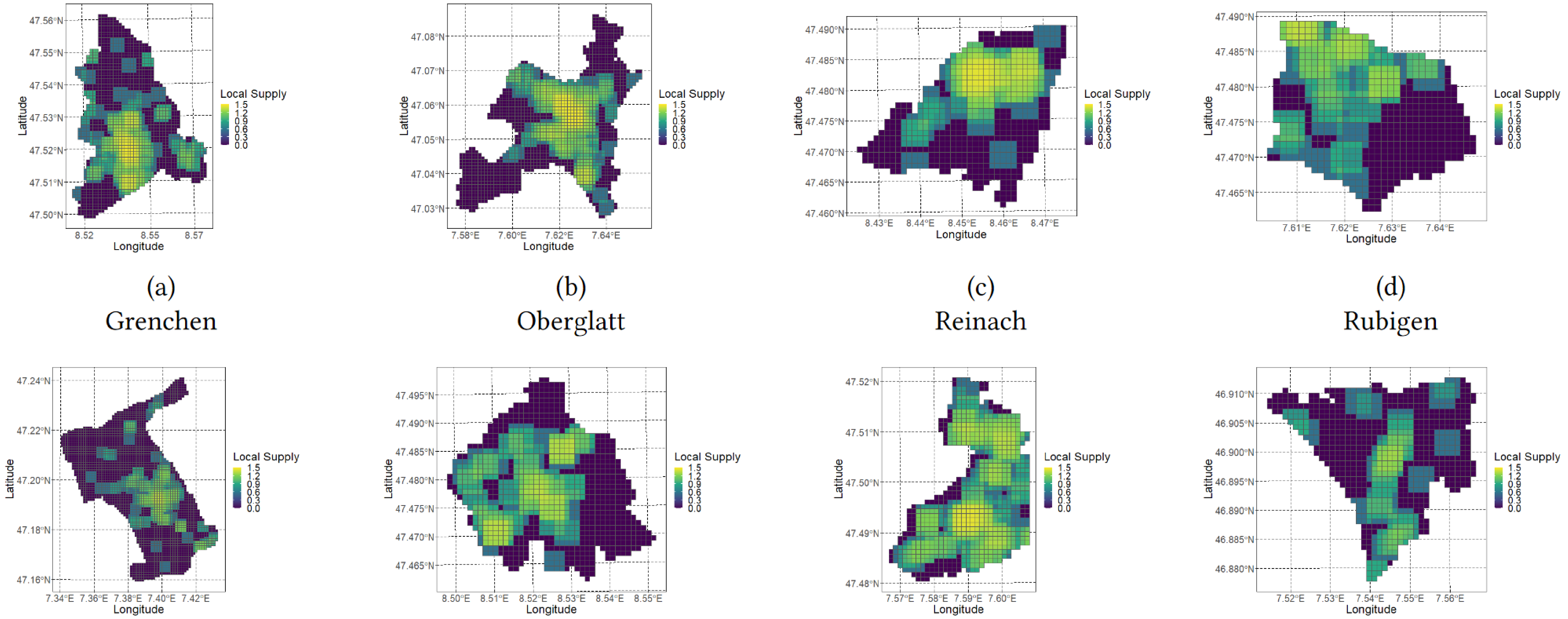
CAS Machine Learning

Unsupervised Learning: Clustering, Anomaly Detection, Dimensionality Reduction and Visualization

Dr. Yves Staudt
PhD in Actuarial Sciences
yves.staudt@fhgr.ch



(a) Grenchen

(b) Oberglatt

(c) Reinach

(d) Rubigen

(e) Buelach

(f) Burgdorf

(g) Dielsdorf

(h) Dornach

Agenda

1. Einführung
2. Kennenlernen
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Deep Clustering
12. Zusammenfassung

Agenda

1. **Einführung**
2. Kennenlernend
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Deep Clustering
12. Zusammenfassung

Lernziel

Nach dem Kurs sind die Teilnehmende in der Lage:

- ➢ Merkmale nach Gleichheiten und Unterschiede zu bestimmen.
- ➢ Unterschiedsmasse zur Messung von Ähnlichkeiten zu beschreiben.
- ➢ Cluster Verfahren zu beschreiben, zu unterscheiden und anzuwenden.
- ➢ Cluster Resultate zu interpretieren.
- ➢ Dimension Reduktionsverfahren zu beschreiben, zu unterscheiden und anzuwenden.

Unterrichtsplanung

Uhrzeit	Thema
9:15 – 10:00	Einführung und Clustering
10:05 – 10:50	K-Means and Determining Optimal Number of Classes
11:00 – 11:45	Hierarchical Clustering Algorithmus
11:40 -12:25	Kommunikation und Interpretation der Resultate
12:25-13:25	Mittagspause
13:25 – 14:10	Dimension Reduktionsverfahren, Principal Component Analysis
14:15 – 15:00	t-Distributed Stochastic Neighbor Embedding
15:05 – 16:00	Deep Clustering
16:05 – 16:45	Zusammenfassung und Abschluss

Code Notebooks

Code Notebooks von Python werden über

- Gitlab

Agenda

1. Einführung
2. **Kennenlernen**
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Deep Clustering
12. Zusammenfassung

Dozent

Profil: Dr. Yves Staudt

Ursprung: Luxemburg

Erfahrung: Umfangreiche praktische Erfahrung in Datenanalysen, Machine and Deep Learning

Interesse: Bergen, Fotografie und Kochen

Sozial: Ein sozialer Austausch ist mir wichtig.

Vorstellung

Wer seid ihr?

Was sind eure Erwartungen an den Kurs?

Agenda

1. Einführung
2. Kennenlernen
3. **Clustering**
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Deep Clustering
12. Zusammenfassung

Goal of Clustering

- The **goal** of **clustering** is to group similar objects or data points together based on their inherent characteristics
- Clustering is a fundamental task in **unsupervised learning**
- Clustering identifies natural groupings or patterns within the data
- Objects within the same cluster are more similar to each other than to those in other clusters
- Clustering algorithms help in understanding the **underlying structure** or organization of the data, revealing insights, and supporting decision-making processes.

Application of Clustering

Clustering has various **applications** across different domains including:

- customer segmentation
- document classification
- anomaly detection
- image analysis
- …

(Generated by Chat GPT 1.1.2023)

Anomaly Detection

- The goal of anomaly detection is to identify **unusual** or **anomalous** patterns or observations in a dataset
- Unusual patterns deviate significantly from the norm or expected behavior
- Clustering aims to group **similar** data points together
- Anomaly detection focuses on identifying the data points that are **dissimilar** or different from the majority
- By clustering the data, we can establish a notion of what is considered normal or expected within each cluster
- Any data point that does not belong to any cluster or deviates significantly from its assigned cluster can be flagged as an anomaly

Distances

Euclidean Distance: $\|x-y\|_2 = \sum_{j=1}^{n} \left(x_i - y_j\right)^2$ (Yin et al., 2021)

Simple Matching Distance: $D(x,y) = \begin{cases} 1,\ if\ x_j \neq y_j \\ 0,\ if\ x_j = y_j \end{cases}$ (Yin et al., 2021)

Dynamic Time Warping: $DTW(m,n) = \ |p_m - q_n| + \min\begin{pmatrix} DTW(m-1,n) \\ DTW(m-1,n-1) \\ DTW(m,n-1) \end{pmatrix}$ (Wong and Chung, 2019)

Agenda

1. Einführung
2. Kennenlernen
3. Clustering
4. **K-Means Algorithmus**
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Deep Clustering
12. Zusammenfassung

K-means Clustering

Task of clusters: Partitioning the dataset into groups, called clusters.

Goal: To split up the data in such way that points within single clusters are very similar and points in different clusters are different.

Algorithm alternates between **two steps**:

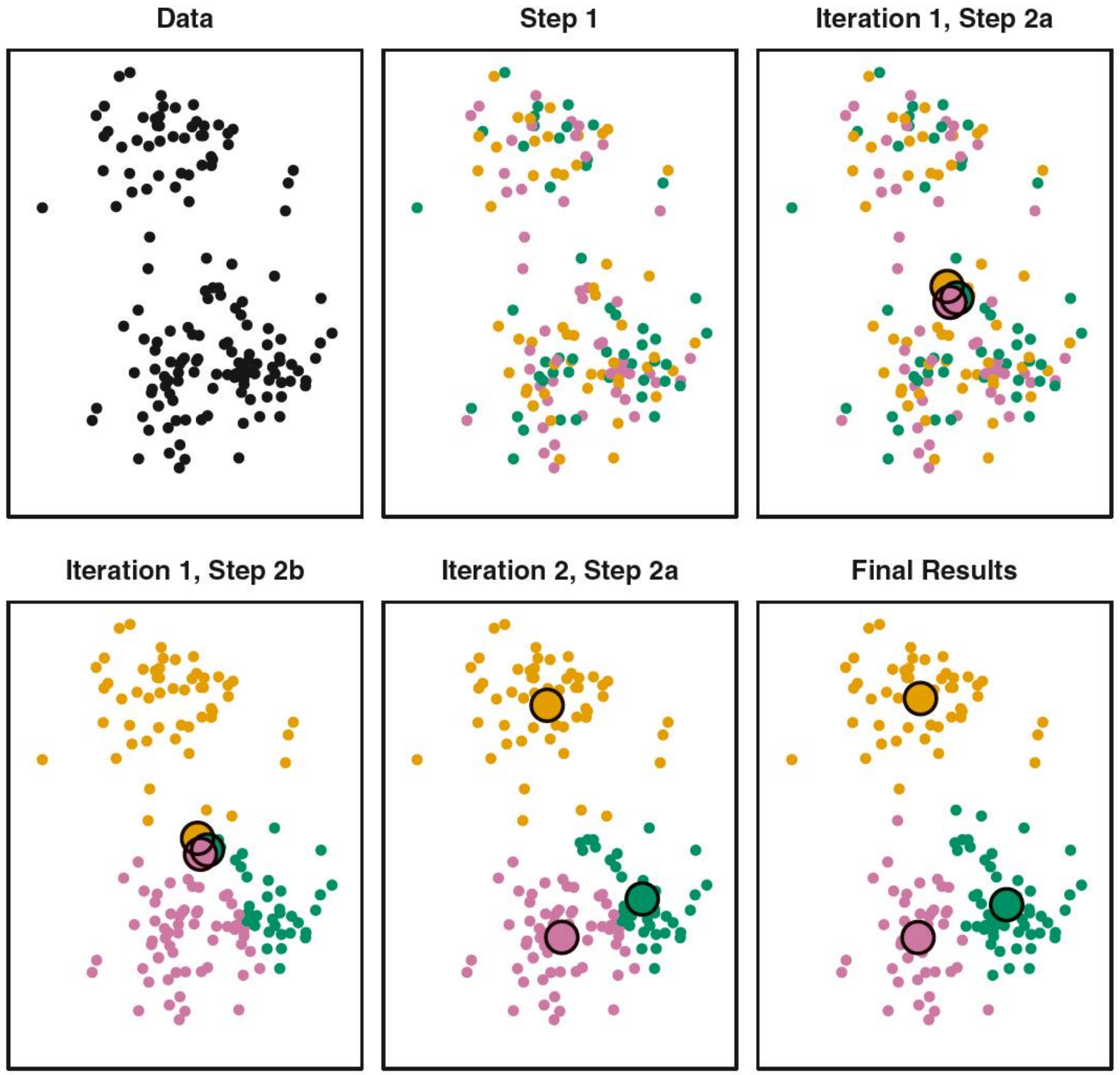
1. Assigning each data point to the closest cluster center.
2. Setting each cluster center as the mean of the data point that are assigned to it.

The algorithm is finished when the assignment of instances to clusters no longer changes.

In k-means clustering the number of clusters k needs to be fixed by the analyst.

(Jamies et al., 2017; Kuhn and Johnson, 2016)

Representation of the k-means algorithm



Representation of the K-means algorithm (James et al. 2013)

Optimization

Idea: A good clustering is one for which the within-cluster variation is as small as possible.

Within-cluster variation for cluster C_k measures the difference between the observations within a cluster.

Goal: To minimize the within-cluster variation

$$minimize_{C_1,\ldots,C_k}\left\{\sum_{k=1}^{K} W(C_k)\right\}$$

$$minimize_{C_1,\ldots,C_k}\left\{\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}\left(x_{ij}-x_{i'j}\right)^2\right\}$$

(James et al., 2013)

Properties of k-means clustering

Let $C_1, C_2, \ldots, C_K$ denote sets containing indices of the observations in each cluster

The sets satisfies the following two properties:

1. $C_1 \cup C_2 \cup \cdots \cup C_k = \{1, \ldots, n\}$
2. $C_k \cap C_l = \emptyset \; \forall \, k \neq l$

(James et al., 2013)

Impact of scale

The scale of the variables has a huge impact in the algorithm.

To scale the data before application of k-means.

