Introduction

Data Science, AU, Fall 2023

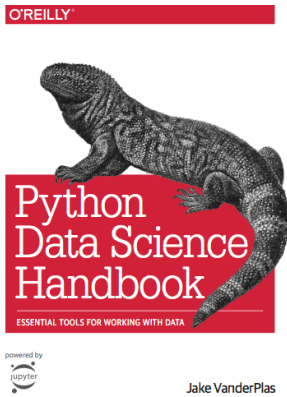
Ira Assent

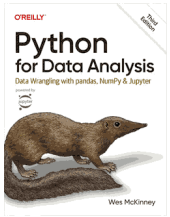
Schedule and topics

1. 11/10 Introduction, data preprocessing, PCA, clustering
2. 13/10 Unsupervised learning, more clustering, outlier detection
3. 23/10 Supervised learning, classical machine learning: DT, SVMs,…
4. 26/10 Neural networks, pitfalls, outlook

- Literature:
  - Data Science: https://jakevdp.github.io/PythonDataScienceHandbook/ Data Science in Python
  - Python: https://wesmckinney.com/book/ Python for Data Science
- Scikit-learn: data science methods in python (machine learning)
  - https://scikit-learn.org
  - Builds on NumPy, SciPy, matplotlib
- Pandas: data analysis, data manipulation in python
  - https://pandas.pydata.org/

Both widely used, many methods, stable, large community, extensive documentation





Staff and setup

- Ira Assent, Professor, Data-Intensive Systems, Department of Computer Science, ira@cs.au.dk
  - Head of Big Data Analysis, DIGIT Aarhus University Centre for Digitalisation, Big Data and Data Analytics
  - Director, Data Analytics and Machine Learning (IAS-8), Research Center Juelich, Germany (part-time)
- Maximilian Egger, PhD student, TA

- Please also discuss in the course forum on brightspace
- In the mornings, we will mostly learn about the different data science techniques and evaluation measures
- In the afternoons, we will mostly work with tutorials

Learning goals

- Data science
  - What it is, what we can do
    - Tools at the analytical level
      - How can I get an overview over data, check its properties, analyze trends, find joint behavior,…
  - Ensure proper data handling and science
    - Data analytics is powerful and can produce impressive results
      - But if you set it up incorrectly, the conclusions may be WRONG!
        - This has profound impact on your research
    - We will focus also on how to choose the right model for your task, how to check data quality and validate findings
      - This requires understanding the underlying principles
        - i.e. how is the data used to reach conclusions
        - Important in order to be able to reason appropriately

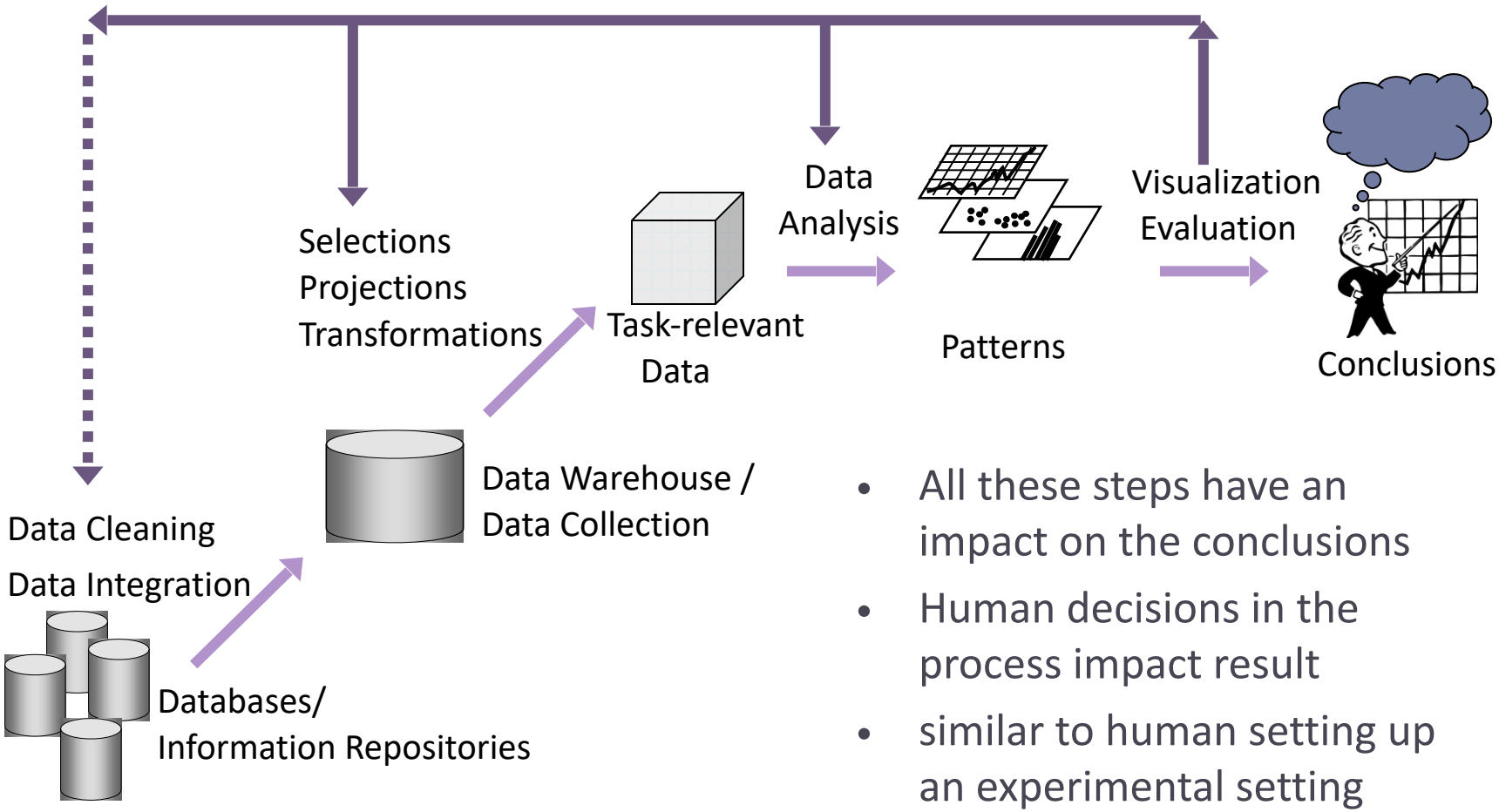
What Is Data Science?

Data science is an **interdisciplinary** field that uses **scientific** methods, processes, algorithms and systems to extract or extrapolate **knowledge and insights** from noisy, structured and unstructured data, and apply knowledge from data across a broad range of application domains. Data science is related to data mining, machine learning and big data.

Wikipedia

- **Data mining** (knowledge discovery in databases)
  - Extraction of interesting (*non-trivial*, *implicit*, *previously unknown* and *potentially useful*) information or patterns from data in *large databases*
- Machine learning
- Artificial intelligence
- Big Data

Data Science as an iterative process

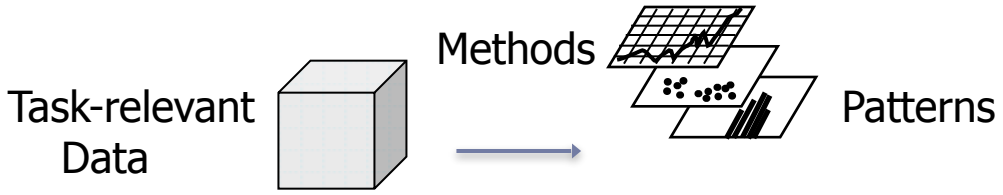


- All these steps have an impact on the conclusions
- Human decisions in the process impact result
- similar to human setting up an experimental setting

Data Science methods

- Clustering
- Outlier detection
- Classification

- Association Rules
- Sequential patterns
- Trends and analysis of changes
- Methods for special data types, e.g., spatial data mining, web mining
- …



Are All the "Discovered" Patterns Interesting?

- Data mining may generate thousands of patterns: Not all of them are interesting
  - Suggested approach: Human-centered, query-based, focused mining
- **Interestingness measures**
  - A pattern is interesting if it is easily understood by humans, valid on new or test data with some degree of certainty, potentially useful, novel, or validates some hypothesis that a user seeks to confirm
- **Objective vs. subjective interestingness measures**
  - Objective: based on statistics and structures of patterns, e.g., support, confidence, etc.
  - Subjective: based on user's belief in the data, e.g., unexpectedness, novelty, actionability, etc.

Data Exploration

Cleaning and profiling

Visualization

Analysis

- A large part of data mining (or data analysis in general) is devoted to data pre-processing
  - Clean data
  - Select relevant data, profile
- Similarly, visualization is crucial for human understanding, feedback and iterative data mining steps
- In explorative analysis, user plays a central role (human-in-the-loop)

Preprocessing

- We characterize the data (=the sample)
- In particular, characterize the data via (statistical) measures

- It is very important to characterize the data
  - Some of the following data science approaches make strong assumptions
    - E.g. classes are balanced
      - Equal number of data samples per class
    - Or attribute ranges are identical
      - Means impact of (dis-)similarity between attributes is the same
  - By characterizing the data, we see whether these assumptions hold
    - Otherwise, we may be producing poor or even incorrect analyses!

Data example

- Iris setosa

- Iris versicolor

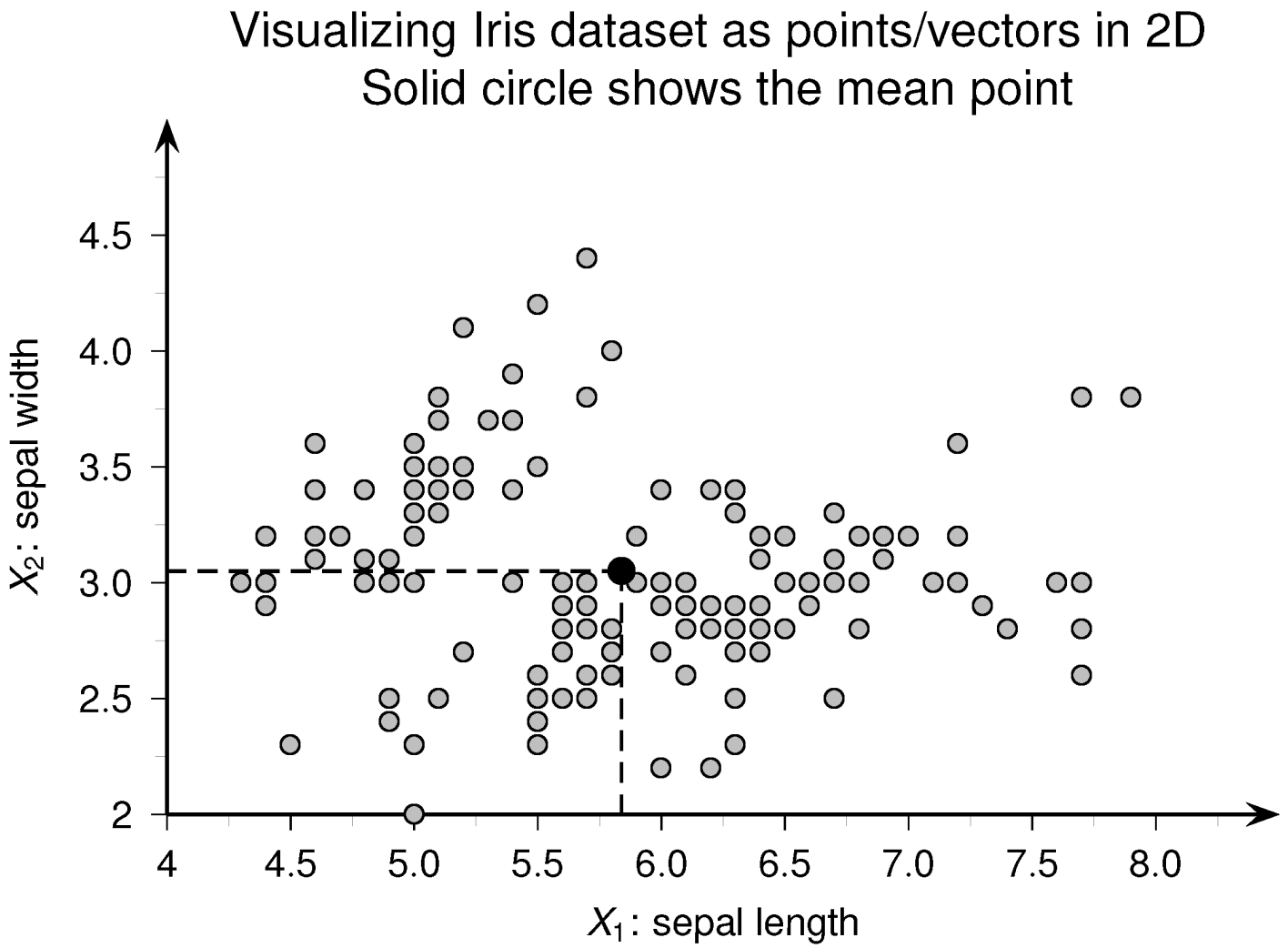
- Iris virginica

- Four attributes:
  - Sepal length, Sepal width, Petal length, Petal width
- Selecting two attributes for visualization, sometimes also reduced to two attributes using PCA (two principle components)

Image sources: Wikipedia

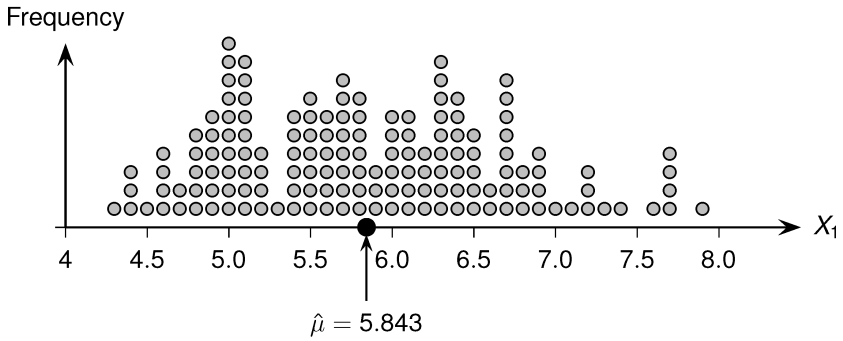
Data: e.g. from Data Mining book at https://dataminingbook.info/book_html/
Mohammed J. Zaki, Wagner Meira, Jr., Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd Edition, Cambridge University Press, March 2020. ISBN: 978-1108473989.

