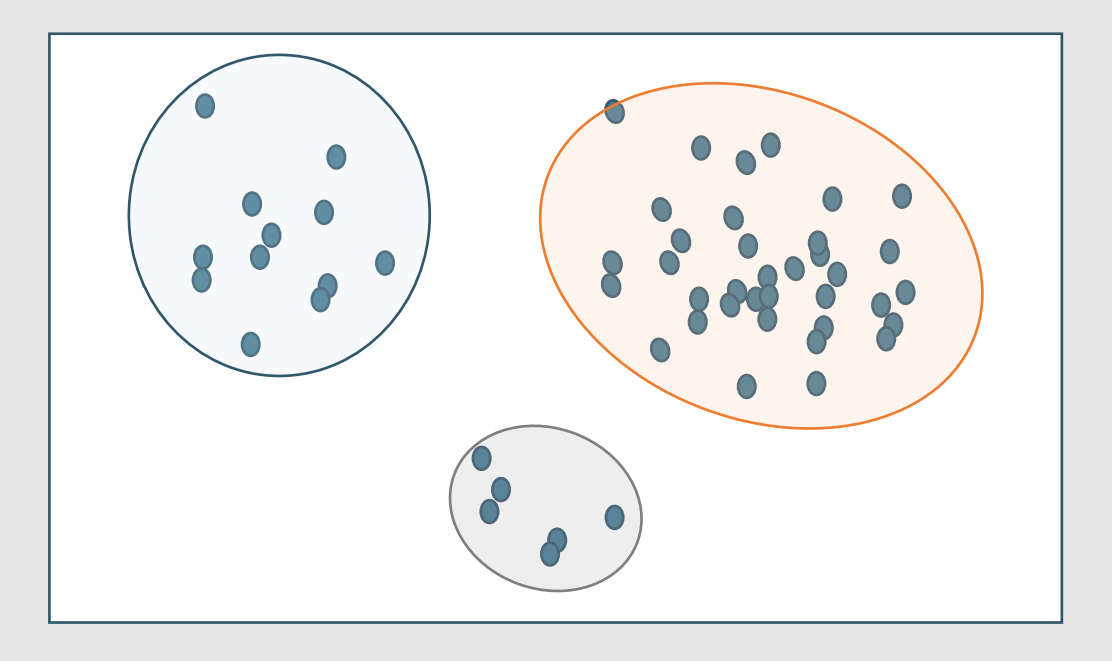
Lecture 9 – Unsupervised Learning

Cluster Analysis

Information Systems for Sustainable Society (is3) | WiSo Faculty | Univ.-Prof. Dr. Wolfgang Ketter | 12/7/22

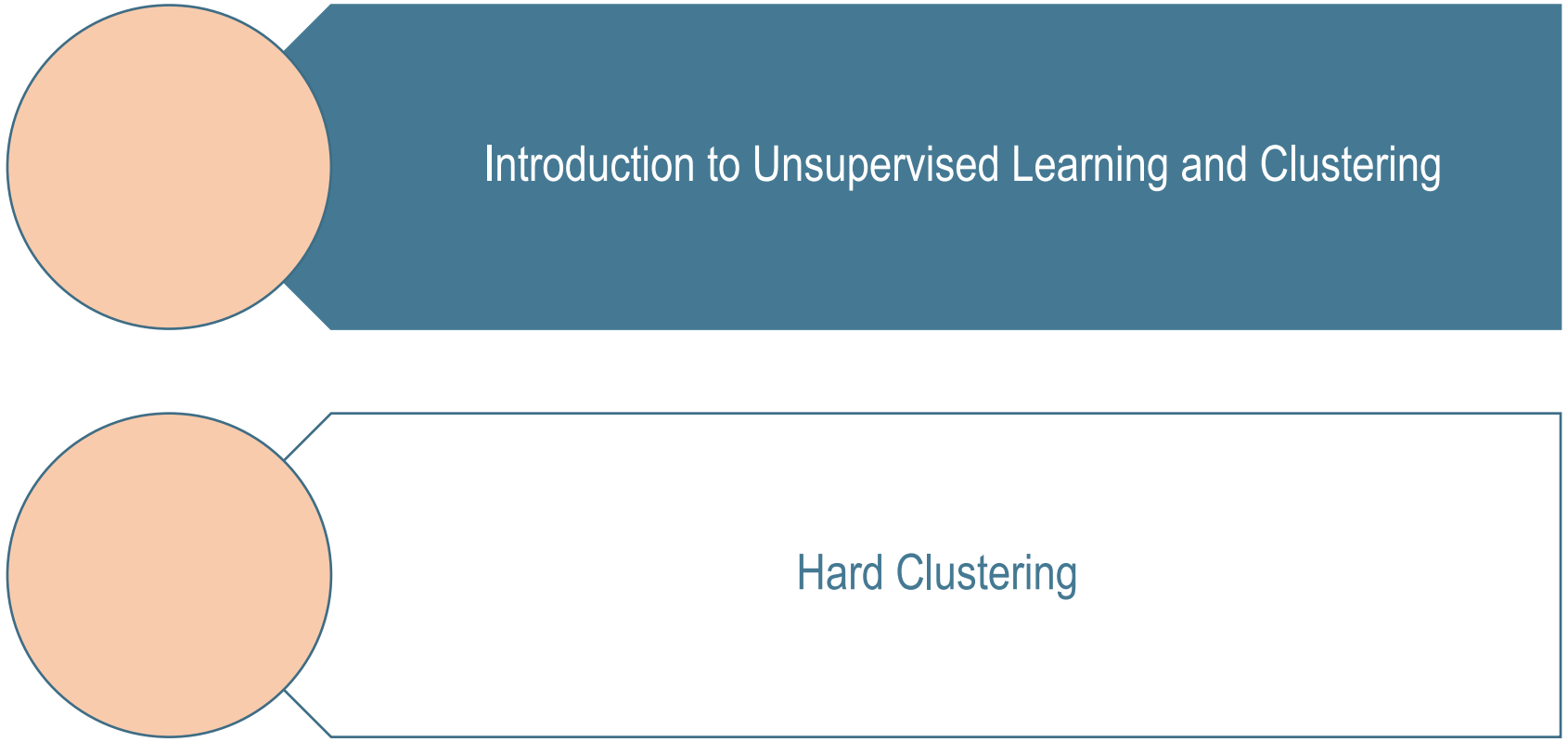
In today's lecture we will investigate unsupervised learning – A paradigm that is distinct from supervised learning tasks discussed thus far

Illustration of Unsupervised Learning



- Data **without** labels
  - E.g., customer clustering – No information number and type of consumer classes (i.e., no labels); just information of certain user characteristics (e.g. avg. basket, number of purchases, etc.)
- Goal to find certain structural **patterns** within the data
- Find **clusters** in data with similar characteristics/attributes
- Model performance **hard** to evaluate

Agenda


Introduction to Unsupervised Learning and Clustering
Hard Clustering

Supervised Learning Paradigm – Learning form labeled data

Training Data	Machine Learning Algorithm	Predictions
$(x^{(1)}, y^{(1)})$ $(x^{(2)}, y^{(2)})$ $(x^{(3)}, y^{(3)})$ $\vdots$	$Hypothesis\ function$ $y^{(i)} \approx h(x^{(i)})$	$New\ example\ \ x$ $\hat{y} = h(x)$

Unsupervised Learning Paradigm – Uncovering patterns in unlabeled data

Training Data	Machine Learning Algorithm	Predictions
$(x^{(1)})$ $(x^{(2)})$ $(x^{(3)})$ $\vdots$	$Hypothesis\ function$ $? \approx h(x^{(i)})$	$New\ example\ \ x$ $? = h(x)$

Three elements of unsupervised learning

- It turns out that **virtually all unsupervised learning** algorithms can be **considered** in the **same manner as supervised learning**:
  1. **Define hypothesis function**
  2. **Define loss function**
  3. Define **how** to **optimize** the loss function
- But, what do a hypothesis function and loss function signify in the unsupervised setting?

Unsupervised learning framework - The canonical ML problem can be directly applied to unsupervised learning

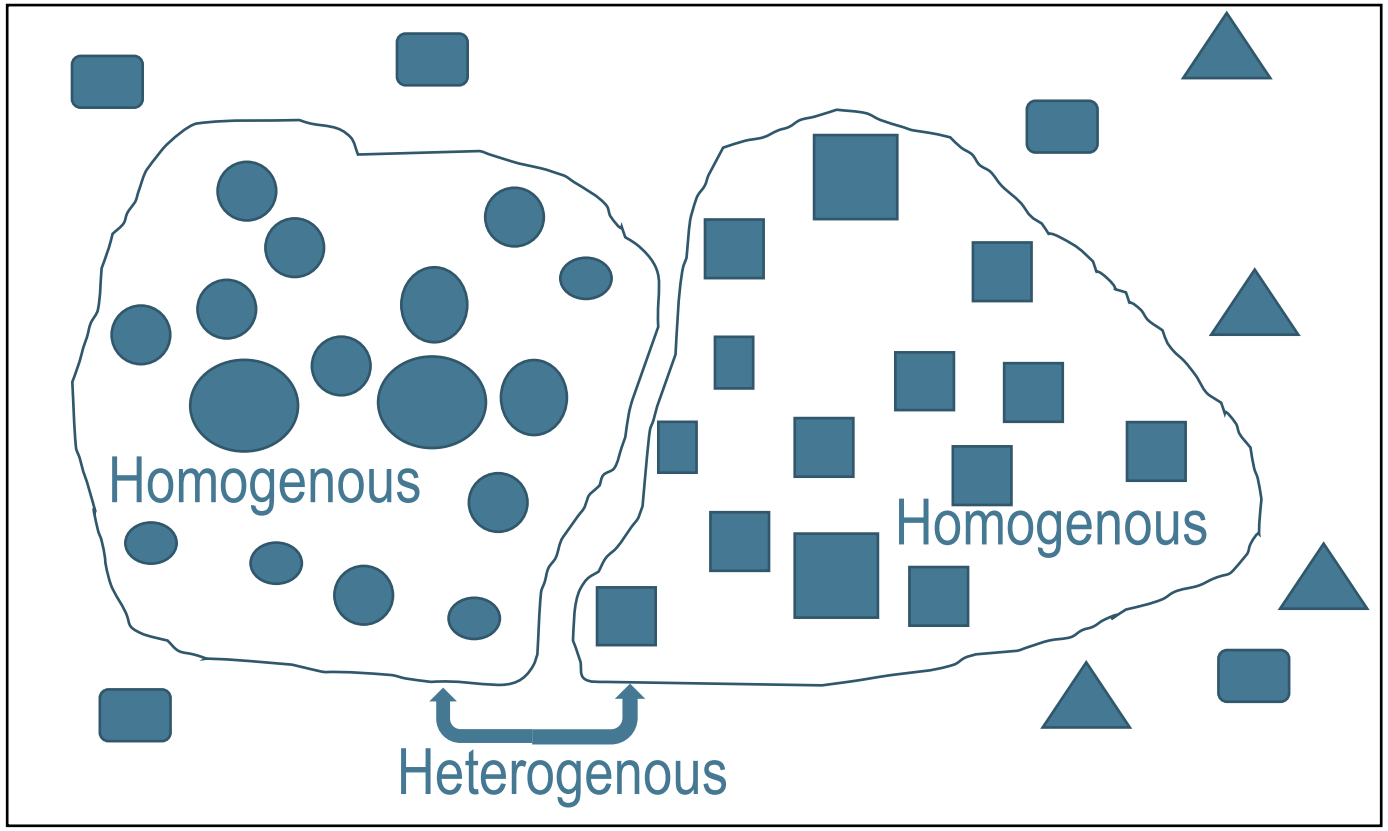
- **Input features:** $x^{(i)} \epsilon \mathbb{R}^n, i = 1, \ldots, m$
- **Model parameters:** $\theta \in \mathbb{R}^k$
- **Hypothesis function:** $h_\theta : \mathbb{R}^n \rightarrow \mathbb{R}^n$, approximates **input** given input
  - i.e. we want $x^{(i)} \approx h_\theta(x^{(i)})$
- **Loss function:** ℓ: $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_+$, measures the difference between a hypothesis and actual input
  - e.g.: $\ell(h_\theta(x), x) = \|h_\theta(x) - x\|_2^2$
- Similar canonical machine learning optimization as before:
  - $Minimize_\theta \sum_{i=1}^{m} \ell\left(h_\theta\left(x^{(i)}\right), x^{(i)}\right)$

Hypothesis and loss functions

- The framework seems odd, **what does it mean to have a hypothesis function approximate the input?**
- **Can't we just pick $h_\theta(x) = x$?**
- The goal of unsupervised learning is to pick some restricted class of hypothesis functions that **extract some kind of structure from the data** (i.e., one that does not include the identity mapping above)
- In this lecture, we will consider a range of different algorithms that fit this framework

Definition of Clustering – A core unsupervised learning task

Illustration of Clustering



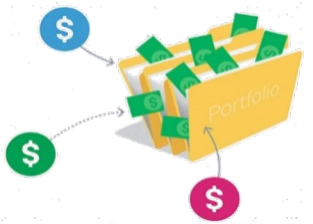
- **Clustering** is another application of our fundamental **notion of similarity**
- The basic idea is that we want to find groups of objects (consumers, businesses, etc.), where the objects within groups are similar (homogenous) but the objects in different groups are not so similar (heteroeneous)

Introduction to Clustering

- **Cluster analysis** is used to form groups or clusters of similar observations based on several measurements made on these observations.
- **Applications:**
  - **Market Segmentation:** Customers are segmented based on demographic and transaction history information, and a marketing strategy is tailored for segment.

