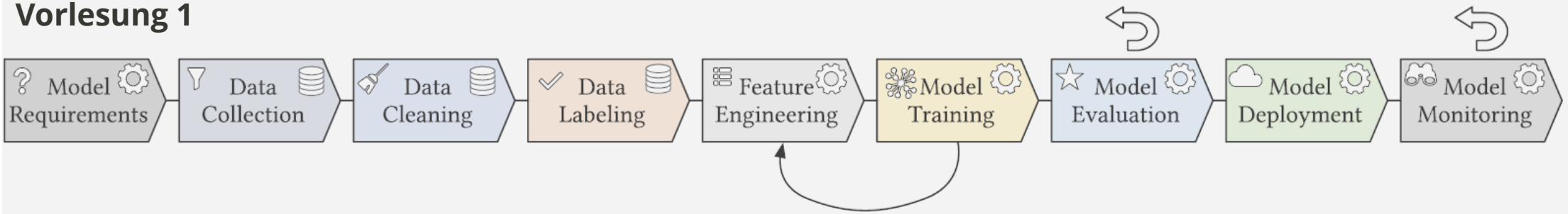
Dr. rer. nat. Valentin Khaydarov
Professur für Prozessleittechnik & Arbeitsgruppe Systemverfahrenstechnik

Clustering

Vorlesung 5, Lehrveranstaltung Experimentelle Prozessanalyse

Einordnung der Vorlesung

Vorlesung 1



Vorlesung 2

Vorlesung 3 – Regr.

Vorlesung 4 – Class.

Vorlesung 5 – Clust.

Vorlesung 6 - Zeitreihenanalyse

Vorlesung 7 – Neuronale Netze

S. Amershi *et al.*, "Software Engineering for Machine Learning: A Case Study," in *Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019*, May 2019, pp. 291–300, doi: 10.1109/ICSE-SEIP.2019.00042.

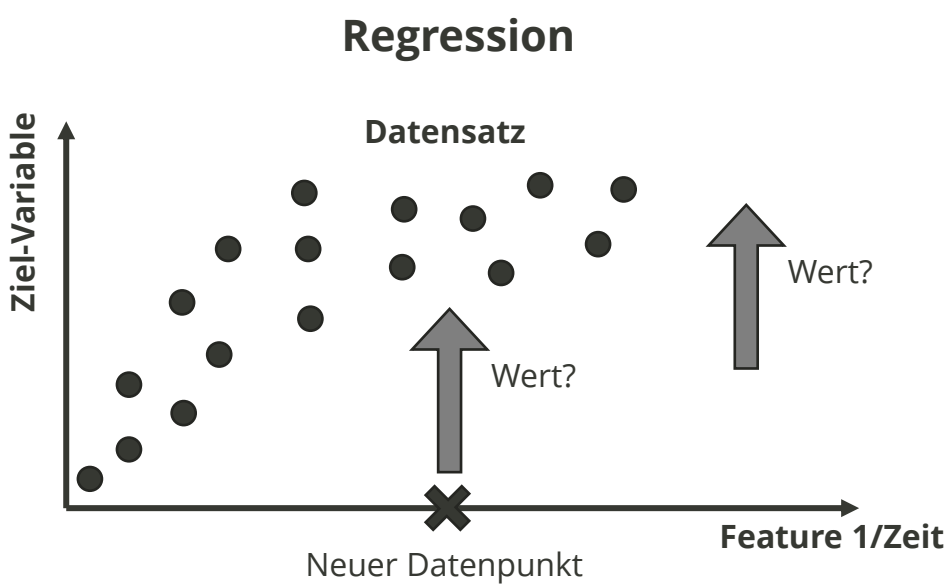
Agenda

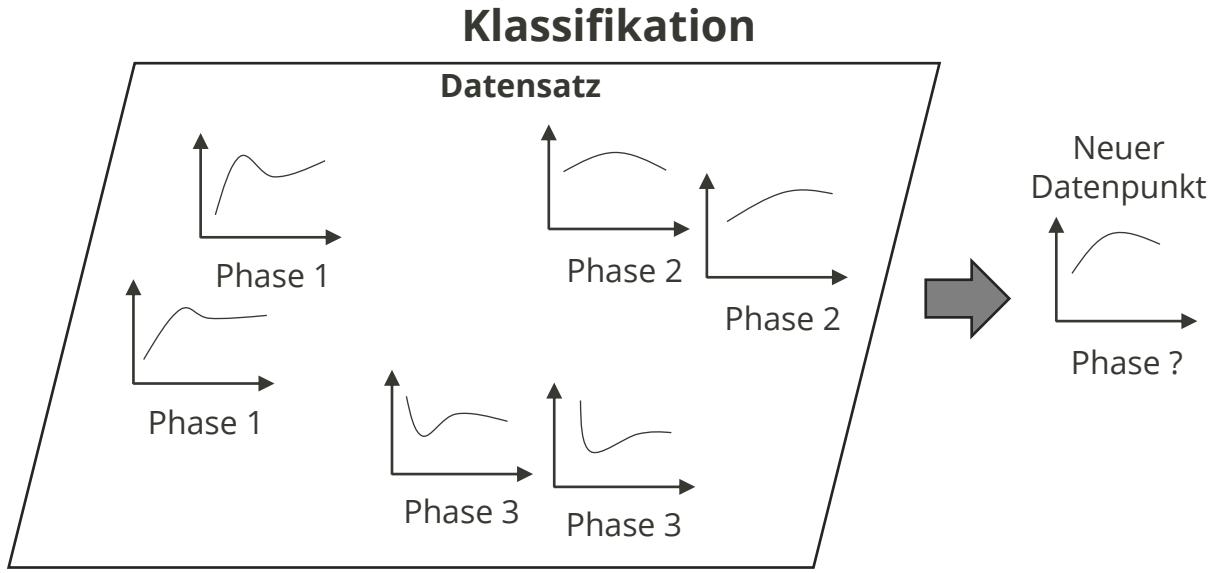
- Motivation und Wiederholung
- Problemdefinition: Clustering
- Clustering-Verfahren
  - K-Means
  - DBSCAN
  - Gaussian Mixtures
  - Übersicht von Verfahren
- Zusammenfassung

Motivation und Wiederholung

Supervised Learning

Erfahrung	Aufgabe	Leistung
Input-Variablen und Ziel-Variablen für alle Datenpunkte	Prädiktion Ziel-Variablen	Regression: MSE, MSLE, MAE Klassifikation: Cross-Entropy, Hinge





Adaptiert aus A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

Supervised Learning: Herausforderungen

Aktuell stellt die Datengewinnung kein Problem dar.

Aber Supervised-Learning-Probleme setzen **Labels** (Klassifikation) oder **Targets** (Regression) voraus.

Herausforderung: **Das Labeling von Daten mit ausreichender Qualität ist enorm aufwändig** (In 90% aller Fälle werden Labels manuell erstellt).

