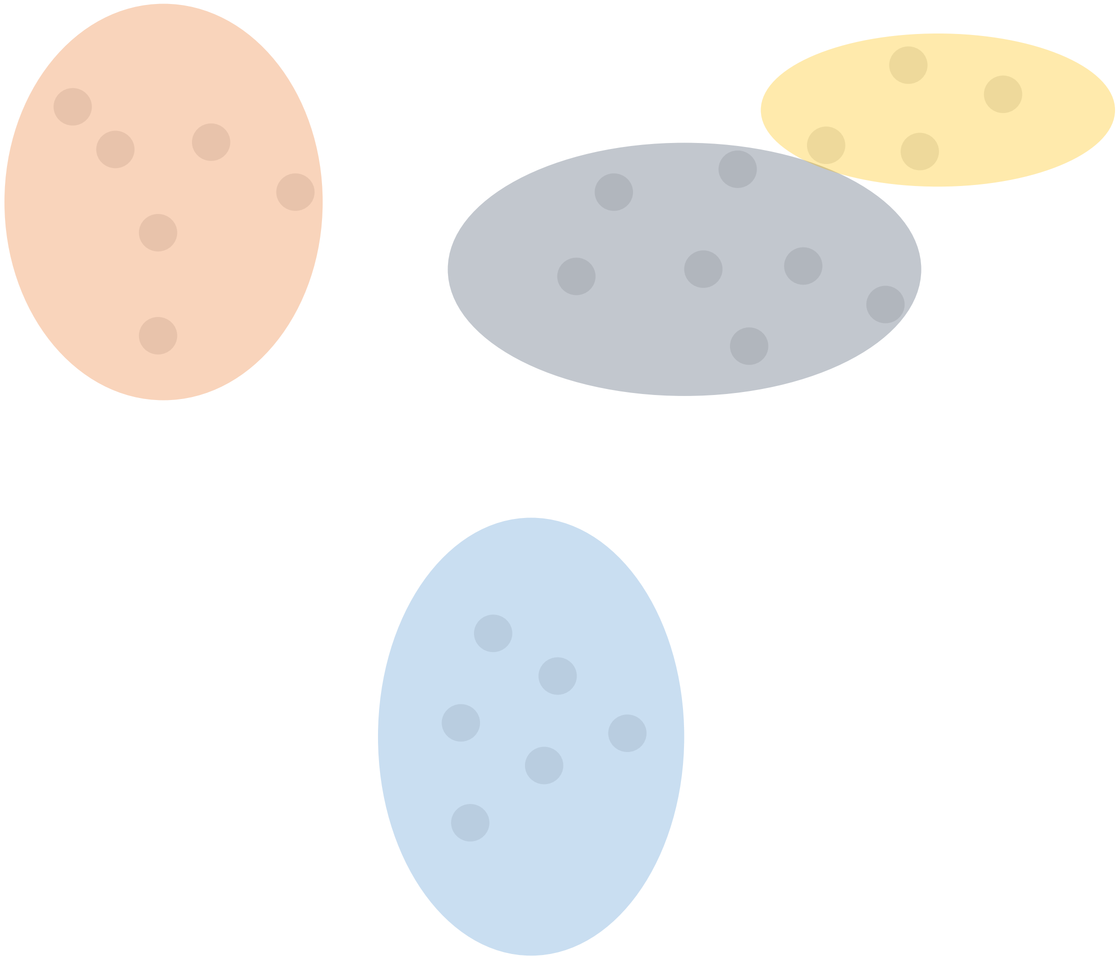
Example (K=3)

$$\sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$

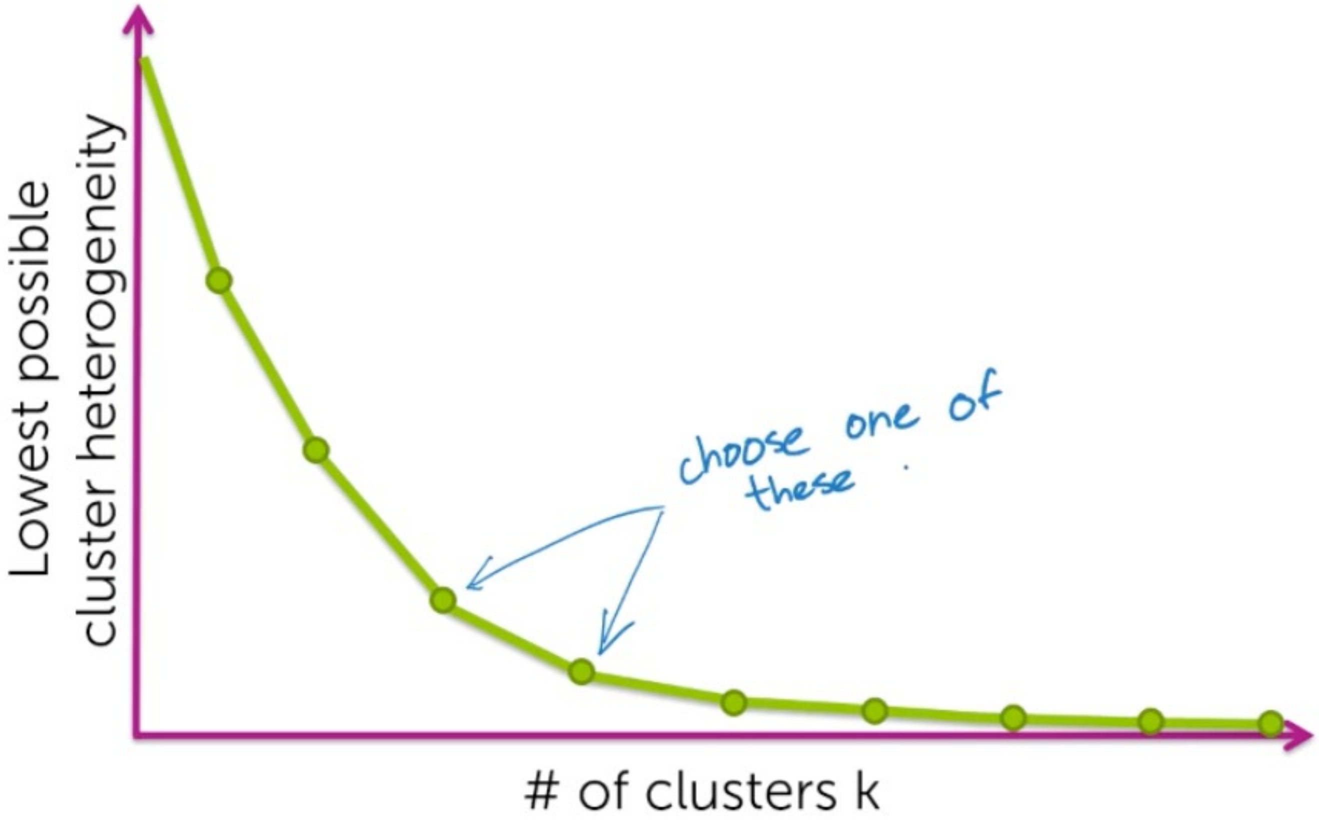


Example (K=4)