2D iris data scatterplot



Measures of central tendency

- Where is the data concentrated?
  - Typical values, etc
- Starting point to understand data
- **Mean / expected value**
  - one-number summary of the location / central tendency for the distribution of X



Sample mean for iris sepal length

  - Discrete

    where f(x) is $\mu = E[X] = \sum_x x \cdot f(x)$ ction of X

  - Continuous

    where f(x) is $\mu = E[X] = \int_{-\infty}^{\infty} x \cdot f(x)\, dx$ of X

- **Sample mean** is st… … “true” mean
- **Not necessarily robust**: outliers can distort sample mean

Median

- Robust measure of centrality, not so much affected by outliers
- An "actual" value in the data set
  - E.g. a mean value of 2.1 kids per family on average is not an actual value in the data
- **Median** m value such that $P(X \leq m) \geq ½$ and $P(X \geq m) \geq ½$
  - So probability of smaller values and larger values about half
  - The median m is the "middle-most" value; half of the values of X are less and half of the values of X are more than m.
  - Simple computation: sort all values
    - Median of n values is at position (n+1)/2 if n is odd, otherwise n/2 or n/2 +1 both median values
- So the median number of kids could be 1
  - Means: most families have one child, few families have many kids
  - So considering both measures provides insights

Digging deeper down into data distributions

- How concentrated are the values around the mean?
- **Variance** provides a measure of how much the values of variable X deviate from the me $\sigma^2 = \text{var}(X) = E\left[(X-\mu)^2\right]$
- **Standard deviation** σ: positive square root of variance
- Looking at more than one variable / attribute: measures of association
  - How do they relate / not relate?
- **Covariance** between two attributes X_1 and X

$$\sigma_{12} = E[(X_1-\mu_1)(X_2-\mu_2)] = E[X_1X_2] - E[X_1]E[X_2]$$

  - measure of association / linear dependence
  - If X_1 and X_2 are independent, then $E[X_1X_2]=E[X_1]\cdot E[X_2]$, so $\sigma_{12} = 0$
- **Correlation** is standardized covariance

$$\rho_{12} = \frac{\sigma_{12}}{\sigma_1\sigma_2}$$

  - normalizing the covariance with standard deviation of each variable
- Provides important information on relationship between attributes – possible bias to data analysis methods

Covariance matrix

The variance–covariance information for the two attributes X_1 and X_2 can be summarized in the square 2×2 *covariance matrix*

$$\begin{aligned}\boldsymbol{\Sigma} &= E[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^T] \\ &= \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{pmatrix}\end{aligned}$$

Because $\sigma_{12} = \sigma_{21}$, $\boldsymbol{\Sigma}$ is *symmetric*.
The *total variance* is given as

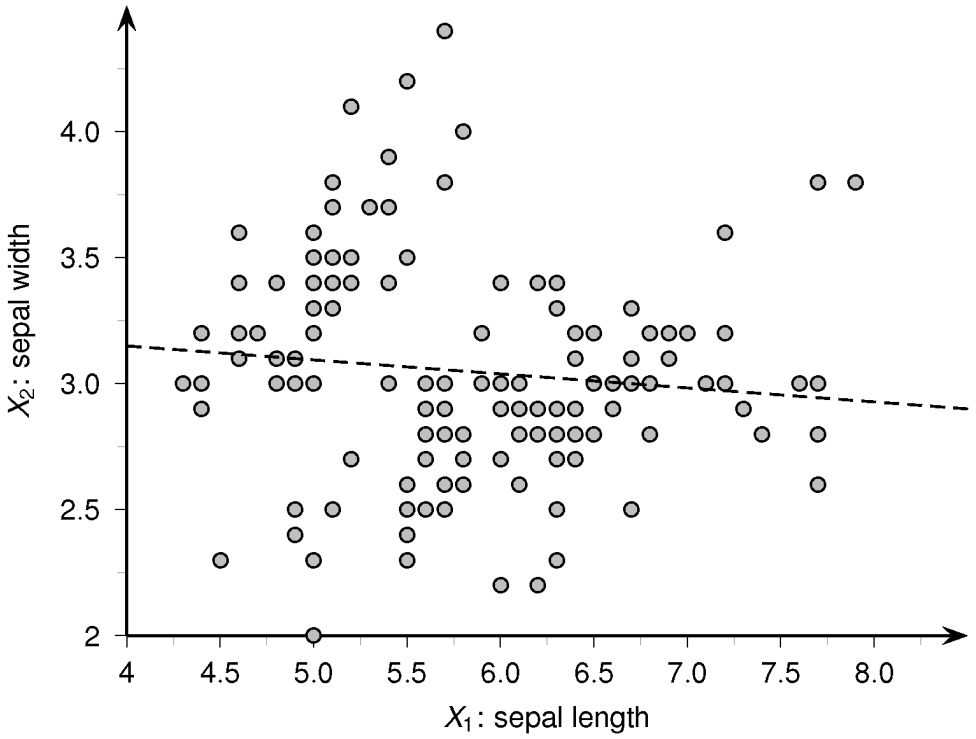
$$var(\mathbf{D}) = tr(\boldsymbol{\Sigma}) = \sigma_1^2 + \sigma_2^2$$

We immediately have $tr(\boldsymbol{\Sigma}) \geq 0$.
The *generalized variance* is

$$|\boldsymbol{\Sigma}| = \det(\boldsymbol{\Sigma}) = \sigma_1^2\sigma_2^2 - \sigma_{12}^2 = \sigma_1^2\sigma_2^2 - \rho_{12}^2\sigma_1^2\sigma_2^2 = (1 - \rho_{12}^2)\sigma_1^2\sigma_2^2$$

Note that $|\rho_{12}| \leq 1$ implies that $\det(\boldsymbol{\Sigma}) \geq 0$.

Correlation sepal length and sepal width



The sample mean is

$$\hat{\boldsymbol{\mu}} = \begin{pmatrix} 5.843 \\ 3.054 \end{pmatrix}$$

The sample covariance matrix is

$$\hat{\boldsymbol{\Sigma}} = \begin{pmatrix} 0.681 & -0.039 \\ -0.039 & 0.187 \end{pmatrix}$$

The sample correlation is

$$\hat{\rho}_{12} = \frac{-0.039}{\sqrt{0.681 \cdot 0.187}} = -0.109$$

Data normalization

If the attribute values are in vastly different scales, then it is necessary to normalize them.

Range Normalization: Let X be an attribute and let $x_1, x_2, \ldots, x_n$ be a random sample drawn from X. In *range normalization* each value is scaled by the sample range $\hat{r}$ of X:

$$x_i' = \frac{x_i - \min_i\{x_i\}}{\hat{r}} = \frac{x_i - \min_i\{x_i\}}{\max_i\{x_i\} - \min_i\{x_i\}}$$

After transformation the new attribute takes on values in the range $[0, 1]$.

Standard Score Normalization: Also called z-normalization; each value is replaced by its z-score:

$$x_i' = \frac{x_i - \hat{\mu}}{\hat{\sigma}}$$

where $\hat{\mu}$ is the sample mean and $\hat{\sigma}^2$ is the sample variance of X. After transformation, the new attribute has mean $\hat{\mu}' = 0$, and standard deviation $\hat{\sigma}' = 1$.

Contingency Analysis

- How to study properties such as (in)dependence of categorical data?
- We look at counts: how often do certain values appear?
- **Contingency analysis** looks at frequencies
  - For variables X_1 and X_2:
    matrix of entries n_{ij}

$$_2 = n \cdot \widehat{\mathbf{P}}_{12} = \begin{pmatrix} n_{11} & n_{12} & \cdots & n_{1m_2} \\ n_{21} & n_{22} & \cdots & n_{2m_2} \\ \vdots & \vdots & \ddots & \vdots \\ n_{m_1 1} & n_{m_1 2} & \cdots & n_{m_1 m_2} \end{pmatrix}$$

Sepal length (X_1)	Sepal width (X_2)			
	Short a_{21}	Medium a_{22}	Long a_{23}	Row Counts
Very Short (a_{11})	7	33	5	$n_1^1 = 45$
Short (a_{12})	24	18	8	$n_2^1 = 50$
Long (a_{13})	13	30	0	$n_3^1 = 43$
Very Long (a_{14})	3	7	2	$n_4^1 = 12$
Column Counts	$n_1^2 = 47$	$n_2^2 = 88$	$n_3^2 = 15$	$n = 150$

Testing independence using contingency

- If X1 and X2 are independent

$$\hat{p}_{ij} = \hat{p}_i^1 \cdot \hat{p}_j^2$$

- Chi-Squared Test for Independence

The expected frequency for each pair of values is

$$e_{ij} = n \cdot \hat{p}_{ij} = n \cdot \hat{p}_i^1 \cdot \hat{p}_j^2 = n \cdot \frac{n_i^1}{n} \cdot \frac{n_j^2}{n} = \frac{n_i^1 n_j^2}{n}$$

The χ^2 statistic quantifies the difference between observed and expected counts

$$\chi^2 = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \frac{(n_{ij} - e_{ij})^2}{e_{ij}}$$

The sampling distribution for the χ^2 statistic follows the *chi-squared* density function:

$$f(x|q) = \frac{1}{2^{q/2}\Gamma(q/2)} x^{\frac{q}{2}-1} \mathrm{e}^{-\frac{x}{2}}$$

where q is the degrees of freedom

$$\begin{aligned} q &= |dom(X_1)| \times |dom(X_2)| - (|dom(X_1)| + |dom(X_2)|) + 1 \\ &= m_1 m_2 - m_1 - m_2 + 1 \\ &= (m_1 - 1)(m_2 - 1) \end{aligned}$$