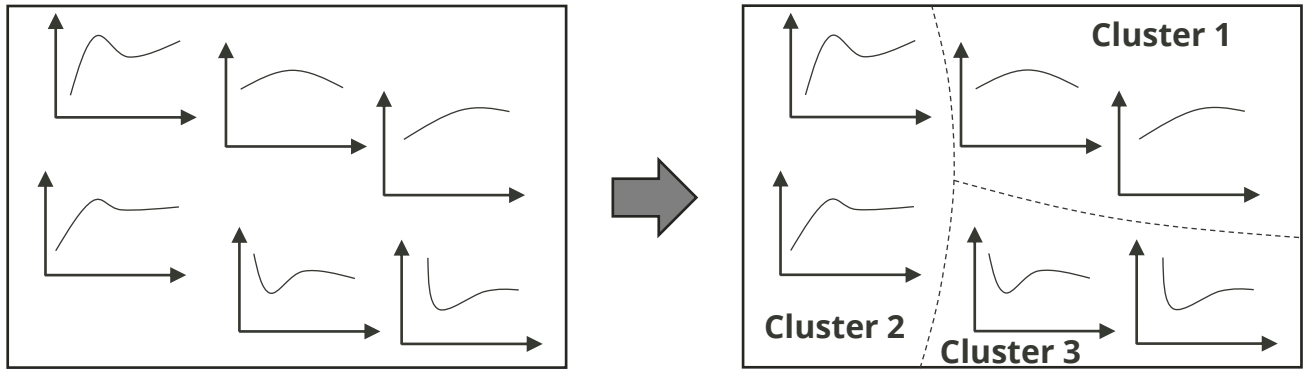
Je komplexer das Modell, desto mehr Daten sollen im Datensatz sein.

Deshalb: Active Learning, Transfer Learning, Semi-Supervised Learning, Self-supervised und Unsupervised Learning

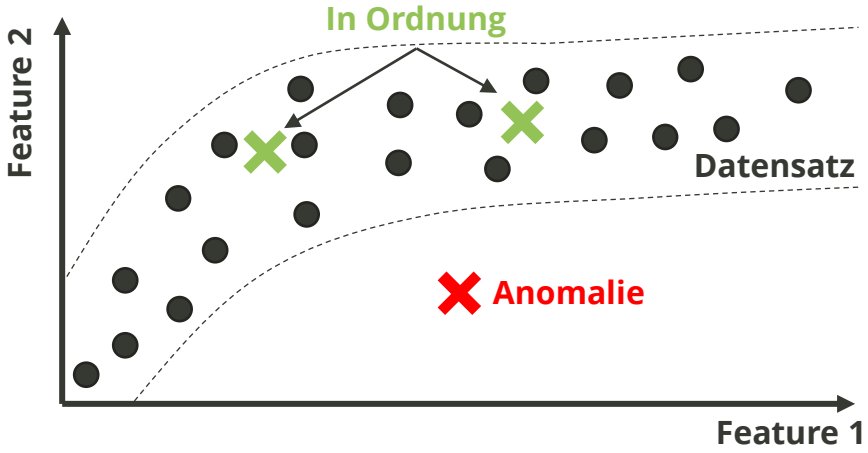
Unsupervised Learning: Typische Anwendungen

- Datensegmentierung und weiter Clusters werden einzeln analysiert
- Reduzierung der Dimensionalität: Affinität zu Clustern als neue Features
- Anomalierkennung
- Semi-Supervised-Learning: Propagation von Clustern als Labels

Clustering



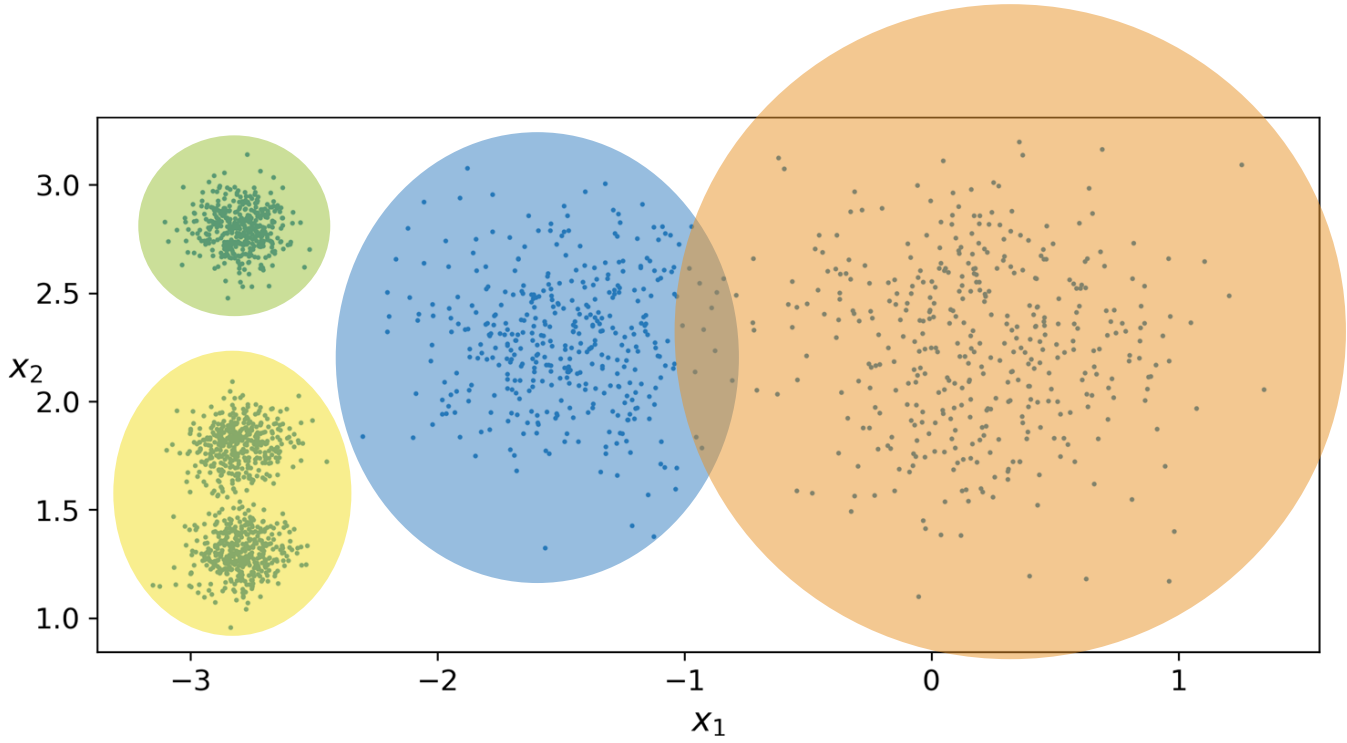
Anomalieerkennung



Adaptiert aus A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

Problemdefinition

Intuition



Ziel:

Minimierung des Abstandes zwischen Instanzen innerhalb der Clusters soll und

Maximierung des Abstandes zwischen den Zentren der Clusters

A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

Problemdefinition

Clustering sucht Aufteilung von Instanzen in Gruppen, welche ähnlichen Mustern gehören.

Gegeben:

- eine Menge von Beobachtungen $\underline{x}$
- ein Parameter K, eine Anzahl zu findender Clusters $C_1, C_2 \dots C_K$
- eine Abstandsfunktion
- eine Qualitätsfunktion q

Finde Clusters $C_1, C_2 \dots C_K$, so dass die Qualitätsfunktion optimiert wird.