  - **Market Structure Analysis:** Identifying groups of products according to competitive measures of similarity.

  - **Creating Balanced Portfolios:** Given data on variety of investment opportunities, one may find clusters based on financial performance variables.



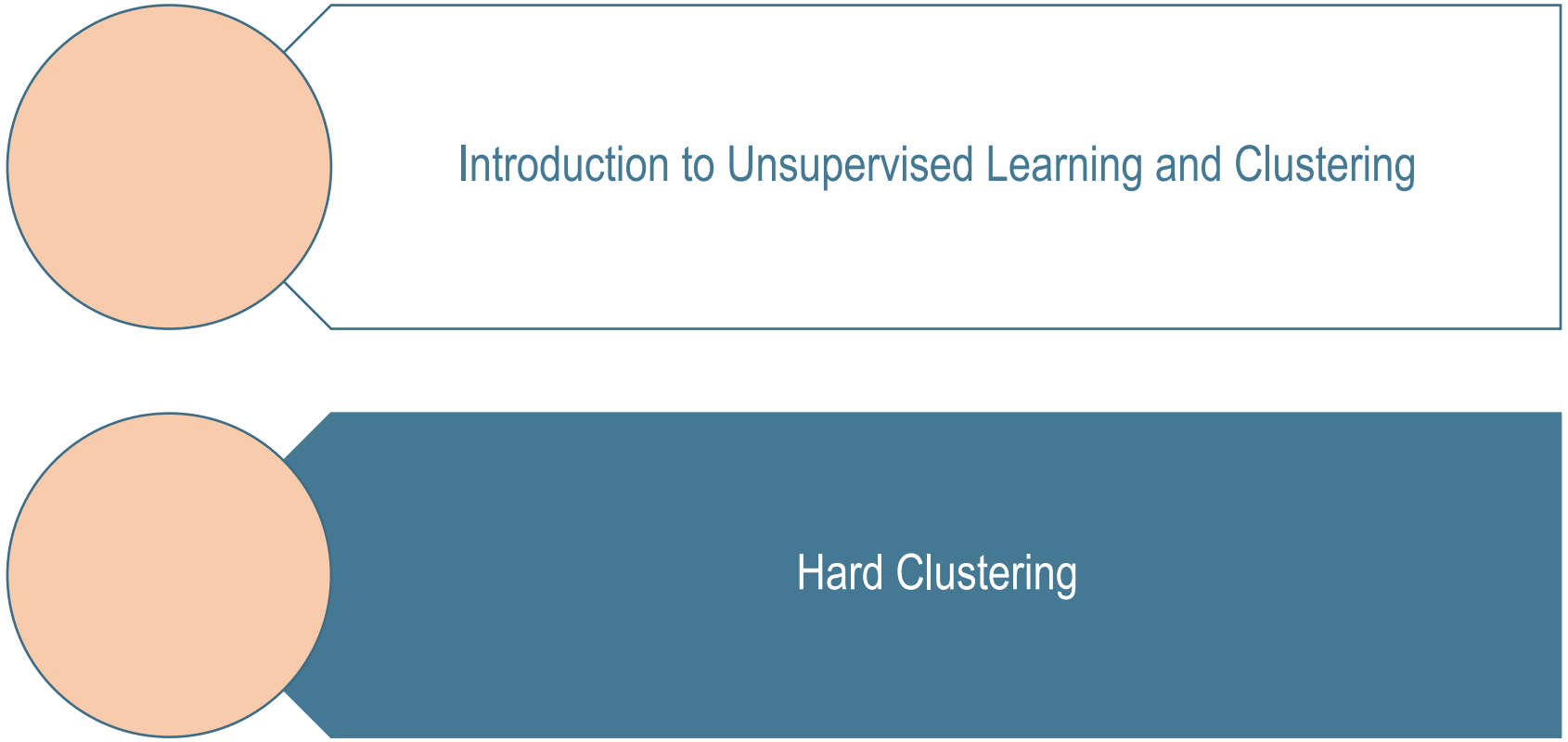
Clustering (unsupervised) or Classification (supervised)?

Data Science Task	Clustering	Classification
Identify archetypical customer groups	✓	
Group cancerous cells into benign or malignant cells		✓
Produce a grouping of animal species	✓	
Decide if a customer is likely to churn or not		✓
Decide on the destination of a carsharing trip (e.g., districts of the city)		✓
Define which geographical regions in a bike-sharing network are similar	✓	

Desirable Properties of Clustering Algorithms

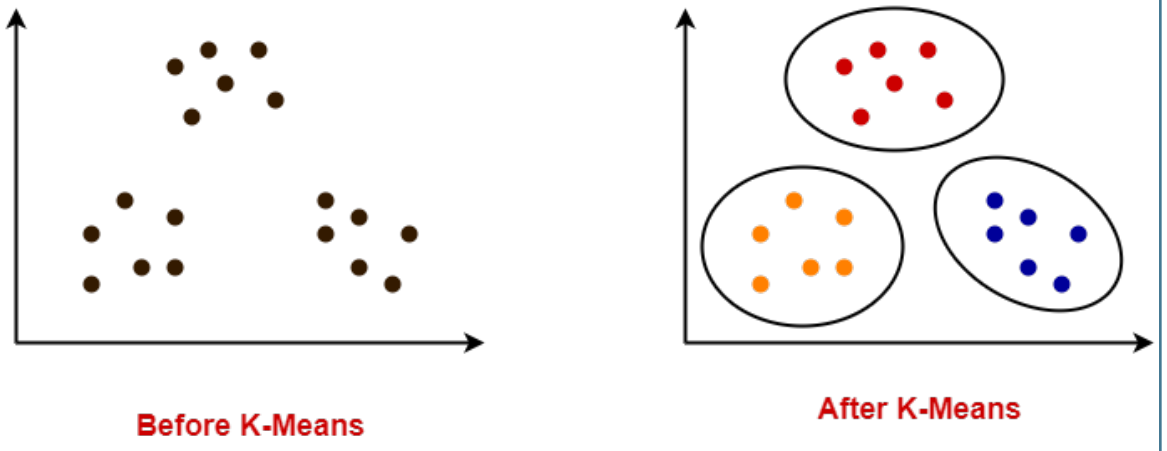
- Scalability (in terms of both time and space)
- Ability to deal with different data types
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- Incorporation of user-specified constraints
- Interpretability and usability

Agenda

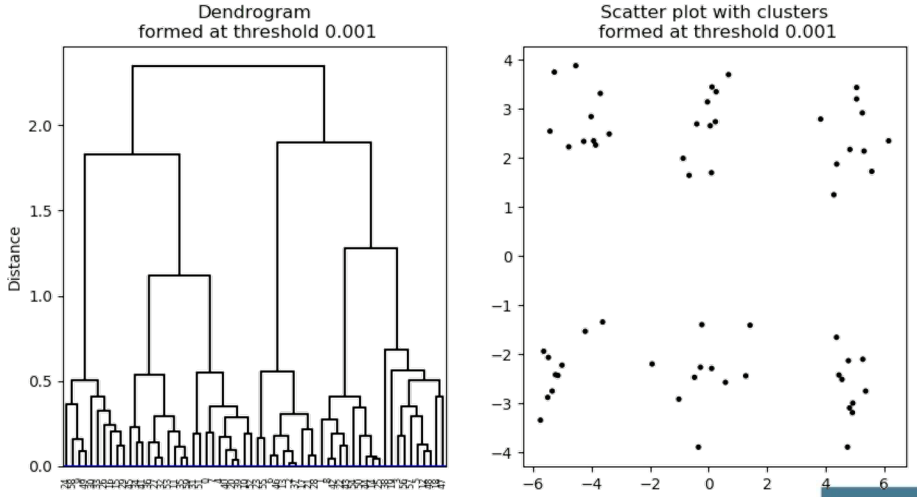

Introduction to Unsupervised Learning and Clustering
Hard Clustering

Two Types of Hard Clustering

- **Partitioning algorithms:** Construct various partitions and then evaluate them by some criterion.
  - k-means
  - k-means ++



- **Hierarchical algorithms:** Create a hierarchical decomposition of the set of objects using some criterion.
  - Hierarchical clustering



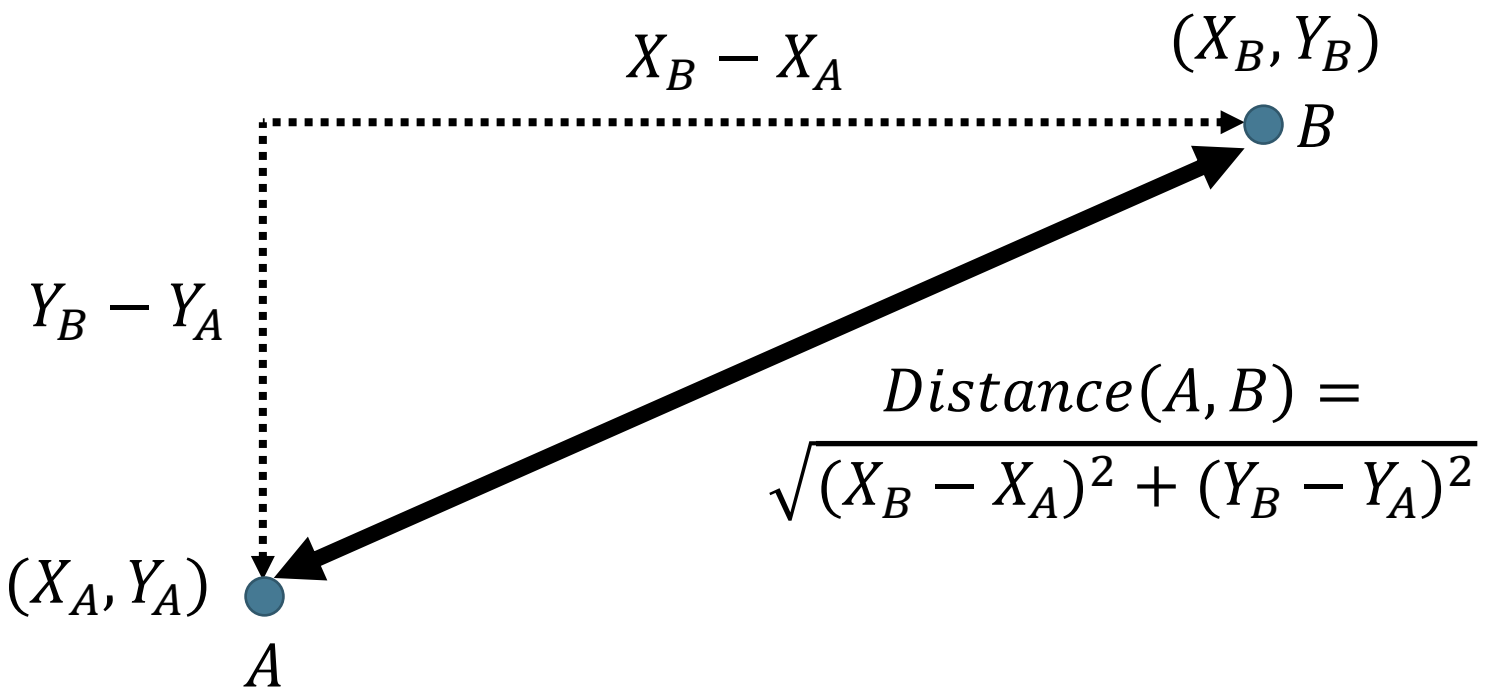
Similarity and distance are core concepts of clustering

Attribute	Person A	Person B
Age	23	40
Years at current address	2	10
Residential status (1= Owner, 2= Renter, 3= Other)	2	1

- If two objects can be represented as feature vectors, then we can compute the distance between them

Distance Between Two Records – Euclidean Distance

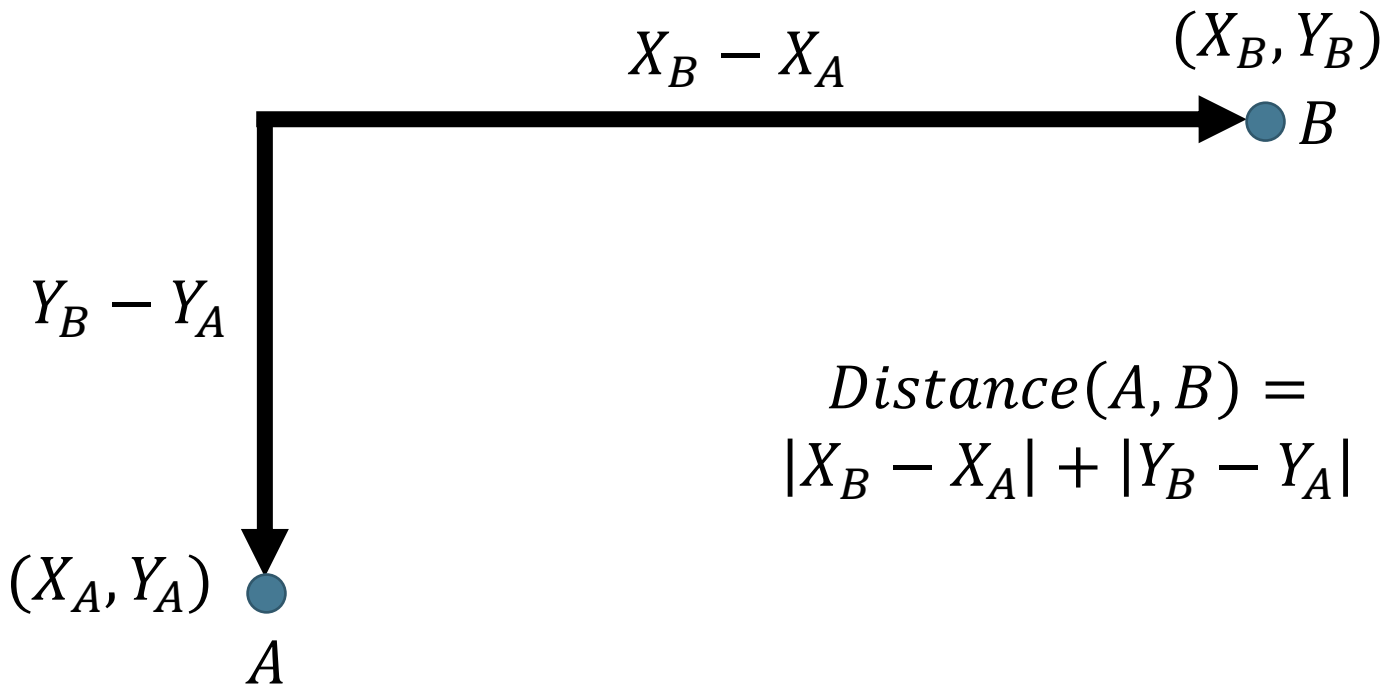
$$d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ip} - x_{jp})^2}$$