Chi-Squared Test: sepal length and sepal width

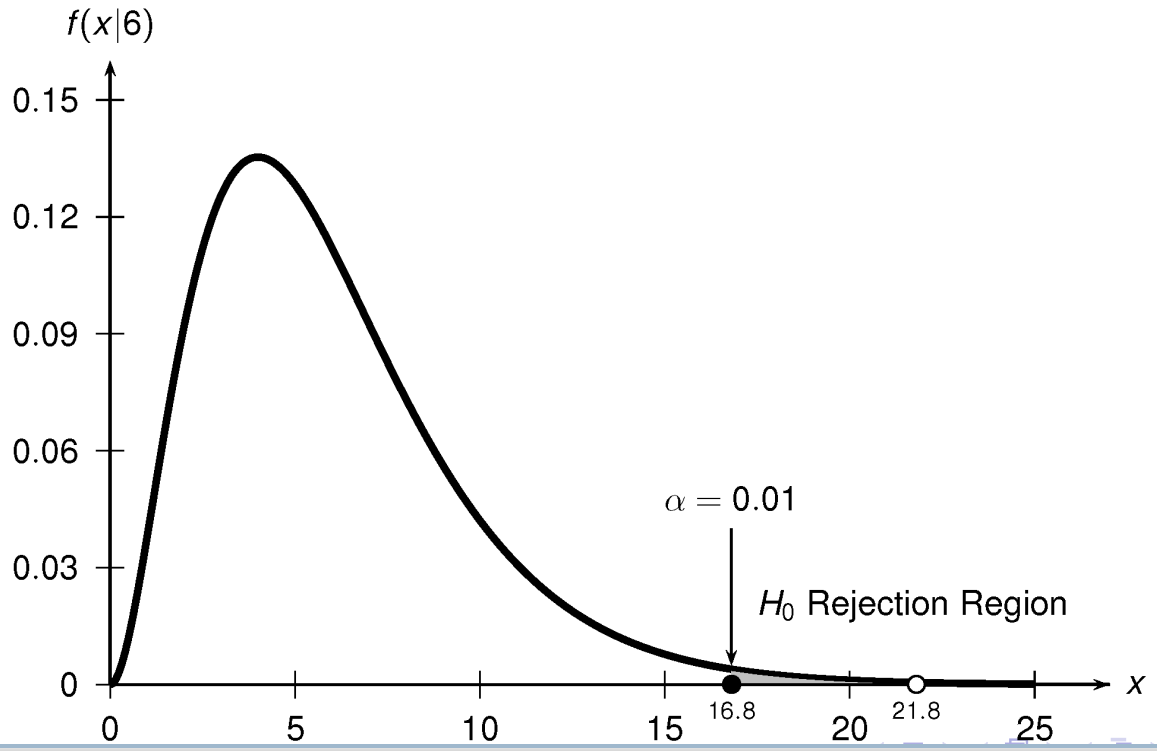
	Expected Counts	X_2		
		Short (a_{21})	Medium (a_{22})	Short (a_{23})
X_1	Very Short (a_{11})	14.1	26.4	4.5
	Short (a_{12})	15.67	29.33	5.0
	Long (a_{13})	13.47	25.23	4.3
	Very Long (a_{14})	3.76	7.04	1.2

	Observed Counts	X_2		
		Short (a_{21})	Medium (a_{22})	Long (a_{23})
	Very Short (a_{11})	7	33	5
	Short (a_{12})	24	18	8
	Long (a_{13})	13	30	0
	Very Long (a_{14})	3	7	2

The chi-squared statistic value is $\chi^2 = 21.8$.

Chi-Squared Distribution (q=6)

The *p-value* of a statistic θ is defined as the probability of obtaining a value at least as extreme as the observed value.
The null hypothesis, that X_1 and X_2 are independent, is rejected if $p\text{-value}(z) \leq \alpha$, say $\alpha = 0.01$. We have $p\text{-}value(21.8) = 0.0013$. Thus, we reject the null hypothesis, and conclude that X_1 and X_2 are dependent.



22

Measuring Similarity

- To measure similarity, often a distance function *dist* is used
  - Measures "dissimilarity" between pairs objects *x* and *y*
    - Small distance *dist*(*x*, *y*): objects x and y are more similar
    - Large distance *dist*(*x*, *y*): objects x and y are less similar
- Definition of a distance function is highly application dependent
  - Typically requires standardization/normalization of attributes!
  - Different definitions for interval-scaled, boolean, categorical, ordinal and ratio variables
- So, you need to choose an appropriate similarity measure if you would like the analysis to reflect how the data should be "understood"
  - An incorrect choice may lead to unusable or incorrect results!

Minkowski distances

- For standardized numerical attributes, i.e., vectors $x = (x_1, ..., x_d)$ and $y = (y_1, ..., y_d)$ from a d-dimensional vector space:
  - General L_p-Metric (Minkowski-Distance)
  - $p = 2$: Euclidean Distance (cf. Pythagoras)
  - $p = 1$: Manhattan-Distance (city block)
  - $p \to \infty$: Maximum-Metric

Other common distance functions

- For sets x and y (Jaccard Index):

- For text documents:
  - A document D is represented by a vector r(D) of frequencies of the terms occuring in D, e.g.,

    where f (t_i, D) is the frequency of term t_i in document D
  - Now call two of these vectors of frequencies A, B, we can compare them using

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

  Typically better for high-dimensional vectors than e.g. Euclidean distance

Discretization

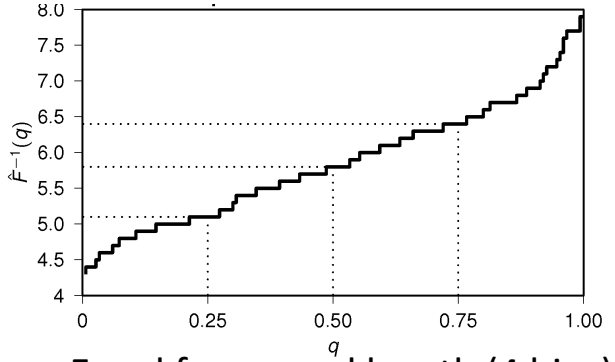
Discretization, also called *binning*, converts numeric attributes into categorical ones.

Equal-Width Intervals: Partition the range of X into k *equal-width* intervals. The interval width is simply the range of X divided by k:

$$w = \frac{x_{\max} - x_{\min}}{k}$$

Thus, the ith interval boundary is given as

$$v_i = x_{\min} + iw, \text{ for } i = 1, \ldots, k-1$$



Equal freq. sepal length (4 bins)

Equal-Frequency Intervals: We divide the range of X into intervals that contain (approximately) equal number of points. The intervals are computed from the empirical quantile or inverse cumulative distribution function

$$\hat{F}^{-1}(q) = \min\{x \mid P(X \leq x) \geq q\}$$

Bin	Width	Count
[4.3, 5.1]	0.8	$n_1 = 41$
(5.1, 5.8]	0.7	$n_2 = 39$
(5.8, 6.4]	0.6	$n_3 = 35$
(6.4, 7.9]	1.5	$n_4 = 35$

We require that each interval contain $1/k$ of the probability mass; therefore, the interval boundaries are given as follows:

$$v_i = \hat{F}^{-1}(i/k) \text{ for } i = 1, \ldots, k-1$$

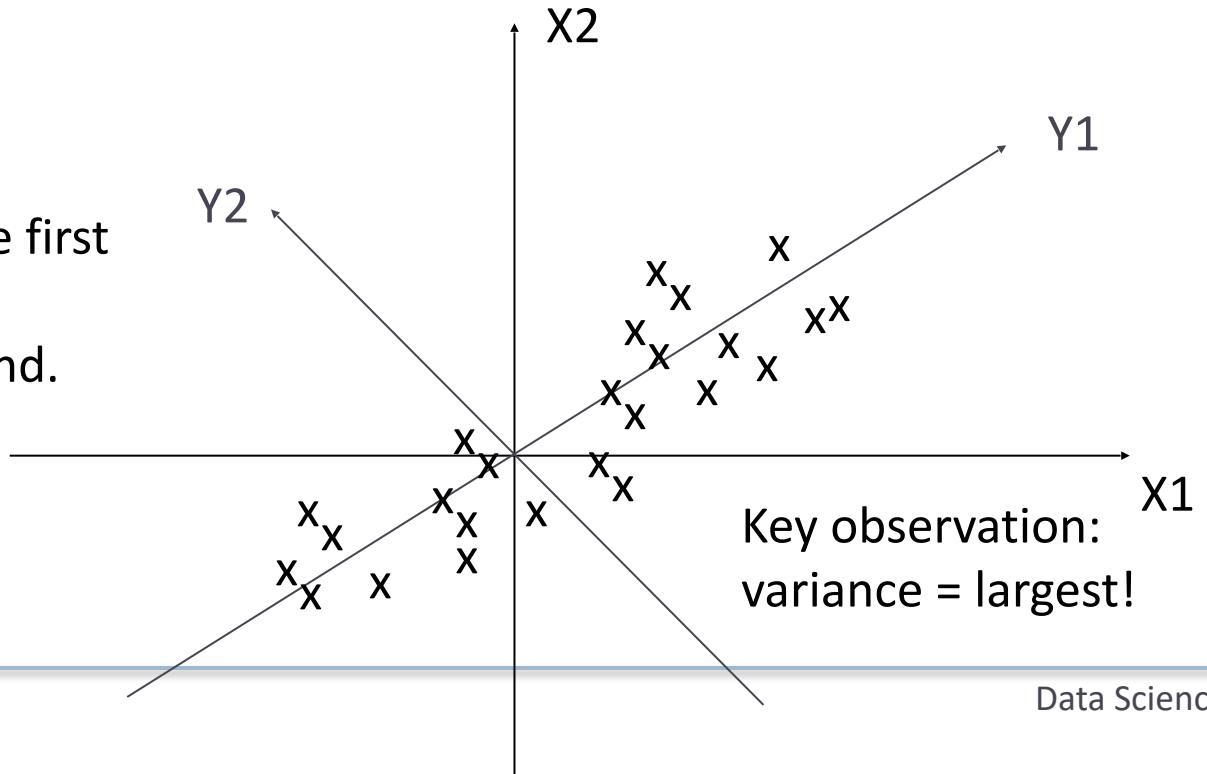
Principal Components Analysis (PCA)

- An exploratory technique used to reduce the dimensionality of the data set to 2D or 3D
- Can be used to:
  - Reduce number of dimensions in data
  - Find patterns in high-dimensional data
  - Visualize data of high dimensionality
- Example applications:
  - Face recognition
  - Image compression
  - Gene expression analysis

Principal Components Analysis Idea

- Does the data set 'span' the whole of d dimensional space?
- For matrix of *m* samples x *n* objects, create new covariance matrix of size *n* x *n*
- Transform some large number of variables into a smaller number of uncorrelated variables called principal components (PCs).
- developed to capture as much of the variation in data as possible

Note: Y1 is the first eigen vector,
Y2 is the second.
Y2 ignorable.



Principal components

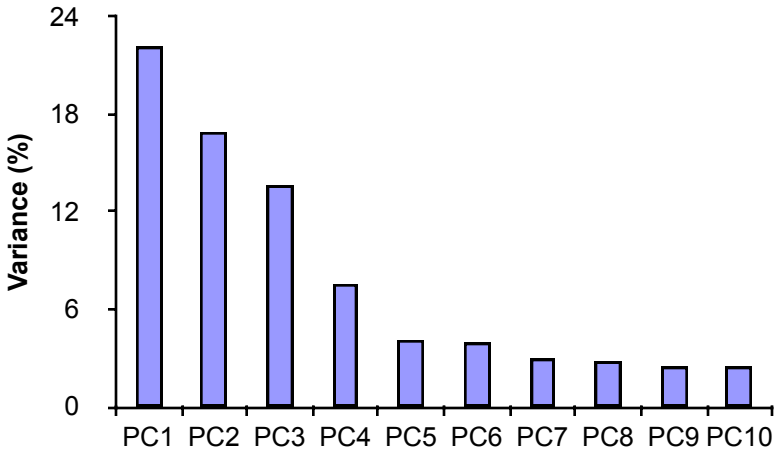
- 1. principal component (PC1)
  - The eigenvalue with largest absolute value indicates data has largest variance along its eigenvector, the direction along which there is greatest variation
- 2. principal component (PC2)
  - the direction with maximum variation left in data, orthogonal to the 1. PC
- …
- In general, only few directions manage to capture most of the variability in the data
  - It is the covariance matrix that captures the variance in multiple dimensions

Steps of PCA

- Let be the mean vector (taking the mean of all rows)
- Adjust the original data by the mean (normalize)

  X′ = X –
- Compute the covariance matrix C of adjusted X
- Find the eigenvectors and eigenvalues of C

- For matrix *C, v*ectors **e** (=column vector) having same direction as *C***e** :
  - *eigenvectors* of *C* is **e** such that *C***e**=λ**e**,
  - λ is called an *eigenvalue* of *C*



Using principal components

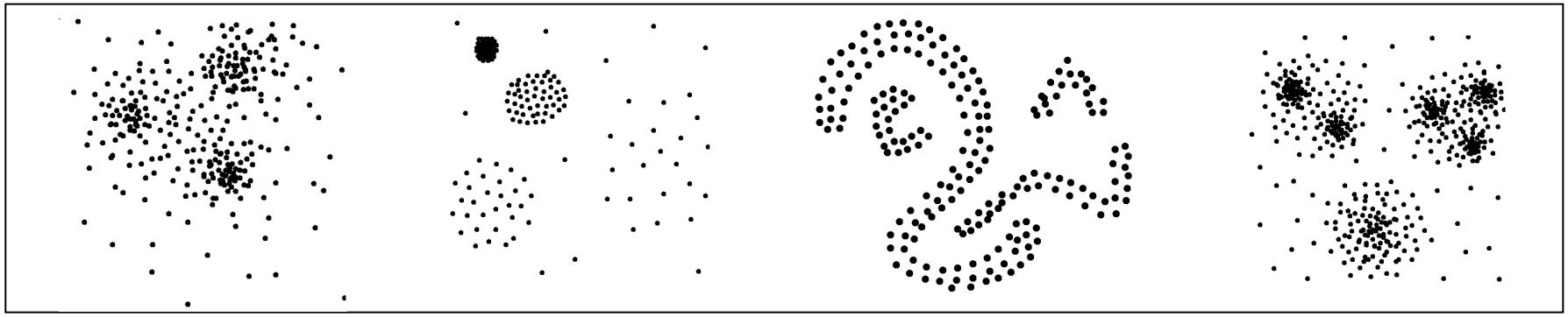
- General about principal components
  - summary variables
  - linear combinations of the original variables
  - uncorrelated with each other
  - capture as much of the original variance as possible

- Retrieving old data (e.g. in data compression)
  - *RetrievedRowData* = (*RowFeatureVector*^{T} x *FinalData*) + *OriginalMean*
  - Yields original data using the chosen components

Breathe deep

What is Clustering?