Abstandsfunktion

Bewertung der Ähnlichkeit von Objekten $\underline{x}_1$ und $\underline{x}_2$ (einzelne Beobachtungen und/oder Clusters):

Ähnlichkeitsmaße für kategoriale Features:

$sim = 0$ - die Objekte haben eine maximale Unähnlichkeit

Distanzmaße für metrische Features:

$dist = 0$ - die Objekte haben einen Abstand von Null (maximale Ähnlichkeit)

Im Allgemein gilt $dist(\underline{x}_1, \underline{x}_2) = 1 - sim(\underline{x}_1, \underline{x}_2)$

Distanzmaße für metrische Daten

L_r - Metrik (Minkowski-Metrik):

$$dist_{ij} = dist(\underline{x}_i, \underline{x}_j) = \left[\sum_{\mu=1}^{m} |x_{i\mu} - x_{j\mu}|^r\right]^{1/r}$$

Spezialfälle:

L_1 - Metrik (Manhattan-Metrik)

$$dist_{ij} = \sum_{\mu=1}^{m} |x_{i\mu} - x_{j\mu}|$$

L_2 - Metrik (Euklidische Metrik)

$$dist_{ij} = \sqrt{\sum_{\mu=1}^{m} (x_{i\mu} - x_{j\mu})^2}$$

Mahalanobis - Metrik

$$dist_{ij} = \sqrt{(\underline{x}_i - \underline{x}_j)^T S^{-1} (\underline{x}_i - \underline{x}_j)}$$

$$S = \frac{1}{n-1} \sum_{v=1}^{n} (\underline{x}_v - \overline{\underline{x}})(\underline{x}_v - \overline{\underline{x}})^T$$

$$\overline{\underline{x}} = \frac{1}{n} \sum_{v=1}^{n} \underline{x}_v$$

Qualitätsfunktion

Abstand zwischen Beobachtungen innerhalb eines Clusters (Within):

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)=k} dist(\underline{x}_i, \underline{x}_j)$$

Abstand zwischen Clusters (Between):

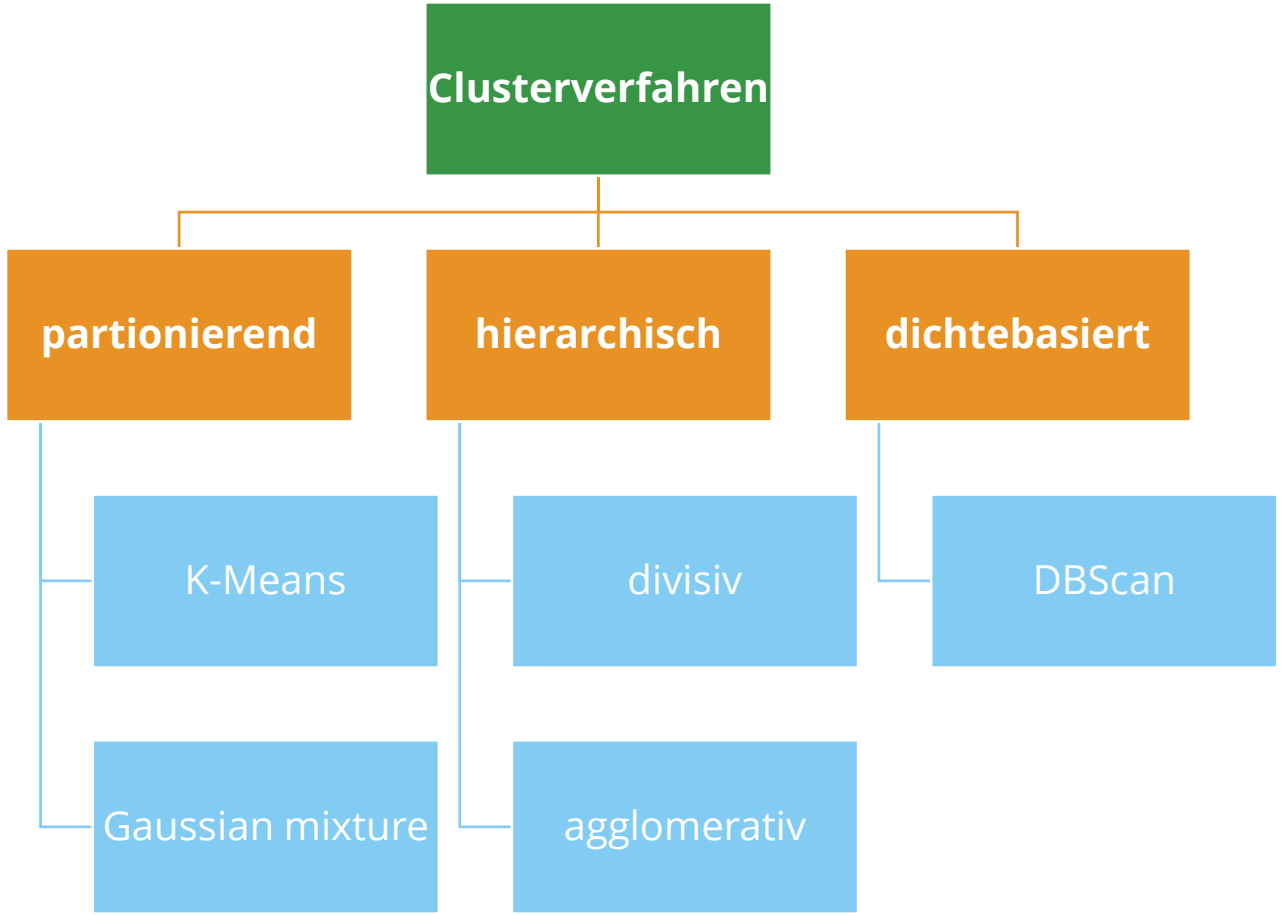
$$B(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)\neq k} dist(\underline{x}_i, \underline{x}_j)$$

Minimierung von $W(C)$ entspricht immer der Maximierung von $B(C)$, denn

$$W(C) + B(C) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} dist(\underline{x}_i, \underline{x}_j) = const$$

Clustering-Verfahren

Taxonomie


Clusterverfahren
partionierend
hierarchisch
dichtebasiert
K-Means
Gaussian mixture
divisiv
agglomerativ
DBScan

K-Means

Algorithmus

Initialisierung: K Clusterzentren $C_1, C_2 \ldots C_K$ werden aus Beobachtungen zufällig ausgewählt

Iterative Schleife:

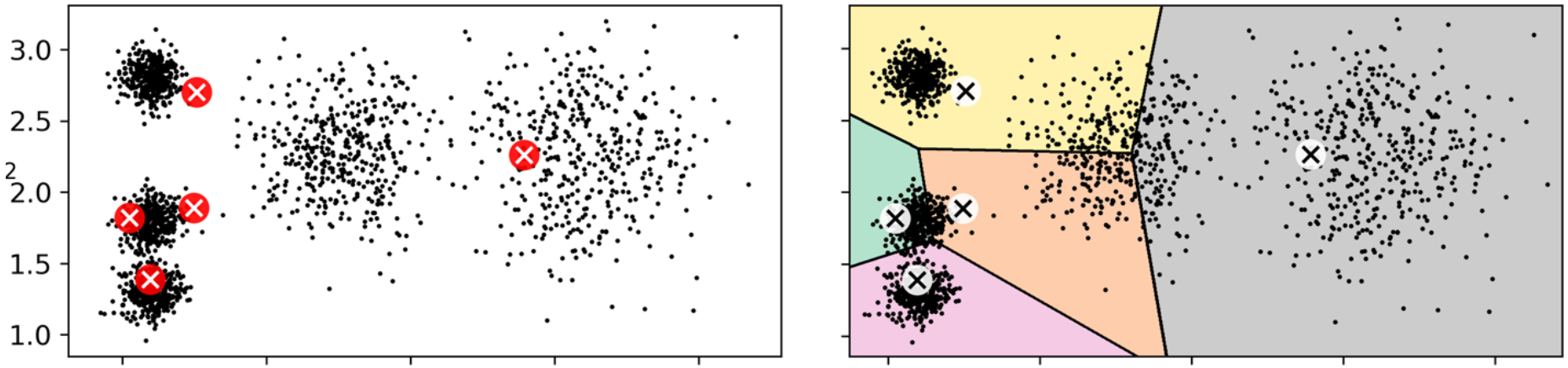
- Zuordnung jeder Beobachtungen dem nächstgelegten Clusterzentrum C_i
- Berechnung der Clusterzentren als Mittelwert aller dem Cluster zugehörigen Punkte

solange, bis:

- Anzahl von maximalen Iterationen erreicht oder
- keine weitere Verbesserung des Zielfunktion (z.B. keine Zuordnungsänderung)