Exercise

- Begeben Sie sich in Gruppen von 2 bis 3 Teilnehmenden
- Öffnen Sie das Code Notebook “clustering_k_means_exercise.ipynb”
- Führen Sie für den gegebenen Datensatz das k-Means Clustering für k=3 durch
- Dokumentieren Sie was Ihnen aufgefallen ist

Agenda

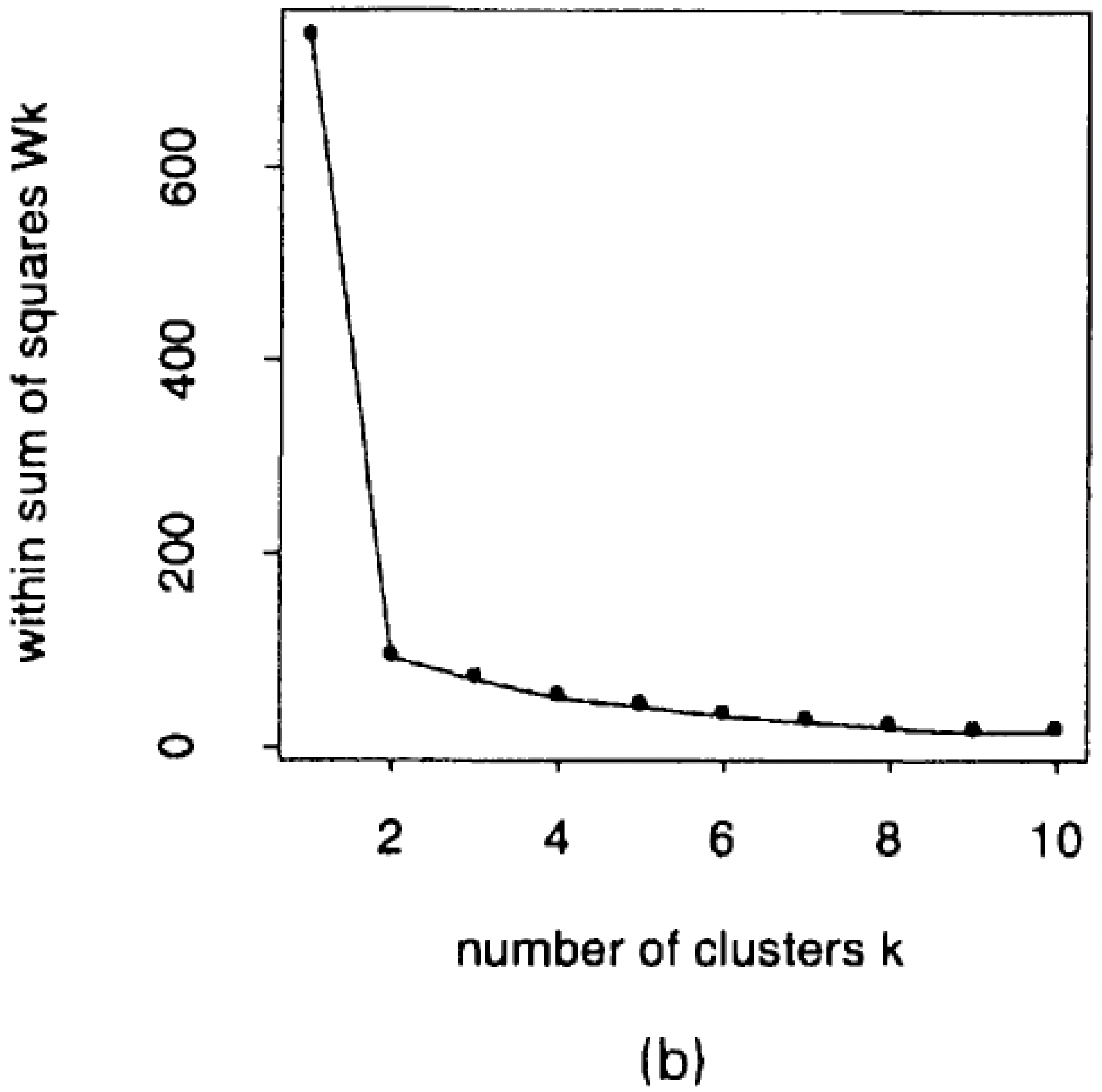
1. Einführung
2. Kennenlernen
3. Clustering
4. K-Means Algorithmus
5. **Determining Optimal Number of Clusters**
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Deep Clustering
12. Zusammenfassung

How to select optimal number of clusters

- But how determine the number of clusters?
- The seen mehtods from Machine Learning cannot be applied
- There exists several methods to assess optimal number of clusters
- We treat three mehtods
  1. Elbow method
  2. Silhouette
  3. GAP Statistic

Elbow Method

- Most popular method
- Idea: Calculating the Within-Cluster-Sum-of Squared (WSS) for different number of clusters
- Elbow method decrease with increasing k



Representation of the elbow method (Tibshirani et al. 2001)

Silhouette

- Silhouette coefficient tells us if individual points are correctly assigned to their clusters.
- Silhouette Coefficient for an observation i is defined as follows

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

  –where $b(i)$ is the smallest average distance of point I to all points in any other cluster
  –and $a(i)$ is the average distance of i from all other points in its cluster

The Silhouette Coefficient for the data set is the average of the Silhouette Coefficient of individual points.

Meaning of $S(i)$:

- $S(i)$ close to 0 means that the observation is between two clusters
- $S(i)$ close to -1 than the point should be assigned to the other cluster
- $S(i)$ close to 1, than the point belongs to the correct cluster

GAP Statistic

Idea: To compare the cluster results with a null reference distribution of the data

The optimal number of cluster is the value of k for which $\log(W_k)$ falls the farthest below the curve of the reference distribution

$$Gap_n(k) = E_n^*\{\log(W_k)\} - \log(W_k)$$

- where W_k is the within-cluster variation
- and $E_n^*\{\log(W_k)\}$ the expectation under a sample of size n form the reference distribution

The optimal $\hat{k}$ is obtained by maximizing $Gap_n(k)$

We assume a null model of a single component and reject it in favor of a k-component model

For each feature a uniformly distributed variable over the range of the observed values is created as reference distribution.

Exercise

- Begeben Sie sich in Gruppen von 2 bis 3 Teilnehmenden
- Öffnen Sie das Code Notebook “clustering_k_means_optimizing_exercise.ipynb”
- Bestimmen Sie die optimale Anzahl Clusters
- Dokumentieren Sie ihre Resultate

Agenda

1. Einführung
2. Kennenlernen
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. **Hierarchical Clustering Algorithmus**
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Deep Clustering
12. Zusammenfassung

Dendogram

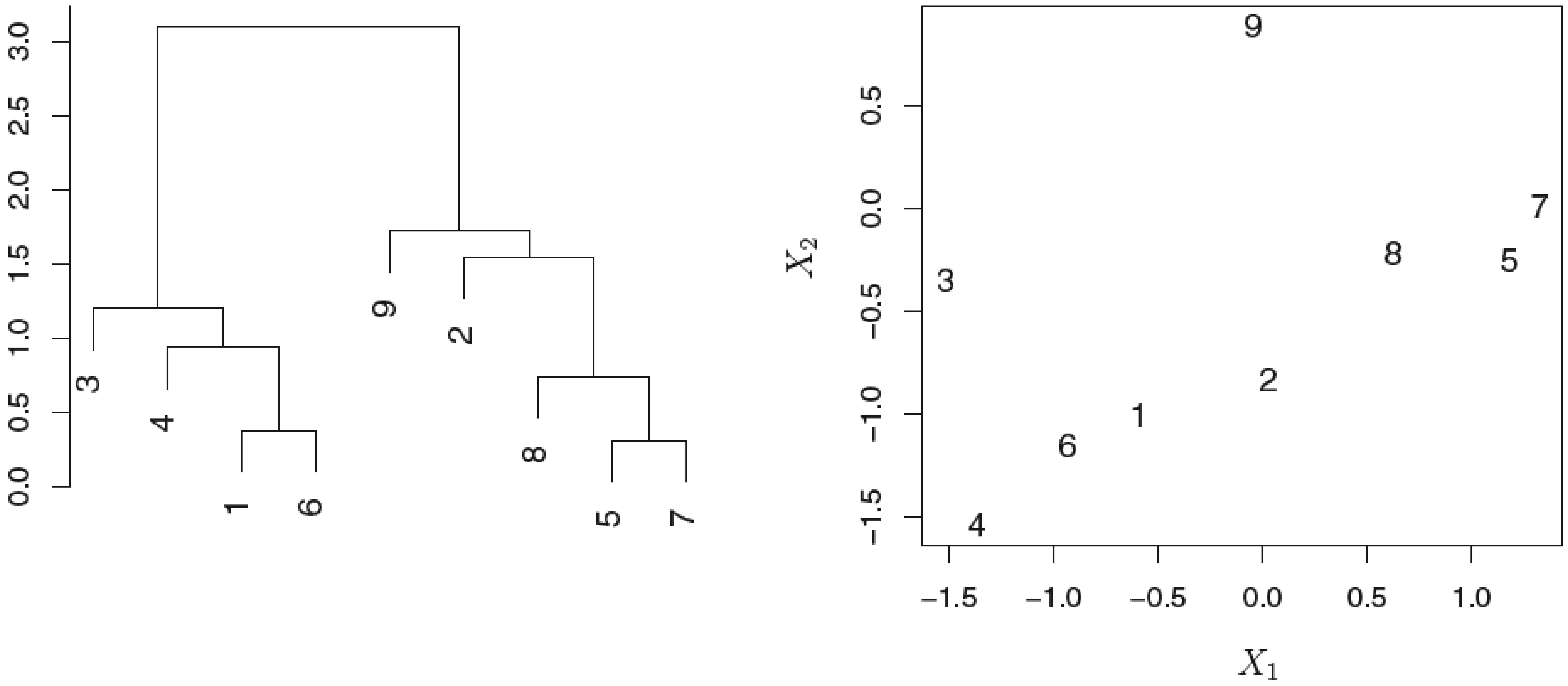


Illustration of a dendogram applied to a simple application (James et al., 2013).

Application example

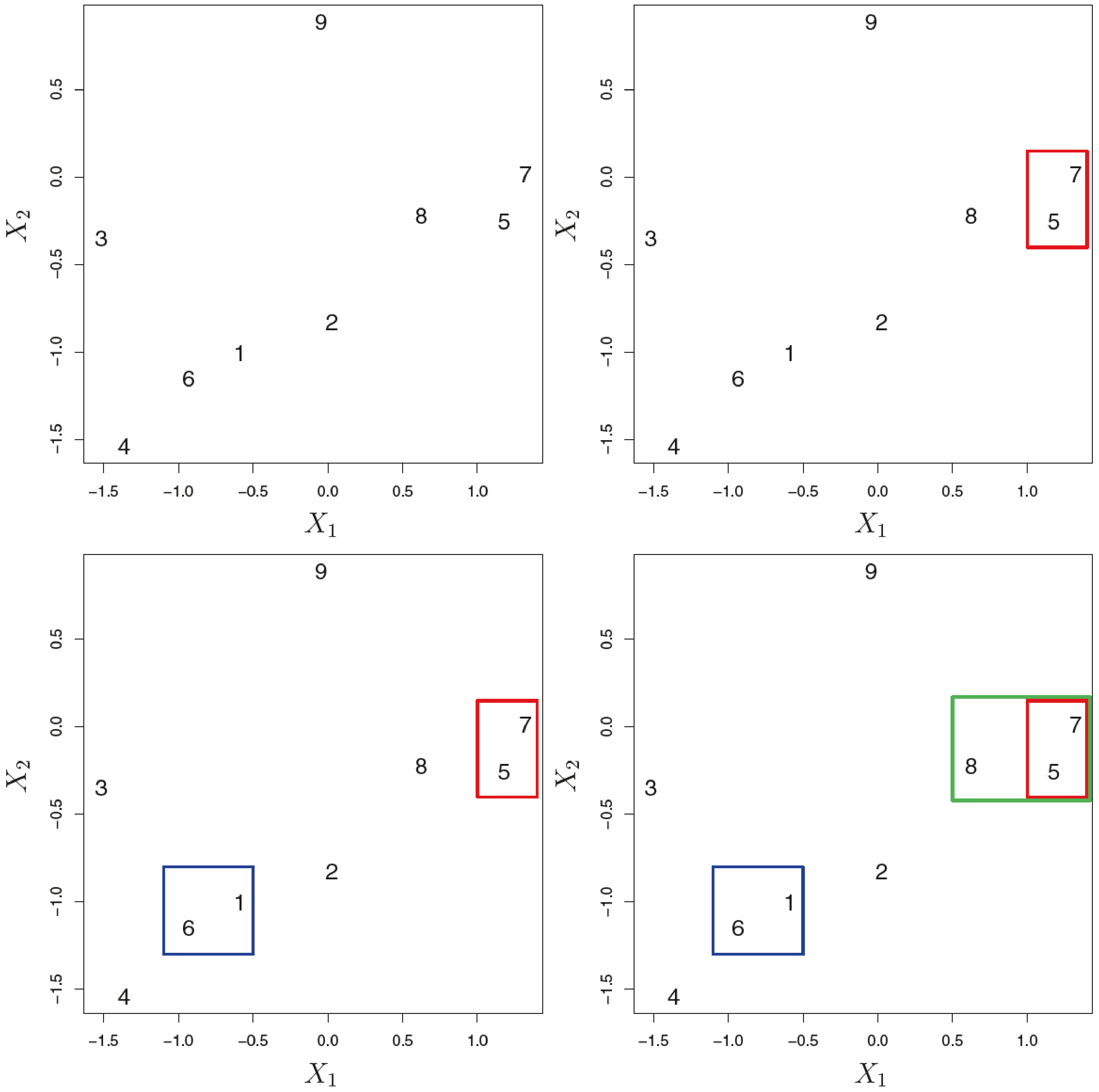
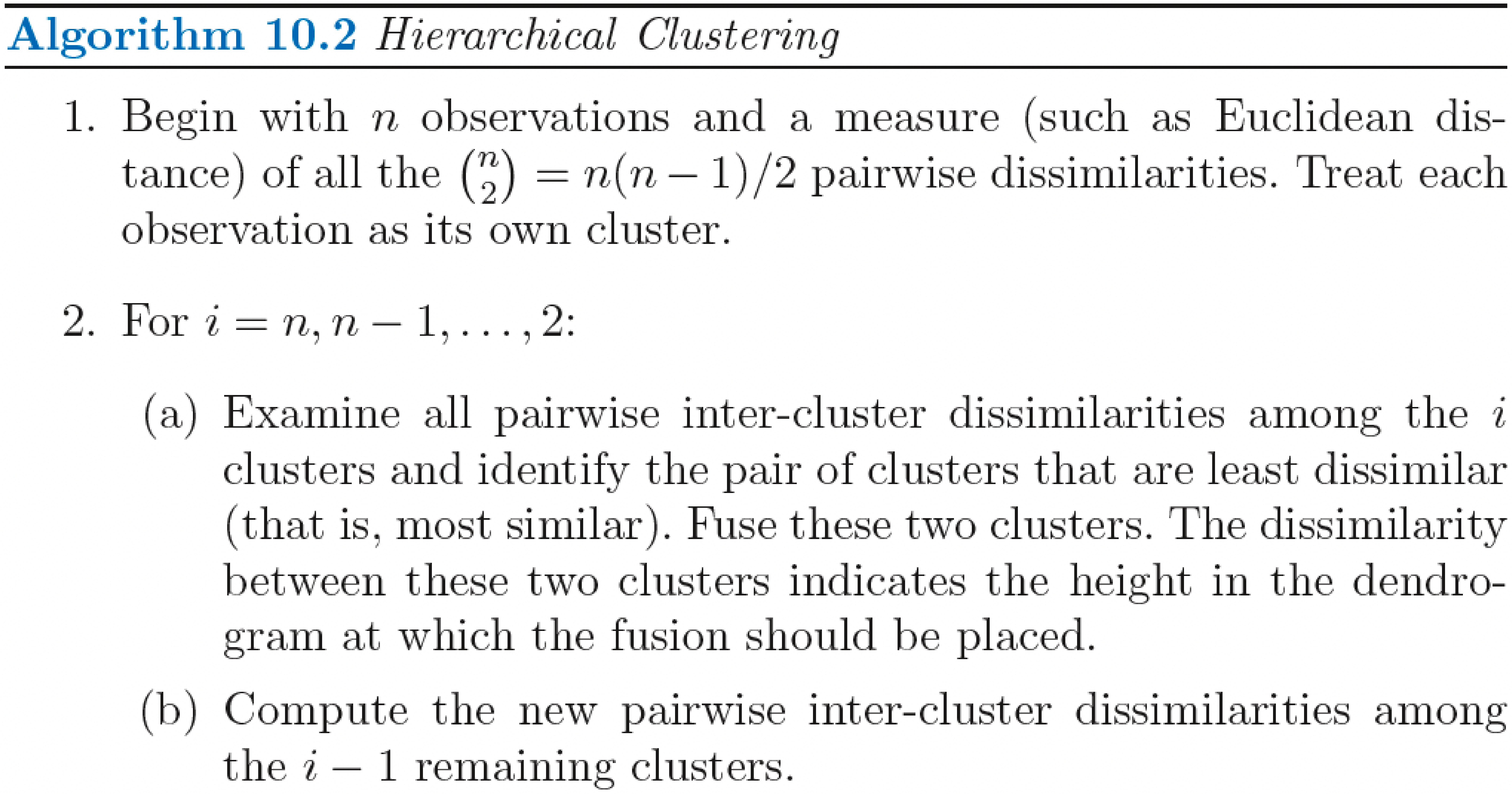