$$\sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$

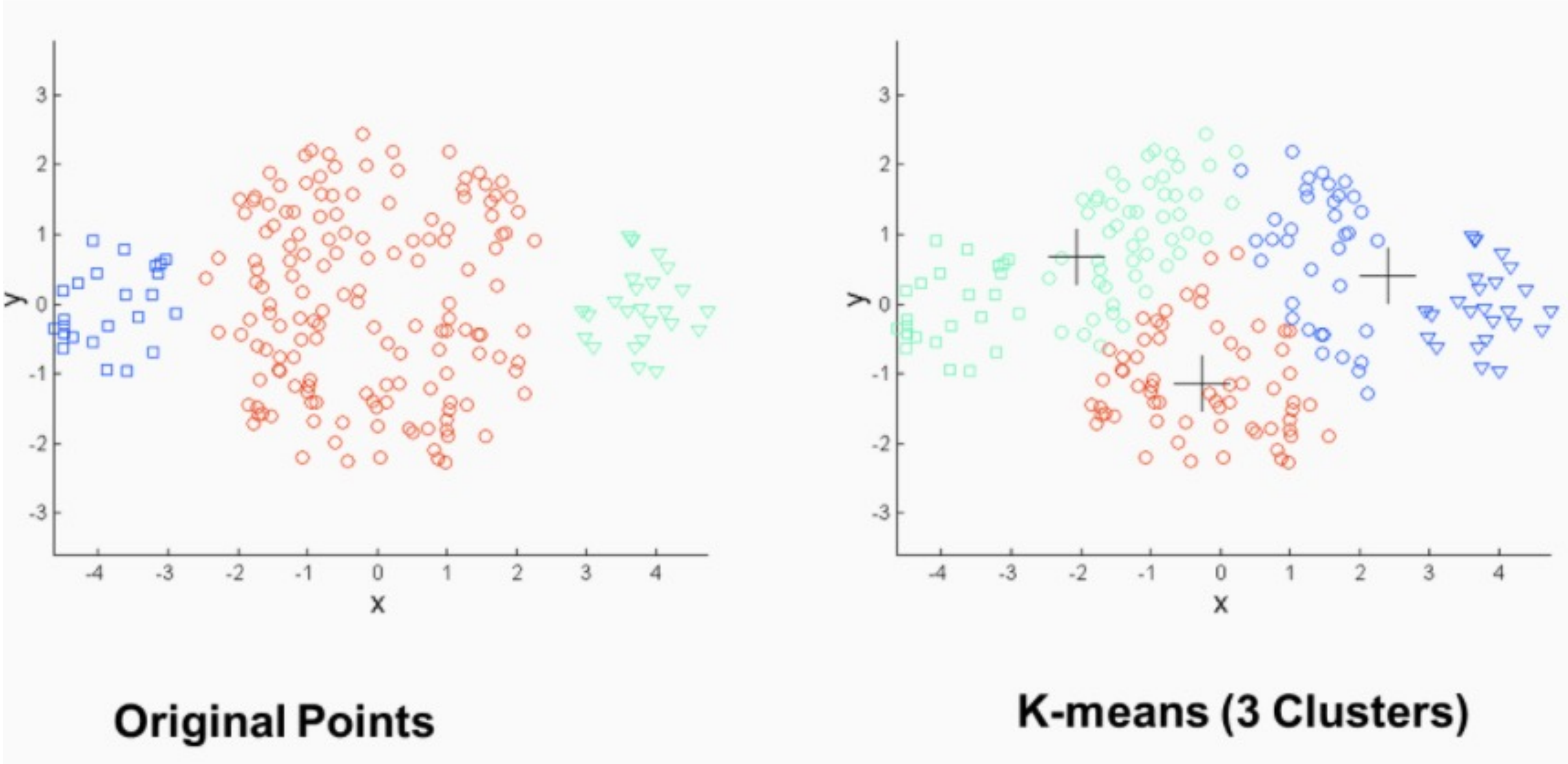


Choosing K – The Elbow Method



Elbow of the curve

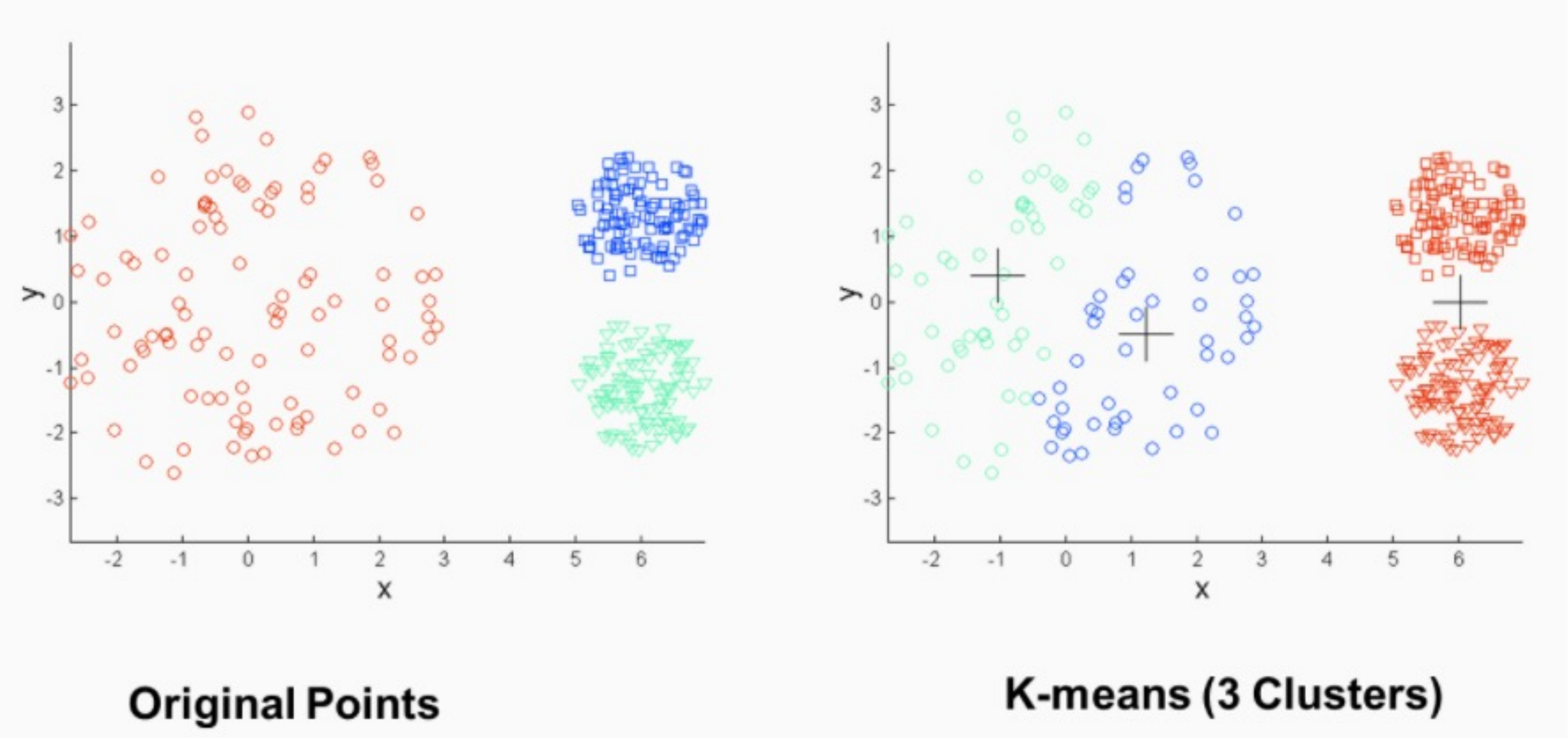
Issues with K-means (1)



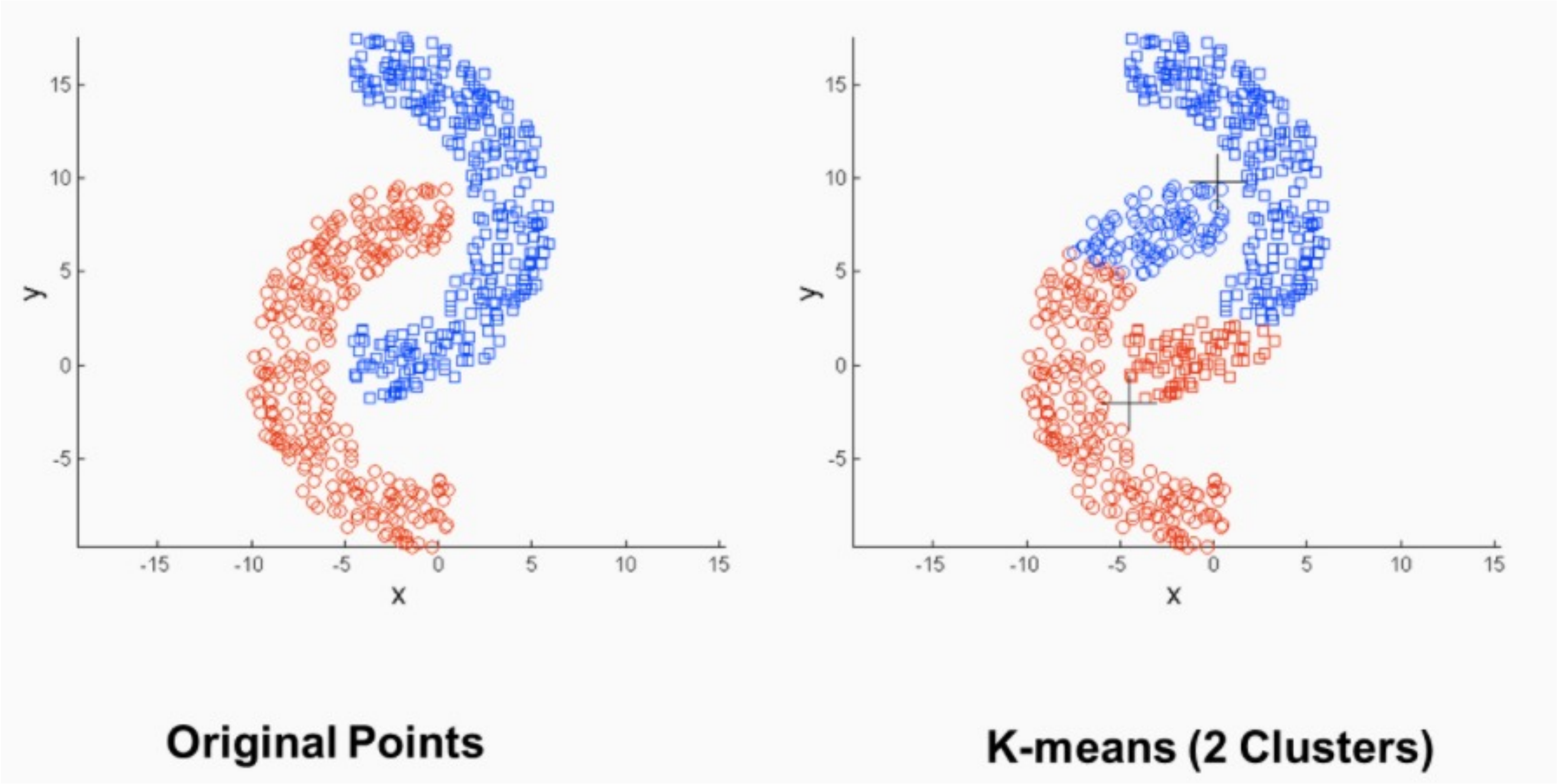
Original Points

K-means (3 Clusters)