Initialisierung

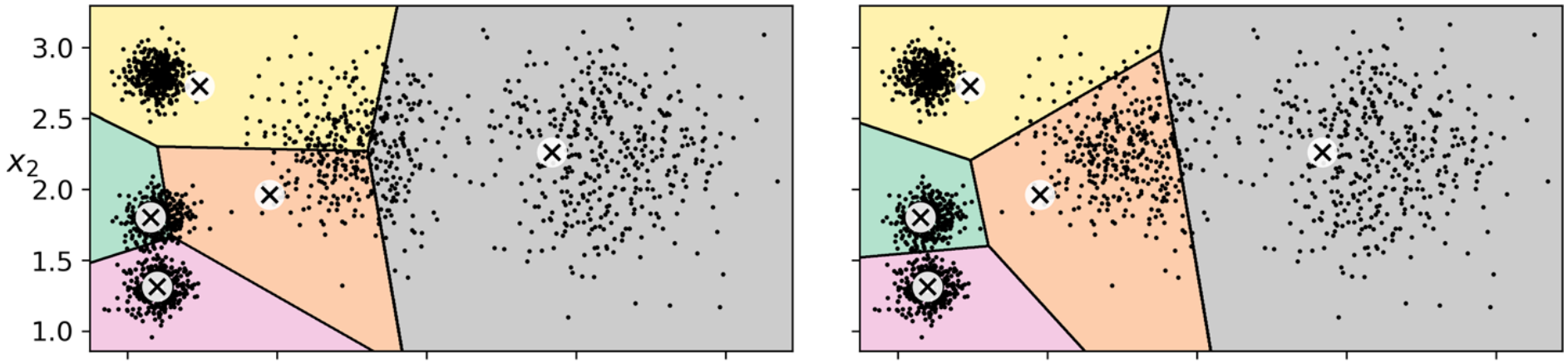
Zufällige Initialisierung von 5 Clusterzentren und Zuordnung:



A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

Iteration 1

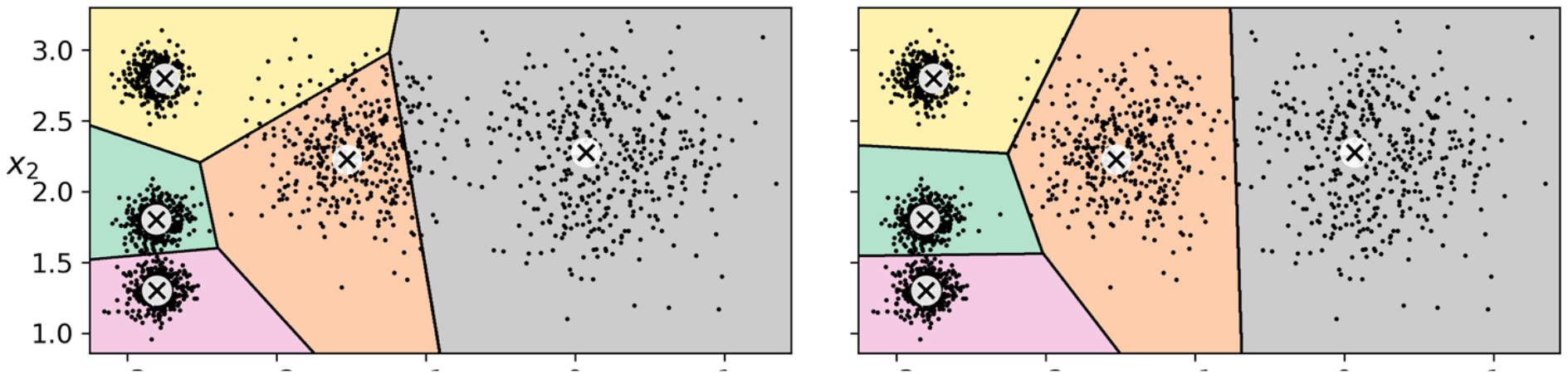
Neue Berechnung von Zentren und Zuordnung von Beobachtungen



A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

Iteration n

Neue Berechnung von Zentren und Zuordnung von Beobachtungen

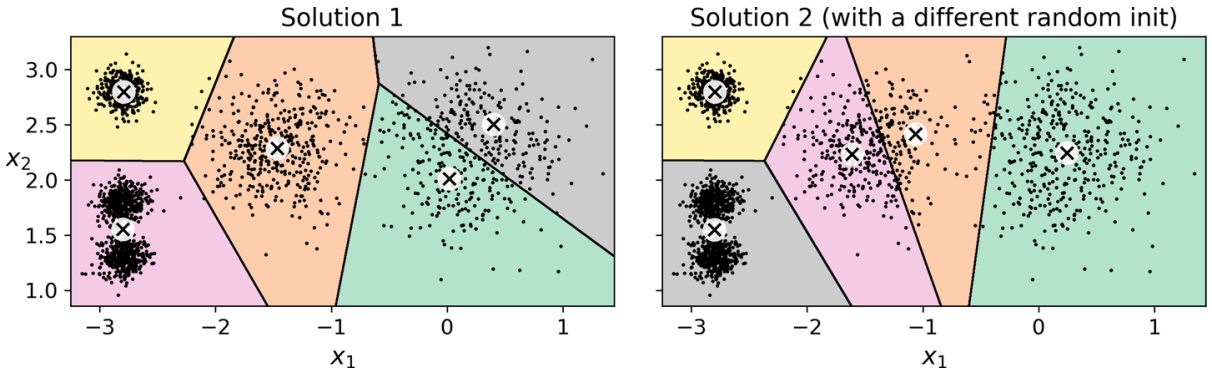


A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

Eigenschaften und Erweiterungen

- Liefert nur ein **lokales** Optimum, wenn zufällig initialisiert
- Zentren als Hyperparameter möglich
- Erzeugt K disjunkte Teilmengen, wobei K vorgegeben soll
- Teilmengen sind disjunkt

Tendenz zu homogenen, kompakten, gleichmäßigen Clusters



A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

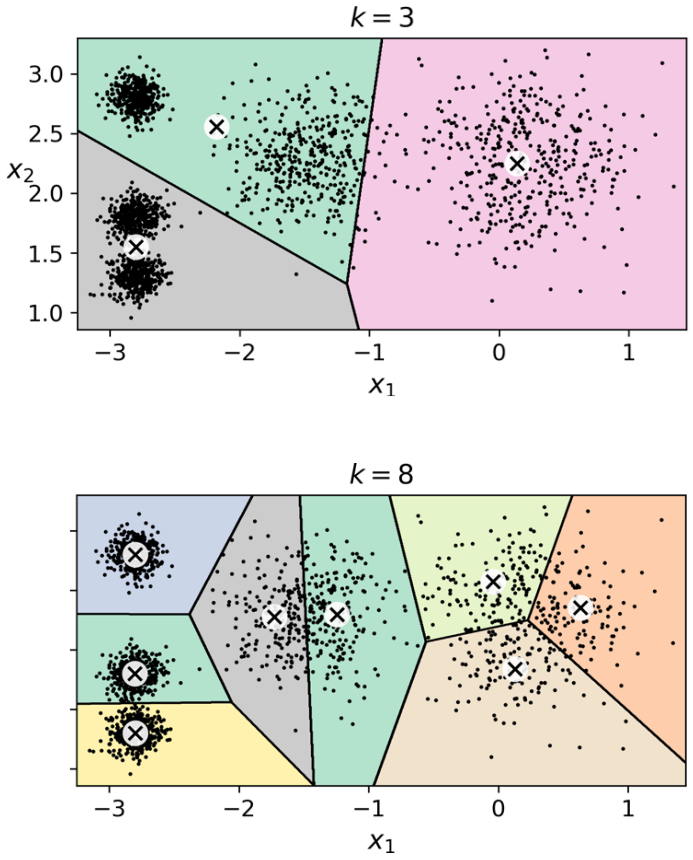
Erweiterungen:

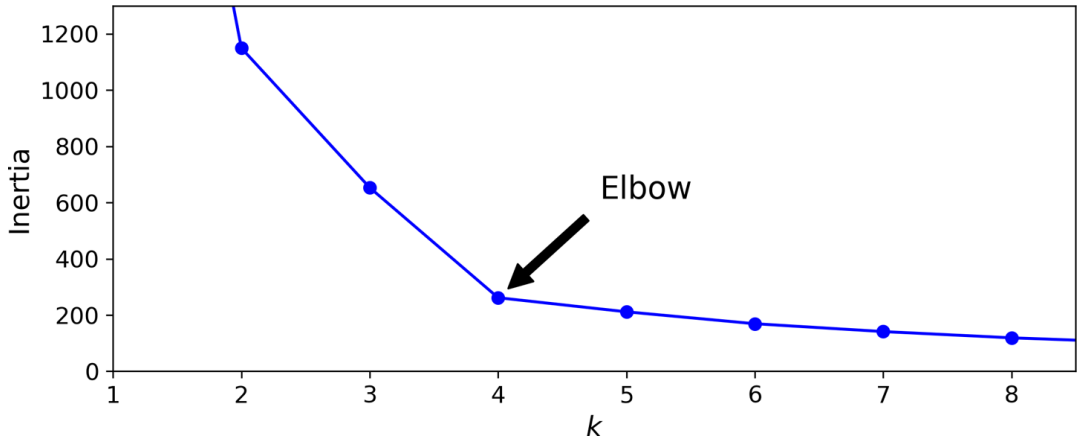
- k-means ++: „smarte" sequenzielle Initialisierung der Clusterzentren
- Minibatch-K-Means

Optimale Anzahl von Clusters

Elbow-Ansatz

Inertia-Wert: $I = \sum_{i=1}^{N}(x_i - C_K)^2$





A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

DBSCAN

Density-based Spatial Clustering of Applications with Noise

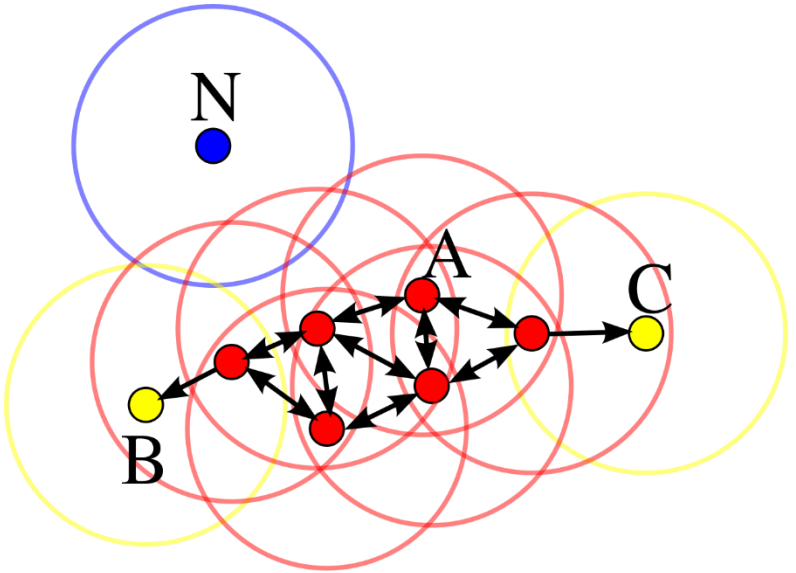
Algorithmus

Annahme: ein Cluster ist eine kontinuerlich wachsende Region mit **einer hohen Dichte** von Beobachtungen

In der ε-Umgebung jeder Beobachtung wird die Dichte berechnet:

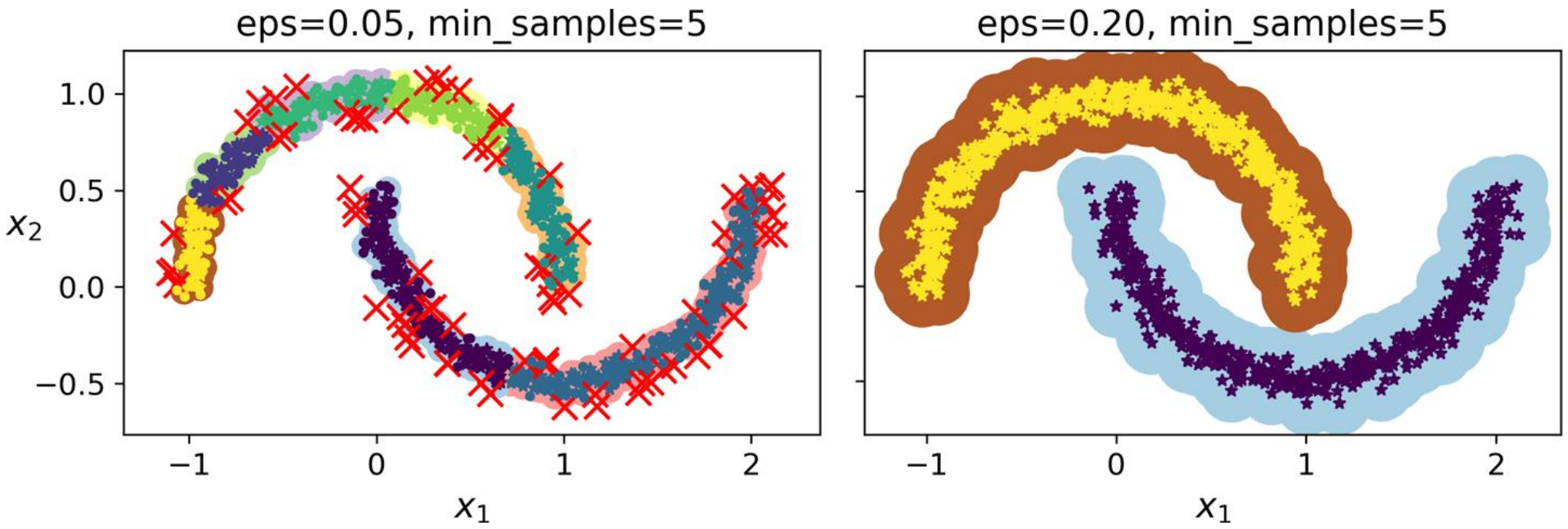
- Kern-Punkt, wenn mehr als $min\_samples$
- Dichte-erreichbare Beobachtung, wenn weniger
- Sonst Rauschpunkt

Einem Cluster gehören die durch dieselben Kernobjekte miteinander verbundenen Beobachtungen.