Illustration of an application of the hierarchical clustering algorithm (James et al., 2013).

Algorithm Hierarchical Clustering



Algorithm 10.2 *Hierarchical Clustering*

1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.

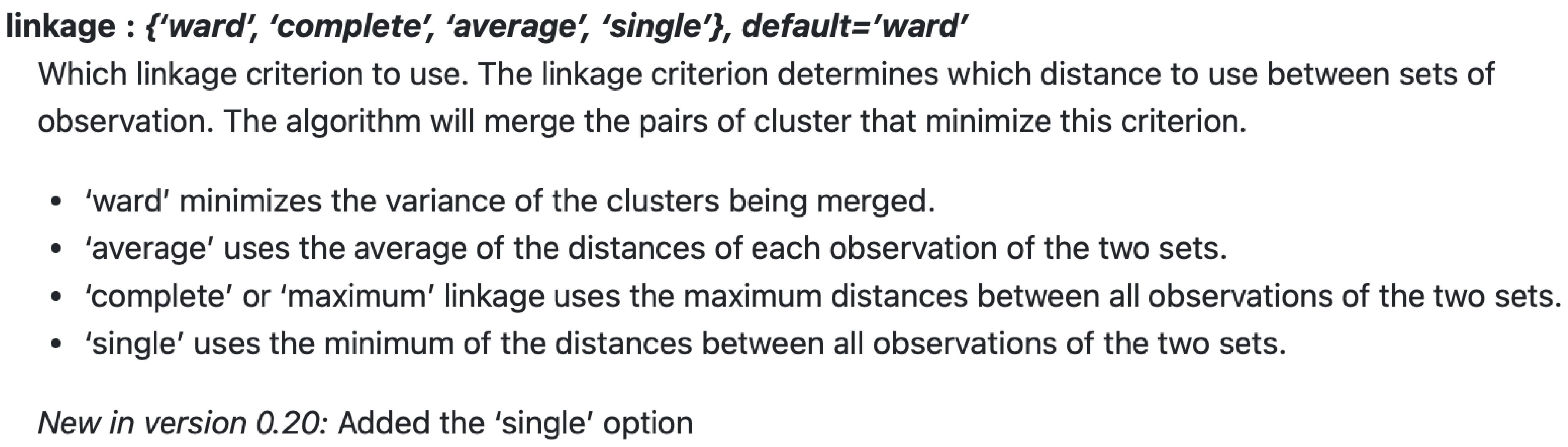
2. For $i = n, n-1, \ldots, 2$:

   (a) Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.

   (b) Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

Illustration of the hierarchical clustering (James et al., 2013).

Linkage functions in hierarchical clustering



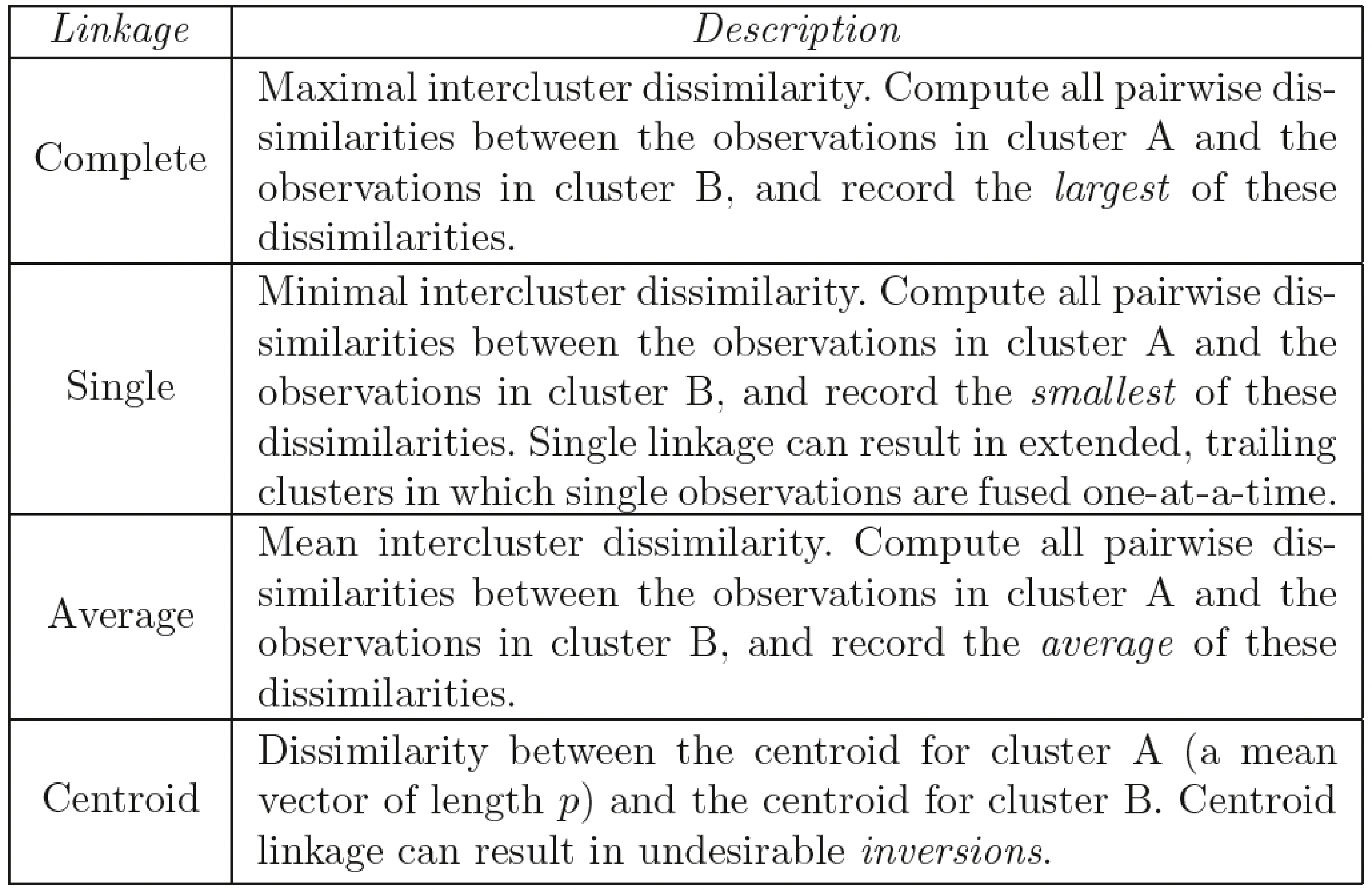
linkage : *{'ward', 'complete', 'average', 'single'}, default='ward'*

Which linkage criterion to use. The linkage criterion determines which distance to use between sets of observation. The algorithm will merge the pairs of cluster that minimize this criterion.

- 'ward' minimizes the variance of the clusters being merged.
- 'average' uses the average of the distances of each observation of the two sets.
- 'complete' or 'maximum' linkage uses the maximum distances between all observations of the two sets.
- 'single' uses the minimum of the distances between all observations of the two sets.

New in version 0.20: Added the 'single' option

Illustration of the different linkage functions applied in hierarchical clustering in Scikit-Learn.



Linkage	*Description*
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

Illustration of the different linkage functions applied in hierarchical clustering (James et al., 2013).

Difference due to the linkage function

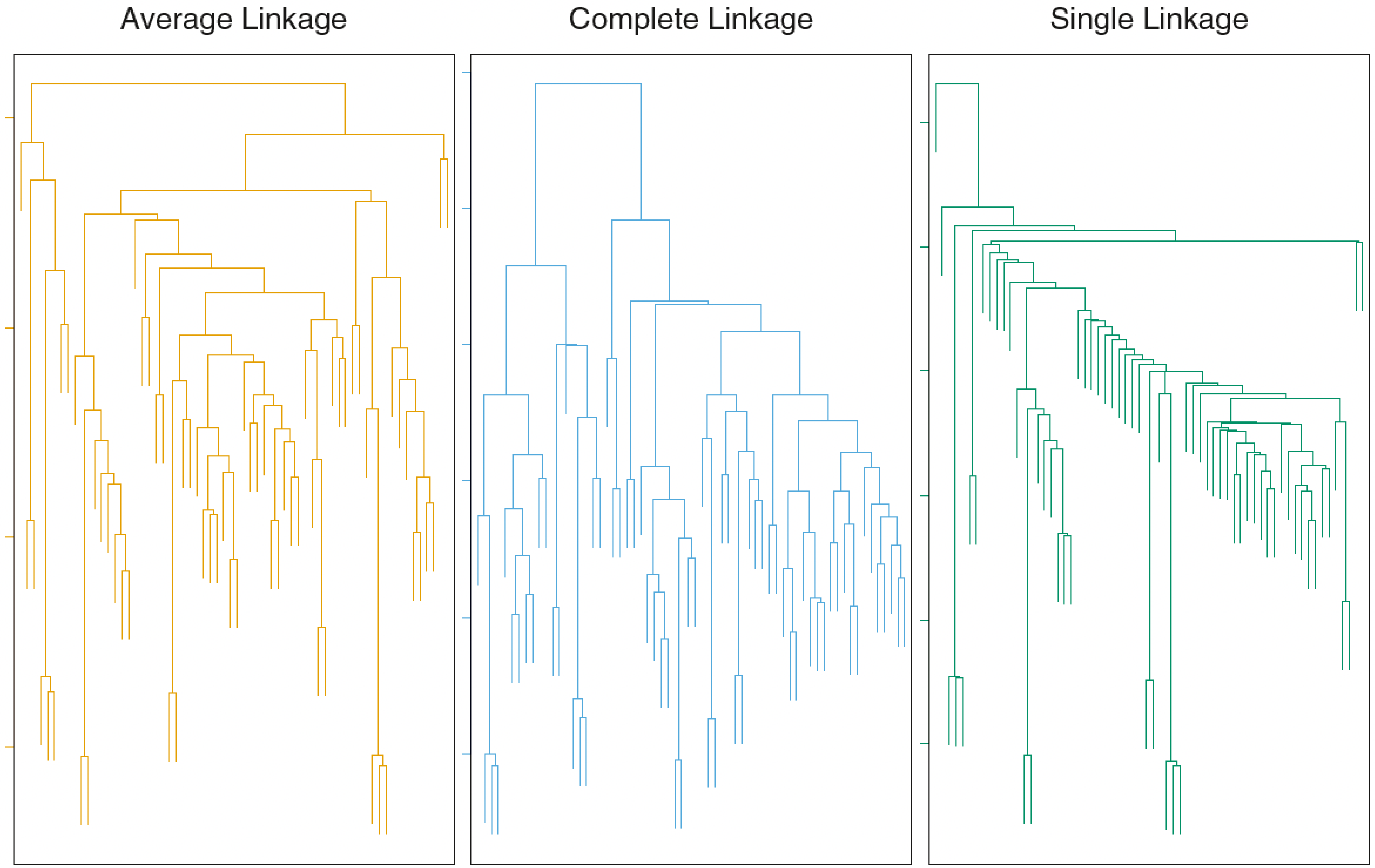


Illustration of differences in the dendogram due to the linkage function (James et al., 2013).