- Grouping a set of data objects into clusters
  - Cluster: a collection of data objects
    - Similar to one another within the same cluster
    - Dissimilar to the objects in other clusters
- Clustering = ***unsupervised classification*** (no predefined classes)
- Typical usage
  - As a *stand-alone tool* to get insight into data distribution
  - As a *preprocessing step* for other algorithms

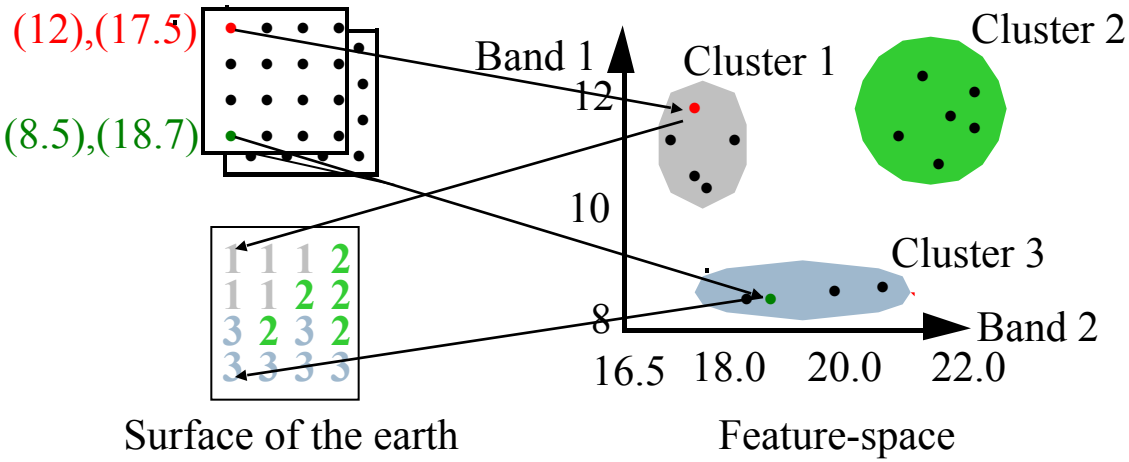


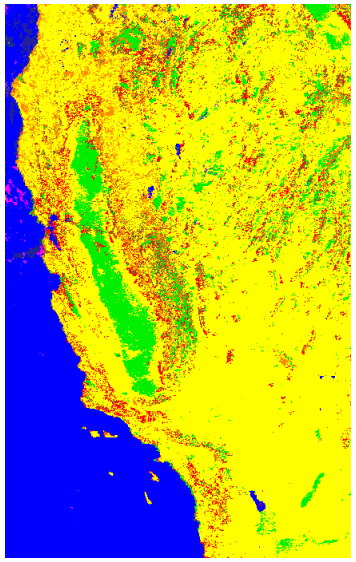
General Applications of Clustering

- Pattern Recognition and Image Processing
- Spatial Data Analysis
  - create thematic maps in GIS (Geographic Information Systems) by clustering feature spaces
  - detect spatial clusters and explain them in spatial data mining
- Biology
  - Clustering of gene expression data
  - taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs

A Typical Application: Thematic Maps

- Satellite images of a region in different wavelengths
  - Each point on the surface maps to a high-dimensional feature vector $p = (x_1, \ldots, x_d)$ where x_i is the recorded intensity at the surface point in band i.
  - Assumption: each different land-use reflects and emits light of different wavelengths in a characteristic way.





Major Clustering Approaches

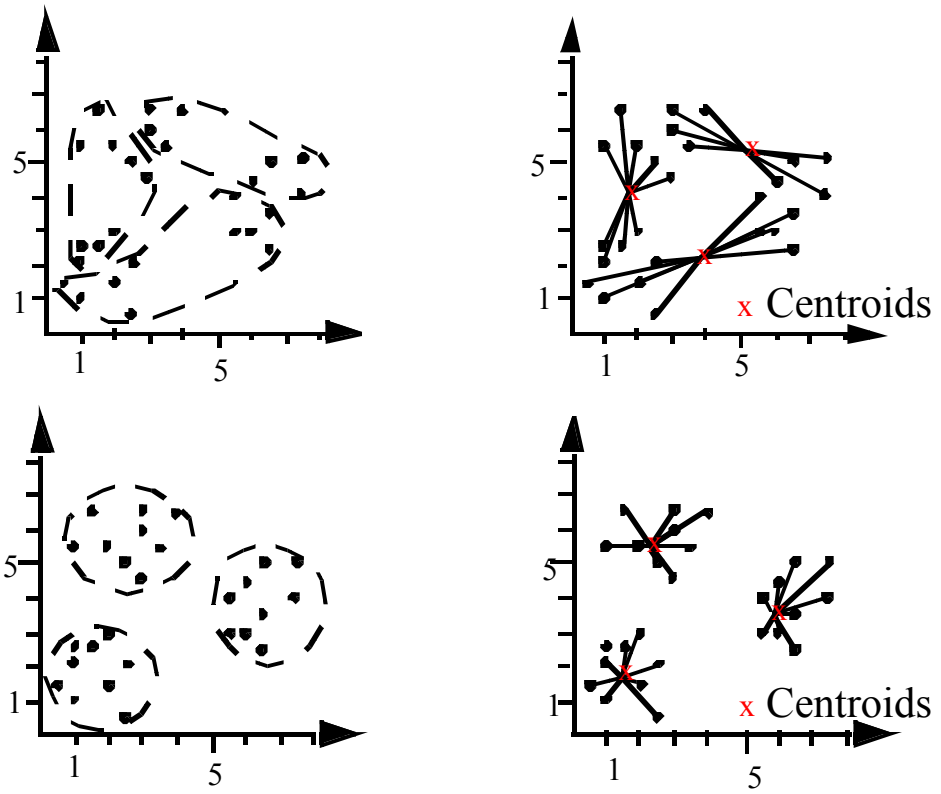
- Representative-based/Partitioning algorithms
  - Find *k* partitions, minimizing some objective function
  - Expectation Maximization
- Density-based
  - Find clusters based on connectivity and density functions
- Hierarchy algorithms
  - Create a hierarchical decomposition of the set of objects
- Subspace Clustering
- Other methods
  - Grid-based
  - Neural networks (SOM's)
  - Graph-theoretical methods
  - . . .

Representative-based approaches

- *Goal:* Construct a partition of a database *D* of *n* objects into a set of *k* clusters minimizing an objective function.
  - Exhaustively enumerating all possible partitions into *k* sets in order to find the global minimum is too expensive.
- Heuristic methods:
  - Choose *k* representatives for clusters, e.g., randomly
  - Improve these initial representatives iteratively:
    - Assign each object to the cluster it "fits best" in the current clustering
    - Compute new cluster representatives based on these assignments
    - Repeat until the change in the objective function from one iteration to the next drops below a threshold
- Types of cluster representatives
  - k-means: Each cluster is represented by the center of the cluster
  - k-medoid: Each cluster is represented by one of its objects

K-Means Clustering: Basic Idea

- Objective: For a given *k*, form *k* groups so that the sum of the (squared) distances between the mean of the groups and their elements is minimal.
  - Poor Clustering
  - Optimal Clustering



K-Means Clustering: Basic Notions

- Objects $p = (x^p_1, ..., x^p_d)$ are points in a d-dimensional vector space (the mean of a set of points must be defined)
- *Centroid* μ_C: Mean of all points in a cluster C,
- Measure for the compactness („Total Distance“) of a **cluster** C_j:
- Measure for the compactness of a **clustering**
  - *The ideal clustering minimizes this objective function*

K-Means Clustering: Lloyd's Algorithm

Given *k*, *k*-means is implemented as Lloyd's algorithm in 2 steps:

- Partition the objects into *k* nonempty subsets

1. Compute the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
2. Assign each object to the cluster with the nearest representative.
    - Go back to Step 1, stop when representatives do not change (substantially).

Pseudocode k-means

K-MEANS ($\mathbf{D}, k, \epsilon$):

$t = 0$

Randomly initialize k centroids: $\boldsymbol{\mu}_1^t, \boldsymbol{\mu}_2^t, \ldots, \boldsymbol{\mu}_k^t \in \mathbb{R}^d$

repeat

  $t \leftarrow t + 1$

  $C_j \leftarrow \emptyset$ for all $j = 1, \cdots, k$

  `// Cluster Assignment Step`

  foreach $\mathbf{x}_j \in \mathbf{D}$ **do**

    $j^* \leftarrow \arg\min_i \left\{ \left\| \mathbf{x}_j - \boldsymbol{\mu}_i^t \right\|^2 \right\}$ `// Assign` $\mathbf{x}_j$ `to closest centroid`

    $C_{j^*} \leftarrow C_{j^*} \cup \{\mathbf{x}_j\}$

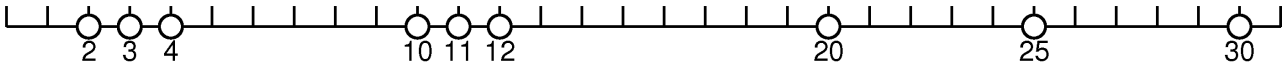
  `// Centroid Update Step`

  foreach $i = 1$ *to* k **do**

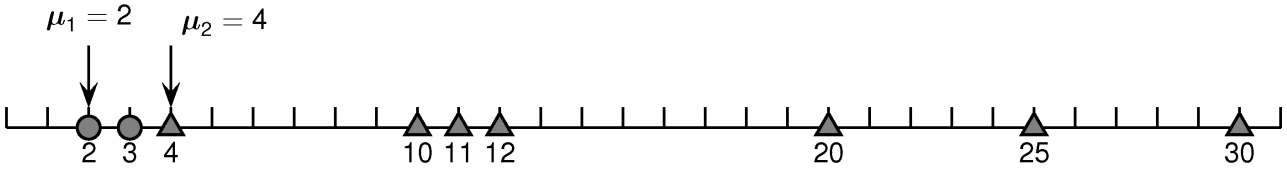
    $\boldsymbol{\mu}_i^t \leftarrow \frac{1}{|C_i|} \sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j$

until $\sum_{i=1}^{k} \left\| \boldsymbol{\mu}_i^t - \boldsymbol{\mu}_i^{t-1} \right\|^2 \leq \epsilon$