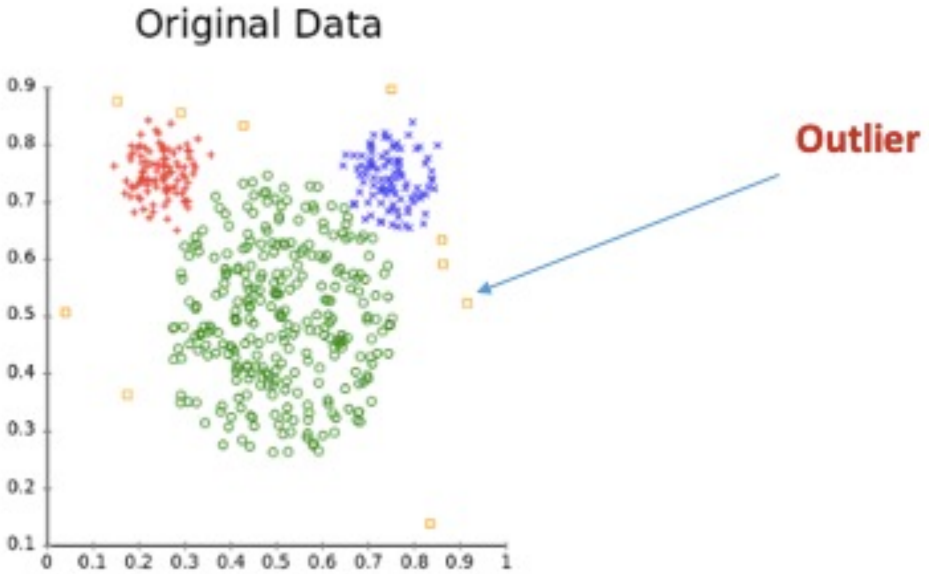
Issues with K-means (2)

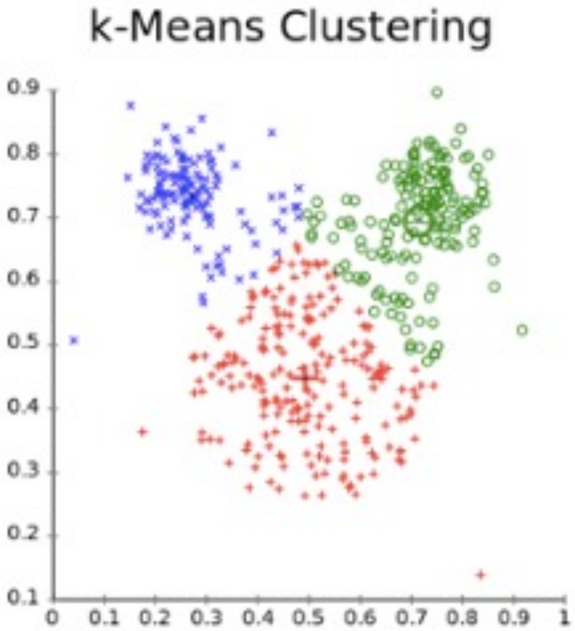


Issues with K-means (3)



Issues with K-means (4)





Issues with K-means

1. Not knowing the optimum value of K

2. Also, k-means clustering does not work well when

   - We want to discover clusters of varying sizes, densities and shapes

   - We do not want to include noisy points (outliers) into any clusters

Are there any Alternatives?

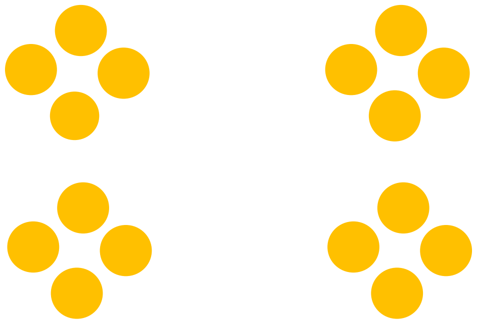
- Many

- But the two that I want to introduce to you are

  - Hierarchical Clustering
  - Density Based Clustering (DBSCAN)

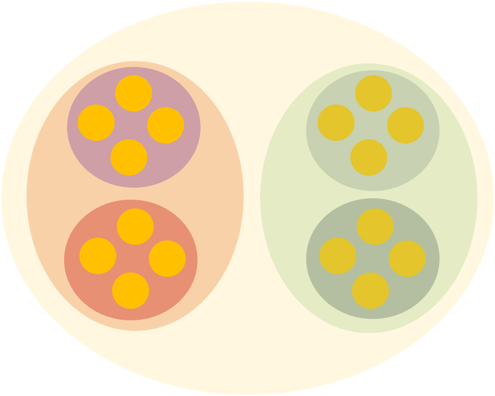
Hierarchical Clustering

How Many Clusters do You See?



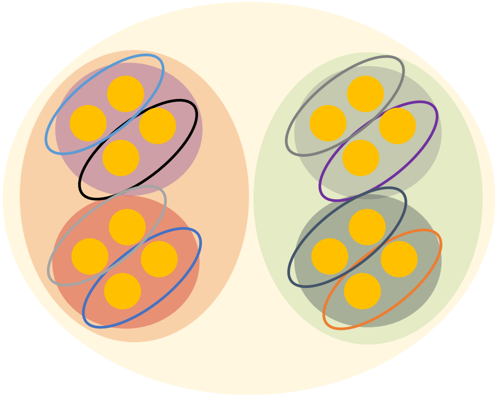
- It all depends on how you are looking at your data, even better – at what granularity do you want to look at your data
- Are you looking for high-level effect or fine-grained details?
- Thus instead of finding a certain number of clusters, we can change our objective – “find a hierarchy of structure”

Ways of Discovering Hierarchy



- Top-down or Divisive Clustering
  - Start with all items in one cluster,
  - Split recursively

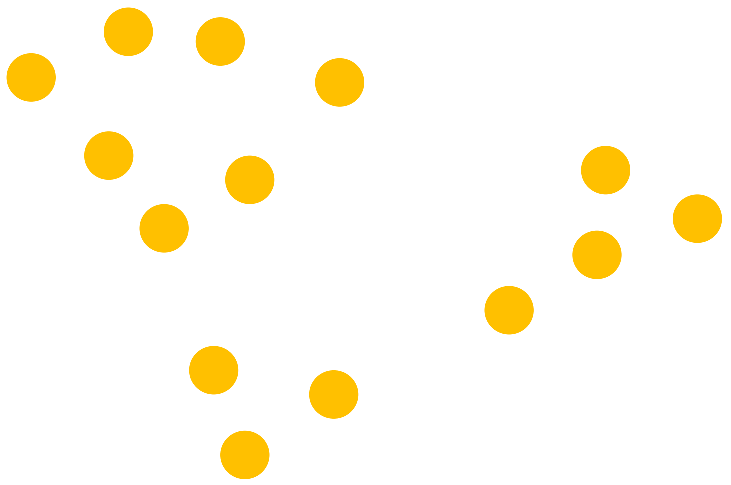
Ways of Discovering Hierarchy (2)



- Bottom-up or Agglomerative Clustering
  - Start with singeltons,
  - Merge clusters using some criteria

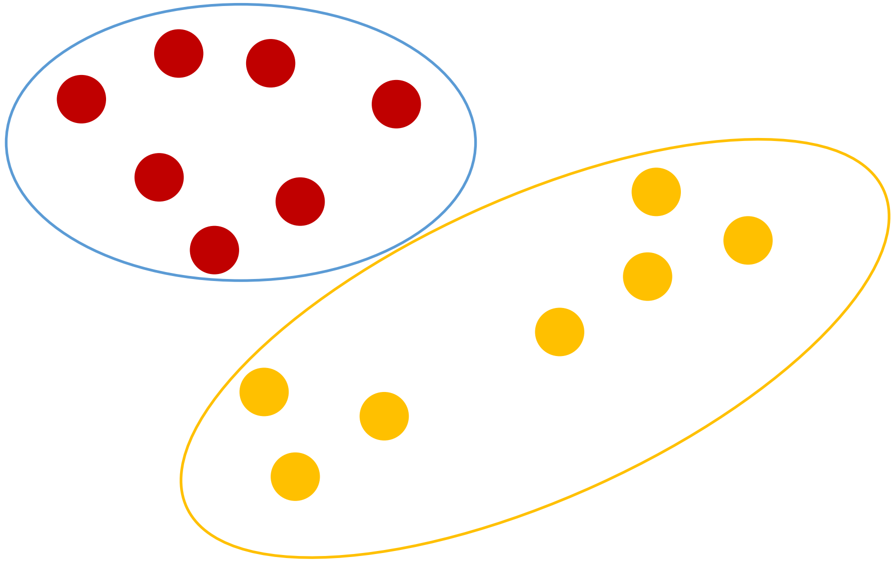
Top-down Hierarchical Clustering (2)

- Simple solution: run k-means recursively



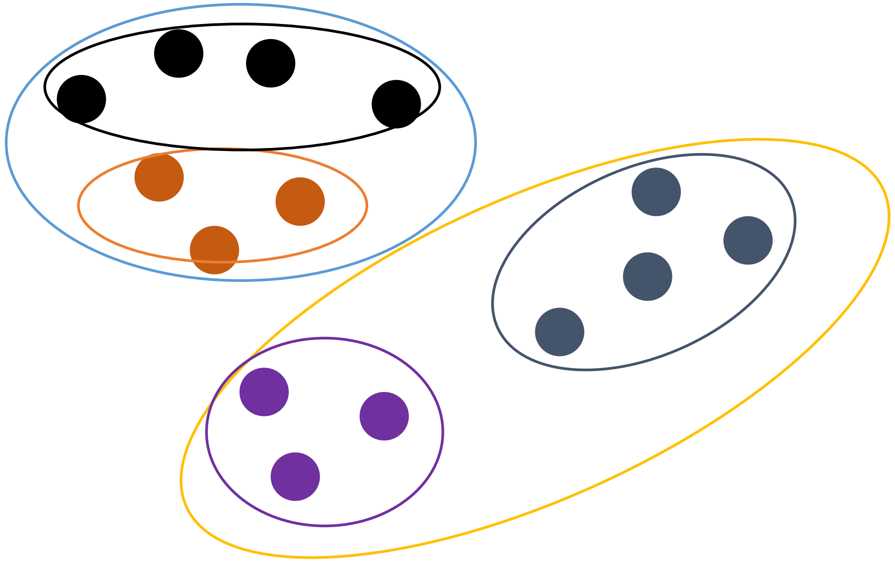
Top-down Hierarchical Clustering (3)

- Simple solution: run k-means recursively



Top-down Hierarchical Clustering (4)

- Simple solution: run k-means recursively



…

Bottom-up Hierarchical Clustering

- Agglomerative clustering algorithms begin with every observation representing a singleton cluster

- At each step the closest two (least dissimilar) clusters are merged into a single cluster

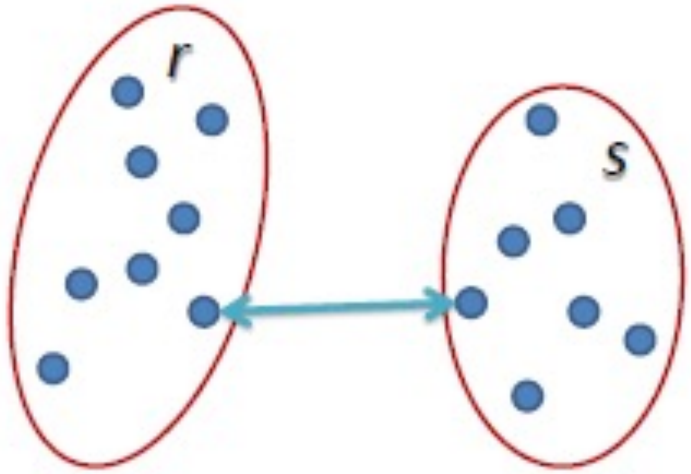
- Therefore, a measure of dissimilarity between two clusters (groups of observations) must be defined.