Application example

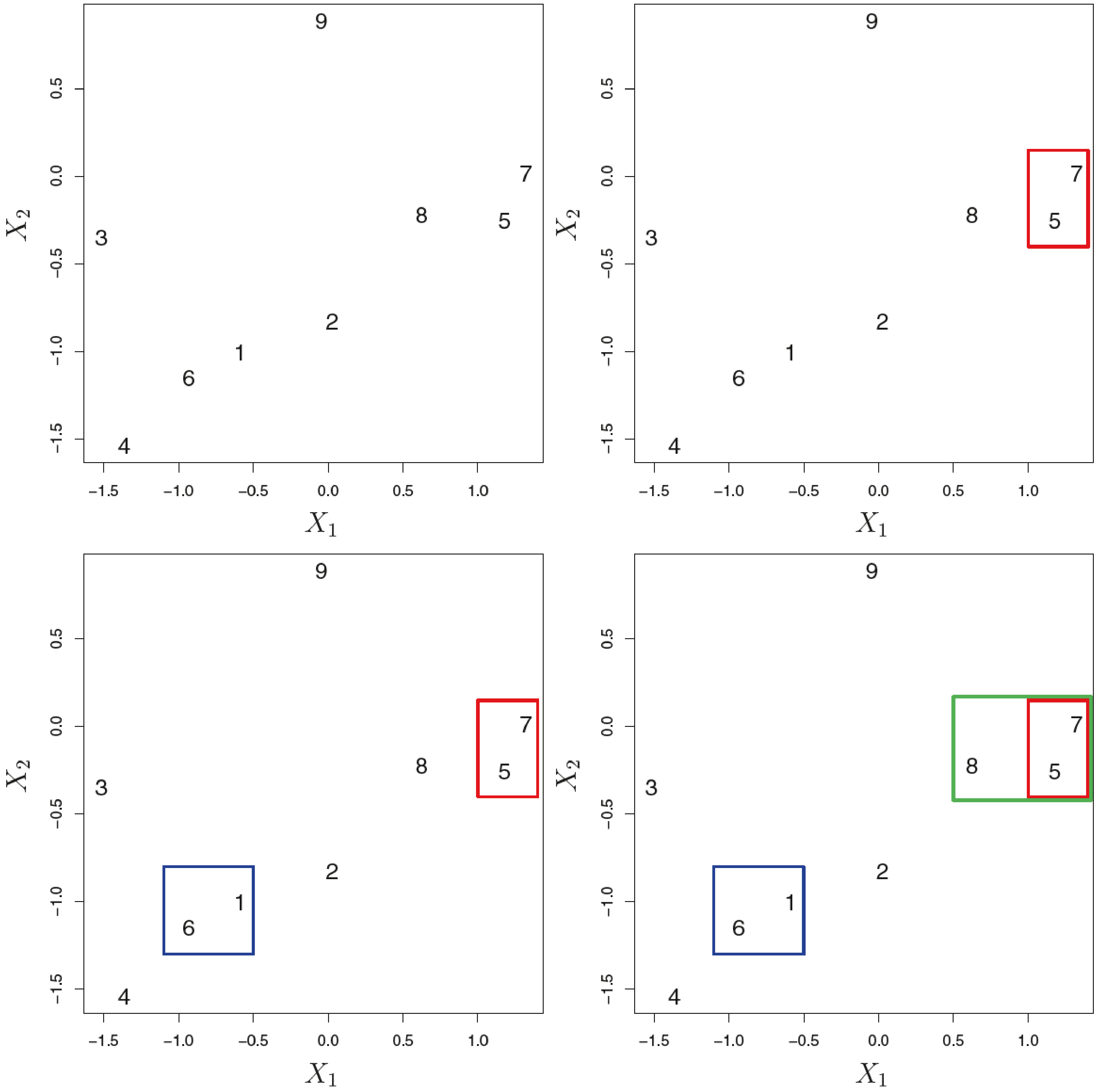


Illustration of an application of the hierarchical clustering algorithm (James et al., 2013).

Small Decisions with Big Consequences

- Standardization of data
- Use of dissamilartiy measure
- Use of linkage
- Number of clusters

Exercise

- Begeben Sie sich in eine Gruppe von 2 bis 3 Teilnehmenden
- Öffnen Sie das Codenotebook “hiearchical_clustering_exercise.ipynb”
- Führen Sie das Hierarchical Clustering vorherigen optimale Clusteranzahl durch
- Dokumentieren Sie ihre Erkenntnisse

Agenda

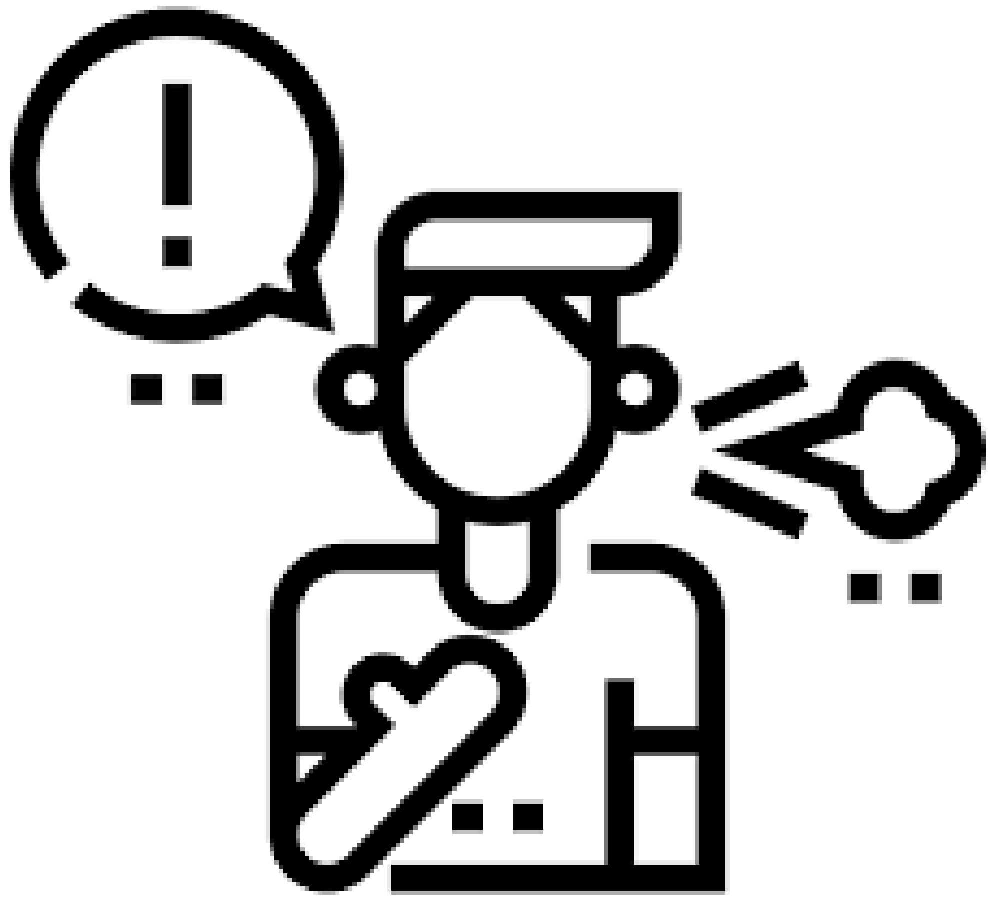
1. Einführung
2. Kennenlernen
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. **Kommunikation der Resultate**
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Deep Clustering
12. Zusammenfassung

Wieso?

Wie unterscheiden sich die Gruppen?

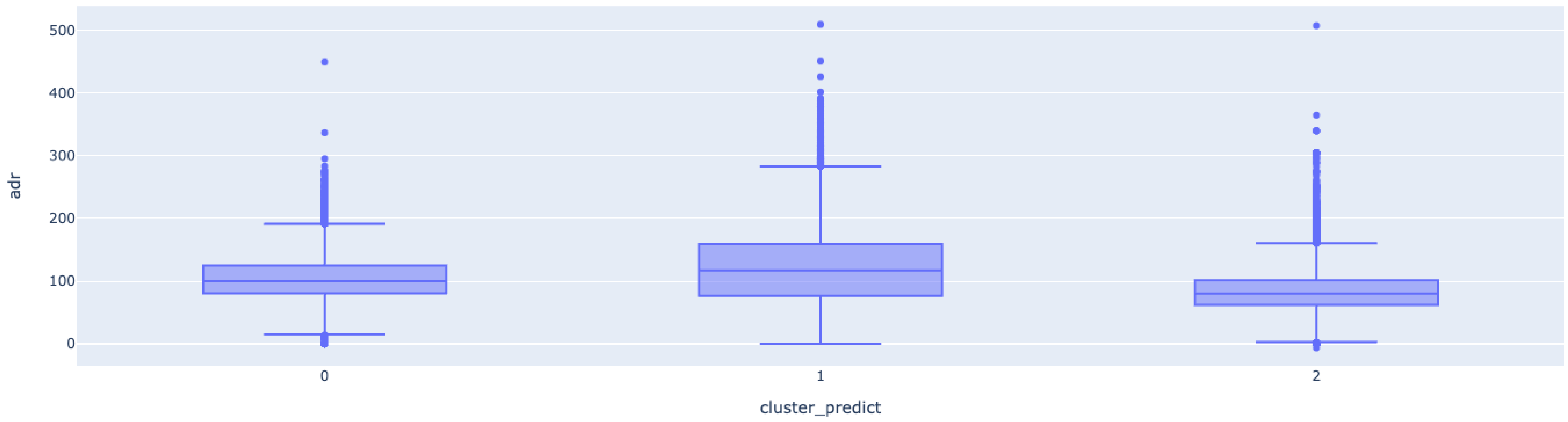
Wie kann ich die Gruppen am besten unterscheiden?

Welche Schwierigkeit ist Ihnen aufgefallen?



Möglicher Weg

- Die Resultate werden auf den skalierten Daten angezeigt
- Wir würden aber gerne den Bezug zu den Ursprungsdaten herleiten
- Lösung: Erhalten Cluster mit den Ursprungsdaten verbinden
- Cluster nach Merkmalen analysieren
- Beste Möglichkeit: Visualisierung und Deskriptive Statistiken



Exercise

- Begeben Sie sich in Gruppen von 2 bis 3 Teilnehmenden
- Beschreiben Sie die Clusters
- Dokumentieren Sie wie sie die Gruppen beschreiben würden

Agenda

1. Einführung
2. Kennenlernen
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. **Dimensions Reduktionsverfahren**
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Deep Clustering
12. Zusammenfassung

Challenge

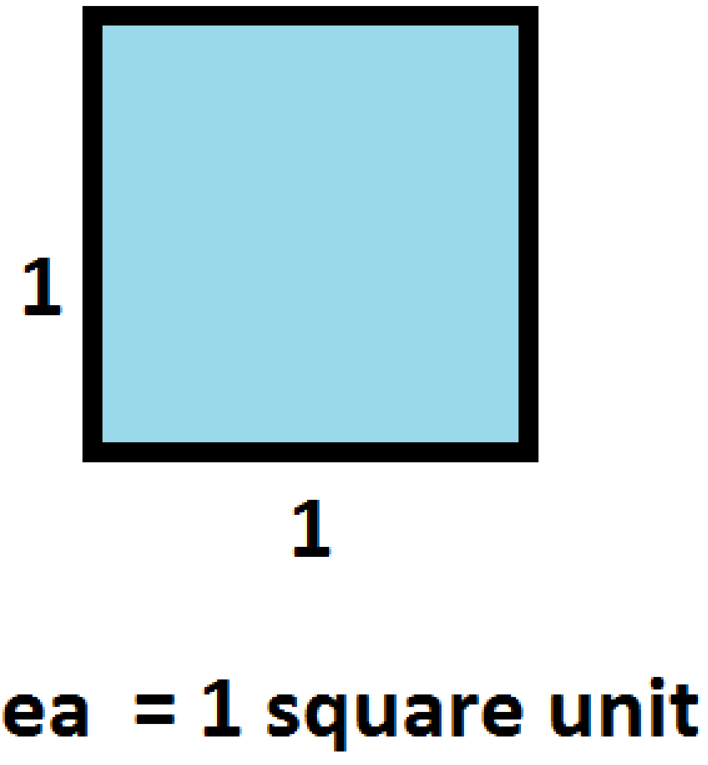
- Many Machine Learning problems involve **thousands** or even **millions** of features for each training instance
- High number of dimensionality can have several **drawbacks**
  - ❖ High dimensionality makes training extremely **slow**
  - ❖ High dimensionality makes it harder to find a **good** solution
  - ❖ High dimensionality makes it harder to **interpret** the results
- The high dimensionality problems are known as **curse of dimensionality**
- Dimensionality reduction reduce the **number of features** considerably
- Dimensionality reductions turns an **intractable** problem into a **tractable** problem
- Dimensionality reduction is extremely useful for **data visualization**