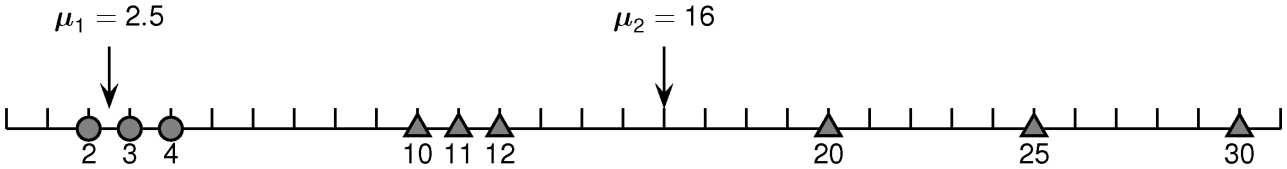
K-Means example in one dimension



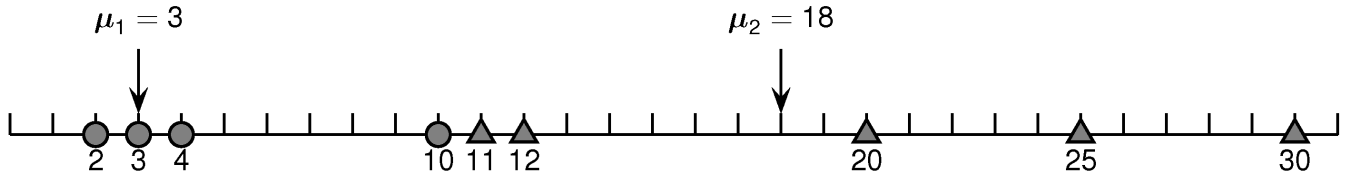
(a) Initial dataset



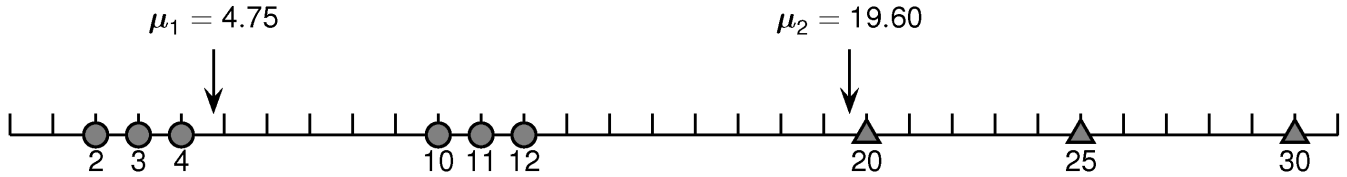
(b) Iteration: $t = 1$



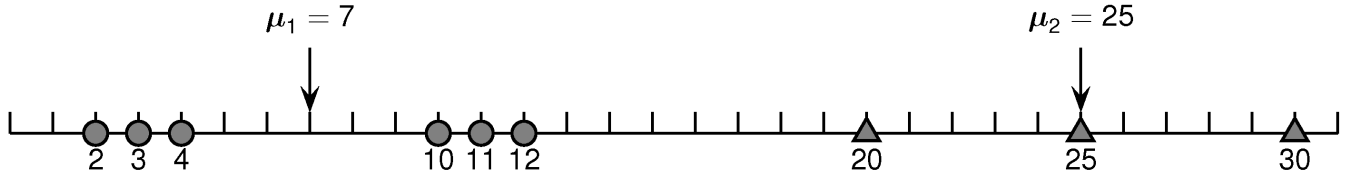
(c) Iteration: $t = 2$



(d) Iteration: $t = 3$

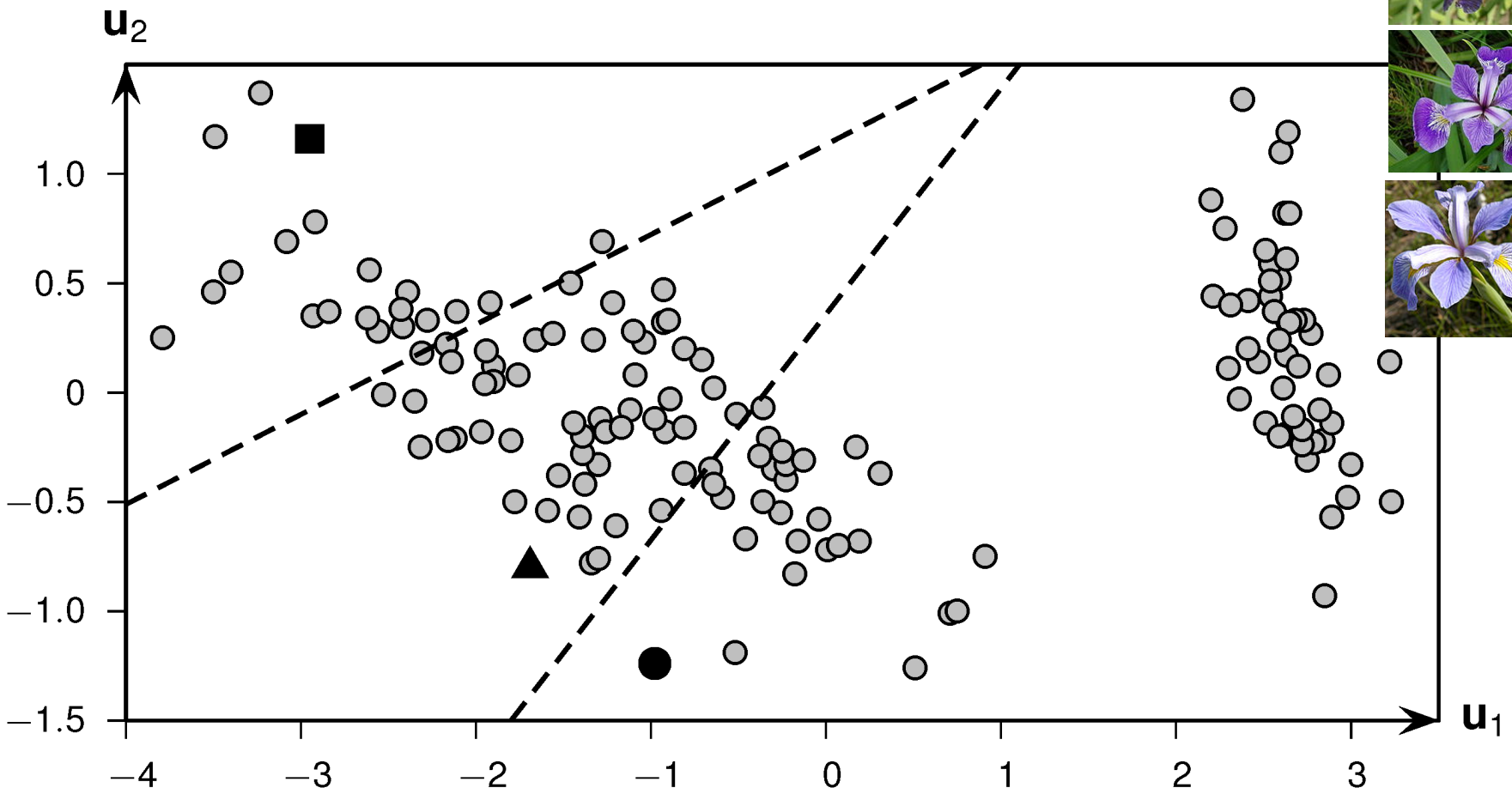


(e) Iteration: $t = 4$



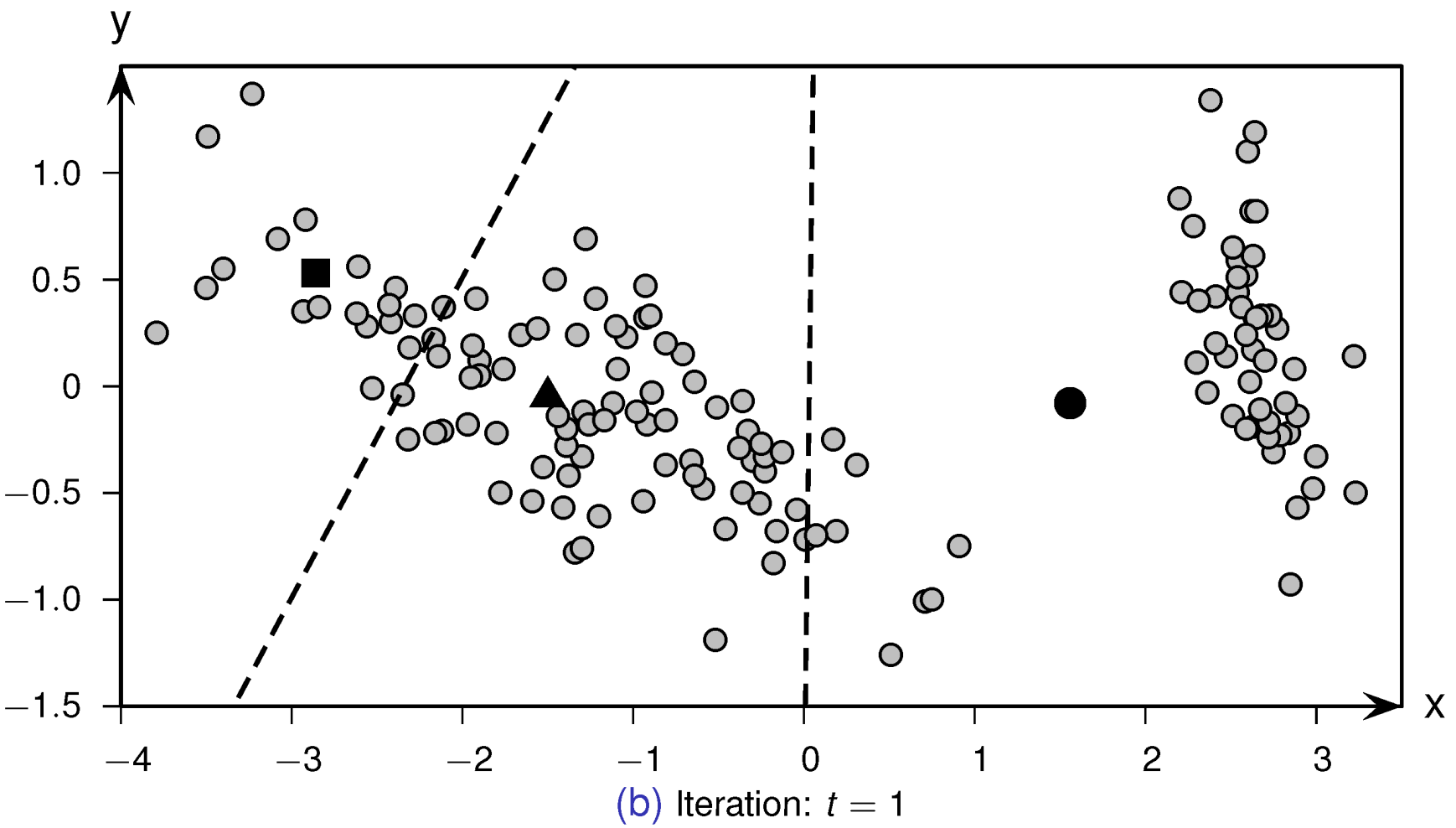
(f) Iteration: $t = 5$ (converged)

K-Means example on 2d principle components of IRIS dataset



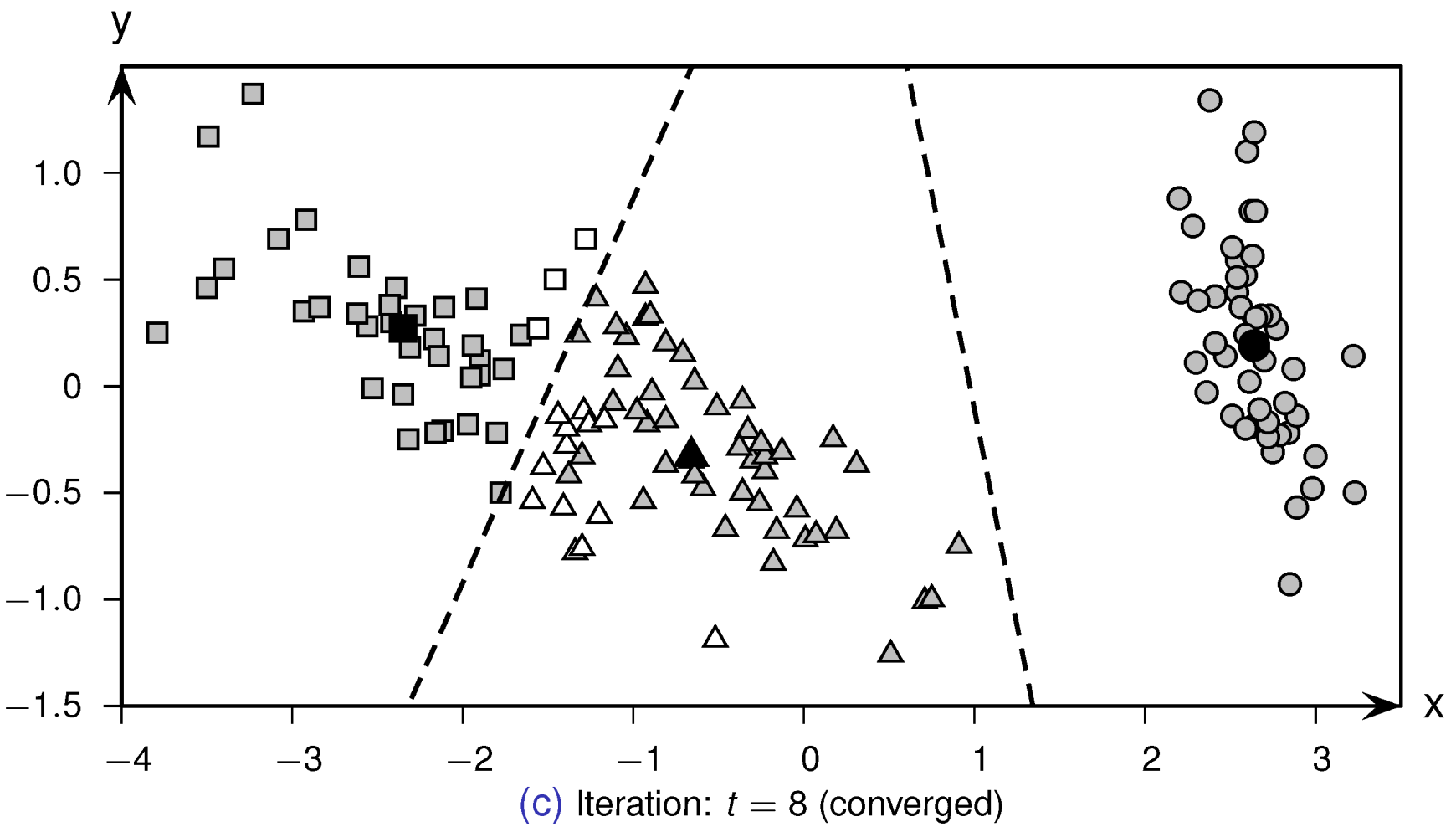
(a) Random initialization: $t = 0$

K-Means example on 2d principle components of IRIS dataset



(b) Iteration: $t = 1$

K-Means example on 2d principle components of IRIS dataset



K-Means Clustering: Discussion

- Strength
  - Relatively efficient: $O(tkn)$, where n is #objects, k is #clusters, and t is #iterations
  - Normally: $k, t << n$
  - Easy implementation
- Weakness
  - Applicable only when mean is defined
  - Need to specify *k,* the number of clusters, in advance
  - Sensitive to noisy data and outliers
  - Clusters are forced to have convex shapes
  - Result and runtime are very dependent on the initial partition; often terminates at a *local optimum* – however: methods for good initialization exist
- Several variants of the *k*-means method exist, e.g. ISODATA
  - Extends *k*-means by methods to eliminate very small clusters, merging and split of clusters; user has to specify additional parameters

