

*** Applied Machine Learning Fundamentals ***

Clustering

M. Sc. Daniel Wehner

SAP SE

Winter term 2019/2020



Find all slides on [GitHub](#)

Lecture Overview

Unit I	Machine Learning Introduction
Unit II	Mathematical Foundations
Unit III	Bayesian Decision Theory
Unit IV	Probability Density Estimation
Unit V	Regression
Unit VI	Classification I
Unit VII	Evaluation
Unit VIII	Classification II
Unit IX	Clustering
Unit X	Dimensionality Reduction

Agenda for this Unit

1 Introduction

What is Clustering?
Clustering Strategies Overview

2 *k*-Means

Introduction
k-Means Algorithm
Use Case: Image Compression
Problems and Issues

3 Hierarchical Clustering

Agglomerative Clustering Algorithm
Agglomerative Clustering: Example

Distance Metrics between Clusters

4 Spectral Clustering

Motivation
A Bit of Graph Theory
Spectral Clustering Algorithm

5 Wrap-Up

Summary
Self-Test Questions
Lecture Outlook
Recommended Literature and further Reading
Meme of the Day

Section:
Introduction



Clustering Introduction

- **Clustering** belongs to the category of **unsupervised learning**
- A clustering algorithm tries to **find structure** in the data
- Once the clusters are found, they first have to be interpreted...
- ...and can then be used for prediction purposes

A cluster must be **internally homogeneous**, but simultaneously **externally heterogeneous**. (Elements of one cluster have to be very similar, but must differ significantly from elements in other clusters.)

Example Use Cases for Clustering

- **Behavioral segmentation**
 - Customer segmentation (e. g. [sinus milieus](#))
 - Creating profiles based on activity monitoring
- **Sorting sensor measurements**
 - Image grouping
 - Detection of activity types in motion sensors
- **Inventory categorization**
 - Grouping inventory by sales activity
 - Grouping inventory by manufacturing metrics
- Many, many more, ...

Clustering Strategies

- ① **EM-based clustering**, e. g.: *k*-Means
- ② **Hierarchical clustering**, e. g.:
 - Agglomerative clustering
 - Divisive clustering
- ③ **Affinity-based clustering**, e. g.:
 - Spectral clustering
 - DBSCAN

Section:
k-Means





k-Means: Procedure

- The algorithm is an instance of **vector quantization**
 - It represent data points by a single vector (here: **centroid**) which is close to them
 - This is useful for **compression**!
- **How to:** Create k partitions ($\hat{=}$ clusters) of the data set \mathcal{D} , such that the sum of squared deviations from the cluster centroids is **minimal**:

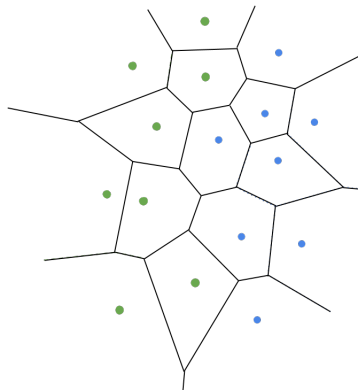
$$\min_{\mu_j} \sum_{j=1}^k \sum_{\mathbf{x}^{(i)} \in \mathcal{D}_j} \|\mathbf{x}^{(i)} - \mu_j\|^2 \quad (1)$$

- With $\mathcal{D}_j \equiv j^{th}$ cluster, $\mu_j \equiv$ centroid of j^{th} cluster



Result: Voronoi Diagram

- The dots represent cluster centroids
- The lines visualize the **cluster boundaries**
- For a new point we can easily determine to which cluster it has to be assigned



k-Means Algorithm

- Input: $\mathcal{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\} \in \mathbb{R}^{n \times m}$, Number of clusters k
- Algorithm:

① $t \leftarrow 1$

② Randomly choose k means $\mu_1^{(1)}, \mu_2^{(1)}, \dots, \mu_k^{(1)}$

③ While not converged:

3a Assign each $\mathbf{x}^{(i)} \in \mathcal{D}$ to the closest cluster:

$$\mathcal{D}_j^{(t)} = \left\{ \mathbf{x}^{(i)} : \|\mathbf{x}^{(i)} - \mu_j^{(t)}\|^2 \leq \|\mathbf{x}^{(i)} - \mu_{j^*}^{(t)}\|^2; \forall j^* = 1, 2, \dots, k; \mathbf{x}^{(i)} \in \mathcal{D} \right\}$$