https://de.wikipedia.org/wiki/Datei:DBSCAN-Illustration.svg

Hyperparameter



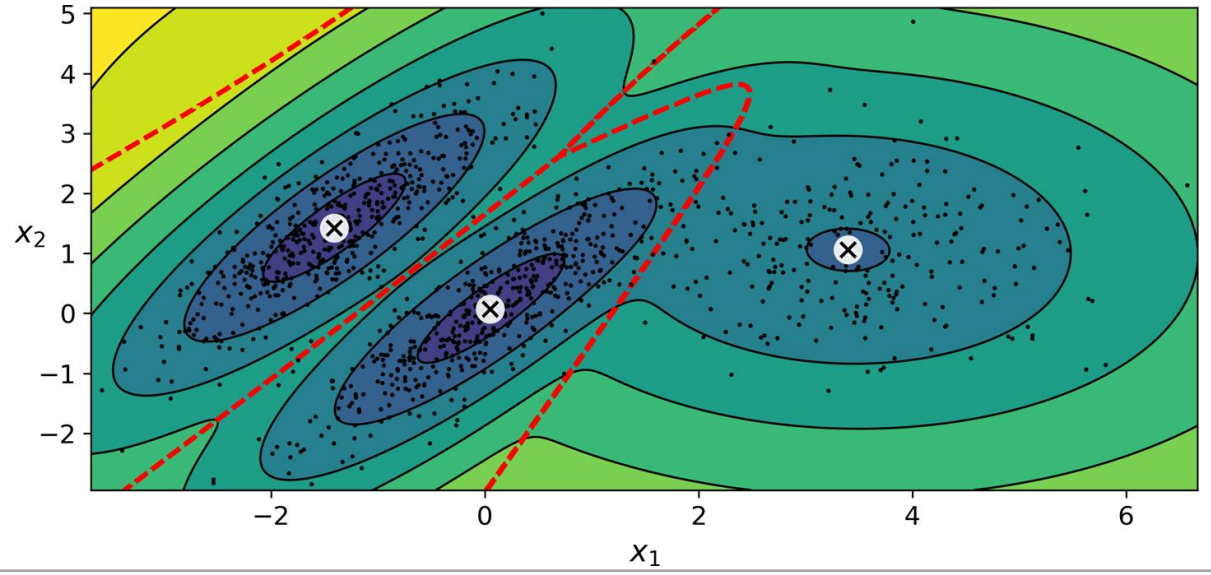
A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

Gaussian Mixtures

Algorithmus

Ein probabilistisches Modell, wo die Normalverteilung zugrunde liegt. Die Verteilung beschreibt die Wahrscheinlichkeit, dass die Beobachtung diesem Cluster gehört.

Ähnlich dem K-Means-Verfahren, aber gefunden werden nicht nur Mittelwerte aber auch Varianzen und Gewichte von einzelnen Teilverteilungen.