- **Euclidean Distance** is most popular
- Recall the Pythagorean Theorem ($a^2 + b^2 = c^2$) – This is Euclidean distance in two dimensions

Distance Between Two Records – Manhattan Distance

$$d_{Manhattan}(X, Y) = \|X - Y\|_1 = |x_1 - y_1| + |x_2 - y_2| + \ \ldots$$



$$Distance(A, B) = |X_B - X_A| + |Y_B - Y_A|$$

- **Manhattan Distance** is another popular distance metric named after the typical grid network of Manhattan

Distance Between Two Records – Cosine Distance

- **Cosine distance** is often used in text classification to measure the similarity of two documents
  - $d_{cosine}(X, Y) = 1 - \frac{X \times Y}{\|X\|_2 \times \|Y\|_2}$
  - Where $\|.\|_2$ represents the L2 norm, or Euclidean length, of each feature vector (for a Vector this is simply the distance from the origin)
  - For example, if $\mathrm{A} = \langle 7,3,2 \rangle$ and $\mathrm{B} = \langle 2,3,0 \rangle$, then Cosine distance between A and B is

$$d_{cosine}(A, B) = 1 - \frac{\langle 7,3,2 \rangle \times \langle 2,3,0 \rangle}{\|\langle 7,3,2 \rangle\|_2 \times \|\langle 2,3,0 \rangle\|_2} = 1 - \frac{7 \times 2 + 3 \times 3 + 2 \times 0}{\sqrt{49 + 9 + 4} \times \sqrt{4 + 9}} = 1 - \frac{23}{28.4}$$

$$\approx 0.19$$

Example: “Whiskey Analytics” (1/2)

Color	yellow, very pale, pale gold, gold, old gold, full gold, amber, etc.	(14 values)
Nose	aromatic, peaty, sweet, light, fresh, dry, grassy, etc.	(12 values)
Body	soft, medium, full, round, smooth, light, firm, oily.	(8 values)
Palate	full, dry, sherry, big, fruity, grassy, smoky, salty, etc.	(15 values)
Finish	full, dry, warm, light, smooth, clean, fruity, grassy, smoky, etc.	(19 values)

- Whiskey can be described along five dimensions
  - Color
  - Nose
  - Body
  - Palate
  - Finish
- **Which Whiskey types are similar to the well-known *Bunnahabhain* Scotch?**

Example: “Whiskey Analytics” (2/2)

Whiskey	Distance[1)]	Descriptors
Bunnahabhain	-	*gold; firm, med, light; sweet, grassy; fresh, sea; full*
Glenglassaugh	0.643	gold; firm, light, smooth; sweet, grassy; fresh, grassy
Tullibardine	0.647	gold; firm, med, smooth; sweet; fruit, full, grassy, clean; sweet; big, arome, sweet
Ardbeg	0.667	sherry; firm, med, full, light; sweet; dry, peat, sea; salt
Bruichladdich	0.667	pale; firm, light, smooth; dry, sweet, smoke, clean; light; full
Glenmorangine	0.667	gold; med oily, light; sweet, grassy, spice; sweet, spicy, grassy, sea, fresh; full, long

1) Euclidian Distance
The dataset is available here.

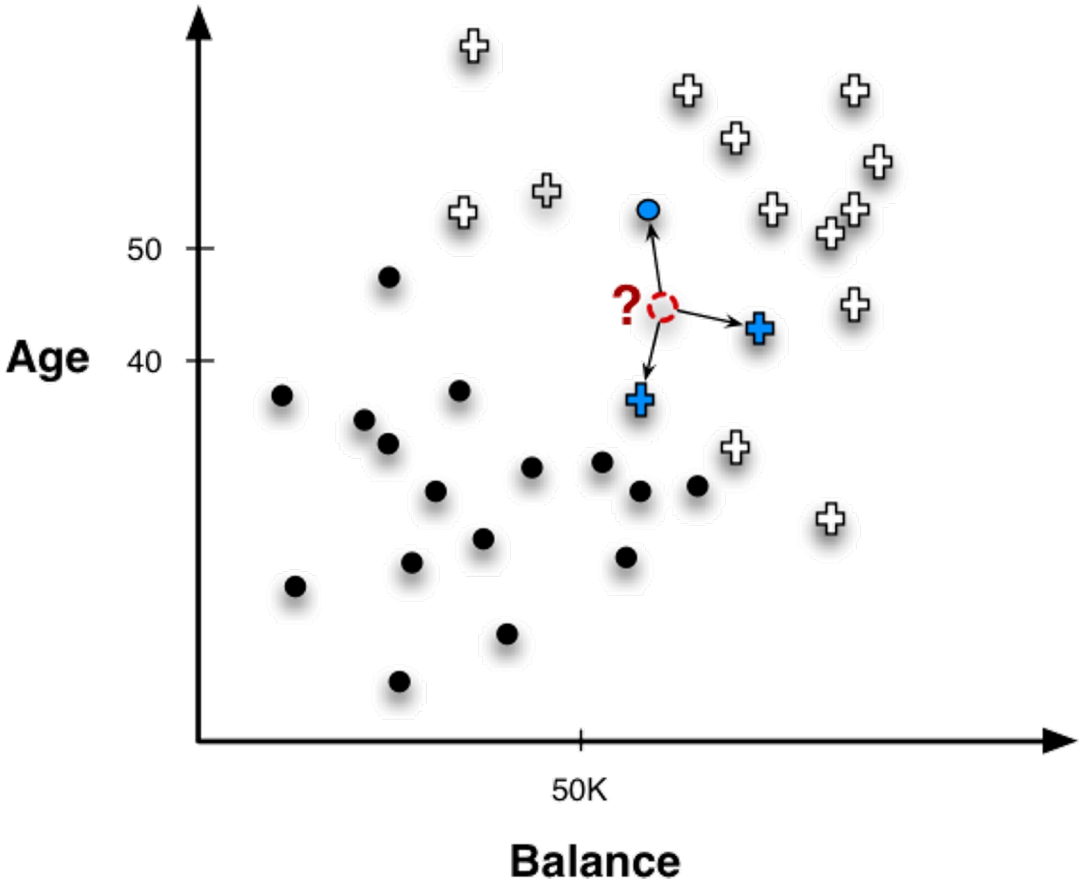
Make sure to scale your features! – For example standardize them

- **Problem:**
  - Raw distance measures are highly influenced by scale of measurements
- **Solution:**
  - normalize (standardize) the data first
    - Subtract mean, divide by std. deviation
    - Also called **z-scores**

Illustrative Example: Classification

Distance measures in a supervised ML scenario (with labels)

To understand the working of distance-based clustering we will consider a supervised algorithm first – K Nearest Neighbors (kNN)



- Instance-based, non-parametric, supervised learning method
- Doesn't construct a general, explicit description of the target function based on training examples, but simply stores training examples
- Generalizing is postponed until a new instance must be classified

K nearest neighbor algorithm can be used both for classification and regression

- In *k-NN classification*, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive, typically small). If *k* = 1, then the object is simply assigned to the class of that single nearest neighbor.

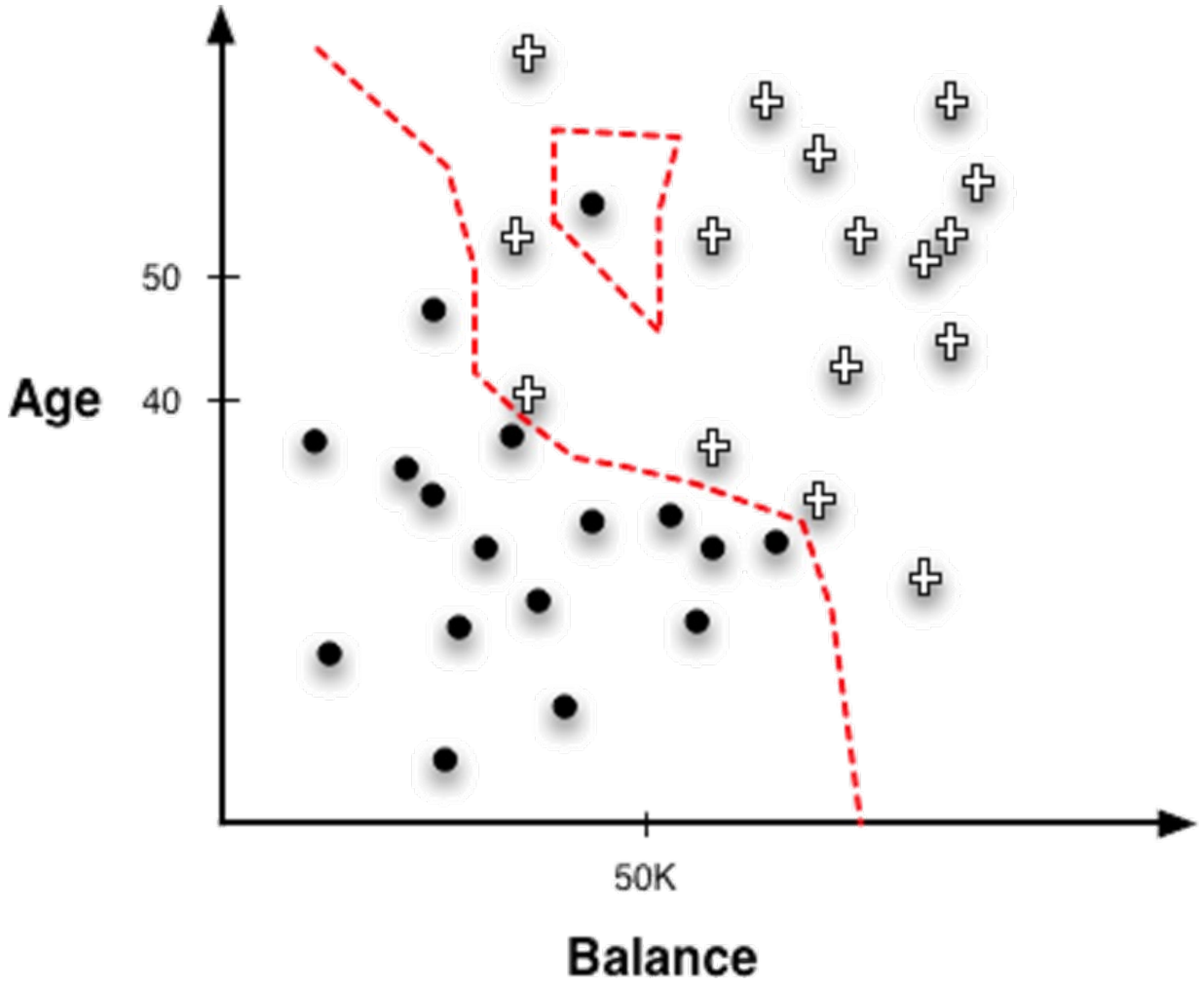
- In *k-NN regression*, the output is the property value for the object. This value is the average of the values of *k* nearest neighbors.