Distance Between Clusters

- Three most common choices are:

   1. Single linkage:

      Distance between the two most similar members of each cluster



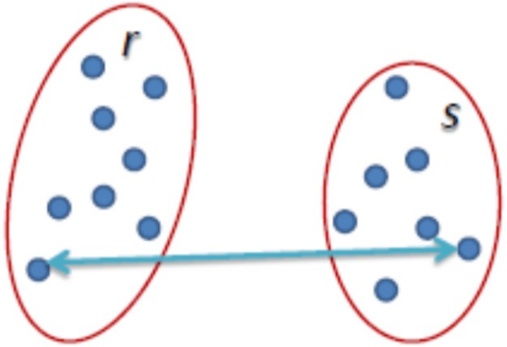
$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$

Distance Between Clusters (2)

- Three most common choices are:
  1. Single linkage:

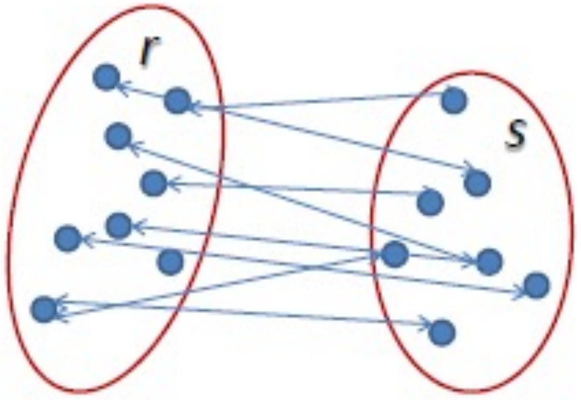
     Distance between the two most similar members of each cluster
  2. Complete linkage:

     Distance between the two most dissimilar members of each cluster



$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

Distance Between Clusters (3)



$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

- Three most common choices are:

  1. Single linkage:

     Distance between the two most similar members of

  2. Complete linkage:

     Distance between the two most dissimilar members

  3. Average linkage:

     Average distance between the members of each cluster

Distance Between Clusters (Summary)

- Three most common choices are:
  1. Single linkage:

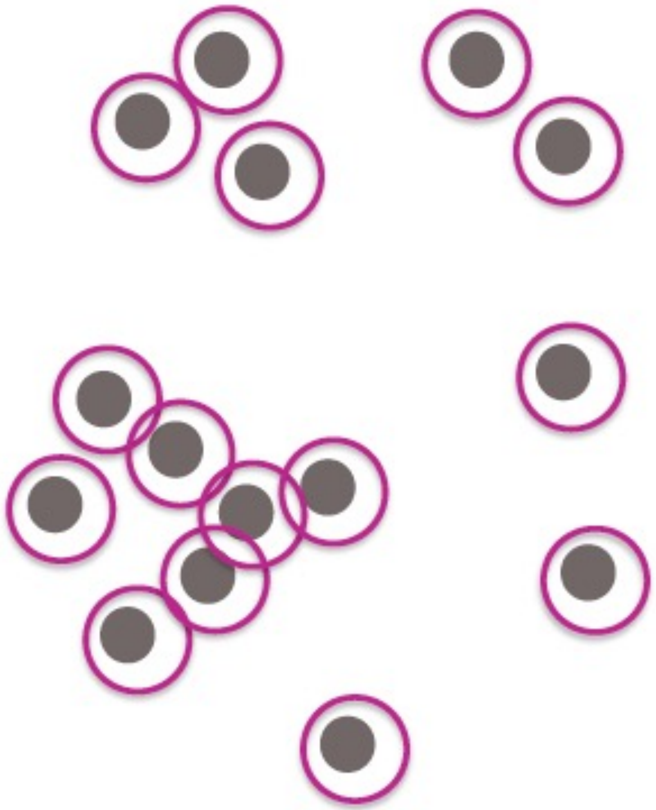
     Distance between the two most similar members of each cluster
  2. Complete linkage:

     Distance between the two most dissimilar members of each cluster
  3. Average linkage:

     Average distance between the members of each cluster

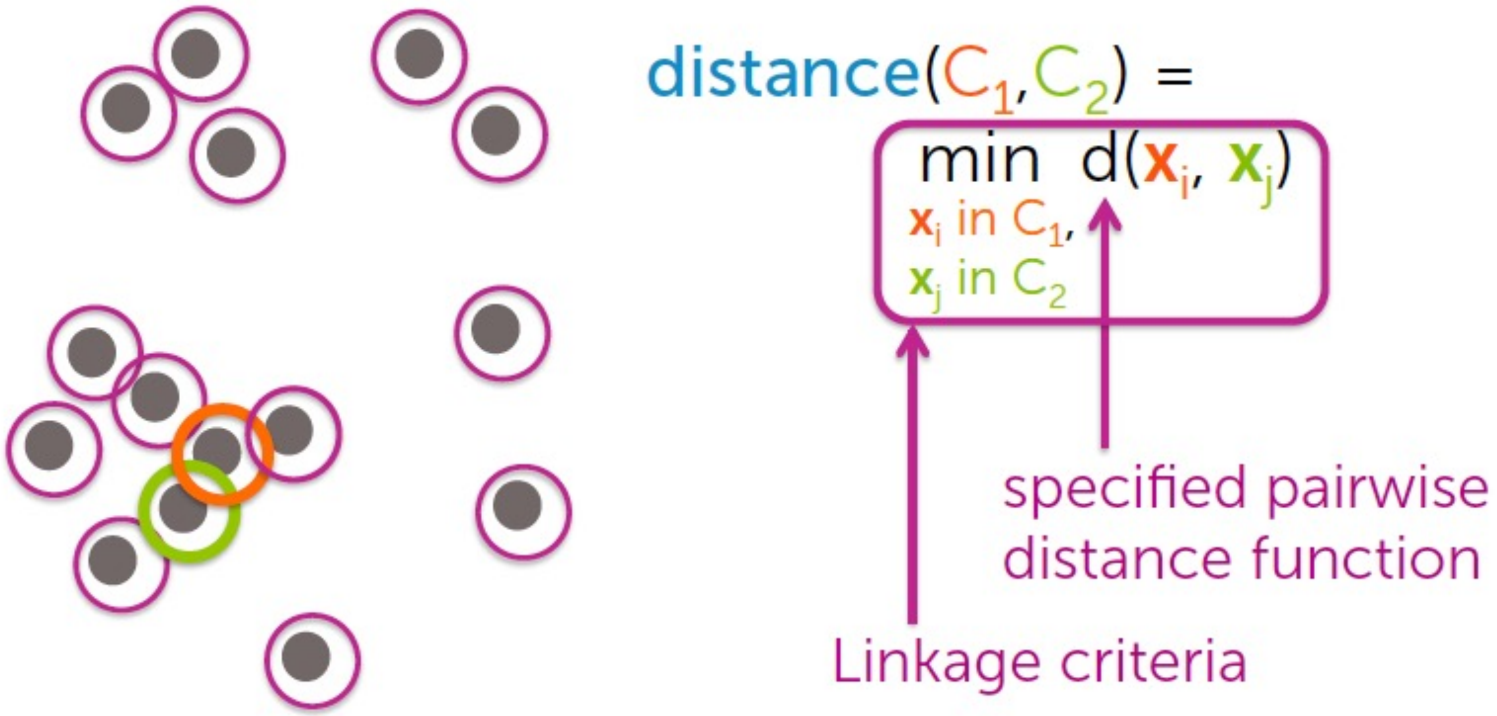
Agglomerative Clustering - Example

1. Make each data point a single cluster



Agglomerative Clustering - Example

2. Compute distance (Single Linkage) between clusters:



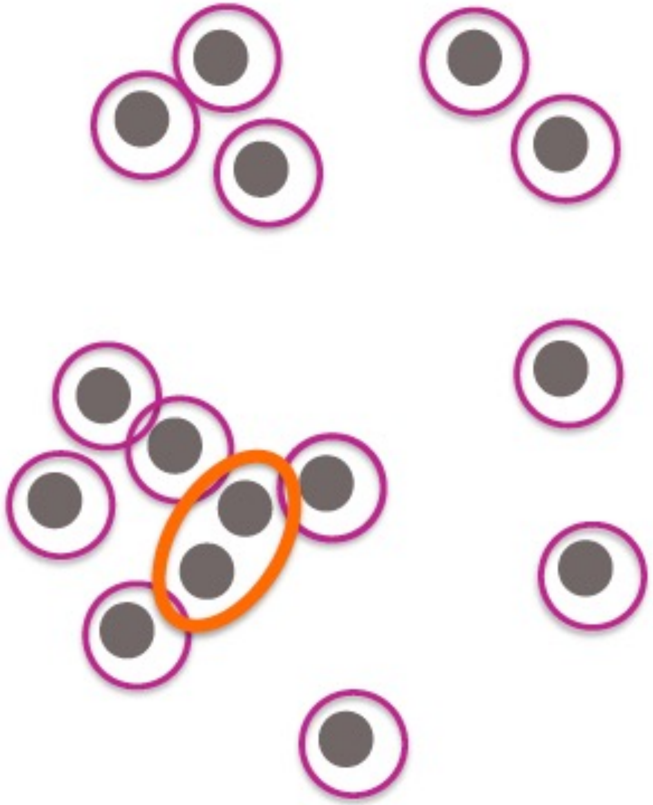
$$\text{distance}(C_1, C_2) = \min_{x_i \text{ in } C_1,\ x_j \text{ in } C_2} d(\mathbf{x}_i, \mathbf{x}_j)$$