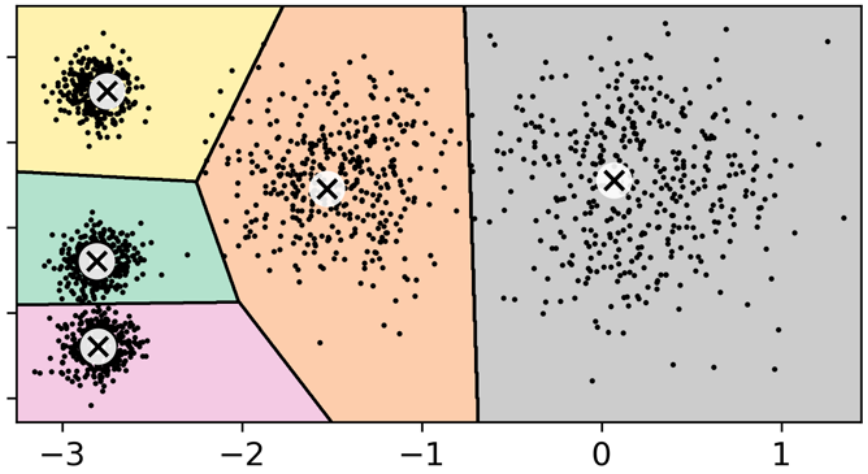


A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

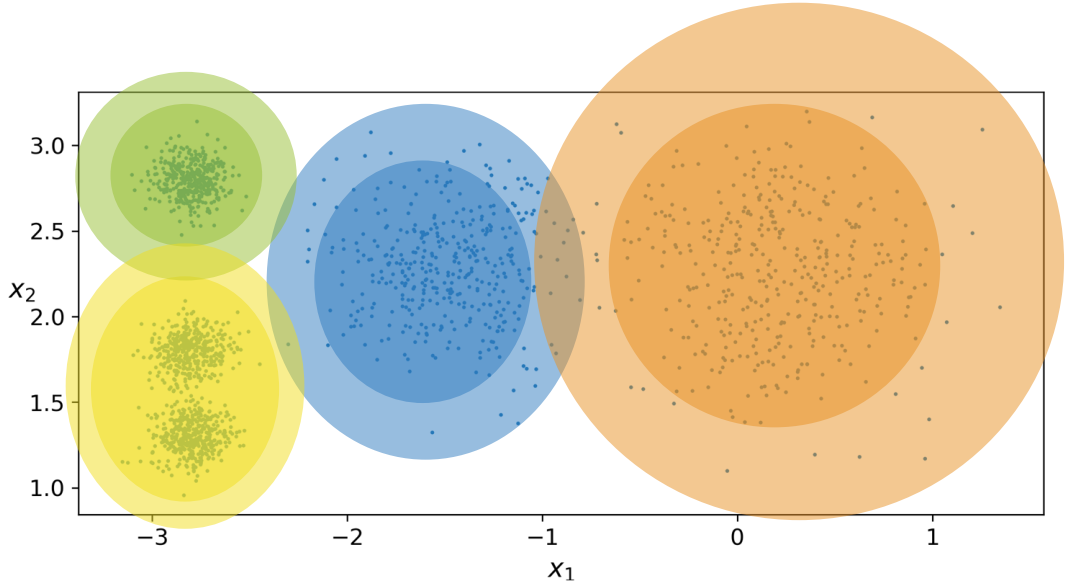
Hard vs Soft clustering

Begriffserklärung

Hard-Clustering:
jede Beobachtung gehört nur einem Cluster

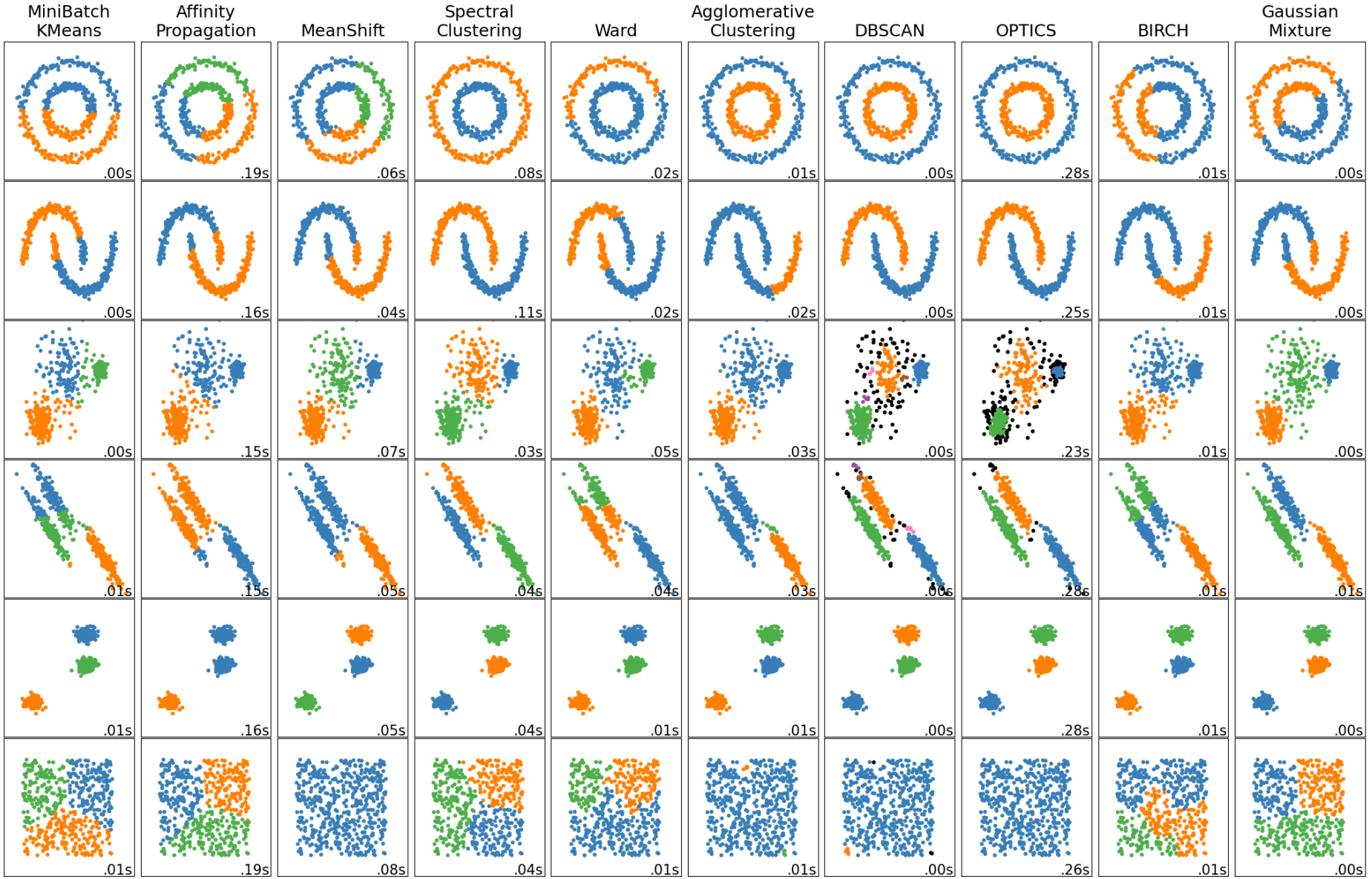


Soft-Clustering:
Beobachtung darf mehreren Clusters gehören



A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

Übersicht



https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

PCA
Principal Component Analysis

Hauptkomponenten- / Faktorenanalyse (PCA / FA)

- Prinzip der linearen Merkmalstransformation
- Merkmalszahl $m \gg 2$

Hauptkomponentenanalyse (principal component analysis - PCA)

- Rückschlüsse auf die Struktur der Originaldaten aus der Lage der Datensätze im transformierten Merkmalsraum ziehen
- die wahre Dimensionalität des Problems zu beurteilen.

Faktorenanalyse

- das Auffinden von Merkmalskomplexen

Hauptkomponentenanalyse

Lineare Transformation der m Merkmalen $x_1, \ldots, x_m$ zu m neuen Merkmale $\tilde{x}_1, \ldots, \tilde{x}_m$

Eigenschaften der neue Merkmale:

- *normiert* (standardisiert)
- untereinander *unkorreliert*

Bestimmung der neuen Merkmale

- $\tilde{x}_1$ (erste Hauptkomponente) ein Maximum der Gesamtvarianz (Gesamtstreuung) aller Merkmale $x_1, \ldots, x_m$ erfasst
- $\tilde{x}_2$ (zweite Hauptkomponente) ein Maximum der nach Extraktion der ersten Komponente verbleibenden Restvarianz, usw.

Dabei gilt:

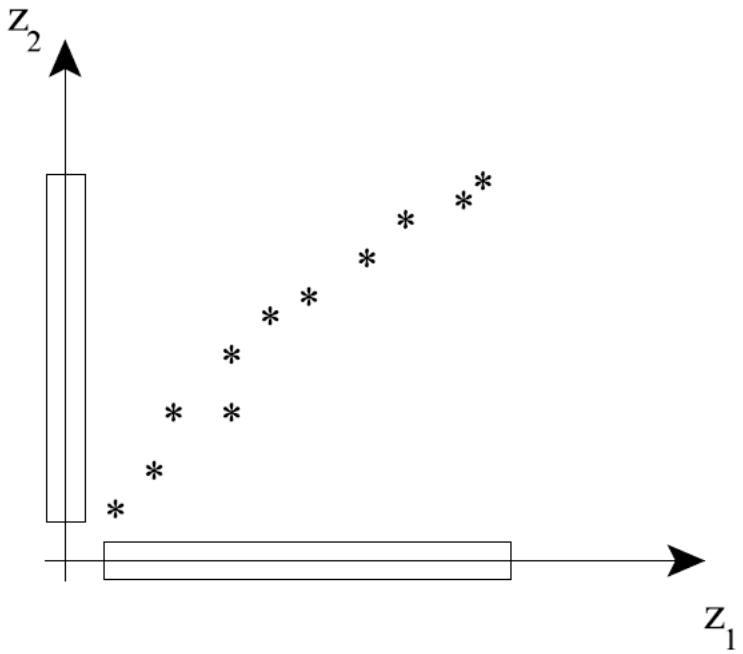
- Gesamtvarianz aller $\tilde{x}_i$ ist gleich der Gesamtvarianz aller x_i
- die Einzelvarianzen sind der Größe nach sortiert. Damit ist die Reduktion der Merkmalsanzahl auf die ersten m' $(m' < \mathrm{m})$ neuen Merkmale $\tilde{x}_1, \ldots, \tilde{x}_{m'}$ ohne größeren Informationsverlust möglich.

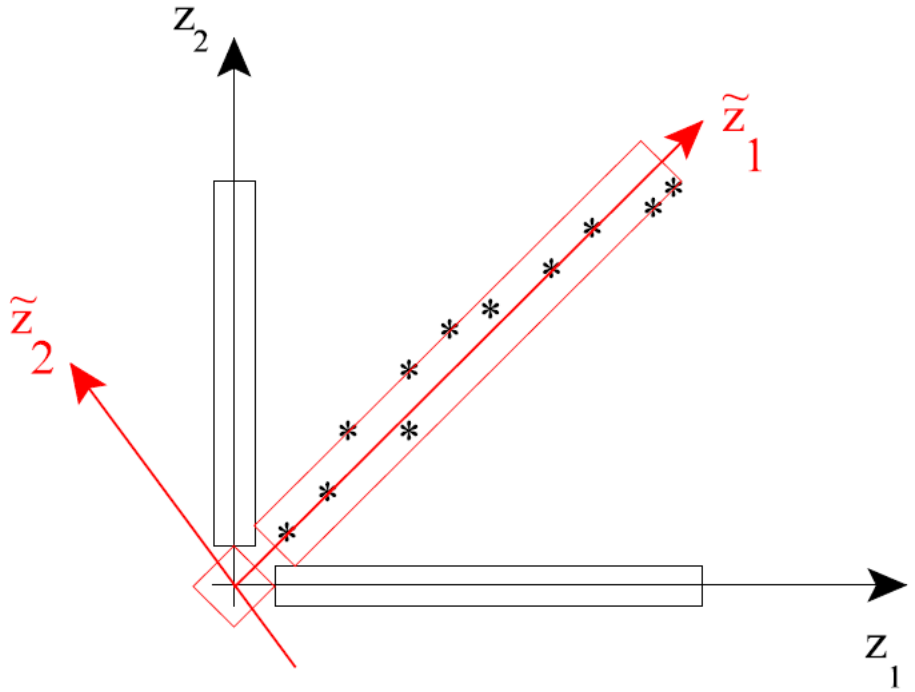
Hauptkomponentendarstellung

- graphische Darstellung des Datenmaterials für $m' = 2$ (transformierten Merkmalen $\tilde{x}_1, \tilde{x}_2$)