- There are only 3 steps for KNN:
  1. Calculate distance (e.g. Euclidean distance, Hamming distance, etc.)
  2. Find k closest neighbors
  3. Vote for labels or calculate the mean

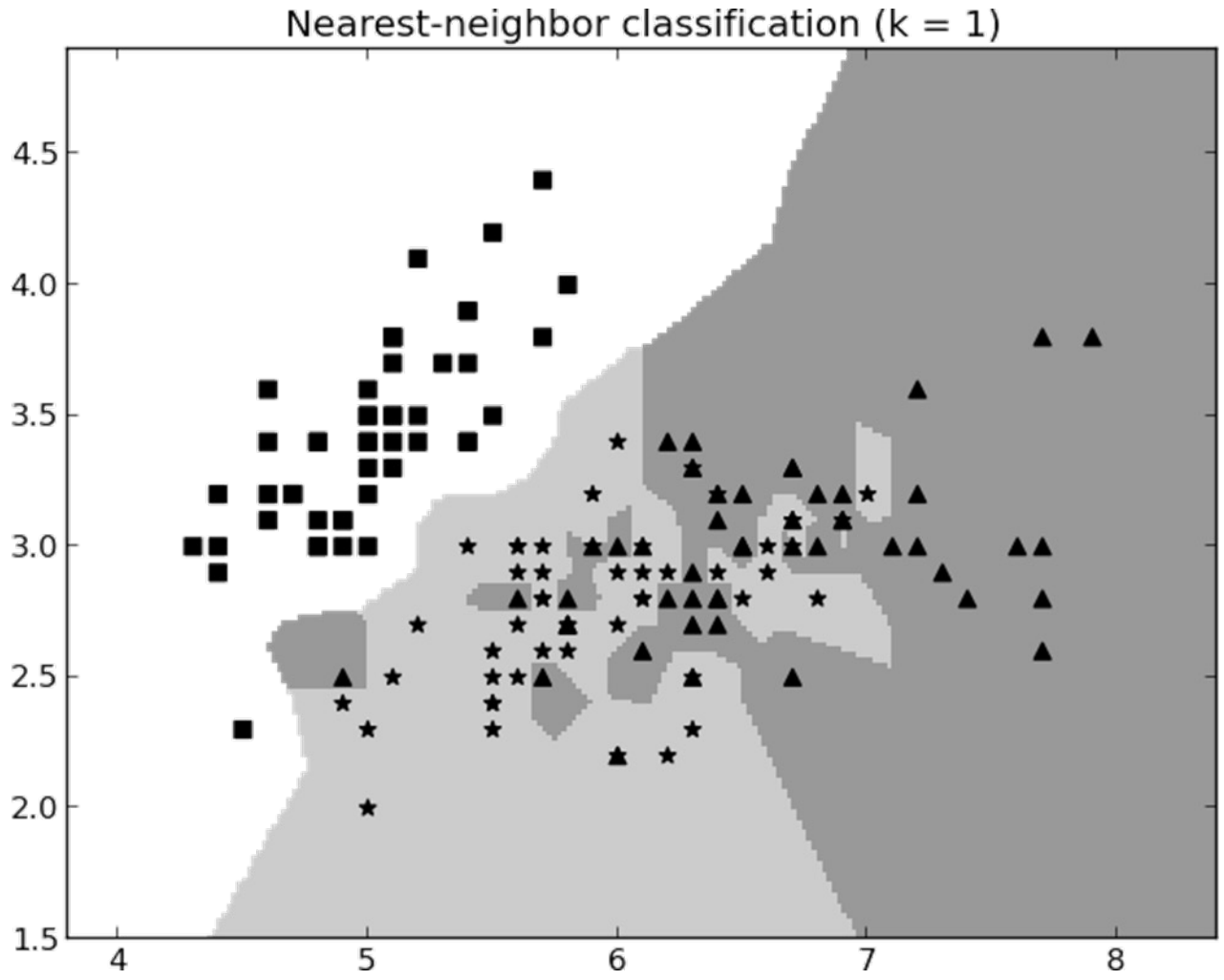
Nearest Neighbors for Predictive Modeling

Customer	Age	Income (1000s)	Cards	Response (target)	Distance (Euclidean) from David
David	37	50	2	?	0
John	35	35	3	Yes	$\sqrt{(35-37)^2+(35-50)^2+(3-2)^2}=15.16$
Rachael	22	50	2	No	$\sqrt{(22-37)^2+(50-50)^2+(2-2)^2}=15$
Ruth	63	200	1	No	$\sqrt{(63-37)^2+(200-50)^2+(1-2)^2}=152.23$
Jefferson	59	170	1	No	$\sqrt{(59-37)^2+(170-50)^2+(1-2)^2}=122$
Norah	25	40	4	Yes	$\sqrt{(25-37)^2+(40-50)^2+(4-2)^2}=15.74$

Geometric Interpretation, Over-fitting, and Complexity

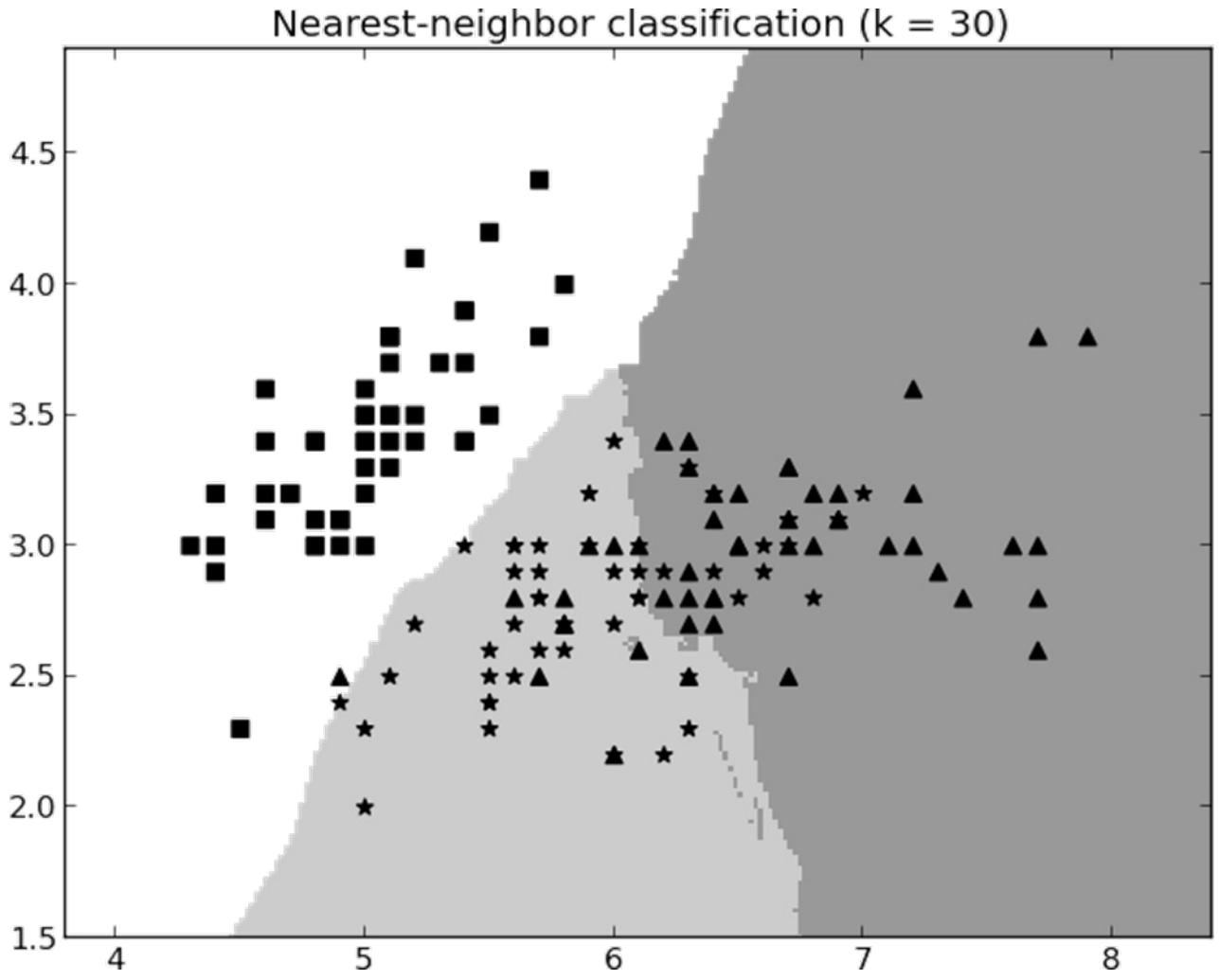


1-Nearest Neighbor



- The most intuitive nearest neighbour type classifier is the one nearest neighbour classifier that assigns a point x to the class of its closest neighbour in the feature space
- The training data will be perfectly predicted, and the bias will be 0 when K=1, however, when it comes to new data (in test set), it has higher chance to be an error, which causes high variance. Hence, A small value of k could lead to overfitting.

30-Nearest Neighbors



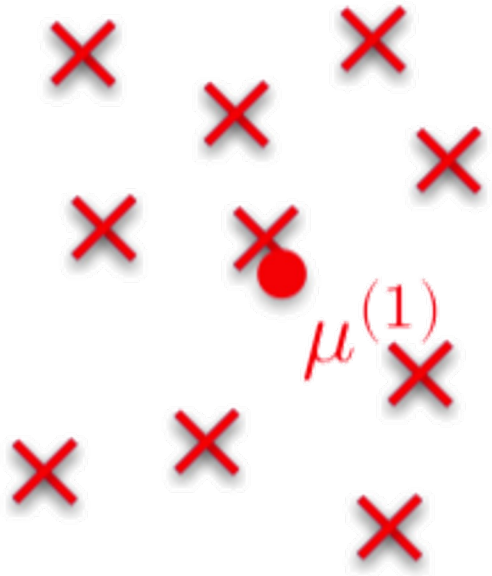
- When we increase K, the training error will increase (increase bias), but the test error may decrease at the same time (decrease variance). We can think that when K becomes larger, since it has to consider more neighbors, its model is more complex. In other words, a big value of K can lead to underfitting.

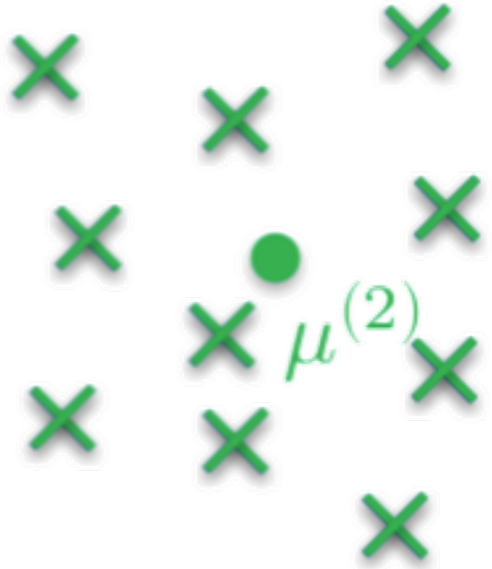
Clustering

Distance measures in an unsupervised ML scenario (without labels)

What happens if we do not have labelled data? – K-means graphically

- The k-means algorithm is easy to visualize: given some collection of data points we want to find k centers such that all points are close to at least one center (i.e. we essentially randomly select our labels and re-select them until our clustering has the lowest error)





K-means in unsupervised framework

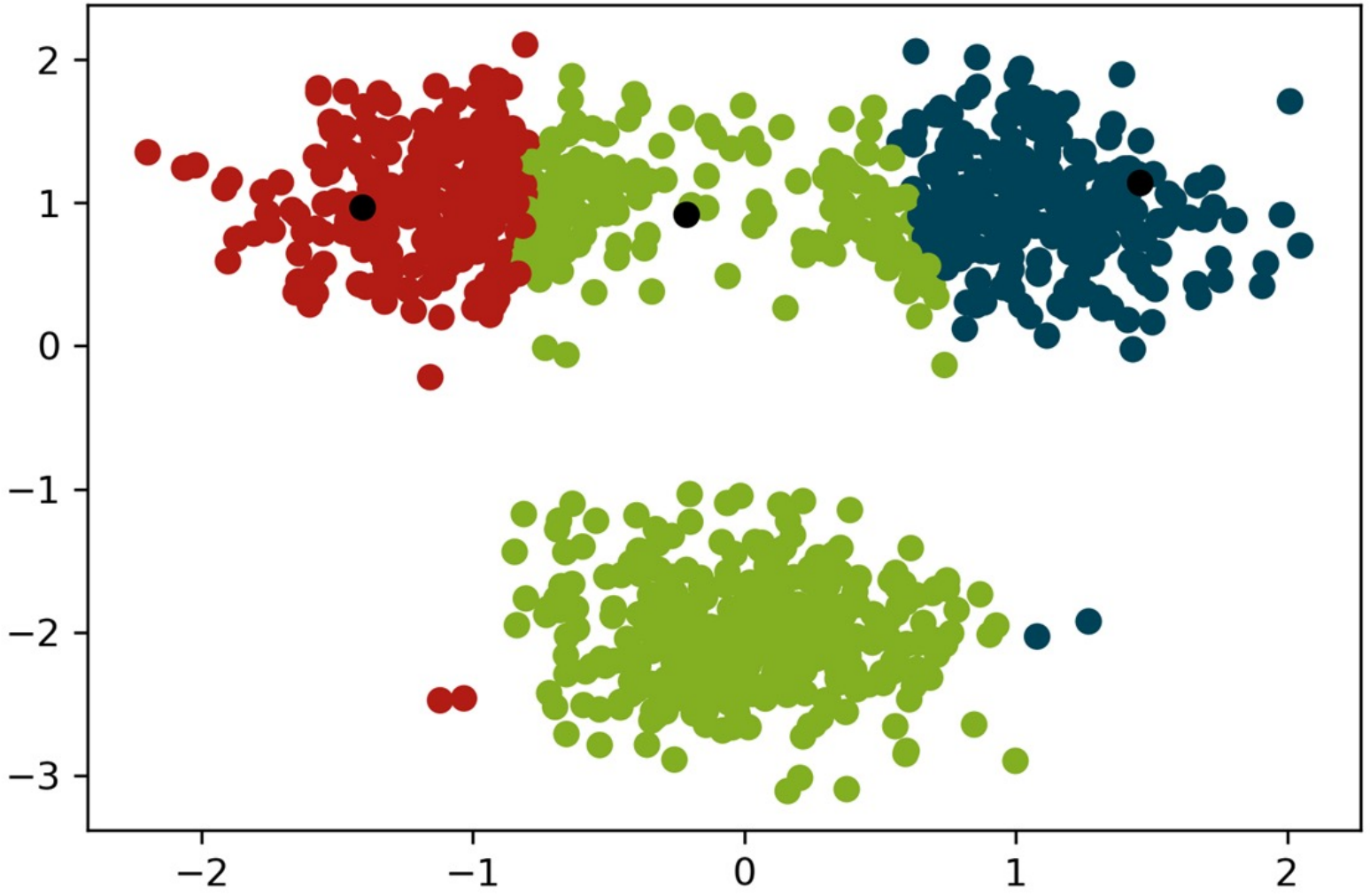
- Parameters of k-means are the choice of centers
  - $\theta = \{\mu^{(1)}, \ldots, \mu^{(k)}\}$ with $\mu^{(i)} \in \mathbb{R}^n$
- Hypothesis function outputs the center closest to a point x
  - $h_\theta(x) = \underset{\mu \in \{\mu^{(1)}, \ldots, \mu^{(k)}\}}{argmin} \|\mu - x\|_2^2$
- Loss function is squared error between input and hypothesis
  - $\ell(h_\theta(x), x) = \|h_\theta(x) - x\|_2^2$
- Optimization problem is thus
  - $\underset{\mu^{(1)}, \ldots, \mu^{(k)}}{Minimize} \sum_{i=1}^{m} \left\|h_\theta(x^{(i)}) - x^{(i)}\right\|_2^2$

