Wirtschaftsinformatik – Informationsmanagement und Data Science

datasciencelab

HOCHSCHULE HEILBRONN

› Data Science Grundlagen
Clusteranalyse

Prof. Dr. Carsten Lanquillon | Fakultät Wirtschaft und Verkehr | Wirtschaftsinformatik

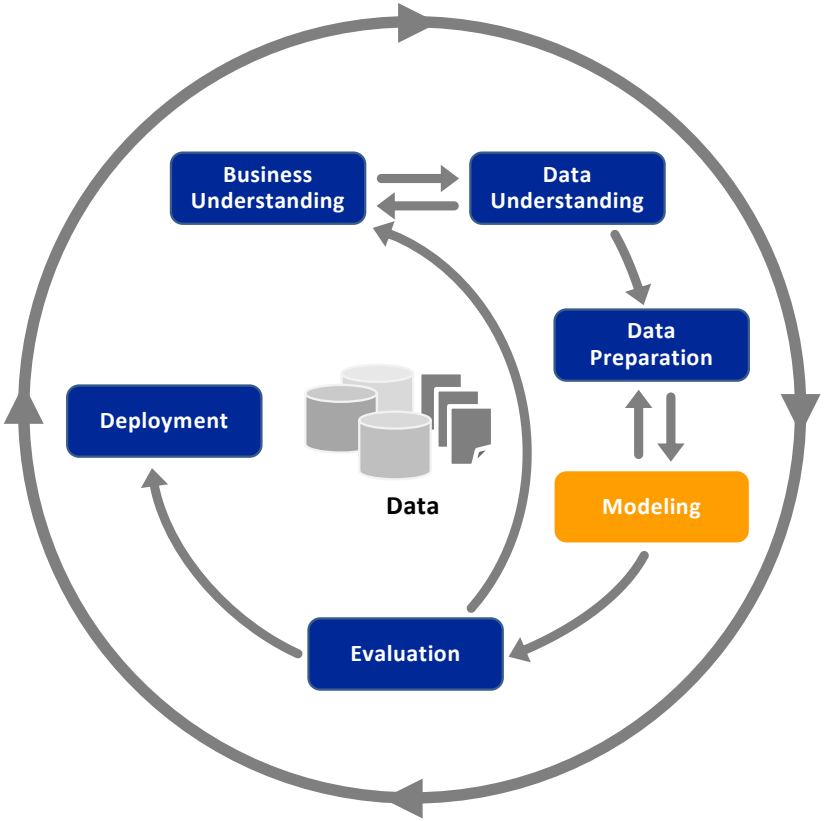
CRISP-DM Phase 4: Modeling

Ziel: Modellerstellung durch maschinelle Lernverfahren

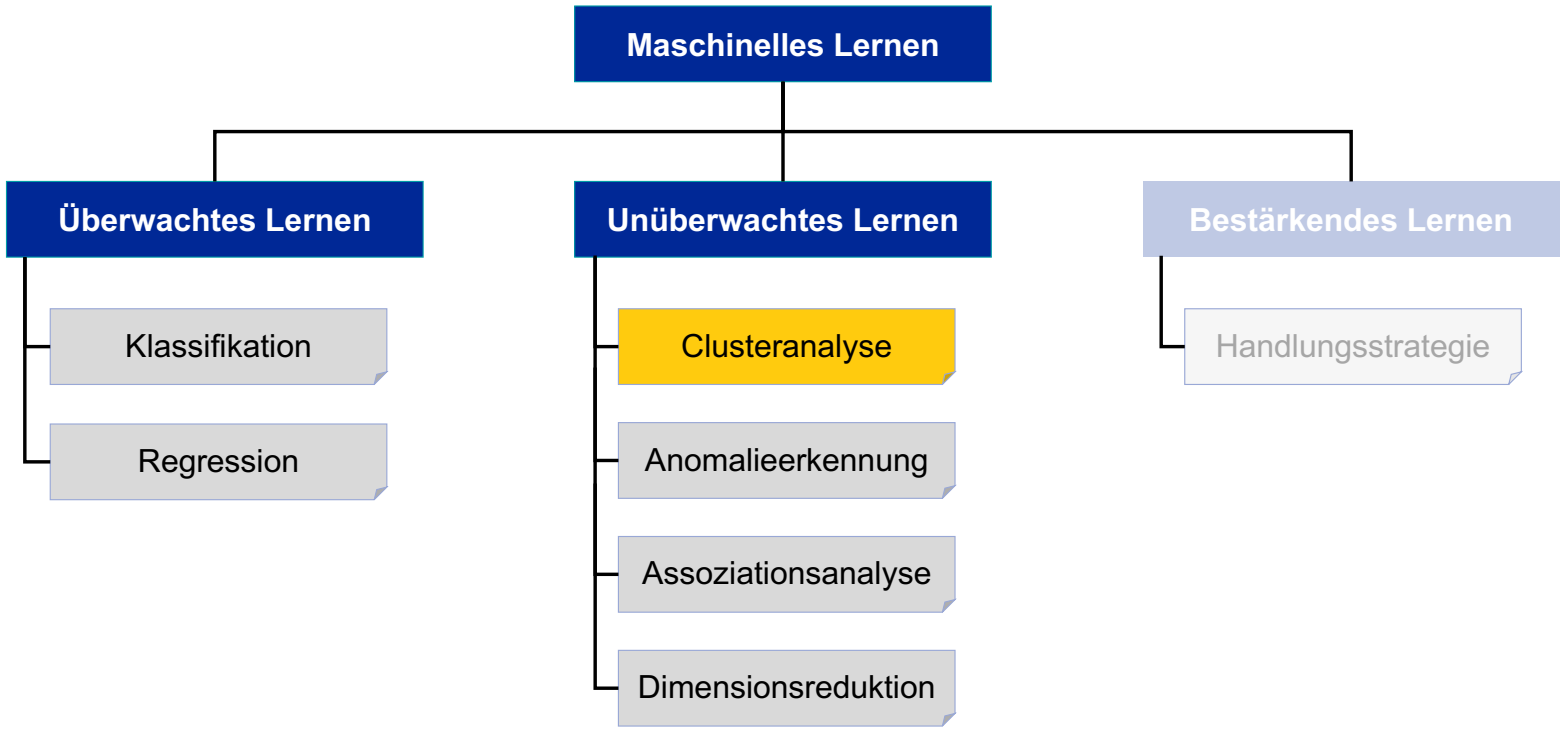
Aufgaben

- Auswahl geeigneter Lernverfahren (Model Selection)
- Nicht auf einen Lösungsansatz (Lernverfahren) festlegen („*no free lunch*“)
- Festlegen von Modellparametern
- Aufbau verschiedener Modelle (Lernen aus Daten)
- Auswahl eines Modells für die Aufgabenstellung

Fokus: Clusteranalyse



Lernformen und Data-Mining-Aufgaben



Agenda

- The Clustering Task
- Clustering Approaches