Choice of the Parameter *k*

- Idea for a method:
  - Determine a clustering for each $k = 2, \ldots n-1$
  - Choose the „best“ clustering
- But how can we measure the quality of a clustering?
  - A measure has to be independent of *k*.
  - The measures for the compactness of a clustering TD^2 and TD are monotonously decreasing with increasing value of *k*.
- Silhouette-Coefficient [Kaufman & Rousseeuw 1990]
  - Measure for the quality of a *k*-means or a *k*-medoid clustering that is independent of *k*.

The silhouette coefficient – basic idea

- How good is the clustering
  - how appropriate is the mapping of objects to clusters
- Elements in cluster should be „similar“ to their representative
  - → measure the average distance of objects to their representative: ***a***
- Elements in different clusters should be „dissimilar“
  - → measure the average distance of objects to alternative cluster

    (i.e. second closest cluster): ***b***

The silhouette coefficient formally

- *a(o):* average distance between object *o* and the objects in its cluster *A*
- *b(o):* average distance between object *o* and the objects in its "second closest" cluster *B*
- The silhouette of *o* is then defined as
- The values of the silhouette coefficient range from –1 to +1

“Reading” the silhouette coefficient

- How good is the assignment of *o* to its cluster
  - *s* (*o*) = –1: bad, on average closer to members of *B*
  - *s* (*o*) = 0: in-between *A* and *B*
  - *s* (*o*) = 1: good assignment of *o* to its cluster *A*

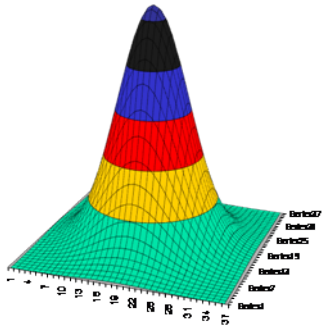
- Silhouette Coefficient s_C of a clustering: average silhouette of all objects
  - $0.7 < s_C \leq 1.0$ strong structure, $0.5 < s_C \leq 0.7$ medium structure
  - $0.25 < s_C \leq 0.5$ weak structure, $s_C \leq 0.25$ no structure

- Absolute values of SC of limited value → difficult to compare across data and algorithms
- Used typically to compare clusterings using different k to determine best value

Expectation Maximization (EM)

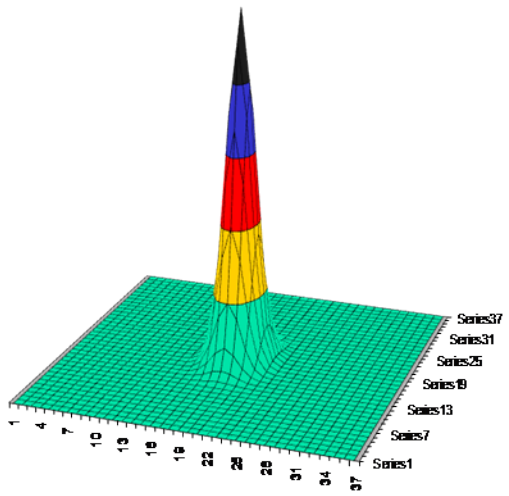
Basic Notions [Dempster, Laird & Rubin 1977]

- Consider points $p = (x_1, ..., x_d)$ from a d-dimensional Euclidean vector space
- Each cluster is represented by a probability distribution
- Typically: mixture of Gaussian distributions
- Single distribution to represent a cluster C
  - Center point μ_C of all points in the cluster
  - d x d Covariance matrix Σ_C for the points in the cluster C
- Density function for cluster C:

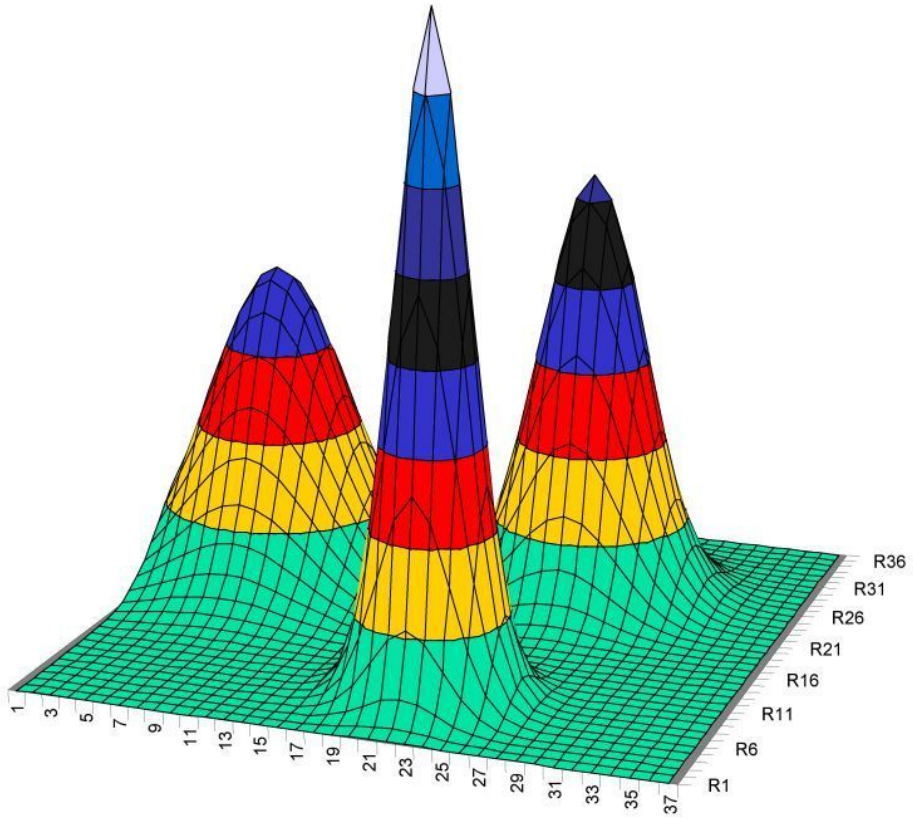


EM: Gaussian Mixture – 2D examples

A single Gaussian density function



A Gaussian mixture model, k = 3



EM – Basic Notions

- Density function for clustering $M = \{C_1, \ldots, C_k\}$
  - Estimate the a-priori probability of class C_i, $P(C_i)$, by the relative frequency W_i, i.e., the fraction of cluster C_i in the entire data set D:
- Assignment of points to clusters
  - A point may belong to several clusters with different probabilities $P(x|C_i)$
- *Maximize E(M), a measure for the quality of a clustering M*
  - *E(M)* indicates the probability that the data D have been generated by the distribution model M

EM – Algorithm

ClusteringByExpectationMaximization (point set *D*, int *k*)

Generate an initial model M' = (C_1', …, C_k')

repeat

// (re-) assign points to clusters – expectation step

For each object x from *D* and for each cluster (= Gaussian) C_i, compute P(x|C_i), P(x) and P(C_i|x)

// (re-) compute the model - maximization step

For each Cluster C_i, compute a new model M = {C_1, …, C_k} by recomputing W_i, μ_C and Σ_C

Replace M' by M

until |E(M) – E(M')| < ε

return M

EM – Recomputation of Parameters (maximization step)

- Weight W_i of cluster C_i
  - *Re-estimate the prior probablity for each cluster as the fraction of weights that contribute to that cluster*

- Center μ_i of cluster C_i
  - *Re-estimate the mean as the weighted average of all points*

- Covariance matrix Σ_i of cluster C_i
  - *Re-estimate the covariance matrix as the weighted covariance over all pairs of dimensions*

EM covariance matrix

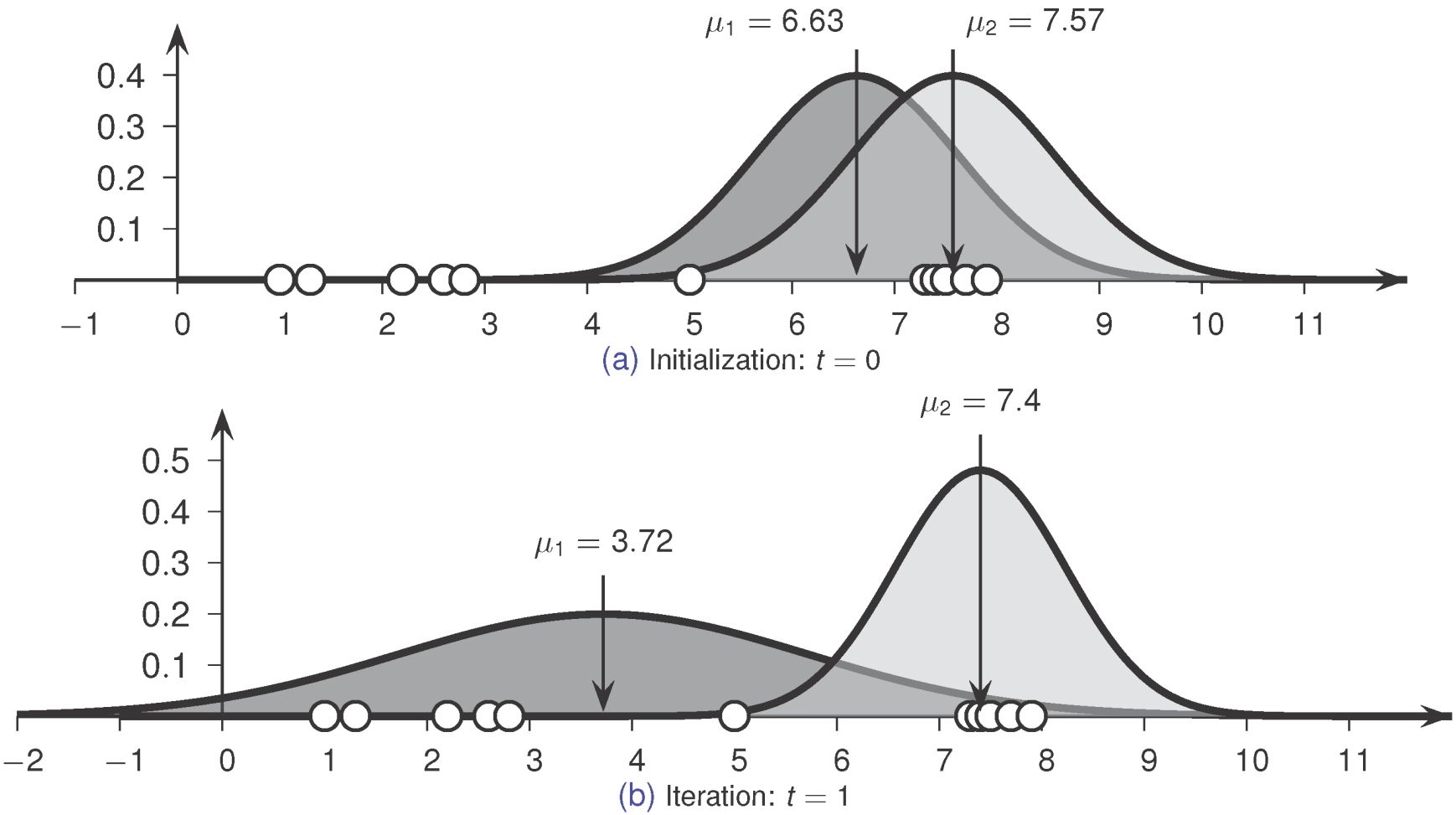
- Full covariance matrix
  - Detailed model
  - Requires estimation of $O(d^2)$ parameters
  - Often not enough data for reliable estimation
  - Costly

$$\boldsymbol{\Sigma}_i = \begin{pmatrix} (\sigma_1^i)^2 & \sigma_{12}^i & \dots & \sigma_{1d}^i \\ \sigma_{21}^i & (\sigma_2^i)^2 & \dots & \sigma_{2d}^i \\ \vdots & \vdots & \ddots & \\ \sigma_{d1}^i & \sigma_{d2}^i & \dots & (\sigma_d^i)^2 \end{pmatrix}$$