Optimizing k-means objective

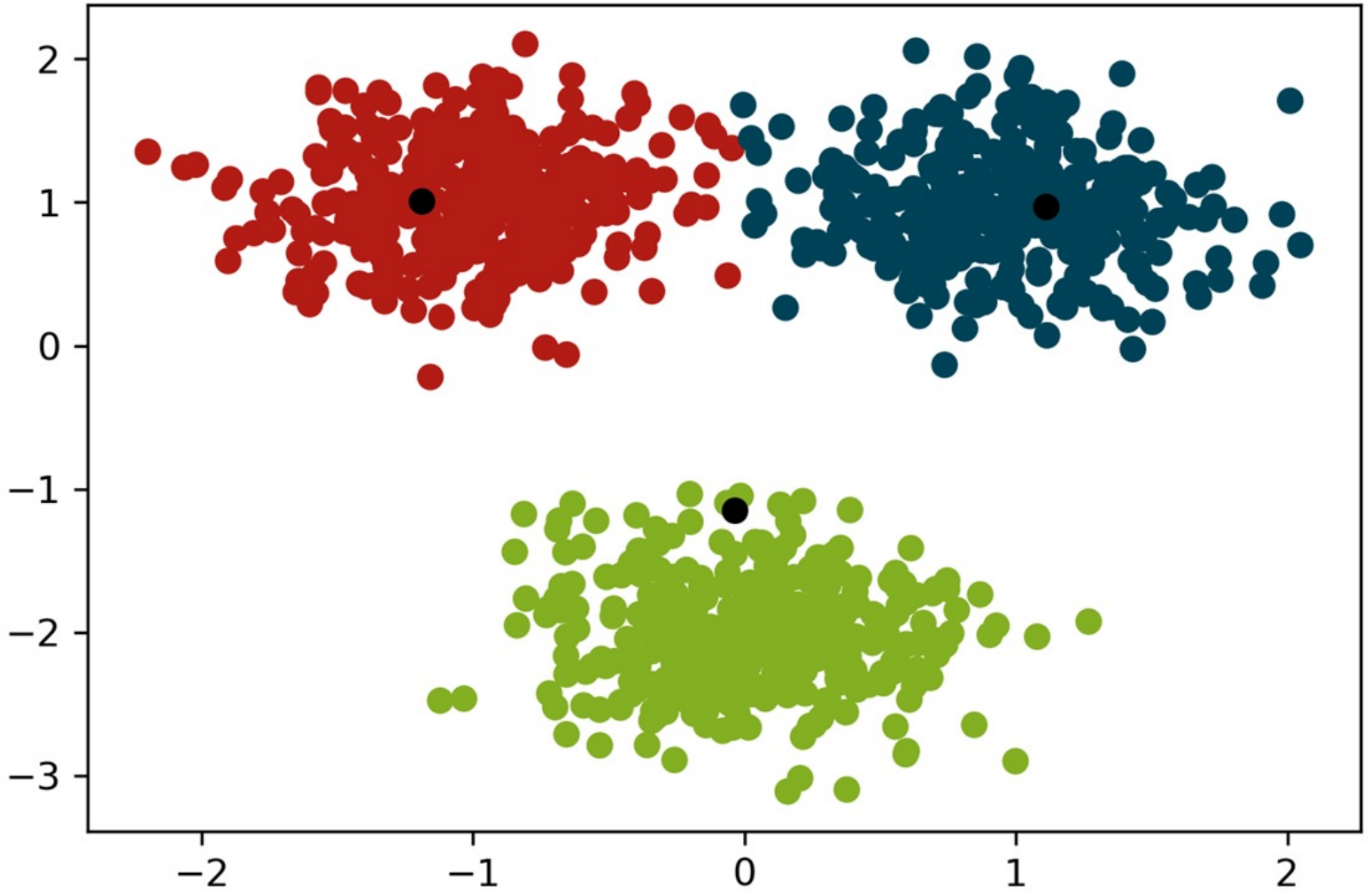
- The k-means objective is non-convex (possibility of local optima), and does not have a closed form solution, so we resort to an approximate method, by repeating the following (Lloyd's algorithm, or just "k-means")

1. Assign points to nearest cluster
2. Compute cluster center as mean of all points assigned to it
   - Given: Data set $(x^{(i)})_{i=1,\dots,m}$, # clusters k
   - Initialize: $\mu^{(j)} \longleftarrow Random\left(x^{(i)}\right),\ j = 1, \dots, k$
   - Repeat until convergence:
     - Compute cluster assignment: $y^{(i)} = \underset{j}{argmin} \left\|\mu^{(j)} - x^{(j)}\right\|_2^2, i = 1, \dots, m$
   - Re-compute means: $\mu^{(j)} \leftarrow Mean\left(\left\{x^{(i)} \middle| y^{(i)} = j\right\}\right), j = 1, \dots, k$

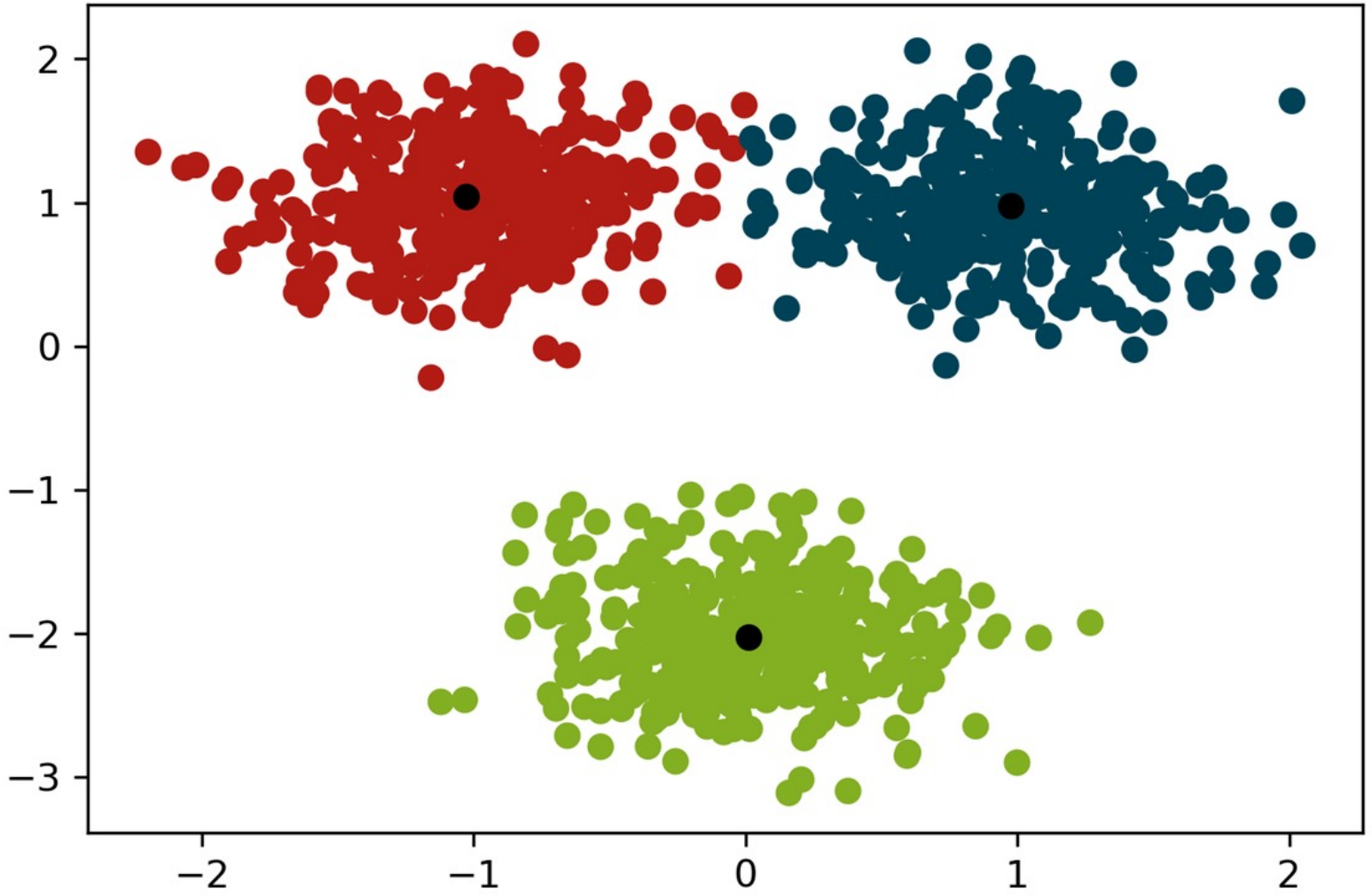
Convergence of k-means



Convergence of k-means



Convergence of k-means



When running the k-means algorithm (compute clusters for each point by finding closest center, set center to be mean of all associated points), what happens to the sum of loss after each iteration,
$\sum_{i=1}^{m} \ell\left(h_\theta\left(x^{(i)}\right), x^{(i)}\right)$?

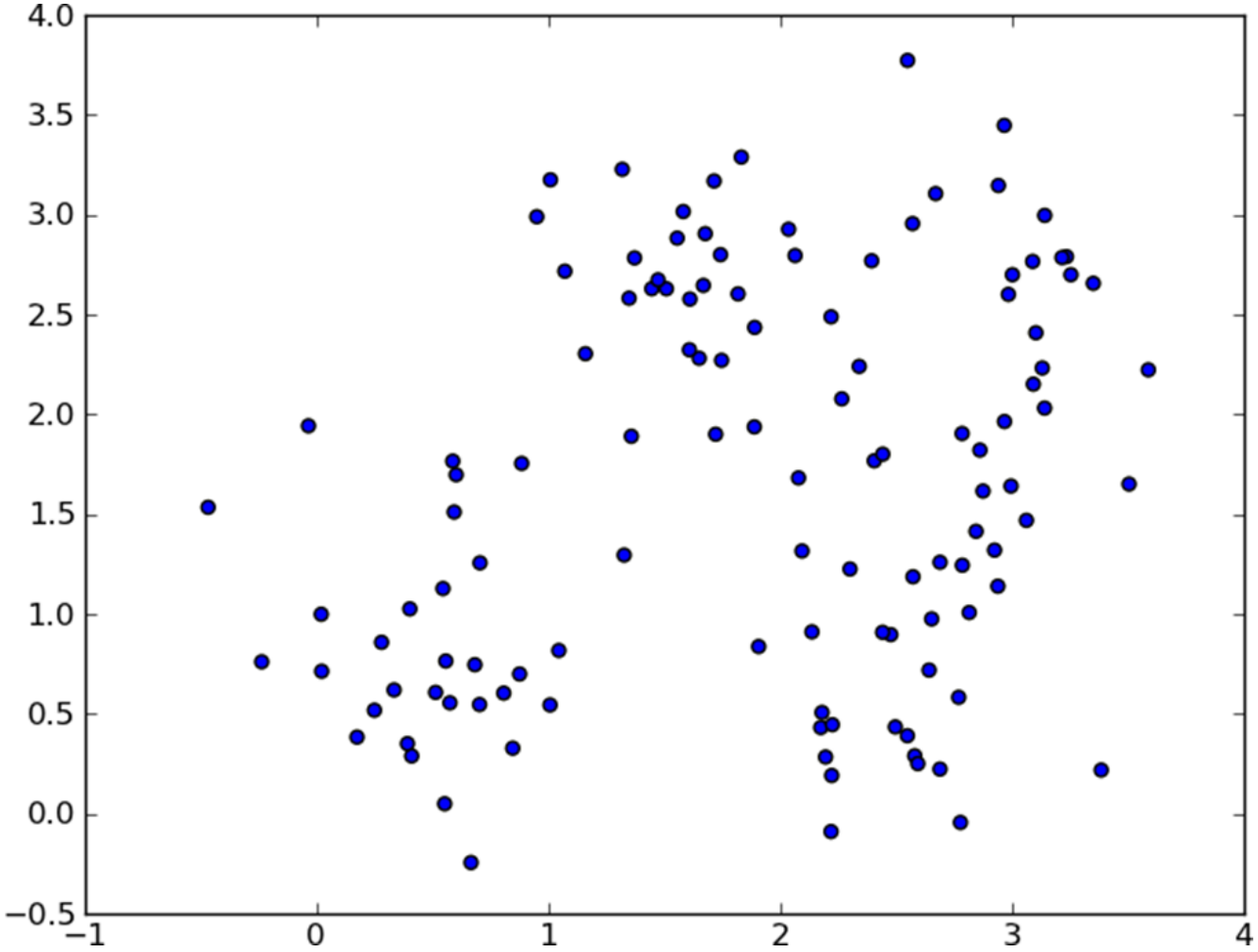
a) Each loss $\ell\left(h_\theta\left(x^{(i)}\right), x^{(i)}\right)$ will decrease for all i
b) The sum of losses will decrease
c) The sum of losses may increase or decrease

When running the k-means algorithm (compute clusters for each point by finding closest center, set center to be mean of all associated points), what happens to the sum of loss after each iteration,
$\sum_{i=1}^{m} \ell\left(h_\theta\left(x^{(i)}\right), x^{(i)}\right)$?