specified pairwise distance function

Linkage criteria

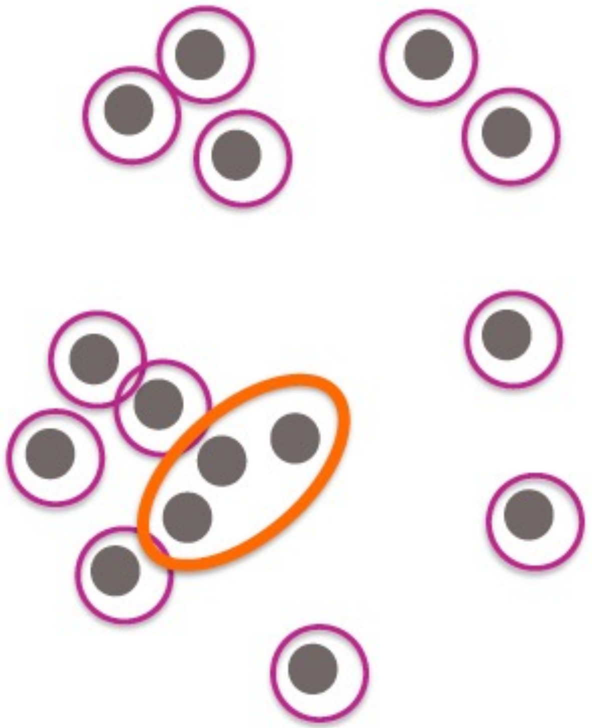
Agglomerative Clustering - Example

3. Merge the two closest clusters



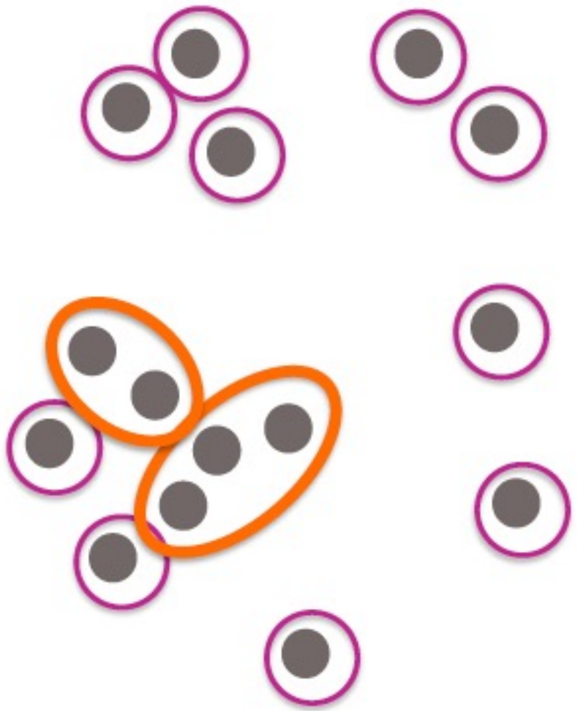
Agglomerative Clustering - Example

4. Repeat step (3) until all points are in one cluster



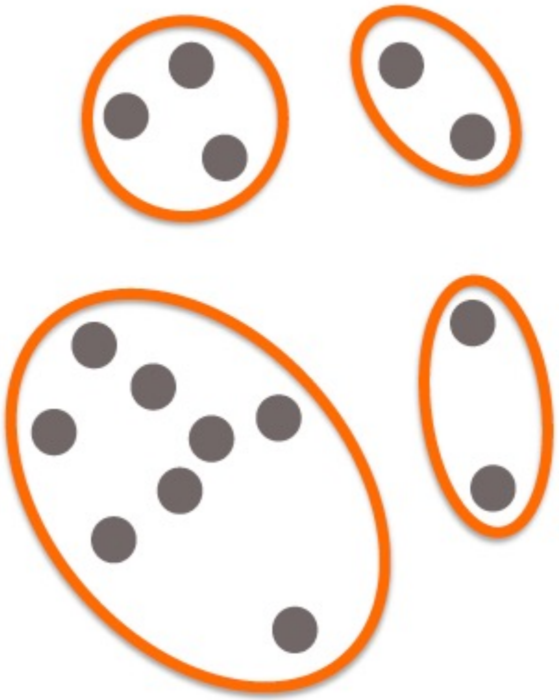
Agglomerative Clustering - Example

4. Repeat step (3) until all points are in one cluster



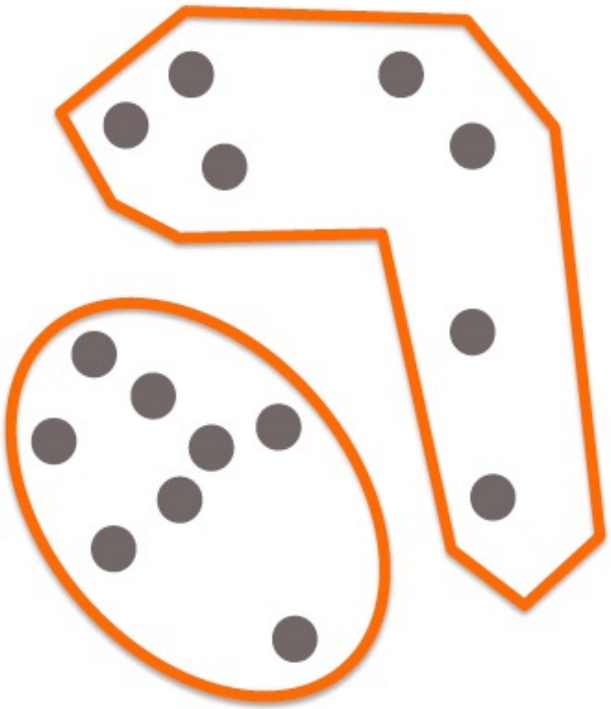
Agglomerative Clustering - Example

4. Repeat step (3) until all points are in one cluster



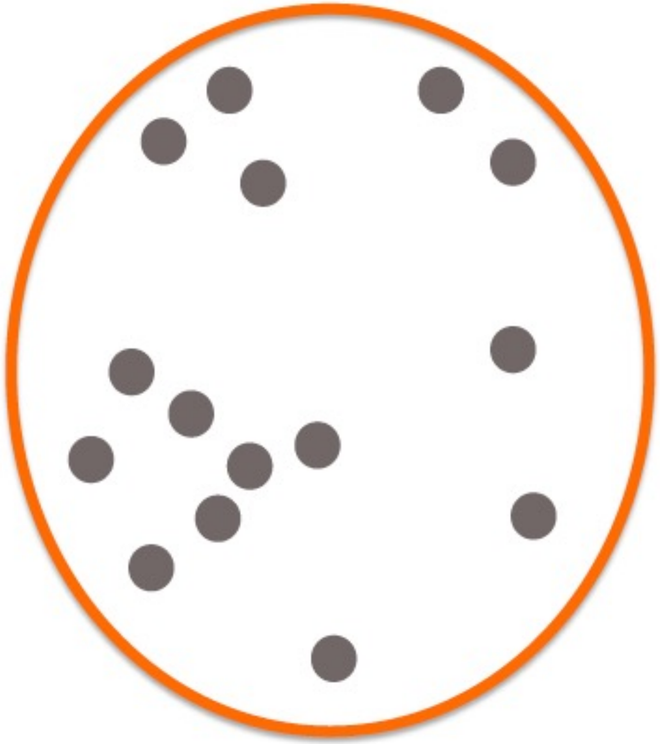
Agglomerative Clustering - Example