3b Update cluster centroids μ_j :

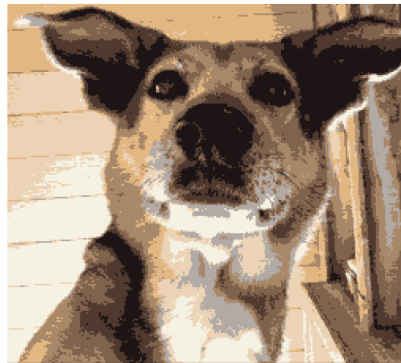
$$\mu_j^{(t+1)} = \frac{1}{|\mathcal{D}_j^{(t)}|} \sum_{\mathbf{x}^{(i)} \in \mathcal{D}_j^{(t)}} \mathbf{x}^{(i)} \quad \text{then update } t: \quad t \leftarrow t + 1$$

Image Compression

Original image



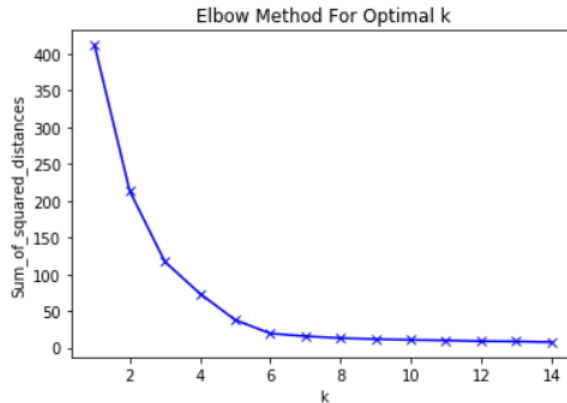
Compressed image



k-Means Issues

- The algorithm assumes all clusters are **spherical** (\neq **affinity-based clustering**)
- Does not have a notion of **outliers** (unlike DBSCAN)
- What is the correct value for k ? \Rightarrow **Elbow-method:**
 - Measure sum of squared distances from data points to cluster centers (inertia)
 - Record results for different values for k and plot them
 - Search for the 'elbow point'

Elbow Method



Section: Hierarchical Clustering



Agglomerative Clustering Algorithm

- 1 Start with one cluster for each instance: $C = \{\{\mathbf{x}^{(i)}\} : \mathbf{x}^{(i)} \in \mathbf{X}\}$
- 2 Compute distance $d(C_i, C_j)$ between all pairs of clusters C_i, C_j
- 3 Join clusters C_i and C_j with minimum distance into a new cluster C_p :

$$C_p = \{C_i, C_j\}$$

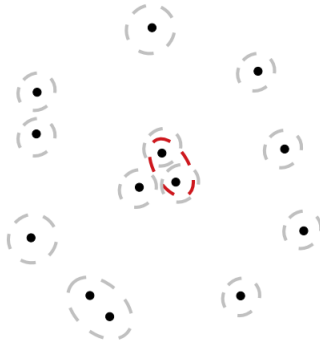
$$C = (C \setminus \{C_i, C_j\}) \cup \{C_p\}$$

- 4 Compute distances between C_p and all other clusters in C
- 5 If $|C| > 1$, goto 3

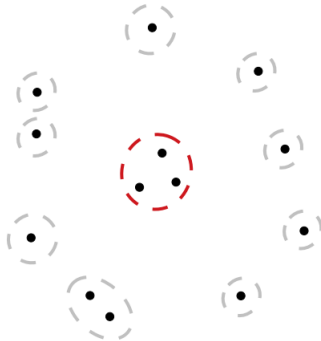
Agglomerative Clustering: Example



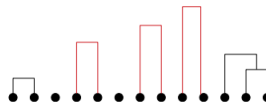
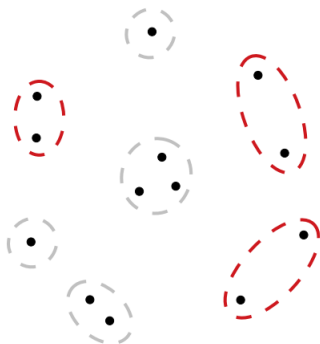
Agglomerative Clustering: Example (Ctd.)