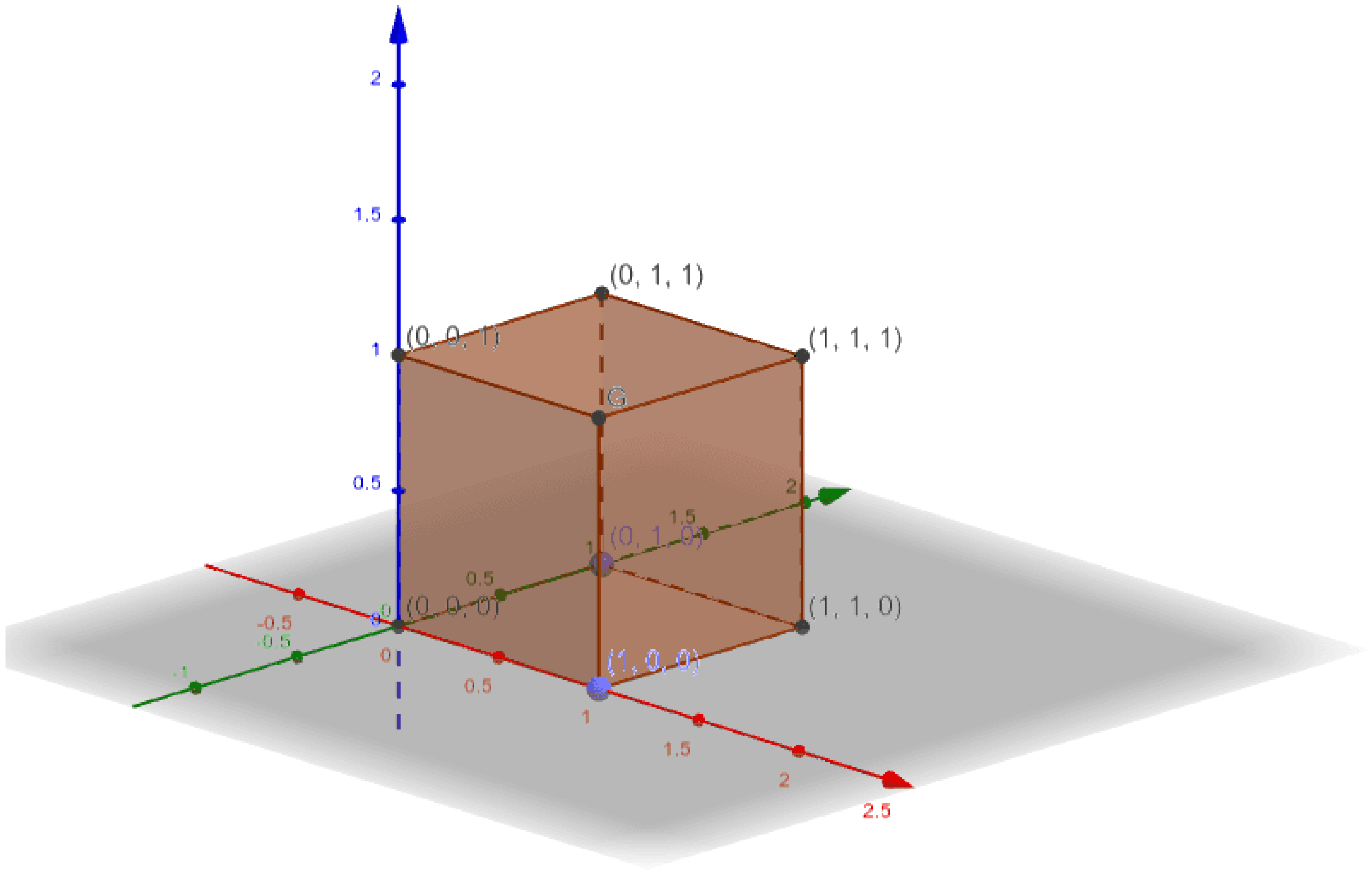
Curse of Dimensionality

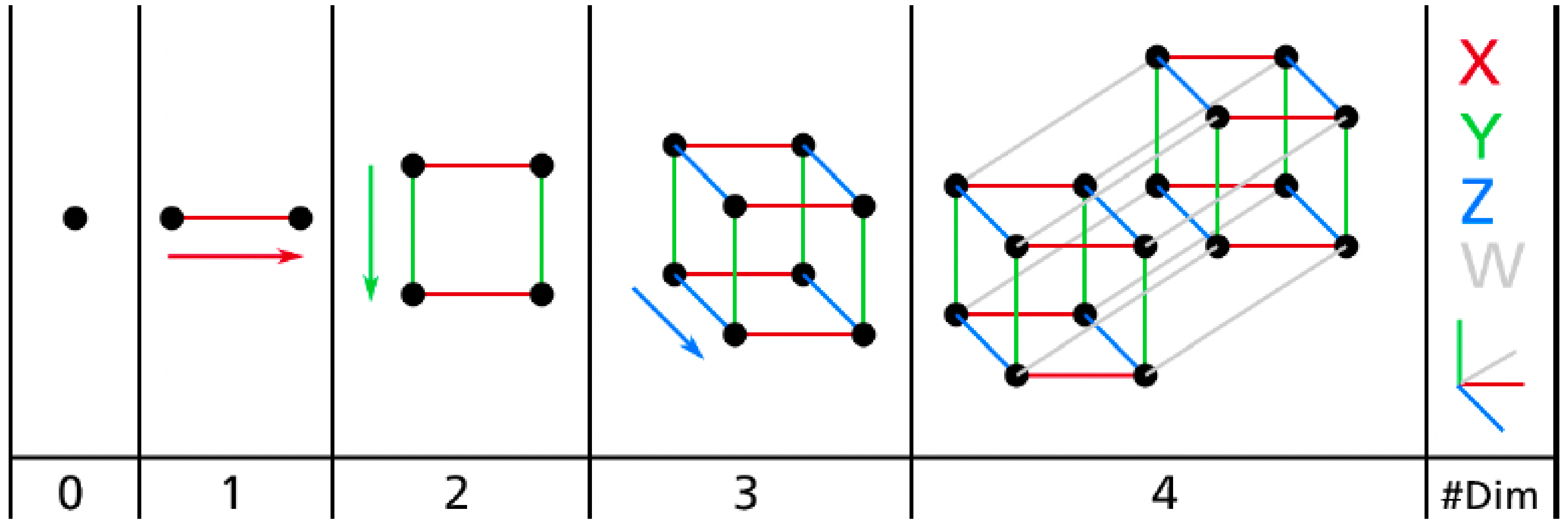
If you pick two points randomly in a **unit square** the distance between these tow points will be, on average, roughly **0.52**

If you pick two points randomly in a **unit 3D cube**, the average distance will be roughly **0.66**

If you pick randomly two points in a **1’000’000-dimensional hypercube**, the average distance will be roughly about **408.25**







Curse of Dimensionality

High dimensional datasets are very **sparse**

Most training instances are likely to be far **away** from each other

New instance will likely be far away from any training instance

Prediction for new instance will be **less reliable** in high dimension than in lower dimension

More dimensions the training set has, the greater the risk of **overfitting**

One solution to the curse of **dimensionality** cold be to increase the size of the training set to reach a sufficient density of training instances

In practices, it is generally unfeasible to increase the training instances as required

Main Approaches for Dimensionality Reduction

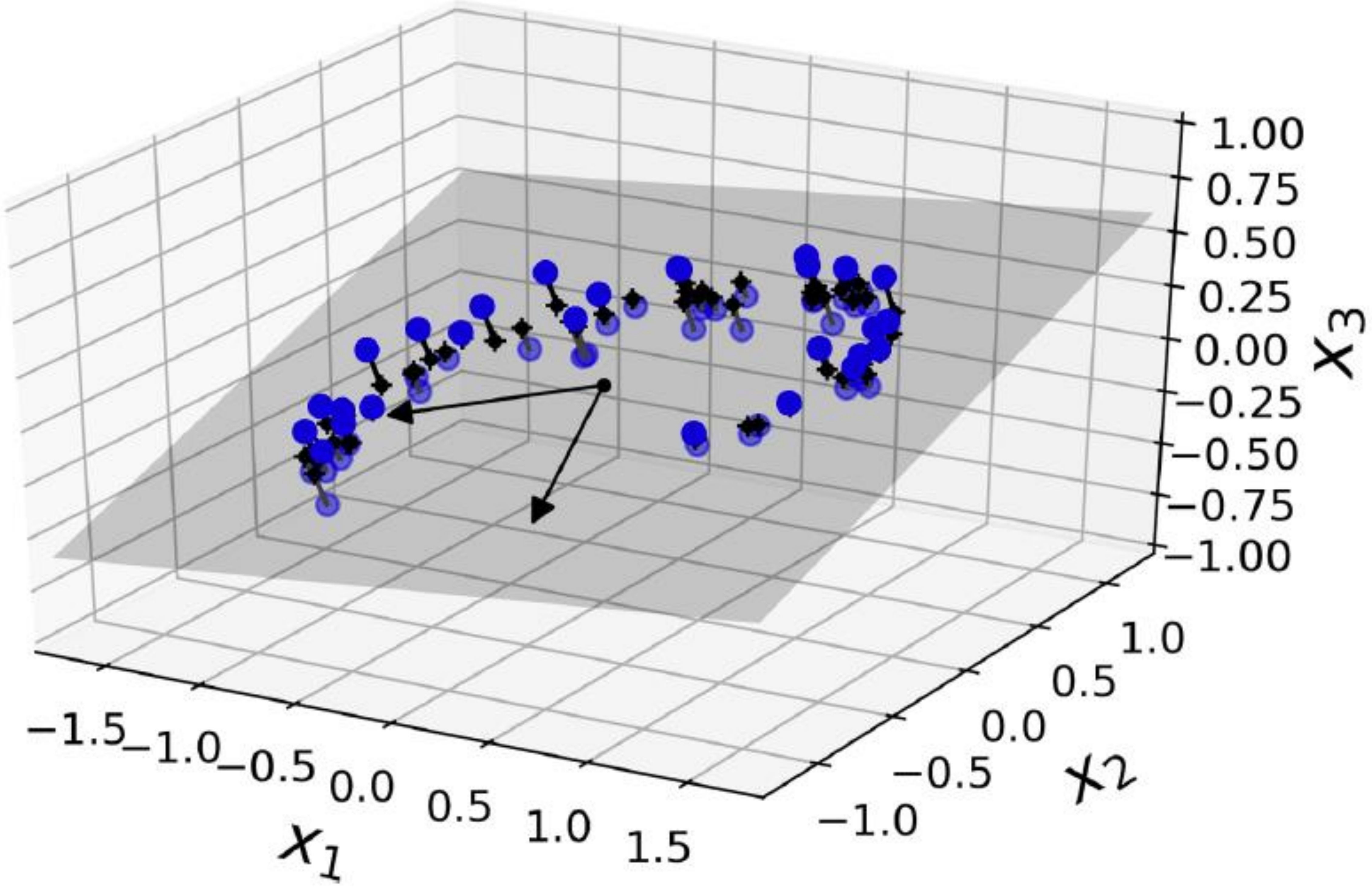
Two main approaches to reducing dimensionality:

- ➢ Projection
- ➢ Manifold Learning

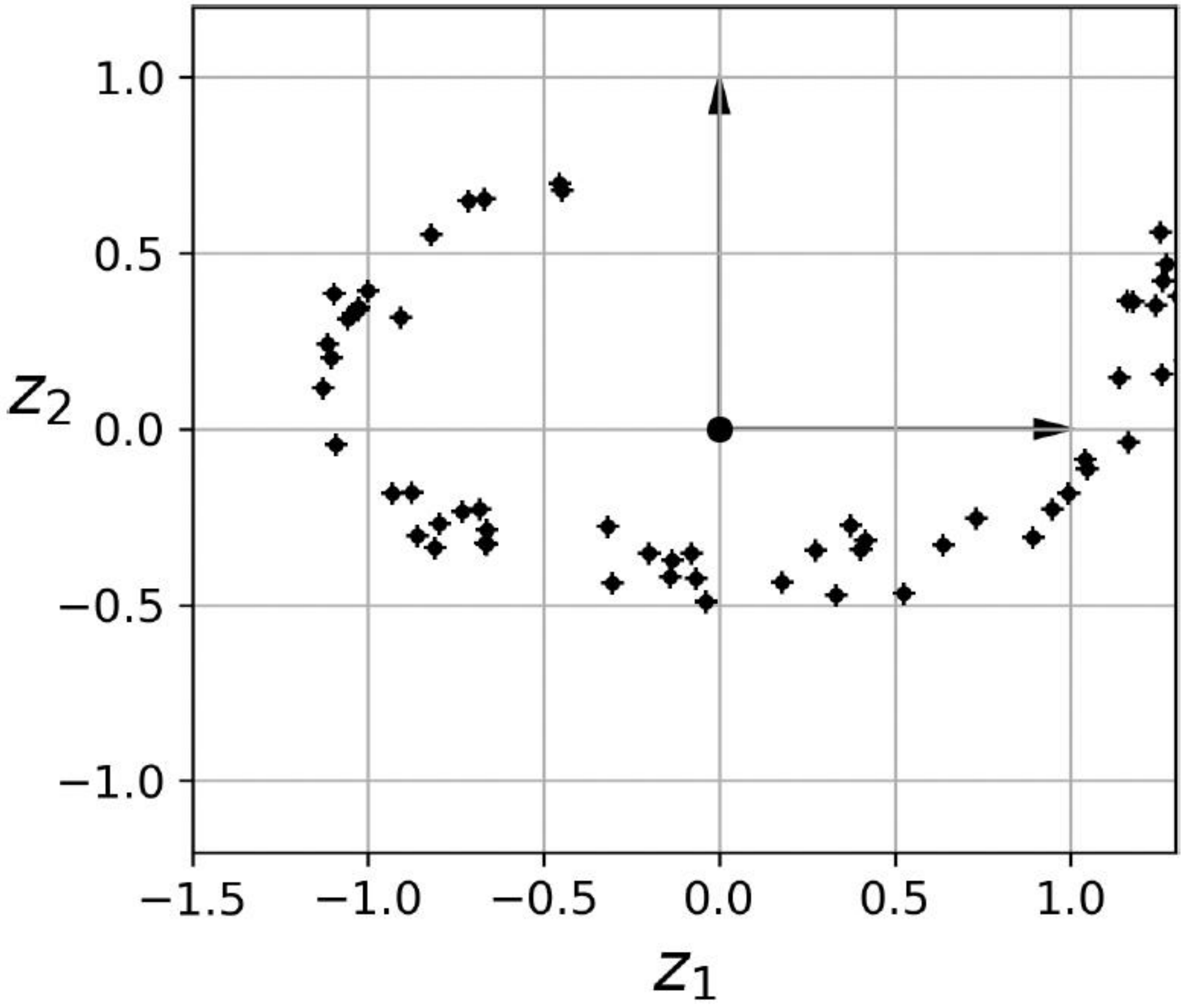
Importance of dimensionality reduction

- In real-world problems, training instances are not spread uniformly across all dimensions
- Many features are constant or highly correlated
- All training instances mostly lie within a much lower-dimensional subspace of the high-dimensional space

Projection

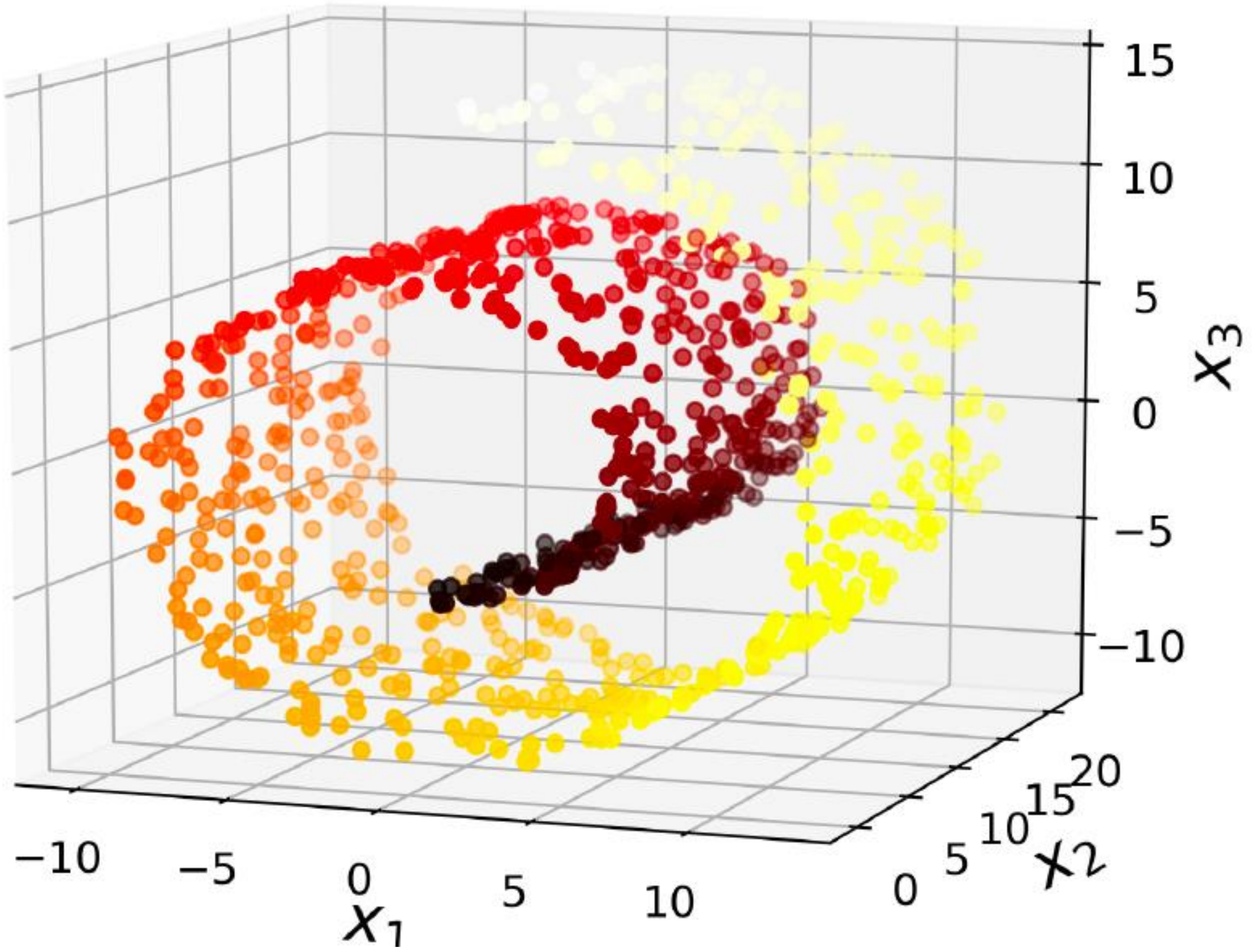


Representation of a 3D dataset (Géron, 2019)



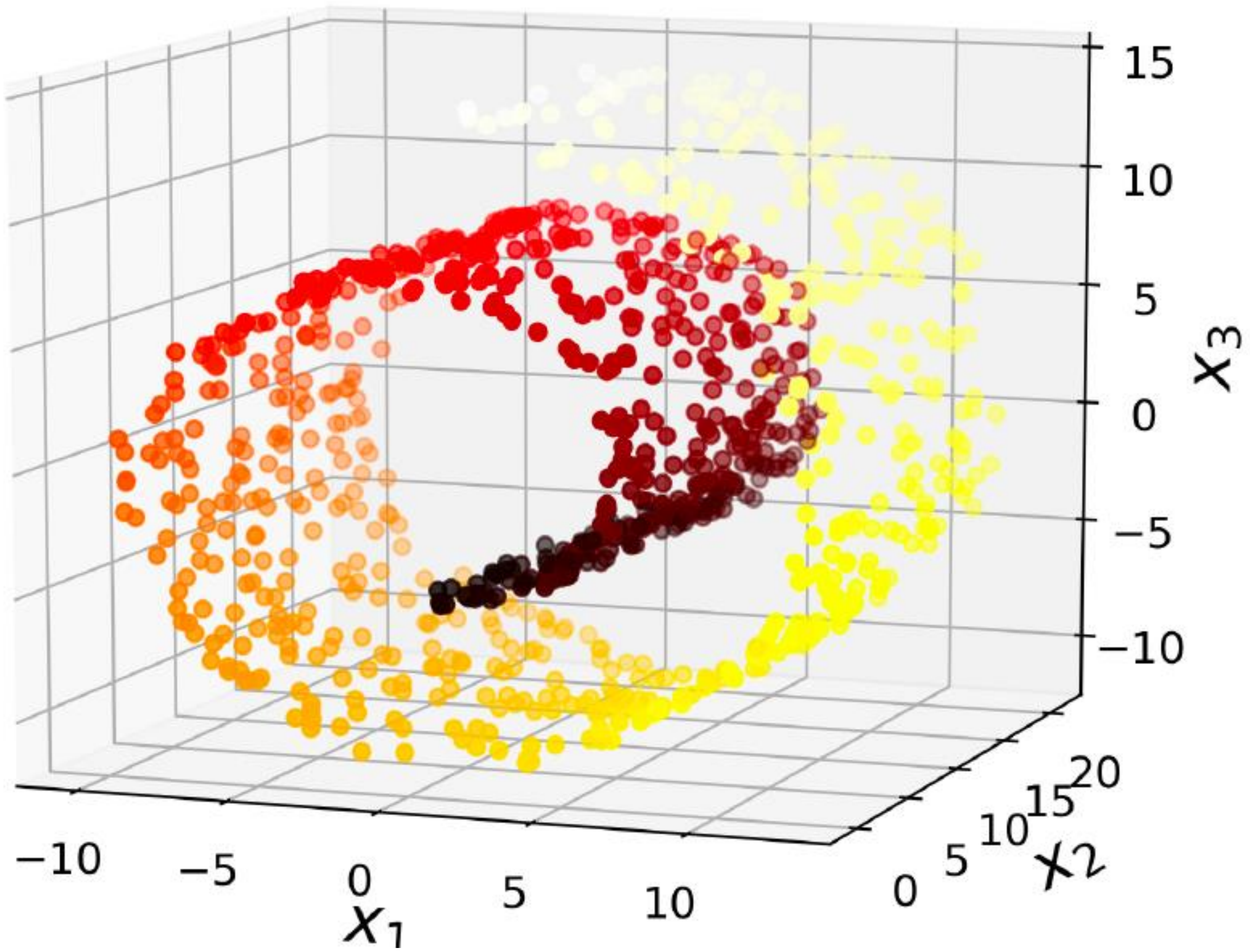
Representation of a 2D project (Géron, 2019)

Possible Drawbacks of Projection

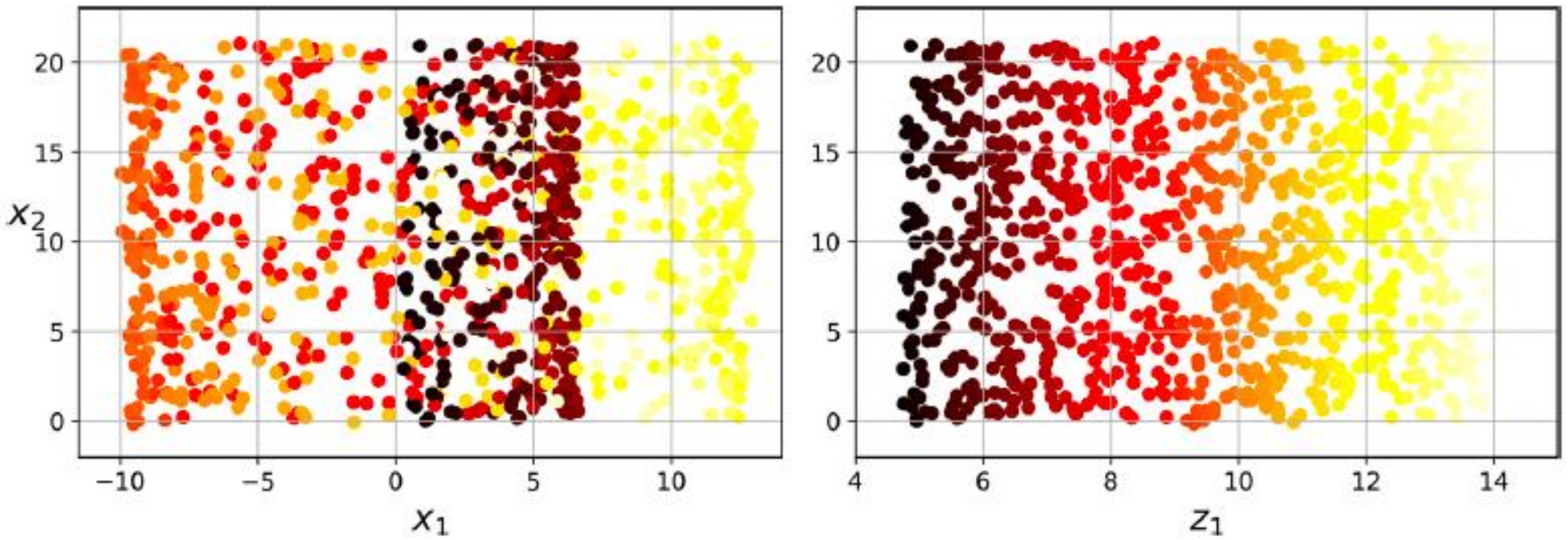


Visualization of the Swiss roll dataset (Géron, 2019)

Possible Drawbacks of Projection



Visualization of the Swiss roll dataset (Géron, 2019)



Visualization of the Swiss roll dataset: Left projection on the plane, right unrolling Swiss roll (Géron, 2019)

Manifold Learning

2D manifold is a 2D shape that can be bent and twisted in higher-dimensional space

d-dimensional manifold is a part of a n-dimensional space (where d-dimensional space)

Manifold Laerning reduces the the dimensionality of the raining instances by modleing the manifold on which the trianing instances lies

Drawbacks of Dimensionality Reduction

- Through reducing dimensionality, we **lose** some information
- The dimensionality reduction can speed up training, however, lead to a slightly **worse** performance
- In some cases, dimensionality reduction of the training data may filter out some **noise** and unnecessary details
- When dimensionality reduction filter out noise, we get a gain in performance
- Always should firs try to train system on the **original** data

Agenda

1. Einführung
2. Kennenlernen
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. **Principal Components Analysis**
10. T-Distributed Stochastic Neighbor Embedding
11. Deep Clustering
12. Zusammenfassung

Principal Component Analysis

Goal: to find a low representation that captures as much of the information as possible (James et al., 2017)

Let $X_1, X_2, \ldots, X_p$ be the p features with n observations.