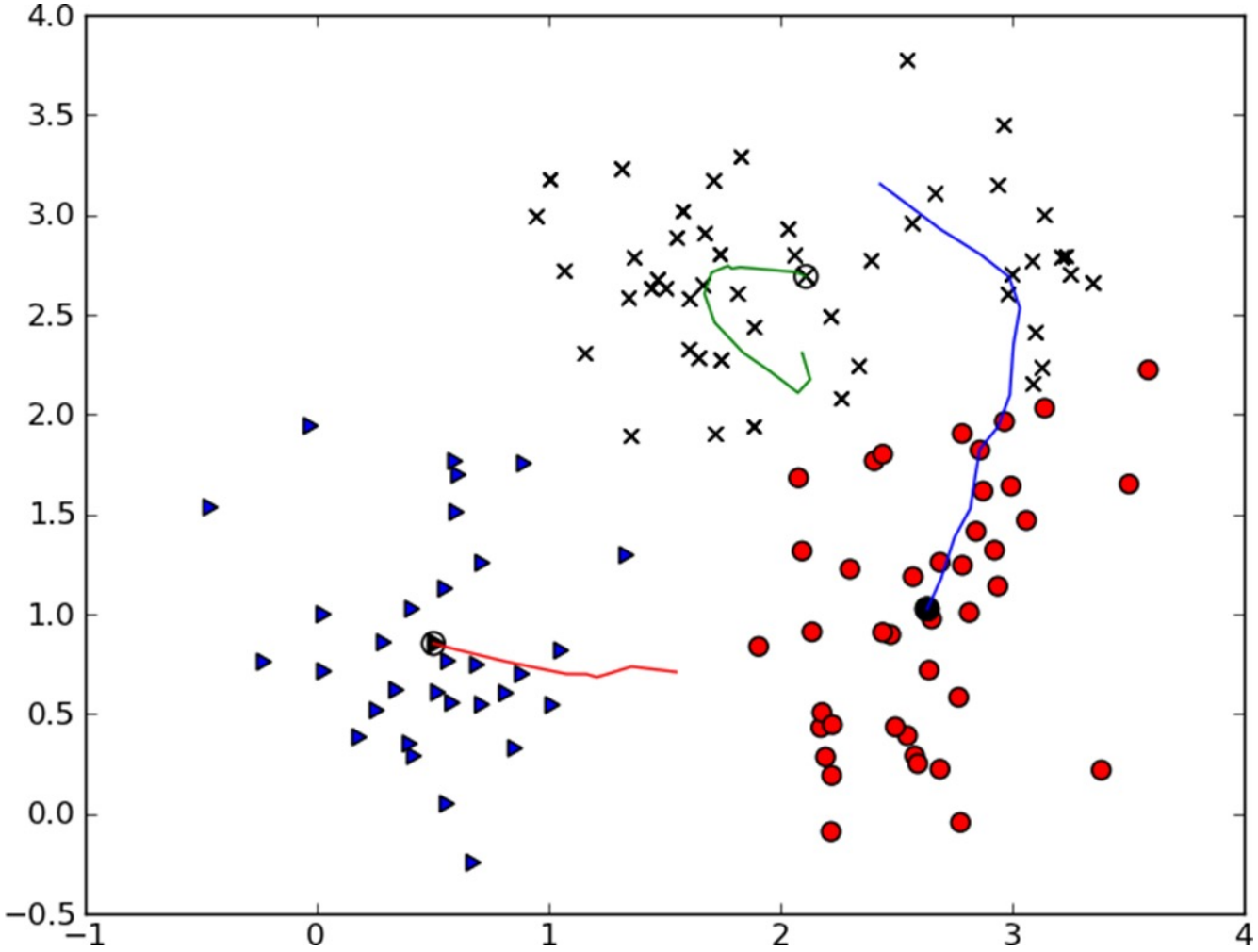
a) Each loss $\ell\left(h_\theta\left(x^{(i)}\right), x^{(i)}\right)$ will decrease for all i
b) The sum of losses will decrease
c) The sum of losses may increase or decrease

Each step of the algorithm is going to decrease the sum of the loss and this the convergence criteria of the algorithm.

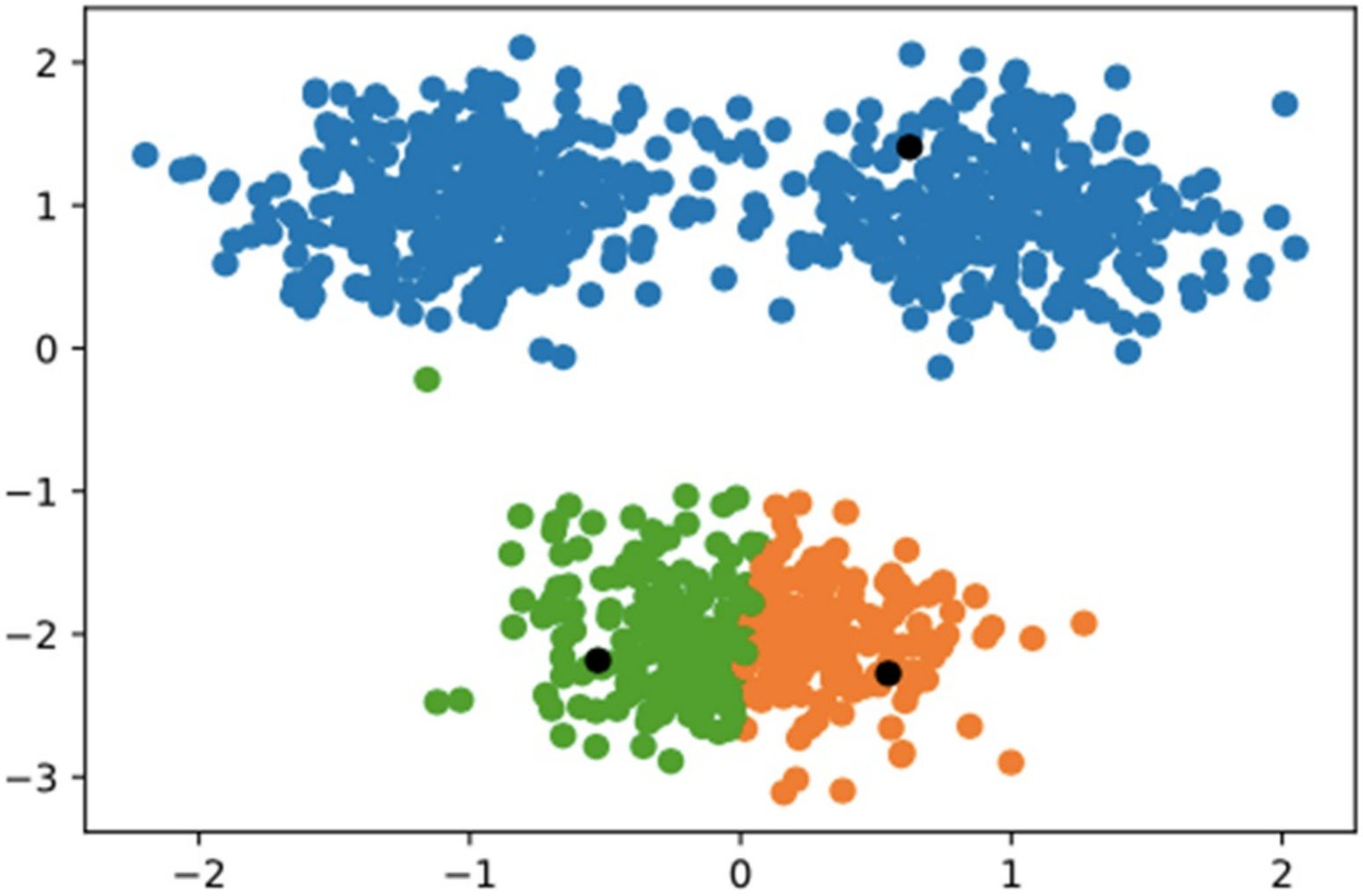
Example: k-means Clustering (1/2)



Example: k-means Clustering (2/2)



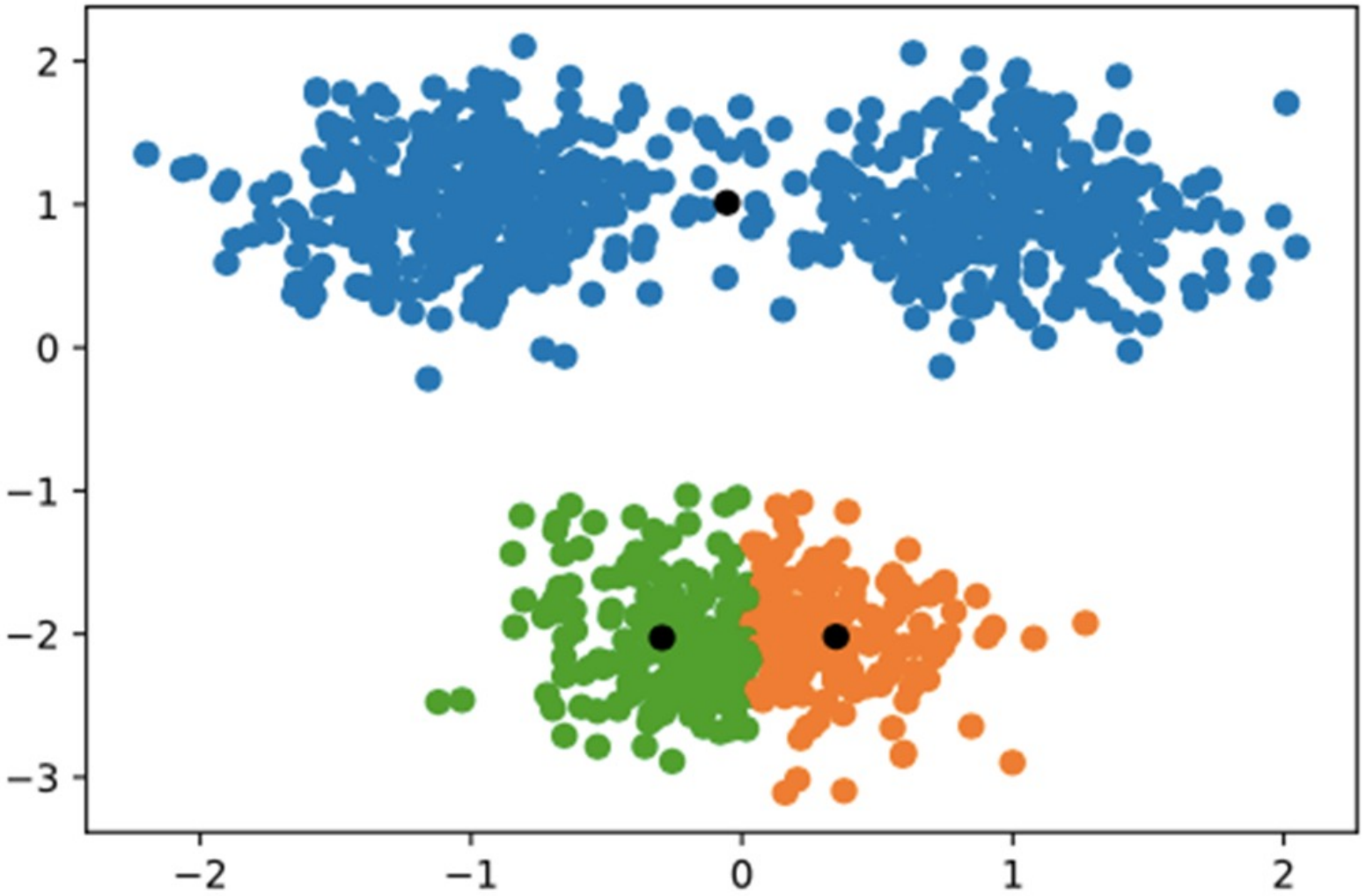
Convergence of k-means (bad)