4. Repeat step (3) until all points are in one cluster



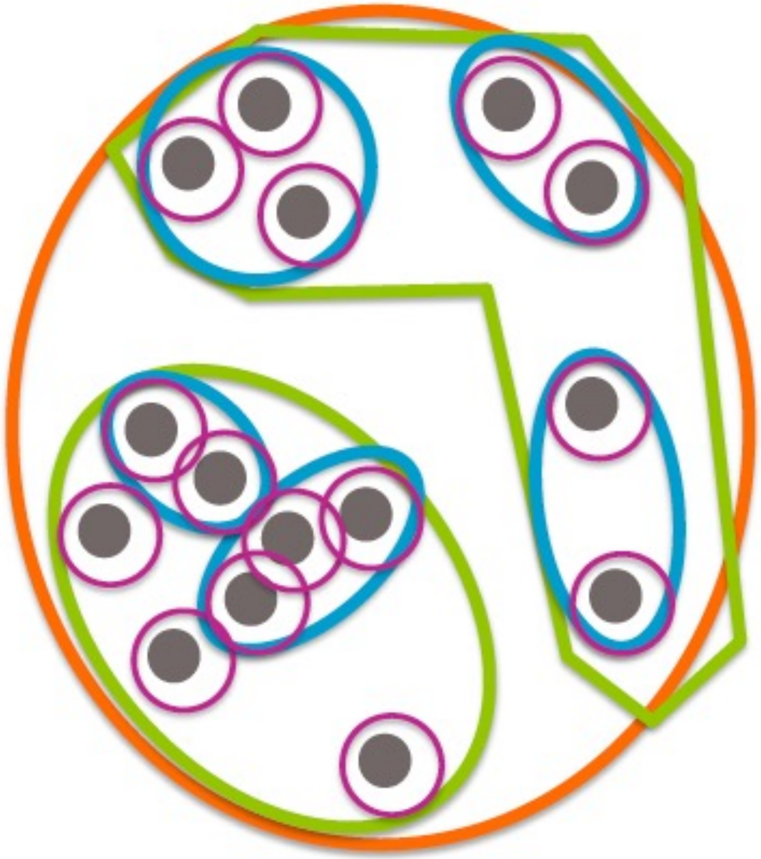
Agglomerative Clustering - Example

4. Repeat step (3) until all points are in one cluster



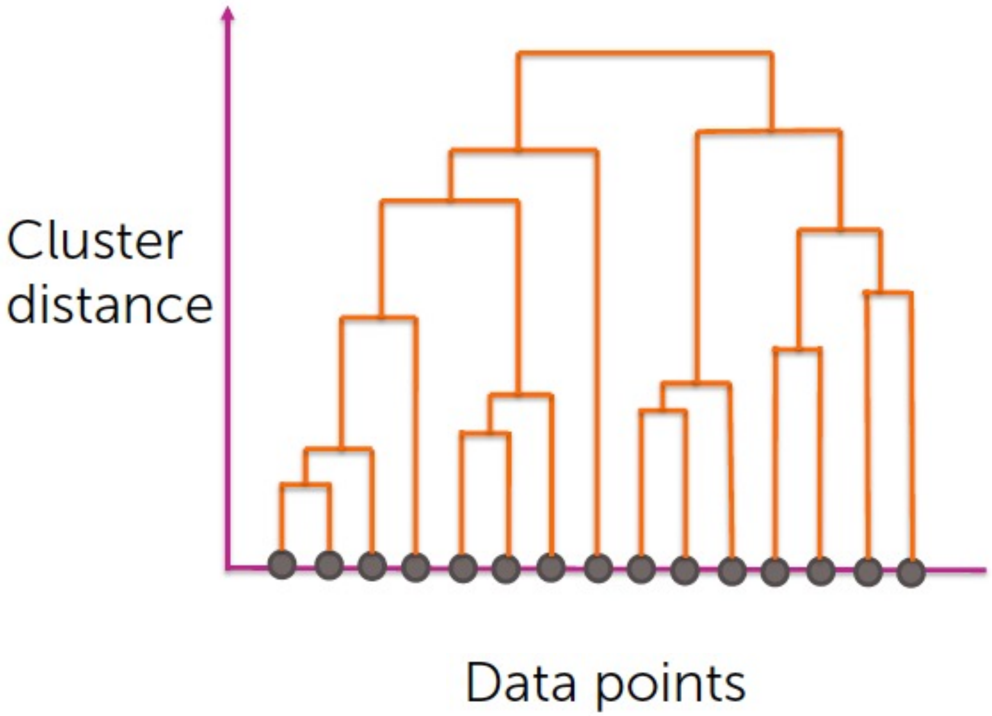
Agglomerative Clustering - Example

- Discovered Hierarchy

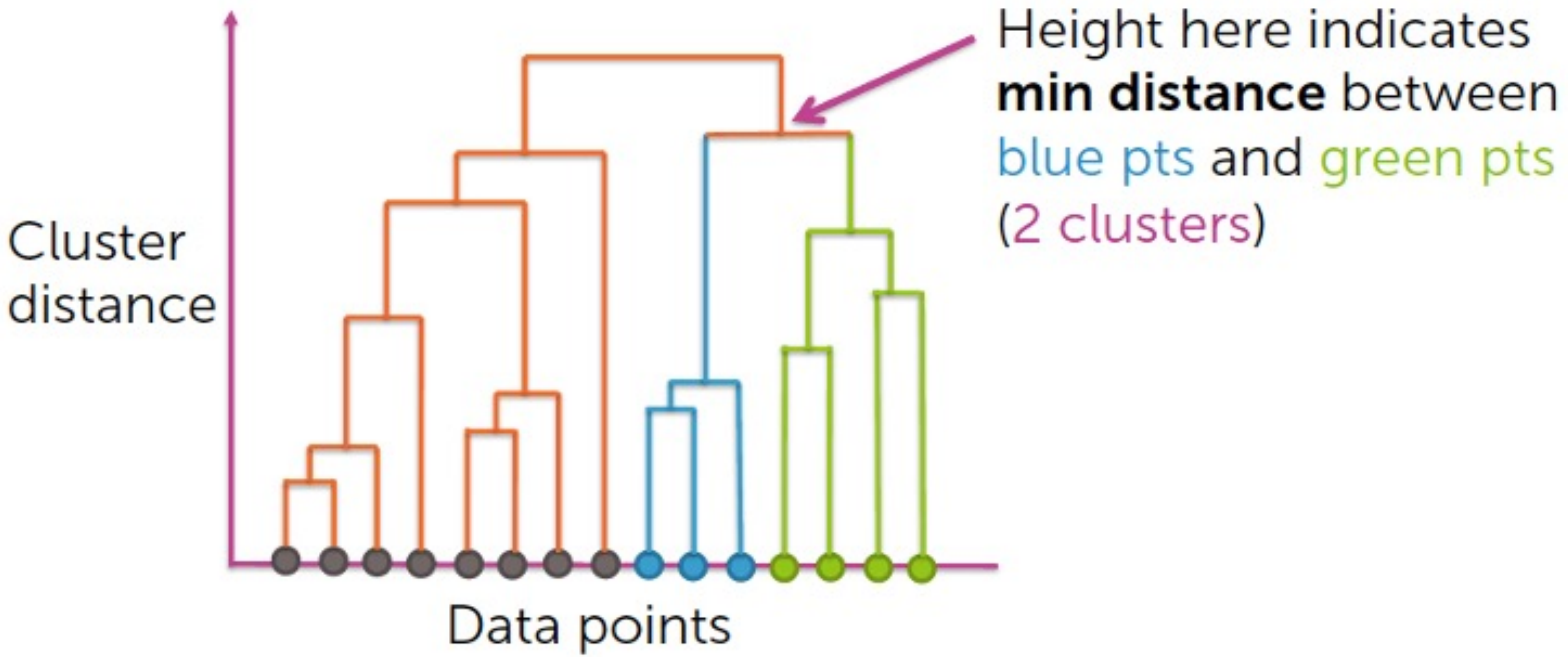


Hierarchy Can Be Viewed as a Tree

- Dendrogram
  - X axis shows data points (ordering is such that it makes the visualization better)
  - Y-axis shows distance between pair of clusters

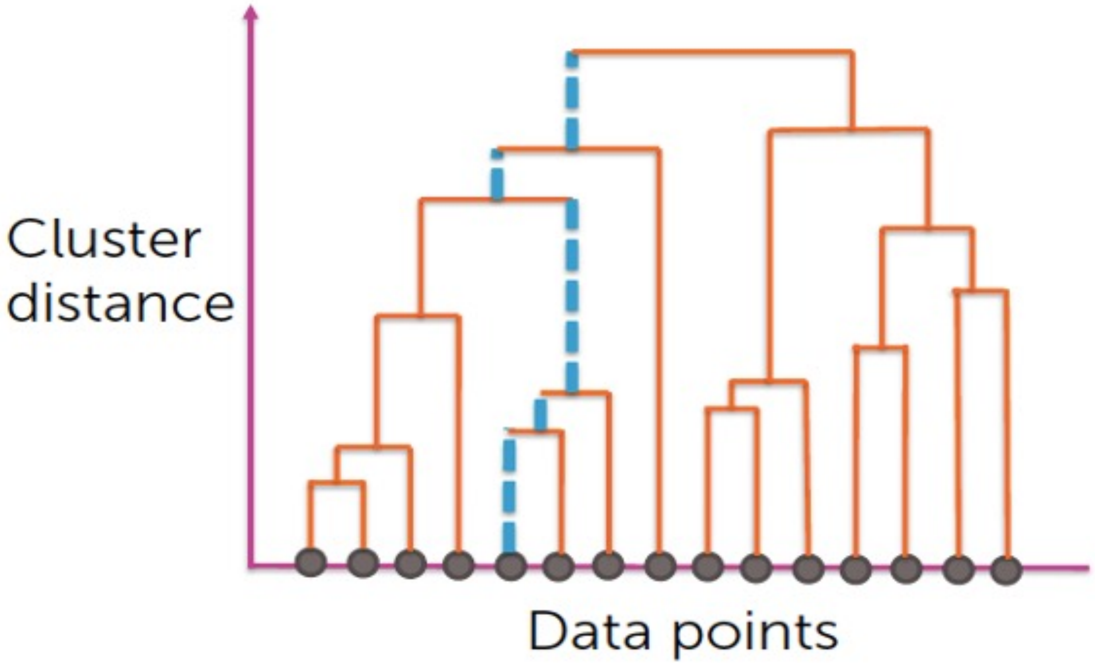


Dendrogram


Height here indicates
min distance between
blue pts and green pts
(2 clusters)
Cluster distance
Data points