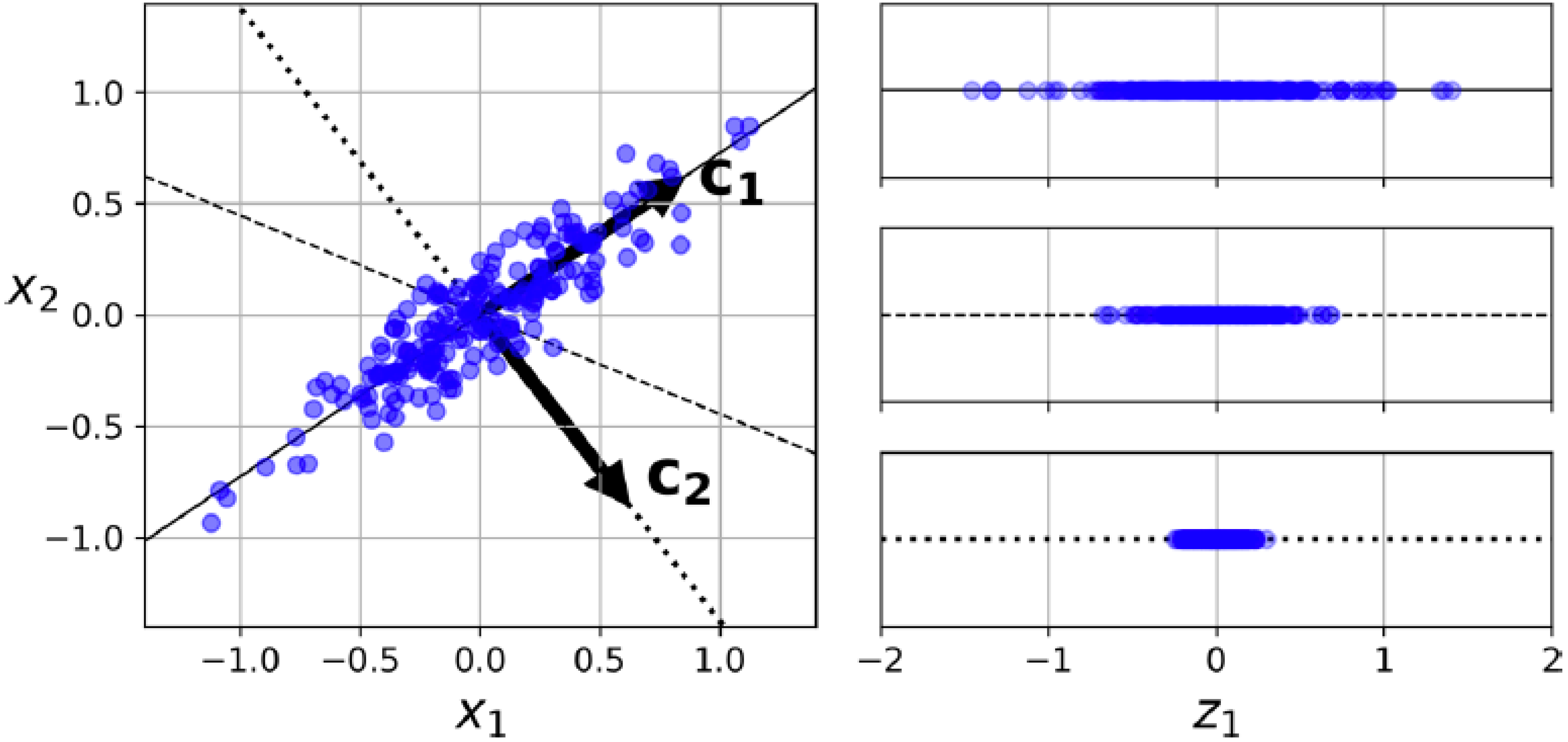
First principal component: $Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \ldots + \phi_{p1}X_p$

- with largest variance and $\sum_{j=1}^{p} \phi_{j1}^2 = 1$
- ϕ_{j1} are the loadings and z_{i1} are the scores.

Optimization Problem for first principal component:

$$\max_{\phi_{11}, \phi_{21}, \ldots, \phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ with } \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

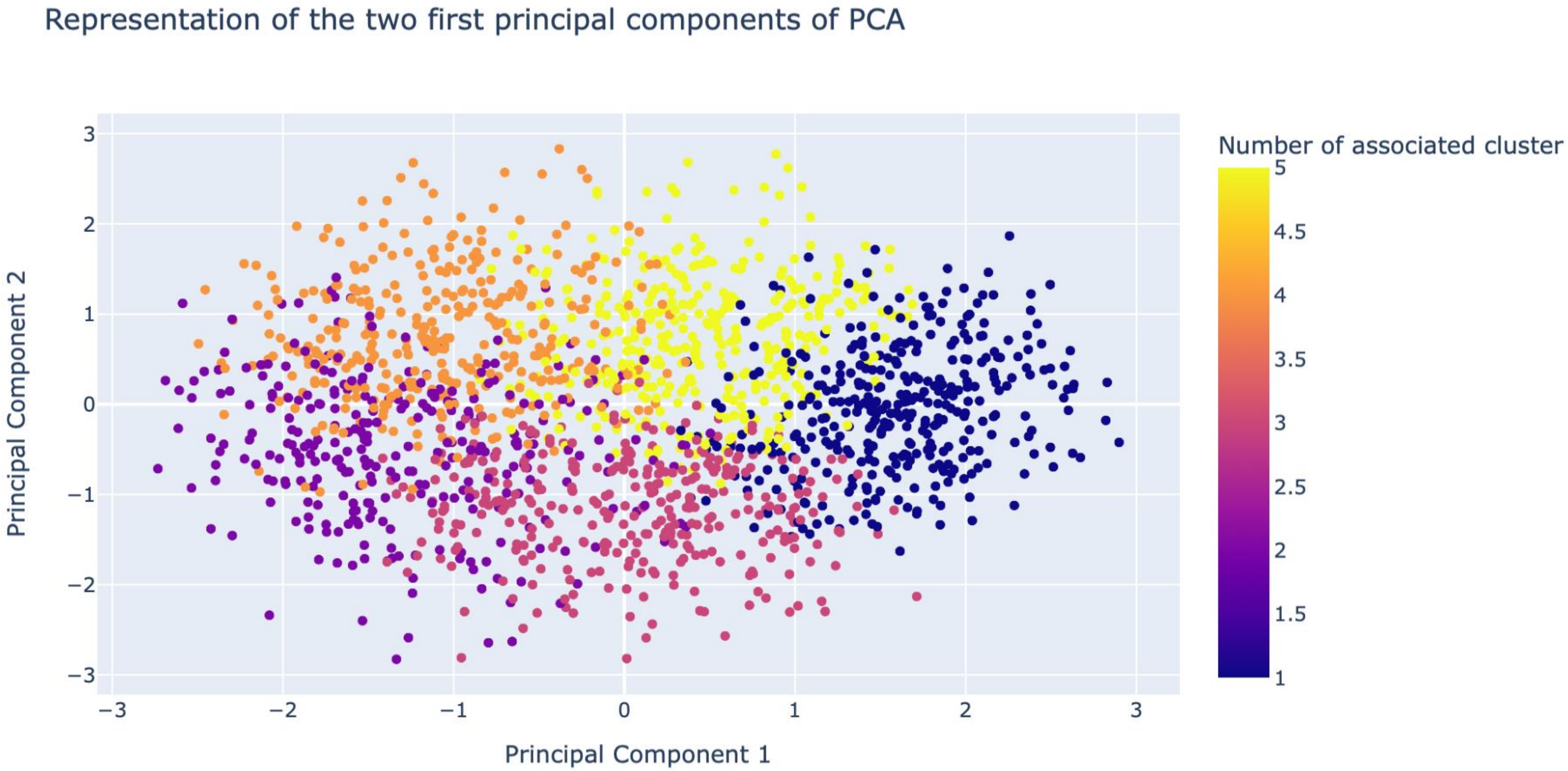
Selecting the subspace to project on



Principal Component Analysis

- When the first principal component Z_1 of the features has been determined, we can find the second principal component Z_2 (James et al., 2017; Kuhn et al., 2016).
- Z_2 uncorrelated with Z_1

Example of PCA



Exercise

- Begeben Sie sich in ein Gruppe von 2 bis 3 Teilnehmenden

- Öffnen Sie das Code Notebook “dimension_reduction_pca_exercise.ipynb”

- Reduzieren Sie den ”Feature” Raum auf zwei Dimensionen

- Dokumentieren Sie ihr Vorgehen und ihre Resultate

Agenda

1. Einführung
2. Kennenlernen
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. **T-Distributed Stochastic Neighbor Embedding**
11. Deep Clustering
12. Zusammenfassung

T-SNE (Stochastic Neighbor Embedding)

Data: data set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$

Cost function parameters: perplexity (perp)

Optimization Parameters:

- Number of iterations T,
- Learning rate η,
- Momentum $\alpha(t)$

Result: Low-dimensional data representation $\mathcal{Y}^t = \{y_1, y_2, \ldots, y_n\}$

(van der Maaten and Hinton, 2008)

t-Distributed Stochastic Neighbor Embedding (t-SNE)

Video: Design at Large – Laurens van der Maaten, Visualizing Data Using Embeddings
https://www.youtube.com/watch?v=EMD106bB2vY

Video explain t-SNE

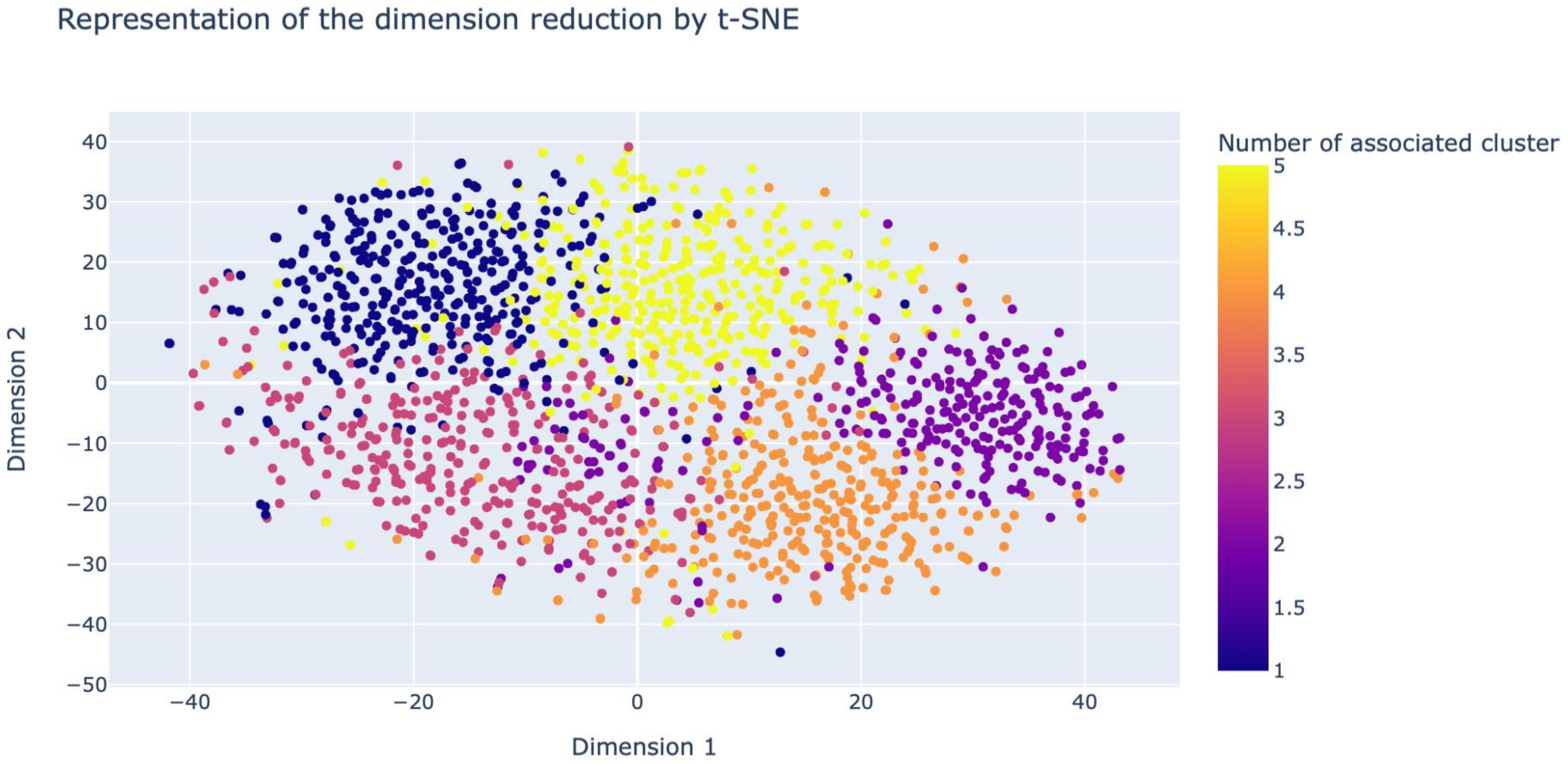
Idea of the method is explained from time 11:15 to 24:00