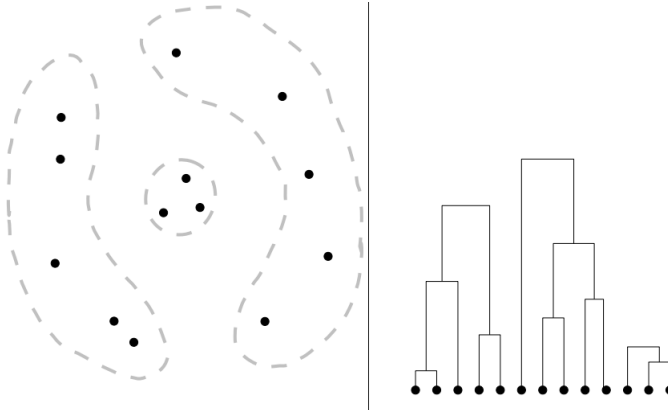
Agglomerative Clustering: Example (Ctd.)



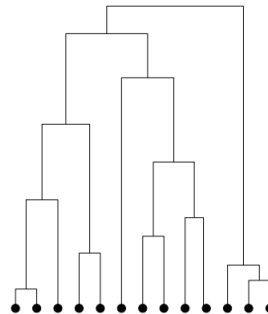
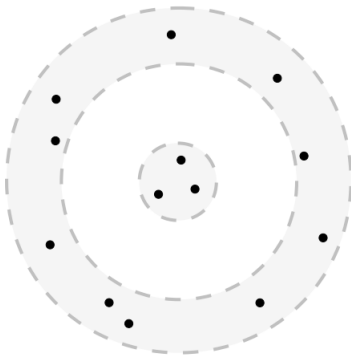
Agglomerative Clustering: Example (Ctd.)



Agglomerative Clustering: Example (Ctd.)



Agglomerative Clustering: Example (Ctd.)

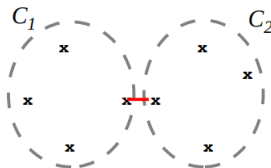


This is a
dendrogram

Single Linkage

- Computing distances between clusters C_1 and C_2
- **Single linkage:**

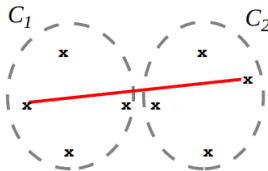
$$d(C_1, C_2) = \min\{d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) : \mathbf{x}^{(i)} \in C_1, \mathbf{x}^{(j)} \in C_2\}$$



Complete Linkage

- Computing distances between clusters C_1 and C_2
- **Complete linkage:**

$$d(C_1, C_2) = \max\{d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) : \mathbf{x}^{(i)} \in C_1, \mathbf{x}^{(j)} \in C_2\}$$



Section:
Spectral Clustering



Spectral Clustering

- Remember the disadvantage of k -Means? (spherical clusters)
- How can we cluster data without this assumption?

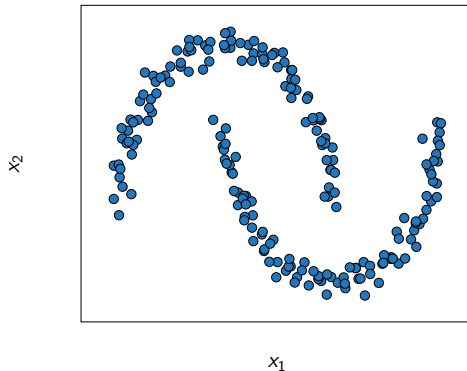
⇒ **Affinity-based clustering**

Affinity-based clustering assumes **no shape** of the resulting clusters. It is based on the **connectedness of the data points**.

- Spectral clustering is affinity-based
- Whenever you hear '*spectral*': It has something to do with eigen-vectors

Example Data Set

What would be
the result of *k*-Means?