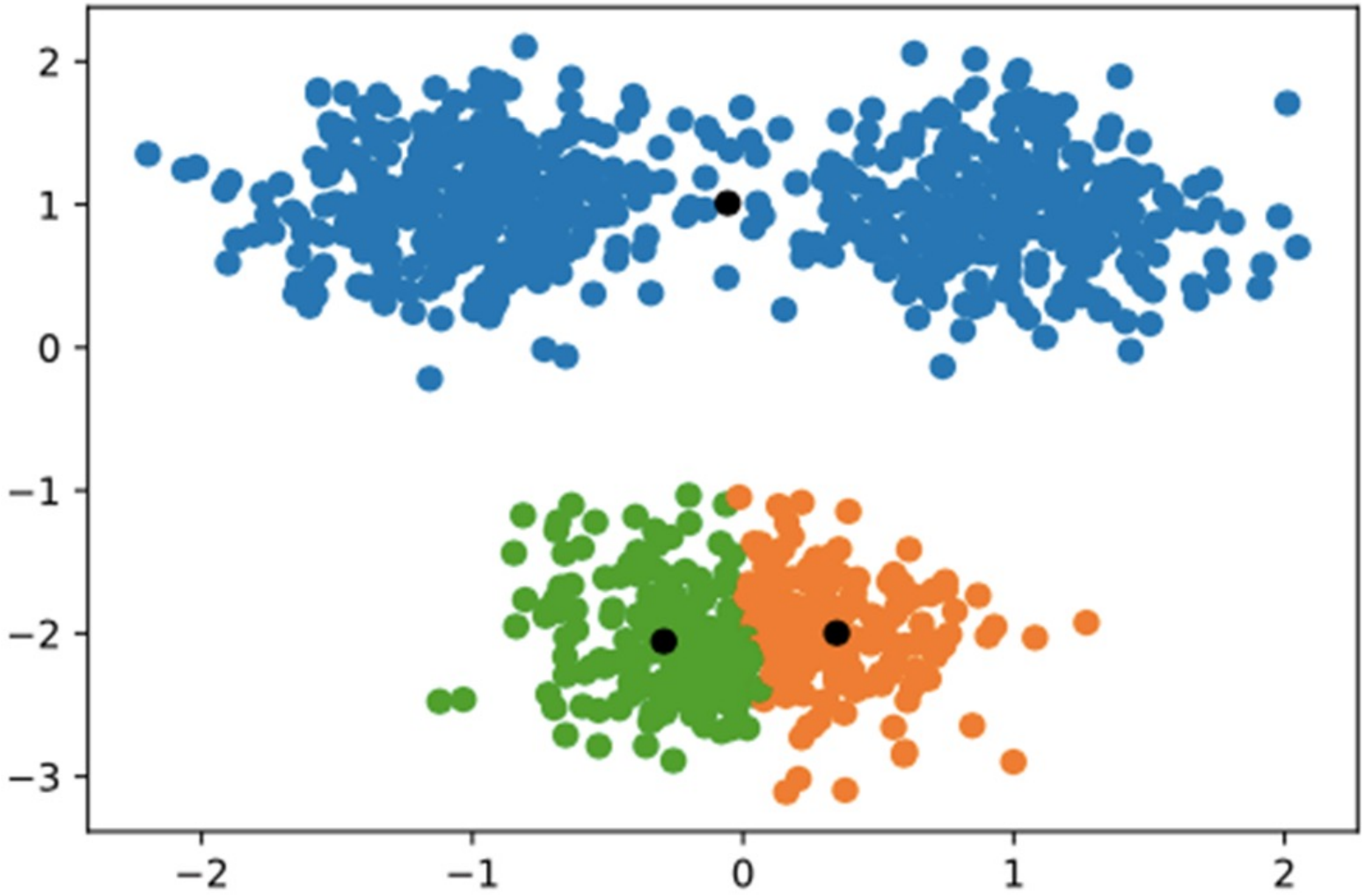
Possibility of local optima

- Since the k-means **objective function has local optima**, there is the **chance that we converge to a less-than-ideal local optima**
- **Especially for large/high-dimensional** datasets, this is not hypothetical: **k- means will usually converge to a different local optima** depending on its starting point

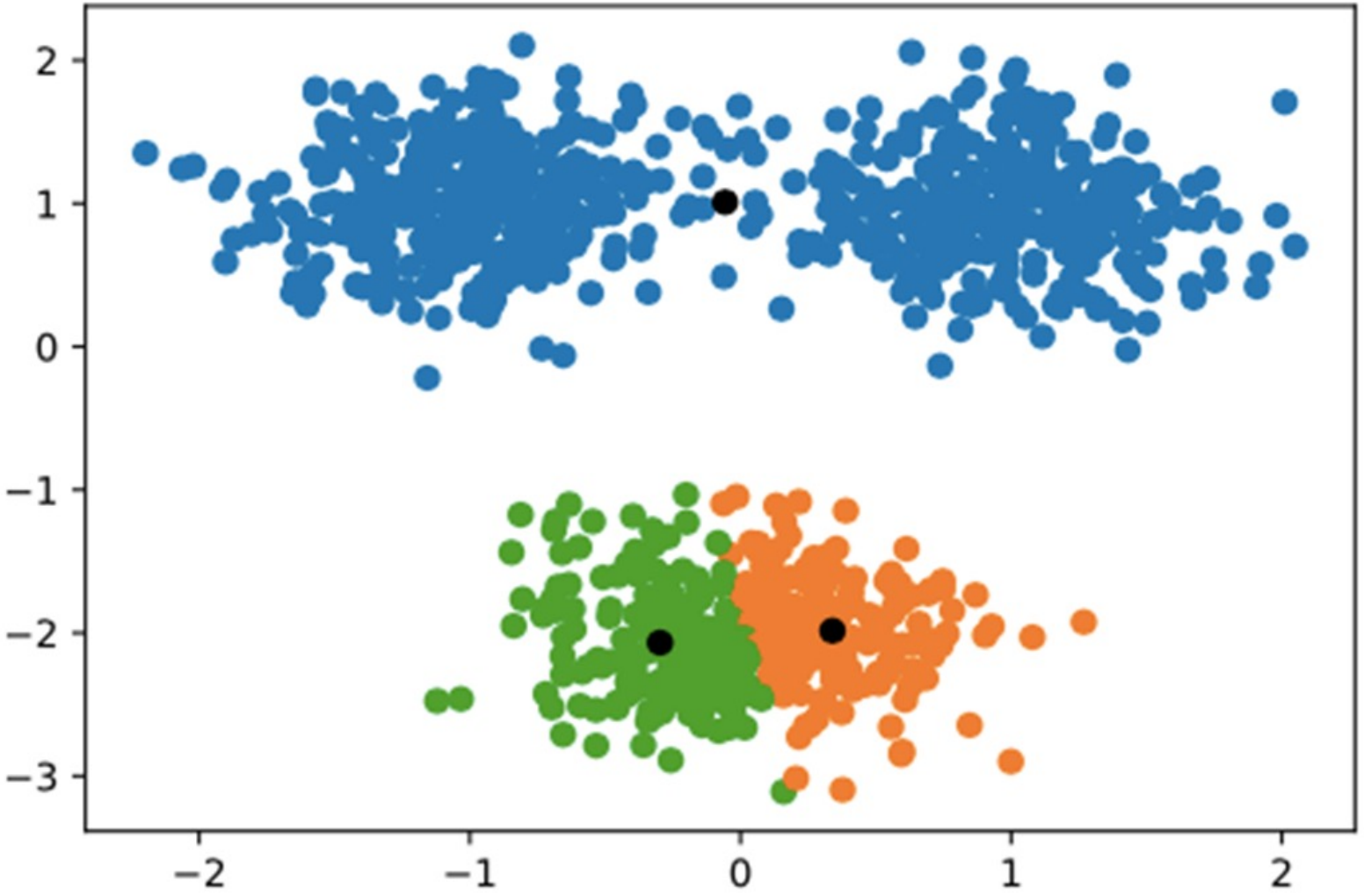
Convergence of k-means (bad)



Convergence of k-means (bad)



Convergence of k-means (bad)



Addressing poor clusters

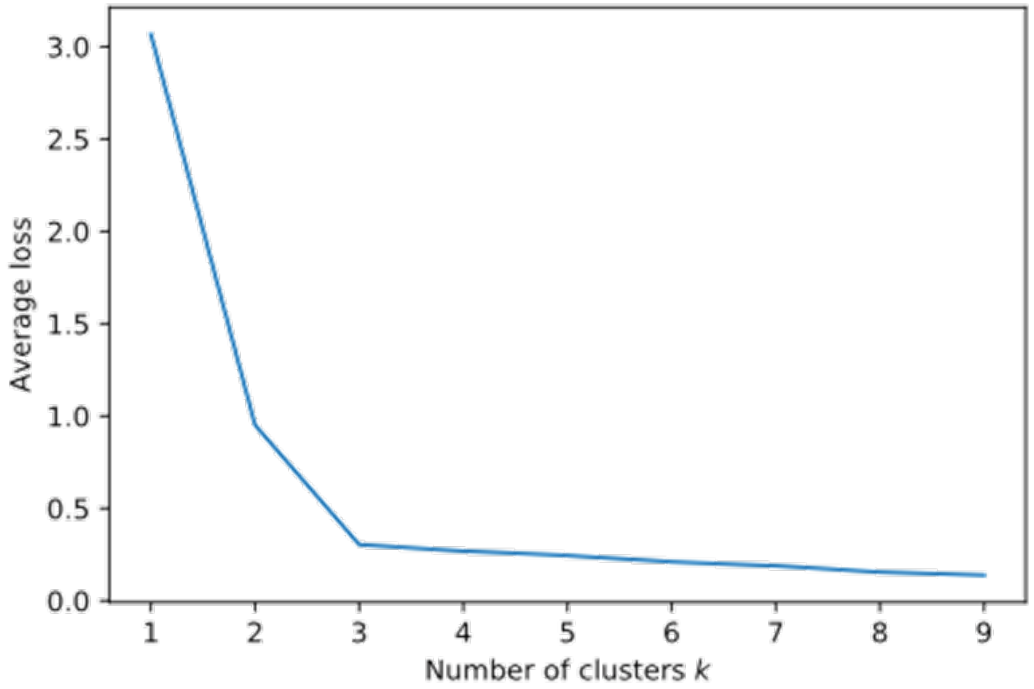
- Many approaches to address potential **poor clustering**:
  - **Randomly initialize many times**
  - **Take clustering with lowest loss**
- A **common heuristic, k-means++**: when initializing means, don't select $\mu^{(i)}$ randomly from all clusters, instead choose $\mu^{(i)}$ sequentially, sampled with probability proportion to the minimum squared distance to all other centroids
- After these centers are initialized, run k-means as normal

K-means++

Intuition

- Spreading out the *k* initial cluster centers is a good thing and helps avoid suboptimal clustering results
- Proceed as follows:
  1. Choose one center uniformly at random from among the data points
  2. For each data point *x* compute the distance between *x* and the nearest center that has already been chosen
  3. Choose one new data point at random as a new center, using a weighted probability distribution where a point *x* is chosen with probability proportional to distance squared
  4. Repeat Steps 2 and 3 until *k* centers have been chosen

How to select k?



- There's no "right" way to select k (number of clusters):
  - larger k virtually always will have lower loss than smaller k, even on a hold out set
- Instead, it's common to look at the loss function as a function of increasing k, and stop when things look "good" (lots of other heuristics, but they don't convincingly outperform this)

What happens to the loss as k approaches N, where N is the number of samples in the dataset?

a) The loss approaches 0
b) The loss approaches infinity
c) The loss converges to a low constant value
d) I am not sure

What happens to the loss as k approaches N, where N is the number of samples in the dataset?

a) The loss approaches 0
b) The loss approaches infinity
c) The loss converges to a low constant value
d) I am not sure

However, note that having k=N clusters (i.e., an identity mapping of x onto N clusters) defeats the purpose of clustering!

Summary of Partitioning Clustering

- Strength:
  - Simple, easy to implement and debug
  - Intuitive objective function: optimizes intra-cluster similarity
  - Relatively efficient: O(tkn), where n is # objects, k is # clusters, and t is # iterations. Normally, k, t << n.
- Weakness:
  - Applicable only when mean is defined, then what about categorical data?
  - Often terminates at a local optimum. Initialization is important.
  - Need to specify K, the number of clusters, in advance
  - Unable to handle noisy data and outliers
  - Not suitable to discover clusters with non-convex shapes

Hierarchical Methods

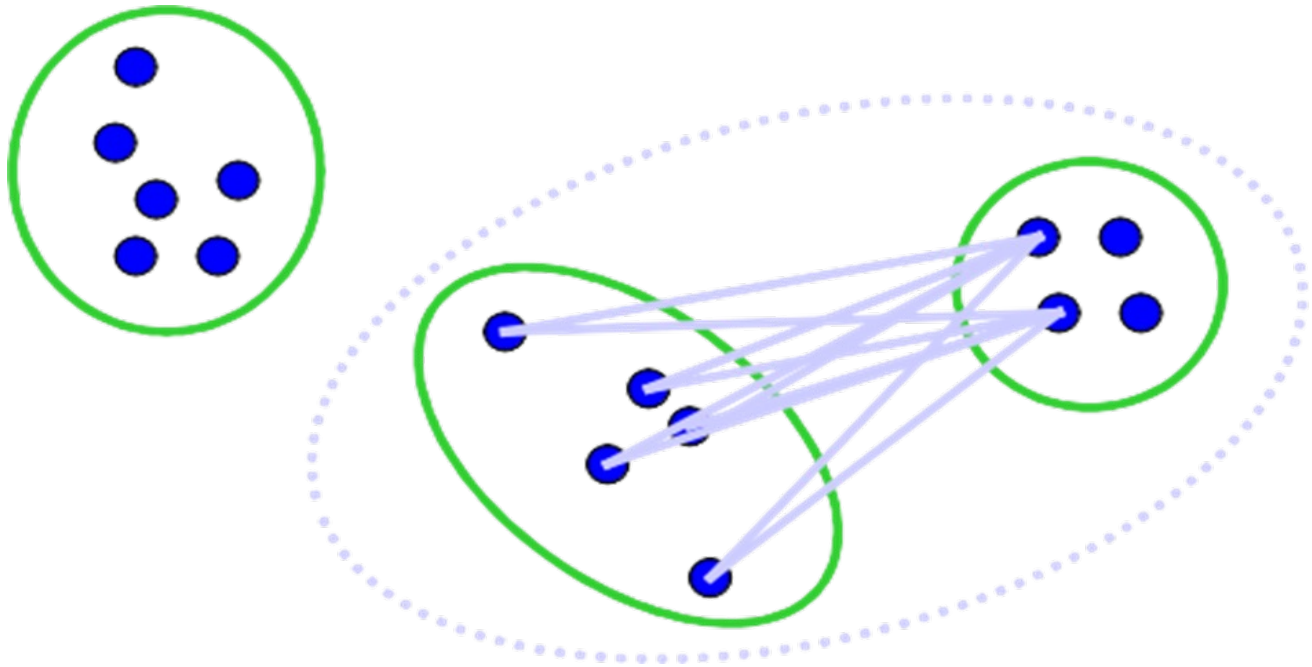
Agglomerative Methods (bottom-up)