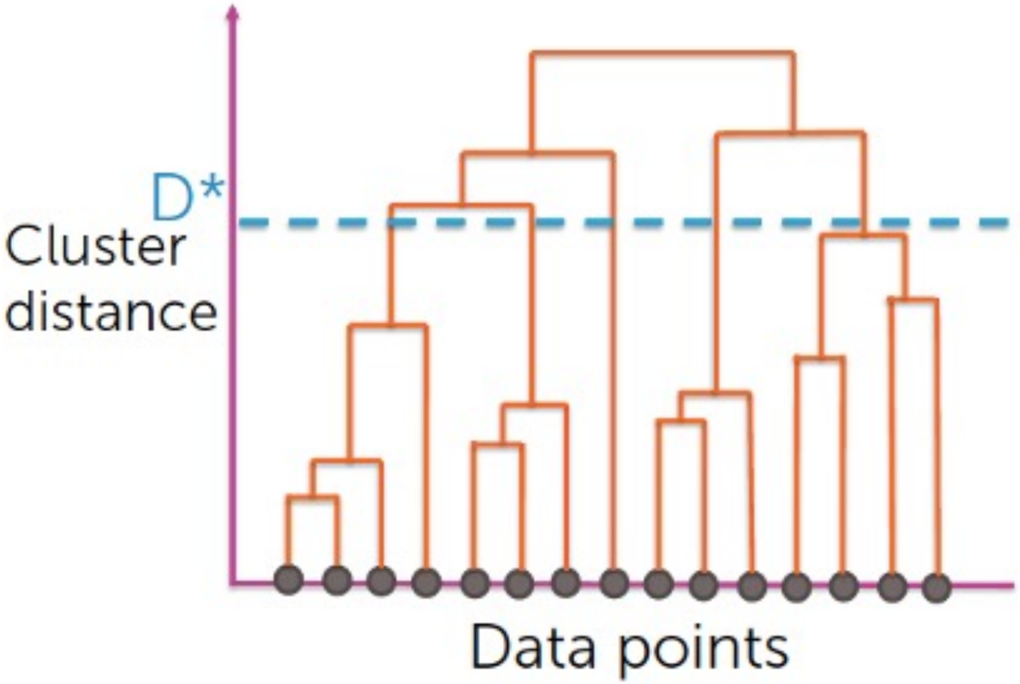
Dendrogram (2)

Path shows all the clusters to which a data point belongs to, and the order in which the clusters merge



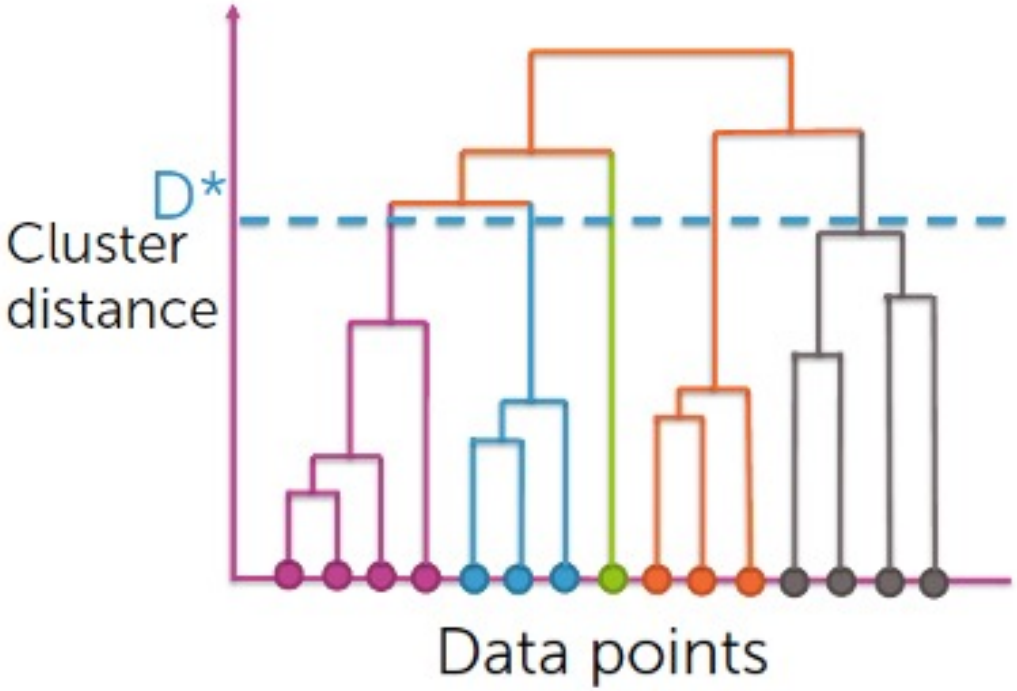
Using Dendrogram

- Choose a distance D^* at which to cut the dendrogram

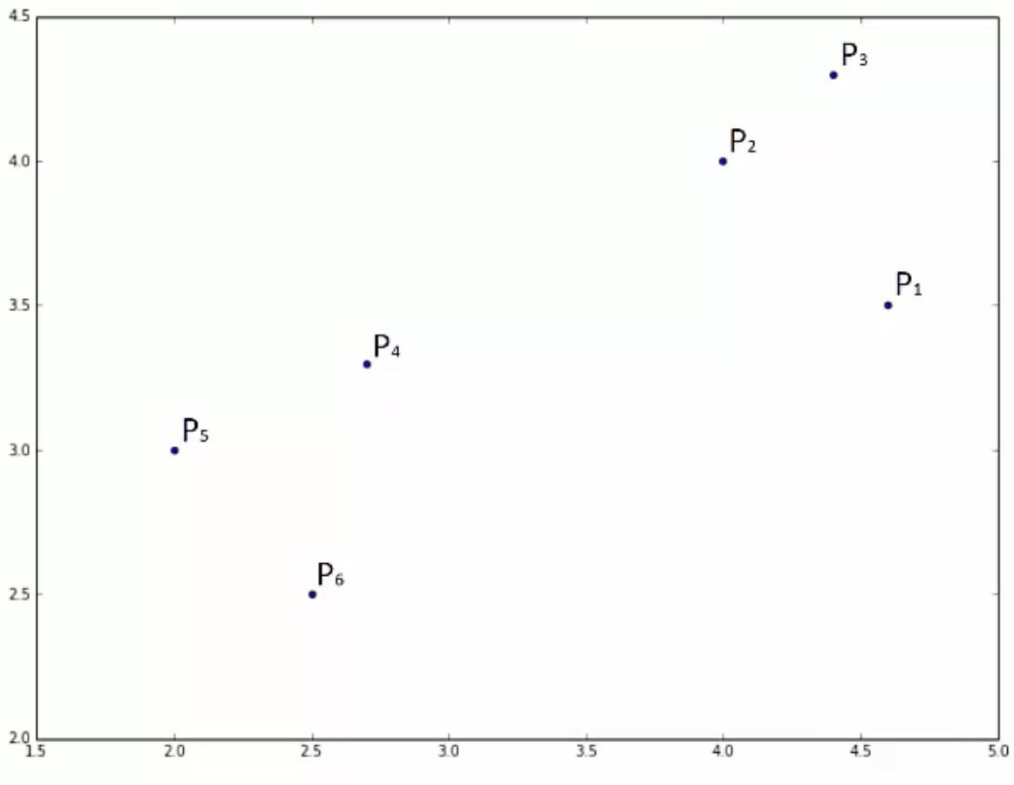


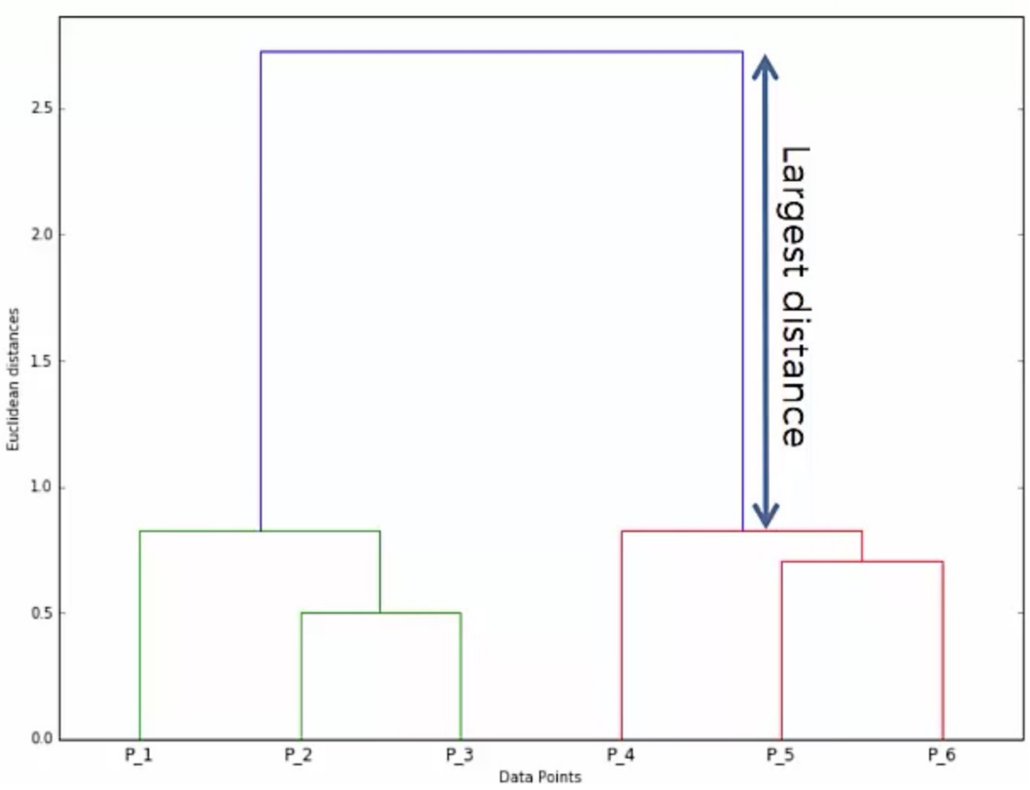
Using Dendrogram (2)