A short Introduction to Graphs

- A graph \mathcal{G} is a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
- $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ is the set of n vertices (nodes)
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the set of edges (connections between nodes)
- **Adjacency matrix A**
 - $A_{ij} = 1$, iff $(v_i, v_j) \in \mathcal{E}$ (v_i is a neighbor of v_j)
 - A is symmetric for undirected graphs, i. e. $A_{ij} = A_{ji}$
- The **degree matrix D** $D = \text{diag}(d_1, d_2, \dots, d_n)$ is a matrix of node degrees

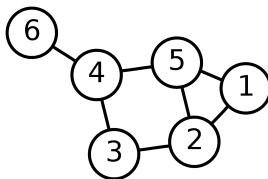
$$d_i = \sum_{j=1}^n A_{ij}$$

A short Introduction to Graphs (Ctd.)

- For graph analysis it is often useful to compute the **graph Laplacian** matrix:

$$L = D - A$$

- Example:



Example: Computation of A , D and L

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$L = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$



How to get the Graph?

- There are at least two possibilities
 - ① **ε -neighborhood graph**
 - Connect all instances whose pairwise distances are smaller than ε
 - **Problem:** How to choose ε ?
 - ② **k -nearest neighbor graph**
 - Connect instance $\mathbf{x}^{(i)}$ with instance $\mathbf{x}^{(j)}$, if $\mathbf{x}^{(j)}$ is among the k nearest neighbors of $\mathbf{x}^{(i)}$
 - Attention: This definition leads to a directed graph (**Why?**)
 \Rightarrow Can be ignored
 - **Problem:** How to choose k ?
- Both approaches are used in practice

Spectral Clustering Algorithm

- Input: $\mathcal{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\} \in \mathbb{R}^{n \times m}$, Number of clusters k
- Algorithm:
 - ① Construct a similarity graph (adjacency matrix \mathbf{A} and degree matrix \mathbf{D})
 - ② Compute the graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$
 - ③ Perform **eigen-decomposition** on \mathbf{L} (to obtain the eigen-vectors \mathbf{Q})

$$\mathbf{L} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$$

- ④ Apply k -Means to the rows of matrix \mathbf{Q} to obtain the clusters $\{C_1, C_2, \dots, C_k\}$

Section:
Wrap-Up



Summary

- Clustering belongs to the category of **unsupervised learning**
- With clustering we try to find **structure in the data**
- Different algorithms make **different assumptions** about the resulting clusters
- **Clustering Strategies:**
 - EM-based clustering (e. g. k -Means)
 - Affinity-based clustering (e. g. spectral clustering, DBSCAN)
 - Hierarchical clustering



Self-Test Questions

- 1 What is clustering?
- 2 What is the definition of a cluster. Which properties should it have?
- 3 Describe the general procedure of *k*-Means. What are disadvantages?
- 4 What is a dendrogram?
- 5 How do we obtain the graphs for spectral clustering
- 6 What is affinity-based clustering. How does it differ from *k*-Means?
- 7 How to calculate the graph Laplacian matrix?

What's next...?

Unit I	Machine Learning Introduction
Unit II	Mathematical Foundations
Unit III	Bayesian Decision Theory
Unit IV	Probability Density Estimation
Unit V	Regression
Unit VI	Classification I
Unit VII	Evaluation
Unit VIII	Classification II
Unit IX	Clustering
Unit X	Dimensionality Reduction

Recommended Literature and further Reading



[1] Pattern Recognition and Machine Learning

Christopher Bishop. Springer. 2006.

→ [Link](#), cf. chapter 9

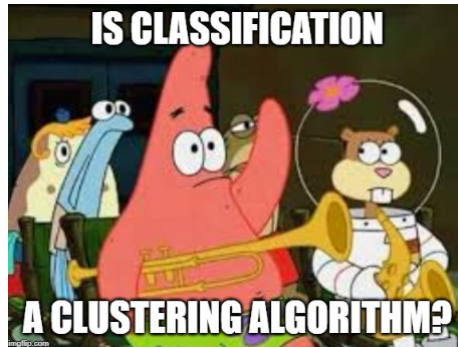


[2] Machine Learning: A Probabilistic Perspective

Kevin Murphy. MIT Press. 2012.

→ [Link](#), cf. chapter 25

Meme of the Day



Thank you very much for the attention!

Topic: *** Applied Machine Learning Fundamentals *** Clustering

Term: Winter term 2019/2020

Contact:

M.Sc. Daniel Wehner

SAP SE

daniel.wehner@sap.com

Do you have any questions?