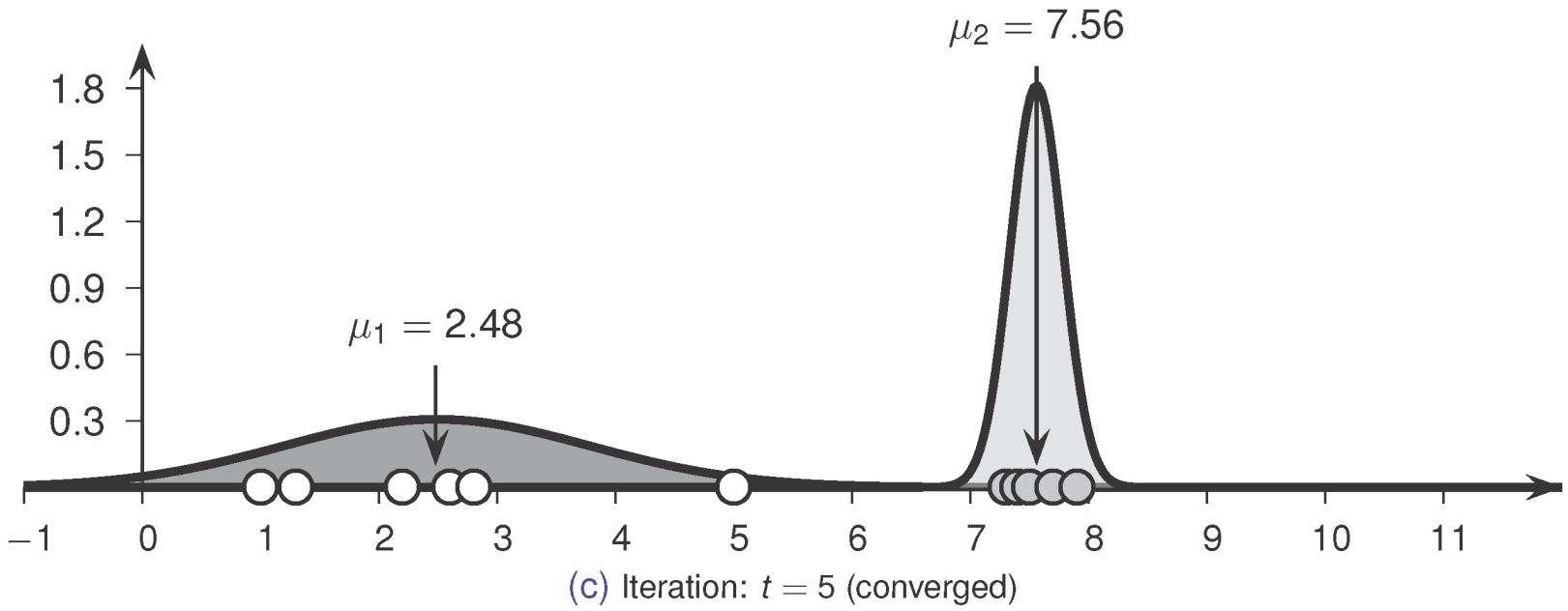
- Diagonal covariance matrix
  - Simplified model
  - Assume all dimensions are independent
  - Only estimate d parameters

$$\boldsymbol{\Sigma}_i = \begin{pmatrix} (\sigma_1^i)^2 & 0 & \dots & 0 \\ 0 & (\sigma_2^i)^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & (\sigma_d^i)^2 \end{pmatrix}$$

EM example in one dimension

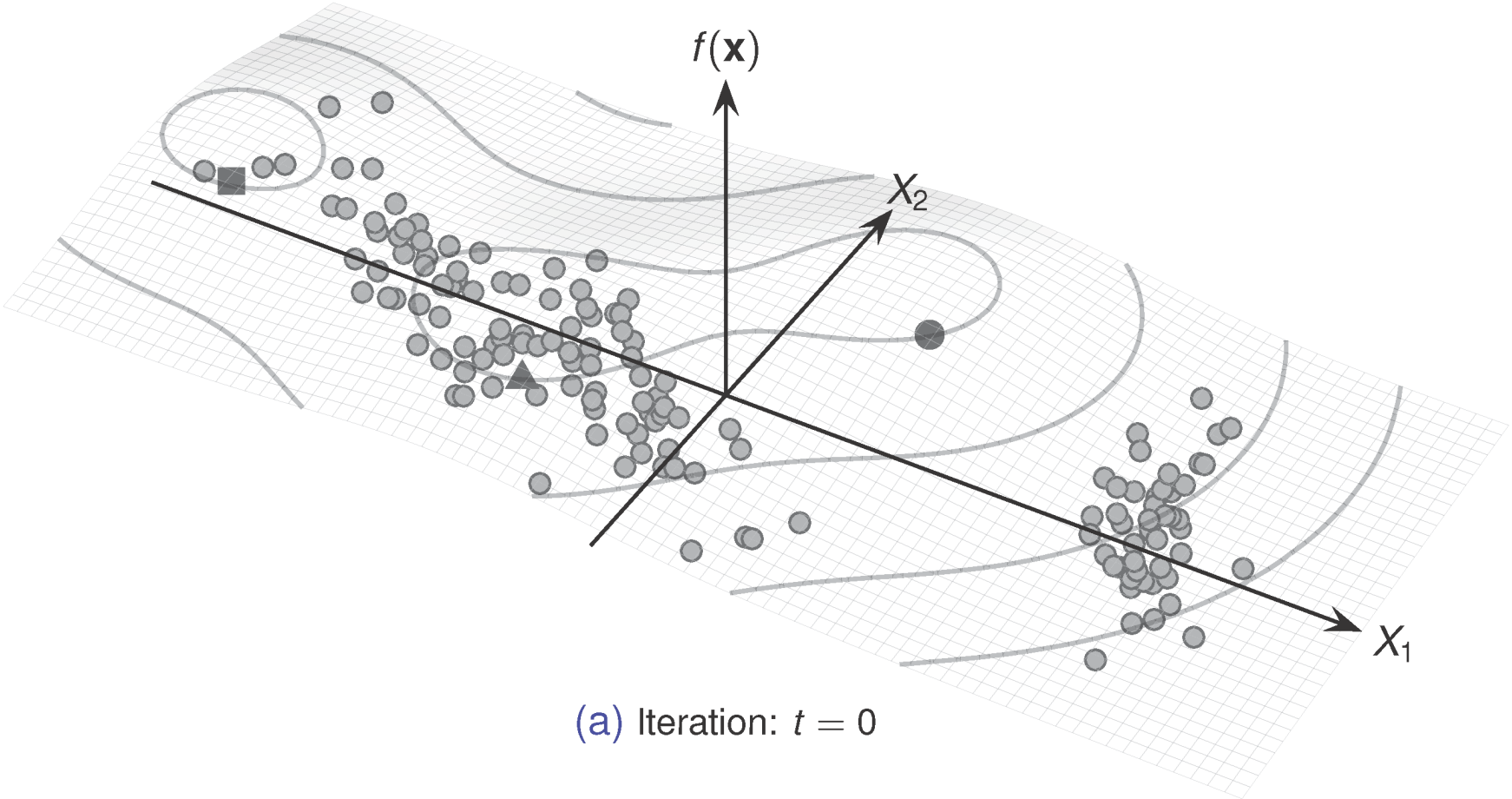


EM example in one dimension



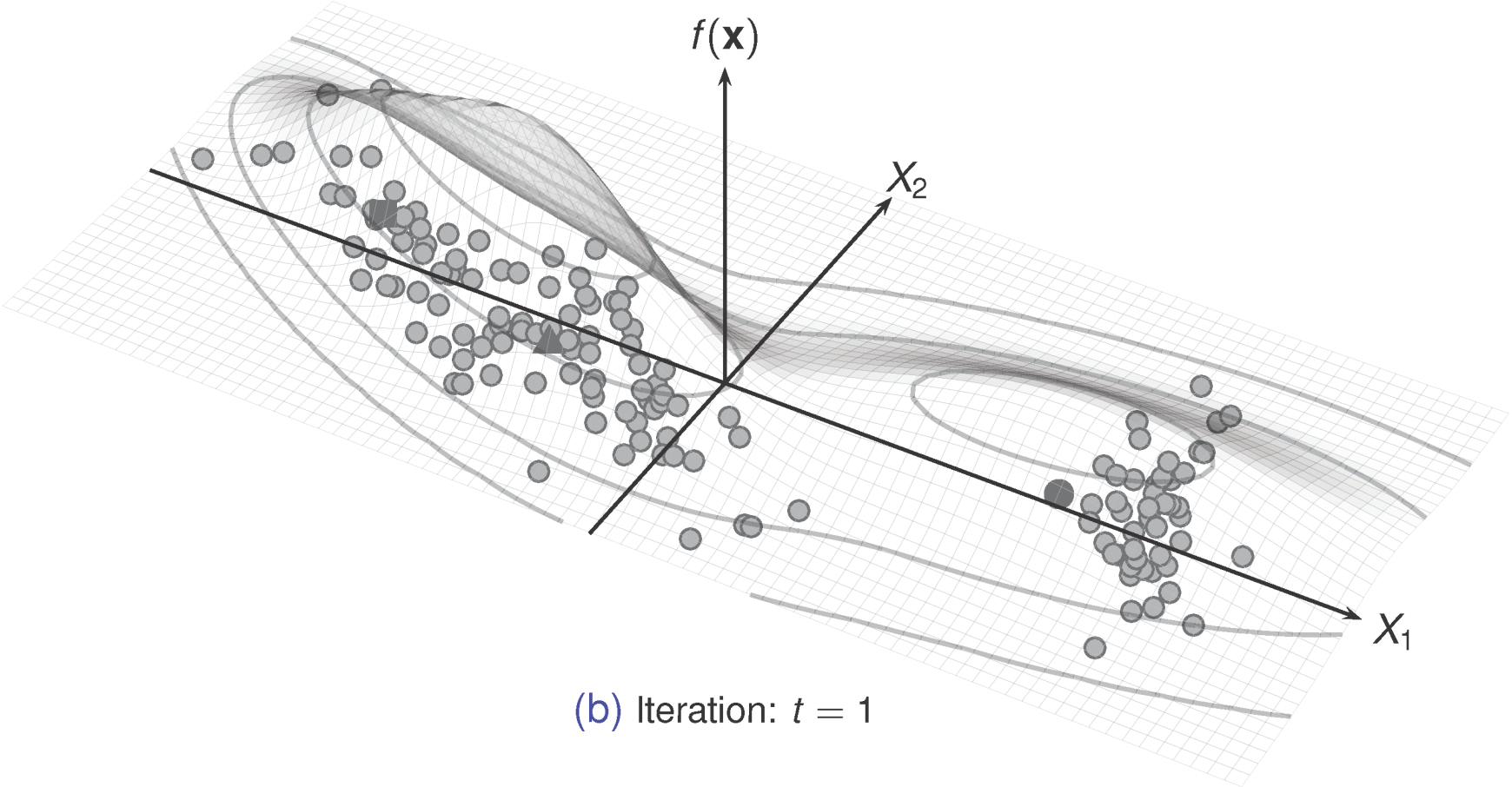
(c) Iteration: $t = 5$ (converged)