- Every branch that crosses D^* becomes a cluster in the final outcome



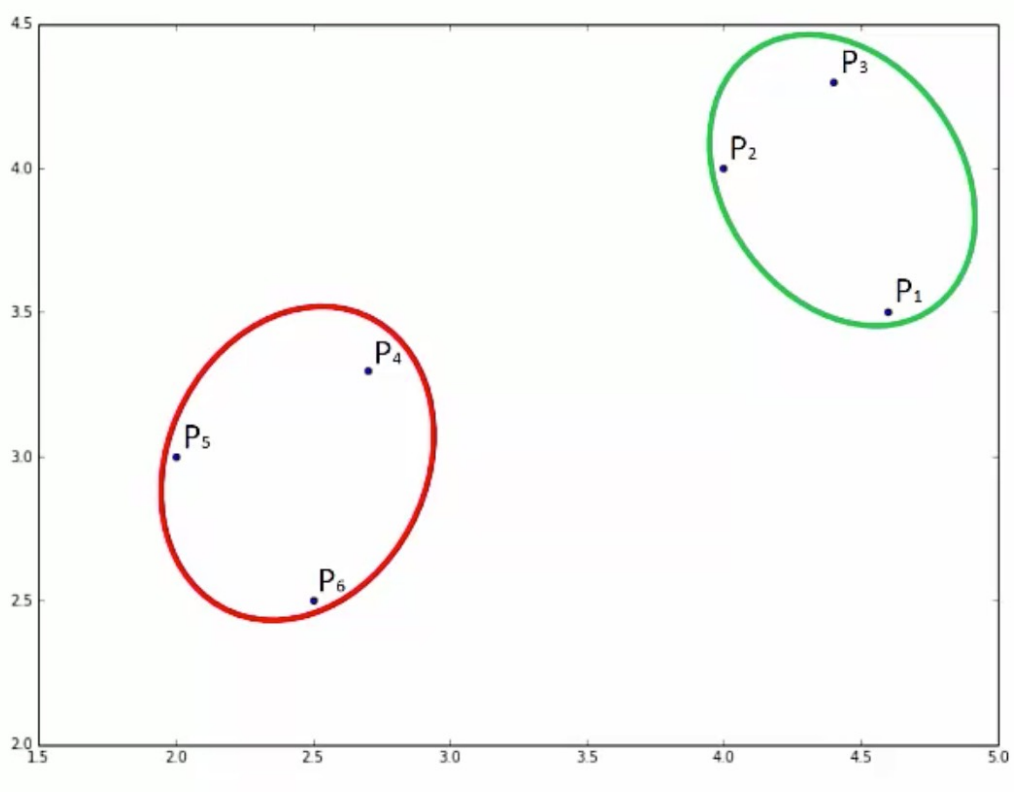
Optimal Cut (Suggestion)

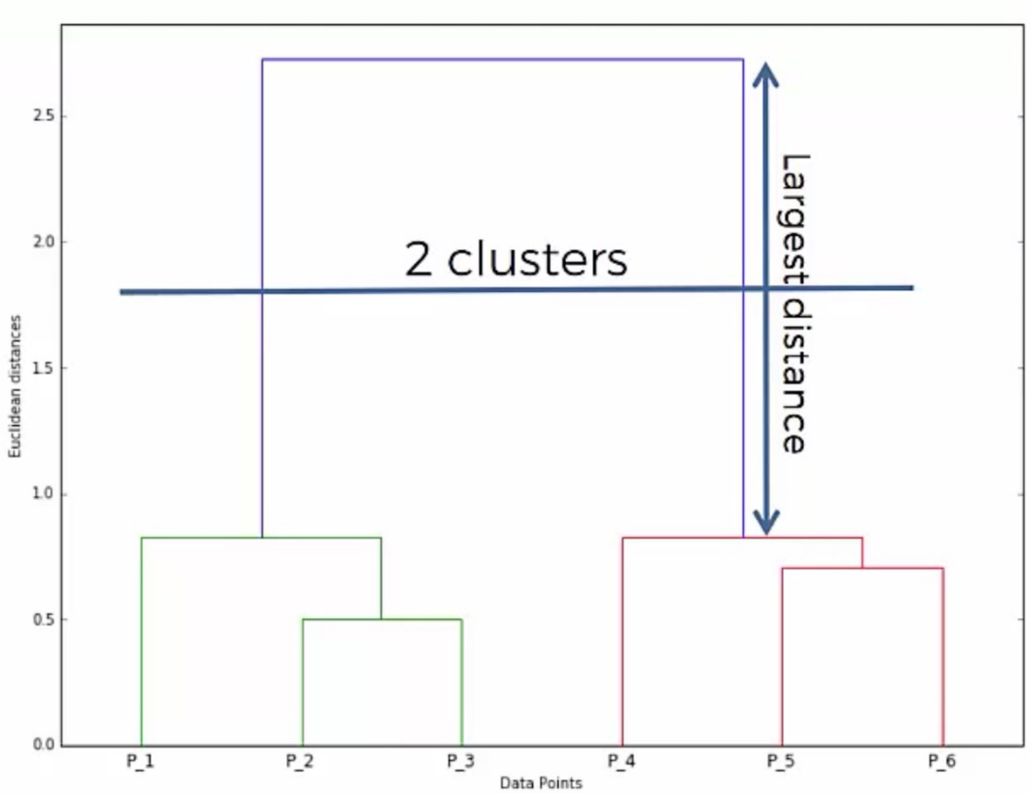




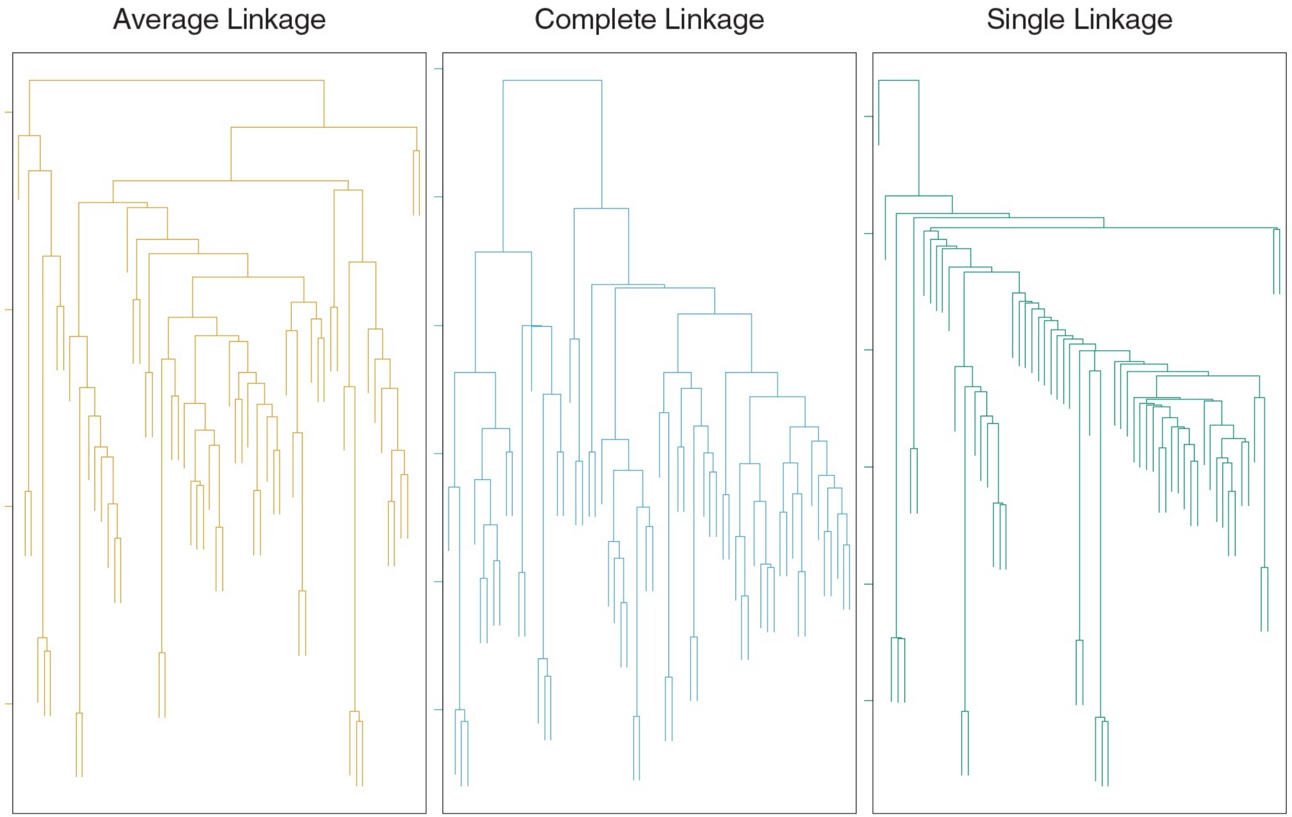
Largest vertical distance uncut by a horizontal line!!

Optimal Cut (Suggestion)





Single vs. Complete vs. Average



Average, complete, and single linkage applied to an example data set.

Average and complete linkage tend to yield more balanced clusters for this example

Explore the Following Topics (Self-Reading)

1. What is DBSCAN?
2. How does it work?
3. What problems with k-means does it help overcome

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd*(Vol. 96, №34, pp. 226–231).*

Summary (2)

1. Clustering: discovering structures/groups in the data
    - An unsupervised learning problem
2. Principle on which clustering works
3. K-means clustering
    - Its objective function
    - How is it motivated
    - How k-means solves it optimization problem
    - Convergence of k-means
    - K-means++
4. Limitations of k-means

Summary (1)

1. K-means clustering
   - K-means ++
2. Hierarchical clustering
   - Divisive
   - Agglomerative
3. DBSCAN
   - Density based – Recommended Reading