- Agglomerative clustering starts with each observation as its own cluster.
- The two closest clusters are joined into one cluster.
- The next closest clusters are grouped together, and this process continues until there is only one cluster containing the entire data set.
- The most popular

Divisive Methods (top-down)

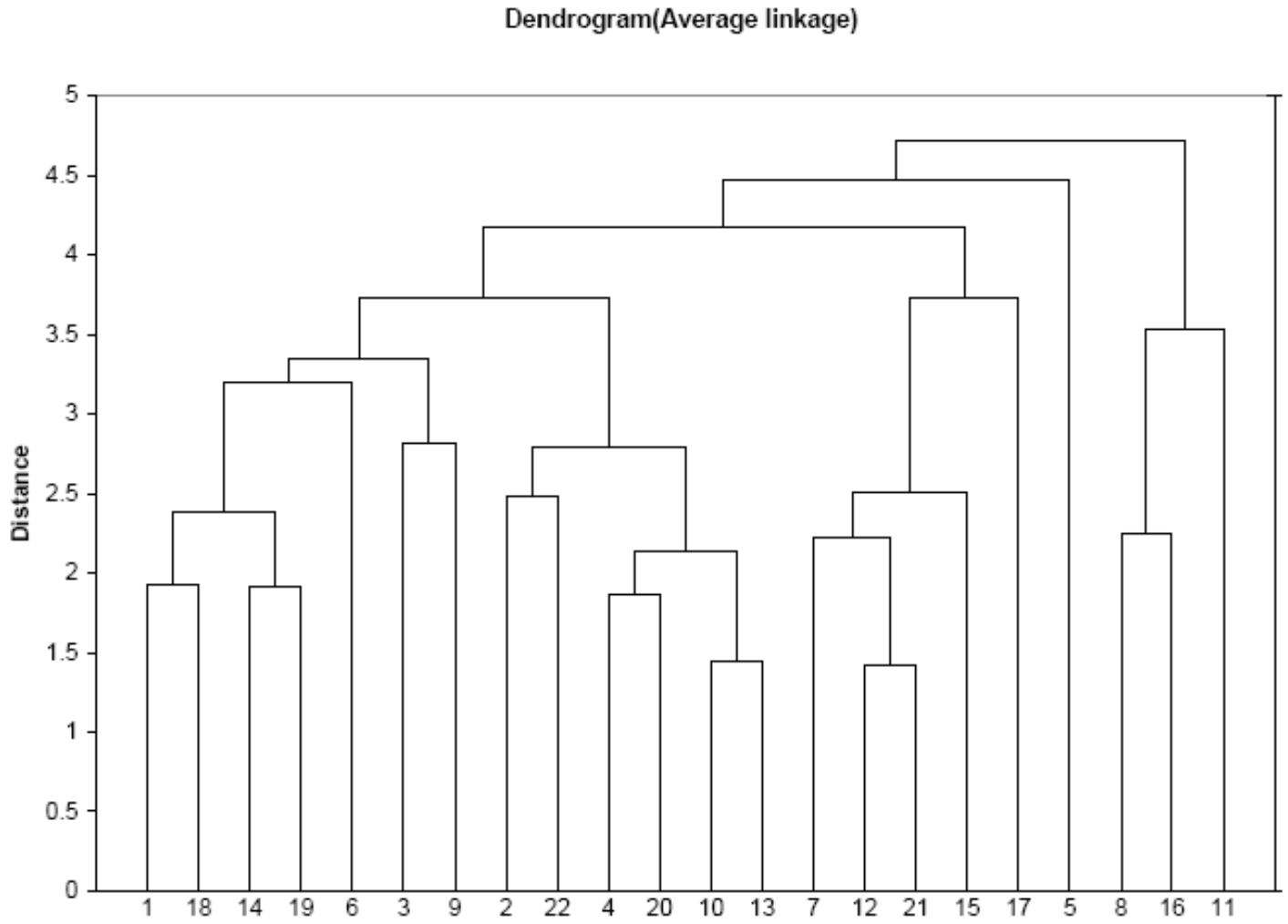
- Start with one all-inclusive cluster
- The observation with the highest average dissimilarity (farthest from the cluster by some metric) is reassigned to its own cluster.
- Any observations in the old cluster closer to the new cluster are assigned to the new cluster.
- This process repeats with the largest cluster until each observation is its own cluster.

Hierarchical Clustering – Measuring similarity

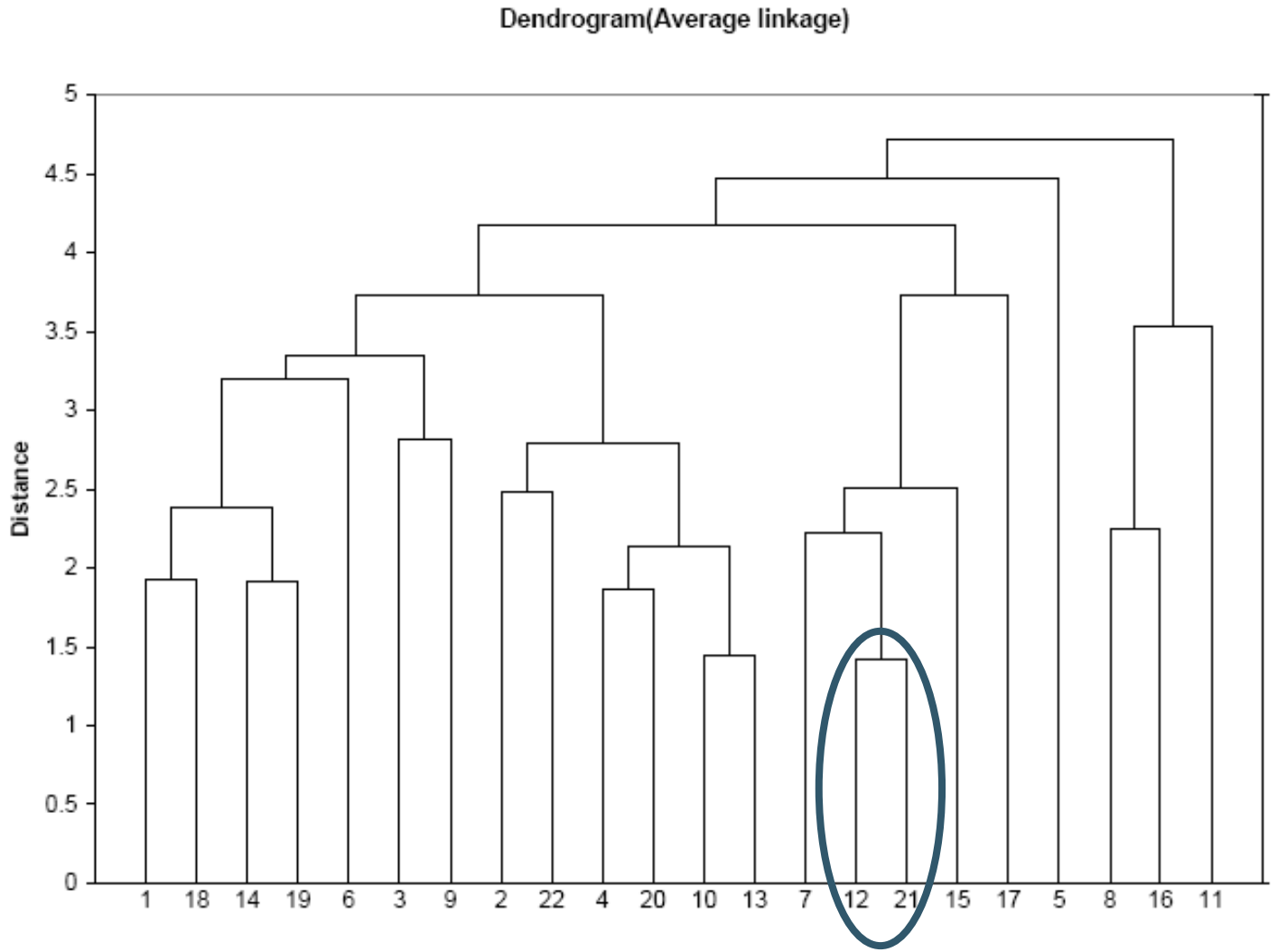


- Cluster similarity = average similarity of all pairs
- The most widely used similarity measure
- Robust against noise

A Dendrogram shows the cluster hierarchy

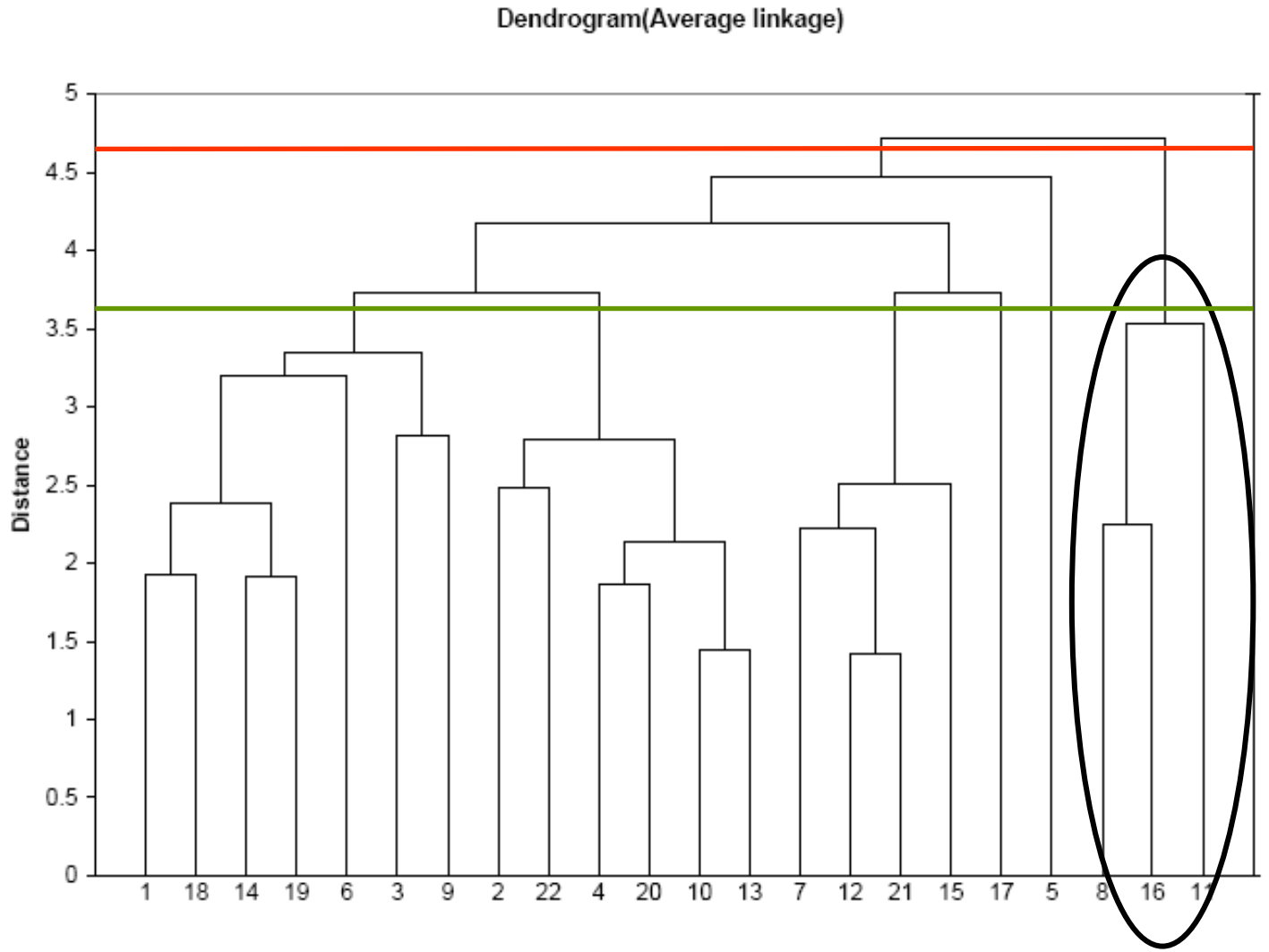


Records 12 & 21 are closest & form first cluster



- See process of clustering: Lines connected lower down are merged earlier
  - 10 and 13 will be merged next, after 12 & 21

Determining number of clusters



- For a given “distance between clusters”, a horizontal line intersects the clusters that are that far apart, to create clusters
- E.g., at distance of 4.6 (red line), data can be reduced to 2 clusters
- At distance of 3.6 (green line) data can be reduced to 6 clusters, including the circled cluster

Summary of **Hierarchical Clustering**

- No need to specify the number of clusters in advance.
- Hierarchical structure maps nicely onto human intuition for some domains
- They do not scale well: time complexity of at least O(n2), where n is the number of total objects. ?
- Like any heuristic search algorithms, local optima are a problem.
- Interpretation of results is (very) subjective.

Summary of **Hard Clustering Methods**

- Cluster analysis is an exploratory tool. Useful only when it produces **meaningful** clusters
- **Hierarchical** clustering gives visual representation of different levels of clustering
  - On other hand, due to non-iterative nature, it can be unstable, can vary highly depending on settings, and is computationally expensive
- **Non-hierarchical** is computationally cheap and more stable; requires user to set *k*
- Can use both methods
- Be wary of chance results; data may not have definitive “real” clusters

Contact

For general questions and enquiries on **research**, **teaching**, **job openings** and new **projects** refer to our website at www.is3.uni-koeln.de

For specific enquiries regarding this course contact us by sending an email to the **IS3 teaching** address at is3-teaching@wiso.uni-koeln.de

Follow us on **Twitter** at **@IS3_UniCologne** to stay up to date with recent publication and presentations of our group