Hauptkomponentenanalyse

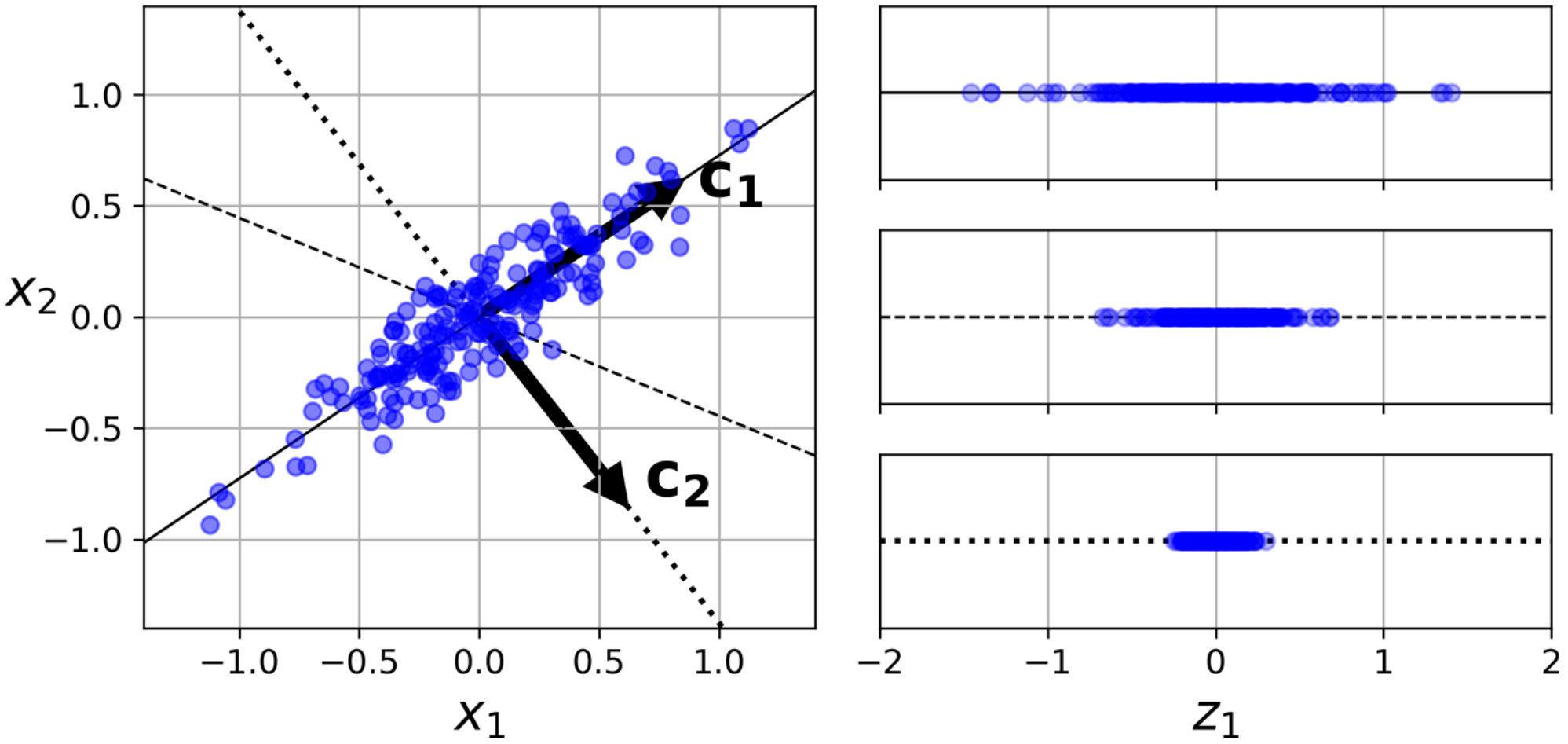
Prinzip der linearen Transformation





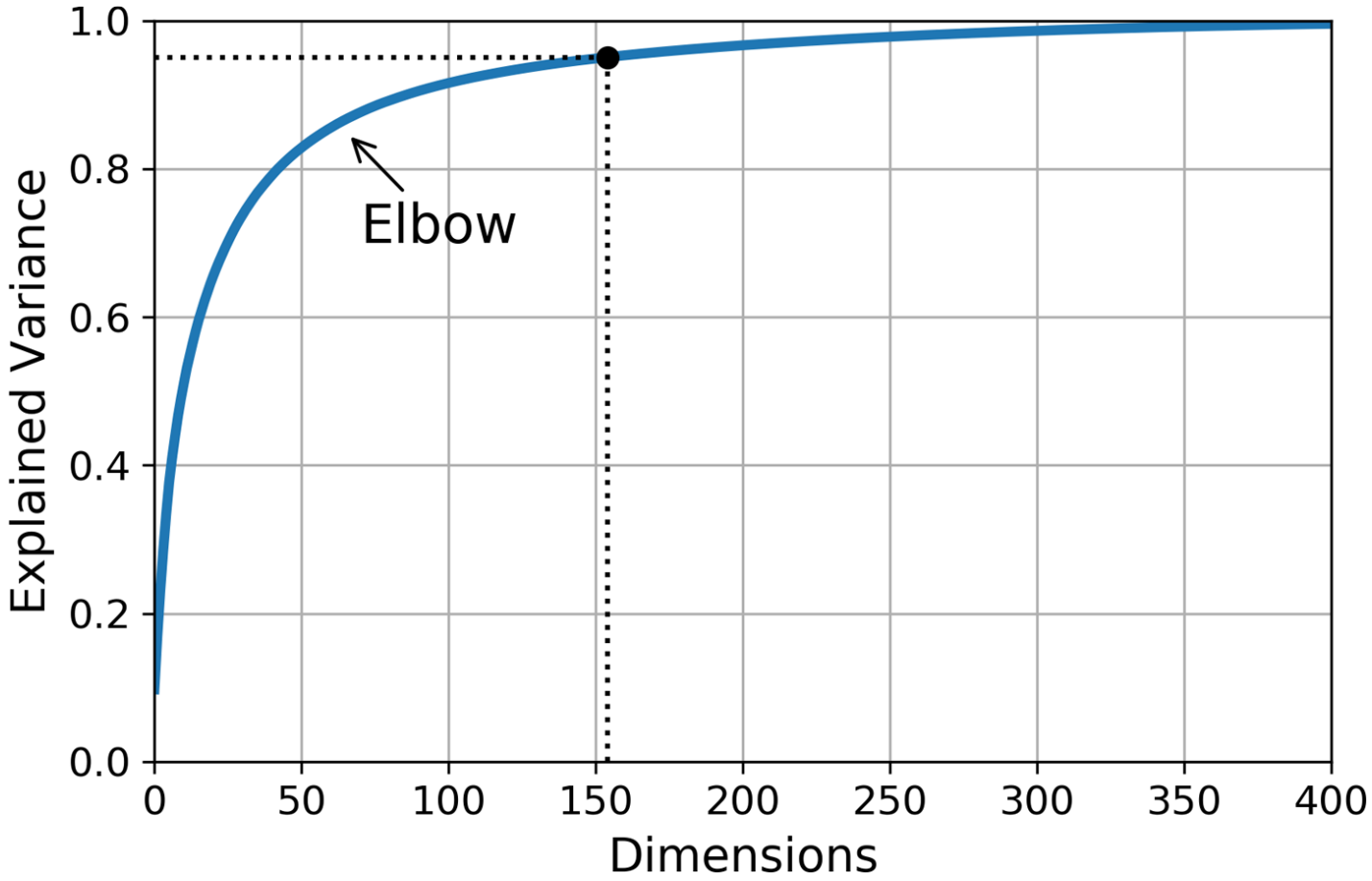
m Orginalmerkmale → m neue Merkmale (standardisiert, unkorreliert, sortiert nach Wichtigkeit)

Maximierung der Varianz



A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

Ermittlung der Anzahl von Hauptkomponenten



A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

Zusammenfassung

Dr. rer. nat. Valentin Khaydarov
Email: valentin.khaydarov@tu-dresden.de
Telefon: 0351 463 33387

Vielen Dank für Ihre Aufmerksamkeit!