EM example in two dimensions using three Gaussians, Iris



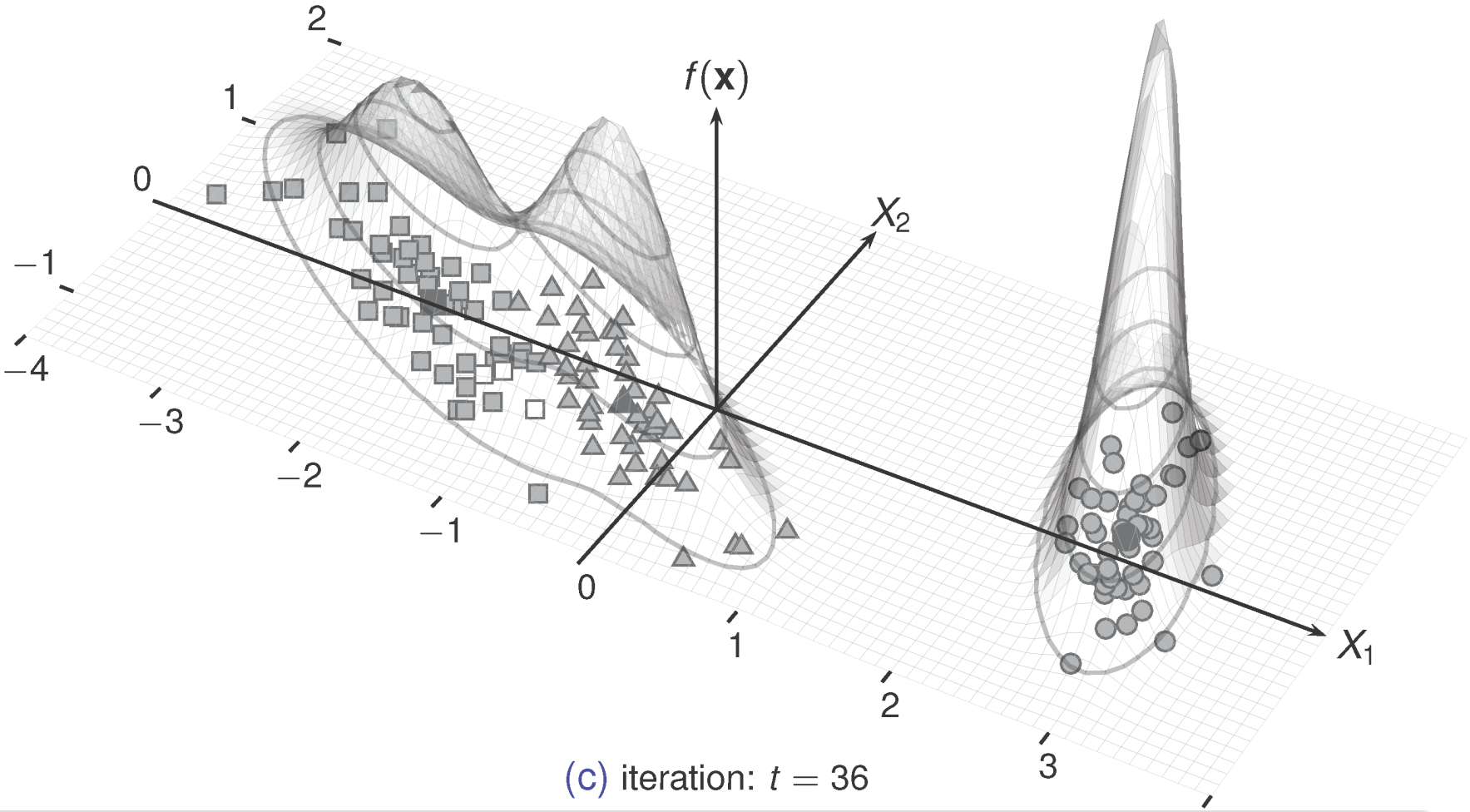
(a) Iteration: $t = 0$

EM example in two dimensions using three Gaussians



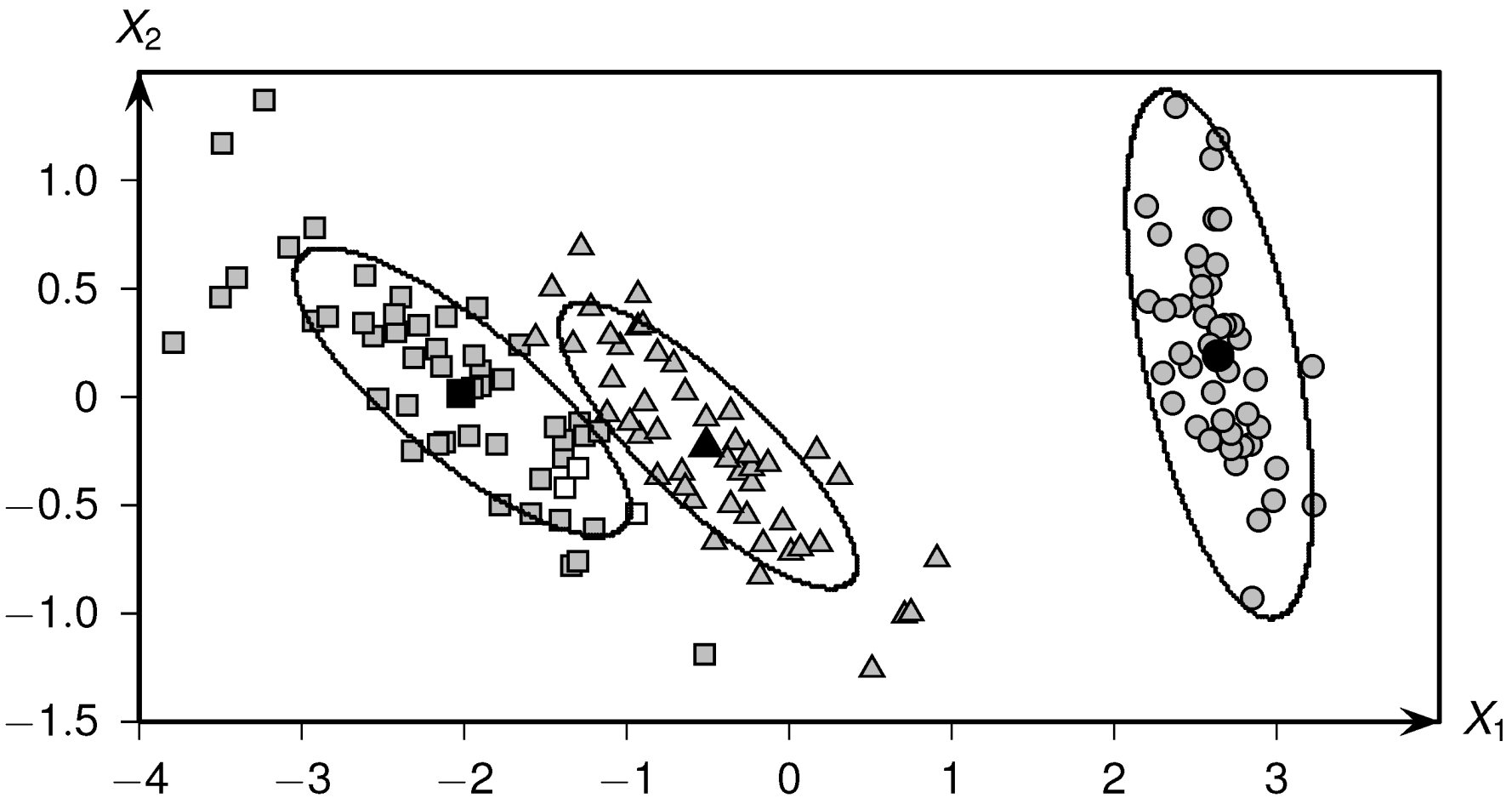
(b) Iteration: $t = 1$

EM example in two dimensions using three Gaussians



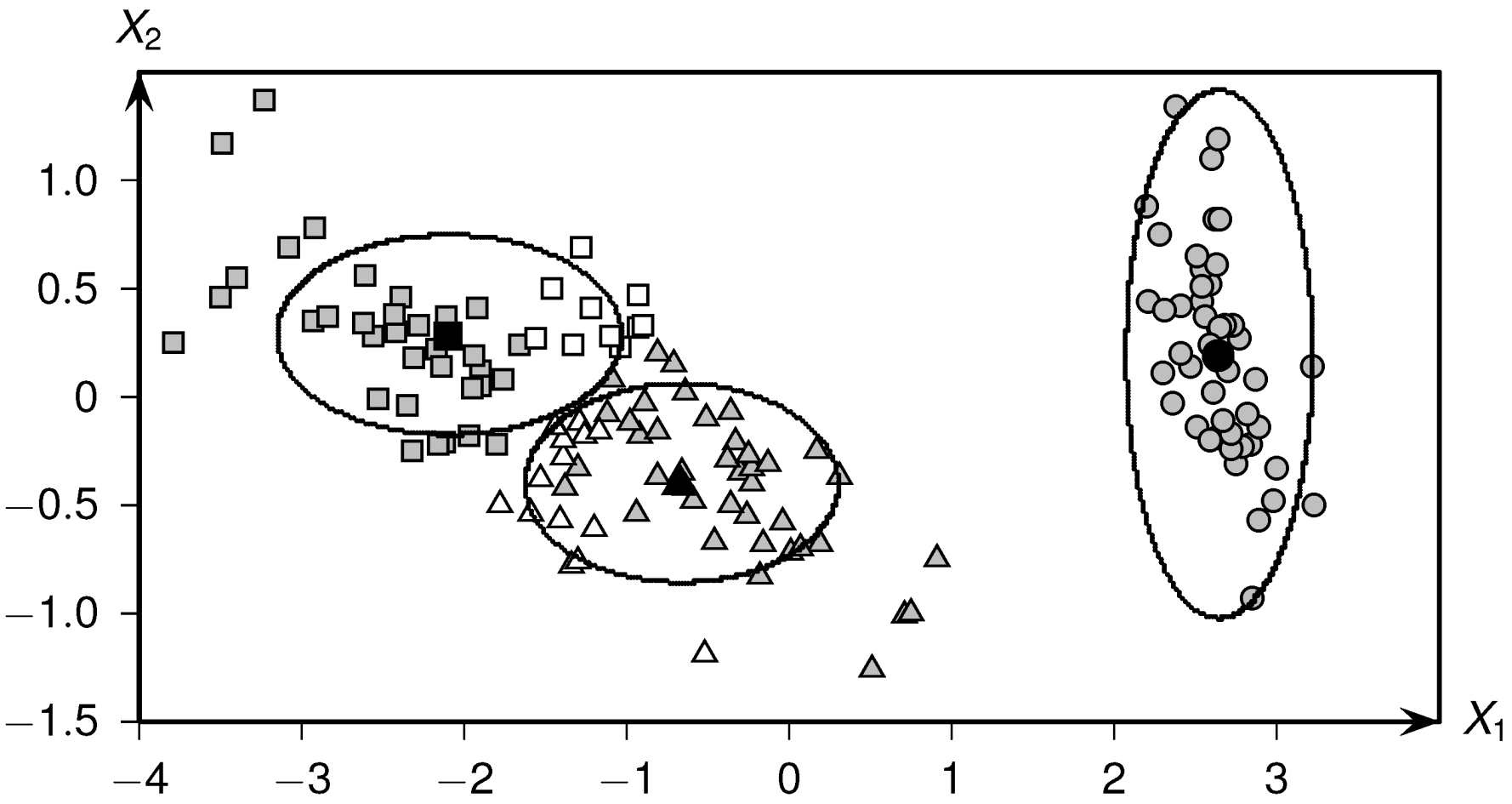
(c) iteration: $t = 36$

EM example in two dimensions using three Gaussians, full covariance matrix



(a) Full covariance matrix ($t = 36$)

EM example in two principle components using three Gaussians, diagonal covariance matrix



(b) Diagonal covariance matrix ($t = 29$)

EM – Discussion

- Convergence to (possibly local) minimum
- Computational effort:
  - O($n \cdot k \cdot$ #iterations)
  - #iterations is quite high in many cases
- Both result and runtime strongly depend on
  - the initial assignment
  - a proper choice of parameter k (= desired number of clusters)
- Modification to obtain a *partitioning* variant
  - Objects may belong to several clusters
  - Assign each object to the cluster to which it belongs with the highest probability

Summary

- Introduction to data science
  - Gain insights from data using scientific methods
- Data preprocessing
  - Crucial for proper setup of data science process
  - Understand your data, bring it to good shape before you run an analysis!
- Principle components analysis
  - Reduction of dimensionality for visualization or analysis
  - Retains as much information as possible
- Clustering
  - Grouping of data based on mutual similarity
    - Similar data in the same cluster, dissimilar in different ones
      - Choice of (dis-)similarity measure matters!
- Representation-based approaches / Partitioning approaches
  - Characterize a cluster by a representative (mean, medoid)
  - Iterative update of clustering
  - Choice of parameter k
    - Evaluation measures important

References

- https://jakevdp.github.io/PythonDataScienceHandbook/ Data Science in Python
- https://wesmckinney.com/book/ Python for Data Science
- https://scikit-learn.org Scikit-learn
- https://pandas.pydata.org/ Pandas
- https://dataminingbook.info/book_html/ Mohammed J. Zaki, Wagner Meira, Jr., Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd Edition, Cambridge University Press, March 2020. ISBN: 978-1108473989.
- J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.
- M. H. Dunham. Data Mining: Introductory and Advanced Topics. Prentice Hall, 2003
- PCA slides, Aidong Zhang, Department of Computer Science and Engineering, University at Buffalo, The State University of New York, US