Pseudo Code T-SNE

```
Begin
    Compute pairwise affinities $p_{j|i} = \frac{exp\left(-\|x_i - x_j\|^2/(2\sigma_i^2)\right)}{\sum_{k\neq i} exp\left(-\|x_i - x_k\|^2/(2\sigma_i^2)\right)}$ with perplexity $Perp(P_i) = 2^{H(P_i)}$
    where $H(P_i)$ is the Shannon entropy $H(P_i) = -\sum_j p_{j|i}\log_2 p_{j|i}$
    Set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2}$
    Sample initial solution $\mathcal{Y}^0 = \{y_1, y_2, \dots, y_n\}$ from $\mathcal{N}(0, 10^{-4}\mathfrak{I})$
    For t = 1 to T do
        Compute low dimensional affinities $q_{ij} = \frac{\left(1+\|y_i - y_j\|^2\right)^{-1}}{\sum_{k\neq l}(1+\|y_k - y_l\|^2)}$
        Compute gradient $\frac{\partial C}{\partial \mathcal{Y}} = 4\sum_j (p_{ij} - q_{ij})(y_i - y_j)\left(1 + \|y_i - y_j\|^2\right)^{-1}$
        Set $\mathcal{Y}^t = \mathcal{Y}^{(t-1)} + \eta\frac{\partial C}{\partial \mathcal{Y}} + \alpha(t)\left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}\right)$
    end
End
```

(van der Maaten and Hinton, 2008)

Example of T-SNE



Exercise

- Begeben Sie sich in ein Gruppe von 2 bis 3 Teilnehmenden
- Öffnen Sie das Code Notebook “dimension_reduction_tsne_exercise.ipynb”
- Reduzieren Sie den Raum
- Dokumentieren Sie ihr Vorgehen und ihre Resultate

Agenda

1. Einführung
2. Kennenlernen
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. **Deep Clustering**
12. Zusammenfassung

Goal of Clustering and Limitations

- Aim: to group a set of unlabeled data based on homogeneous patterns in a given feature space

- Clustering is very suitable for exploring the underlying structure of data sets that get a well-classified subset of objects (Huang et al., 2021)

- Clustering is an important preprocessing technique in machine learning, and it is the core of many data driven based learning technologies (Huang et al., 2021)

- Traditional clustering algorithms attain a limited performance as the dimensionality goes higher (Huang et al., 2021)

Deep Clustering

- Traditional clustering methods first map the objects from the raw feature space to low-dimensional feature space or after feature transformation and then clustering (Huang et al, 2021)

- Deep clustering is a new research direction that combines deep learning and clustering.

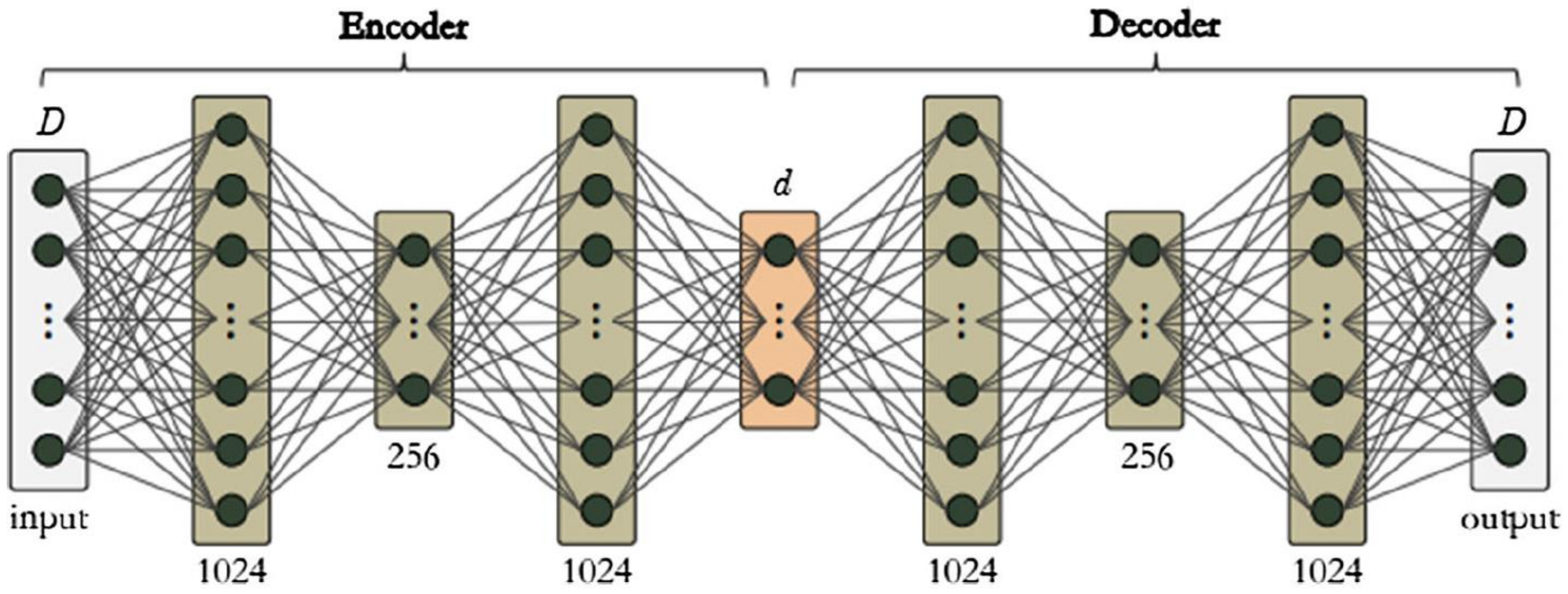
- Deep clustering approaches extract the higher-level features form the raw features of the input through DNNs

- The auto-encoder is a neural network model, which can learn the hidden features of the input object to achieve

Deep-Autoencoder and Deep Convolutional Autoencoders

Deep-autoencoders (DAEs) and deep convolutional auto-encoders are unsupervised models for representation learning

The training is performed in an unsupervised manner by minimizing the differences between original data and reconstructed data with distance metrics.



Autoencoder and Clustering

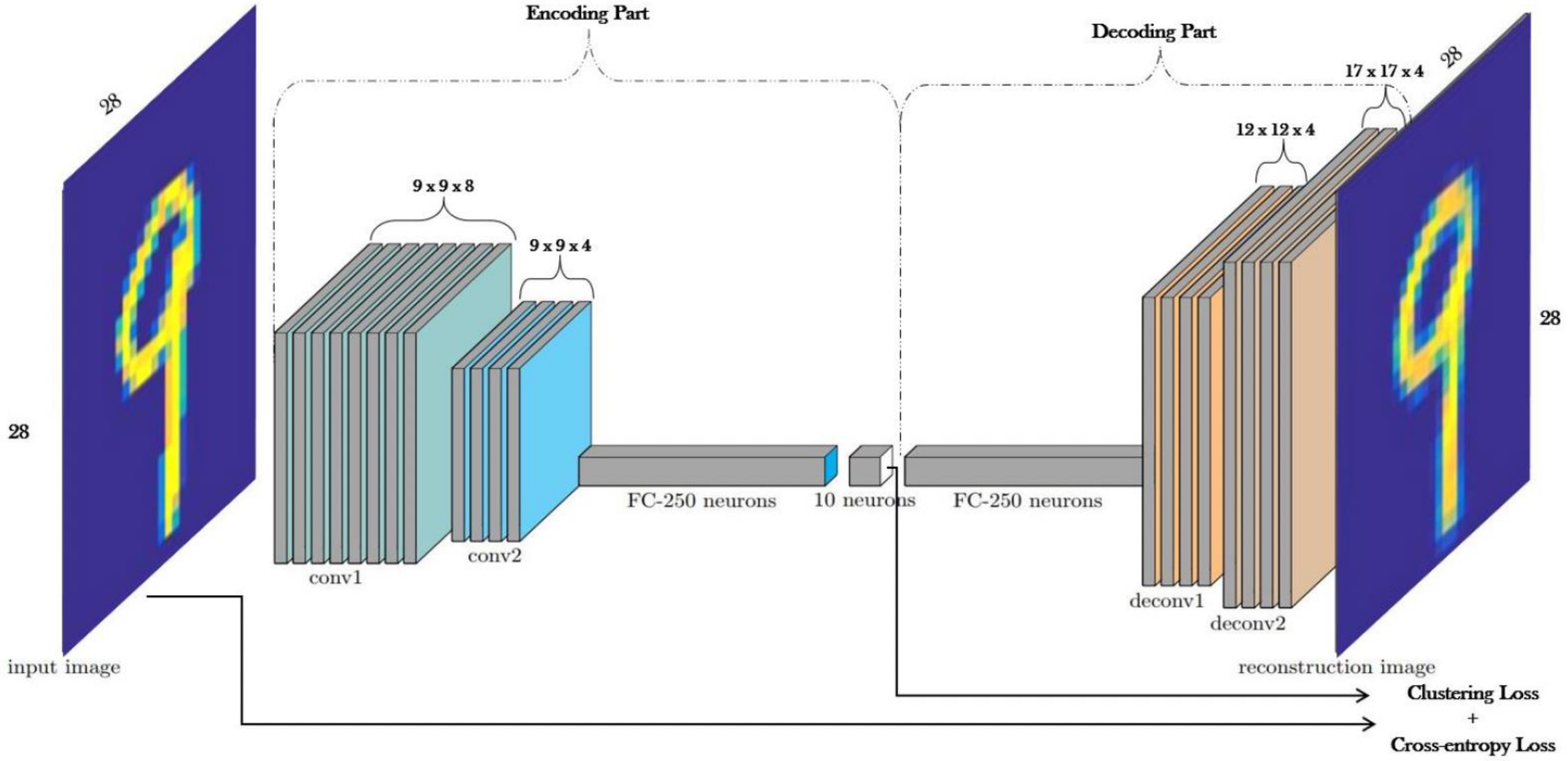
- Deep networks (e.g. AE or CAE) provide an abstracted latent representation

- Latent representation is used for clustering analysis.

- AEs to learn a lower dimensional representation space

- Separated Clustering:
  - two separate procedures: feature space learning and clustering
  - both procedures are not jointly optimized

Extension Embedded Clustering – Alqahtani et al. (2018)



Visualisation of the model archtiecture of Alqahtani et al. (2018)

The **goal** of our clustering model is to learn feature representations and cluster assignments simultaneously

The **overall** cost function is a combination of two parts:

1. the first part is essentially the cross-entropy loss minimizing the reconstruction error,
2. while the second part is clustering objective function minimizing the distance between data representations in the latent space and their corresponding cluster centers

$$\min_{W,b} E_1 + E_2$$

Extension Embedded Clustering – Alqahtani et al. (2018)

- DCAE extracts latent representations through its internal layer by minimizing the reconstruction error.

$$E_1 = -\frac{1}{N}\sum_{n=1}^{N}(y_n \log \widehat{y}_n + (1-y_n)\log(1-\widehat{y}_n))$$

- The clustering objective minimizes the distance between data samples and assigned centroids in latent space as follows

$$E_2 = \lambda \cdot \frac{1}{2N}\sum_{n=1}^{N} \parallel h^t(x_n) - c_n^* \parallel^2$$

- N denotes the number of samples,
- λ is clustering weight-parameter that control the contribution percentage of clustering cost function in the overall cost function
- $h^t(x_n)$ is the latent representation at the t^{th} iteration and n^{th} sample
- c_n^* is the assigned cluster center

Agenda

1. Einführung
2. Kennenlernen
3. Clustering
4. K-Means Algorithmus
5. Determining Optimal Number of Clusters
6. Hierarchical Clustering Algorithmus
7. Kommunikation der Resultate
8. Dimensions Reduktionsverfahren
9. Principal Components Analysis
10. T-Distributed Stochastic Neighbor Embedding
11. Deep Clustering
12. **Zusammenfassung**

Zusammenfassung

Was haben wir gelernt?

Feedback

Vielen Dank für euer Feedback auf folgender Seite

http://www.evaluationszielscheibe.ch/?disc=7ebb64

Fragen

Darstellung eines Fragesymbol aufgerufen von der Webseite https://www.qnigge.de/news/detail/modul-v/#images am 12.07.2021.

Referenzen

Galli, S. (2017). Feature Engineering for Machine Learning, https://github.com/solegalli/feature-engineering-for-machine-learning, Accessed on 15.09.2022.

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn and TensorFlow

Gonick, Larry and Woollcott Smith. 2005. The Cartoon Guide to Statistics. HarperCollins Publishers.

Huang, Z. (1997), Clustering Large Data Sets With Mixed Numeric And Categorical Values, Computer Science.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013). An Introduction to statistical learning with Applications in R. Springer.

Kuhn, M. and Johnson, K. (2013). Applied Predictive Modeling. Springer.

Staudt, Y. and Wagner, J. (2022). Assessing the Performance of Random Forests for Modeling Claim Severity in Collision Car Insurance. Risks. Vol 9. No. 53. DOI: 10.3390/risks9030053

Tibshirani, R., Walther, G. and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic.

Wong, K. Y., and Chung, F., 2019; Visualizing Time Series Data with Temporal Matching Based t-SNE.

Yin, S., Gan, G., Valdez, E and Vadiveloo, J. (2021). Applications of Clustering with Mixed Type Data in Life Insurance, arXiv.

HSLU Hochschule Luzern

Danke!

Datum

FH Zentralschweiz