  - Hierarchical Approaches
  - Prototype-based Approaches
  - Density-based Approaches
- Cluster Evaluation

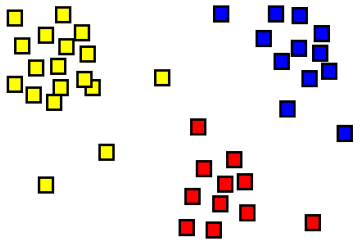
Klassifikation vs. Clustering

Klassifikation

- Klassen und Zuordnung von Objekten zu Klassen sind gegeben
- Aufgabe: Klassen für neue Objekte vorhersagen

Clustering

- Keine Klassen, keine Zielgröße gegeben
- Aufgabe: Klassenstruktur in Daten entdecken

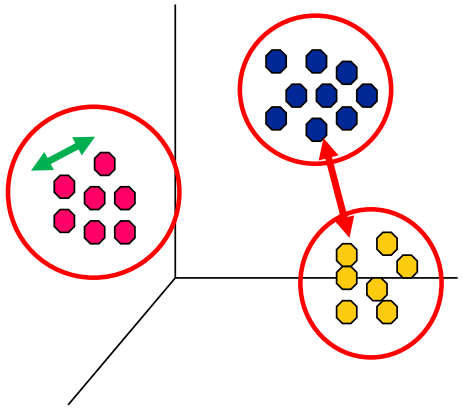


Clusteranalyse

Erzeuge Gruppen (Cluster, Klassen, Segmente)
von Objekten mit folgenden Eigenschaften:

- **“Within-Cluster Homogeneity”**
  Objekte in einer Gruppe sind ähnlich zueinander

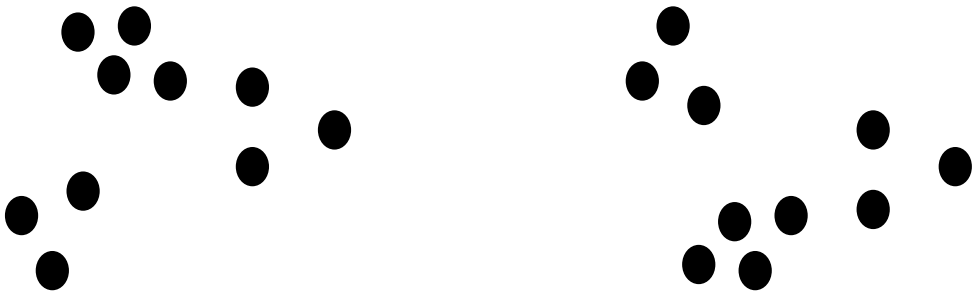
- **“Between-Cluster Heterogeneity”**
  Objects in verschiedenen Gruppen sind unähnlich



Wan sind Objekte ähnlich?

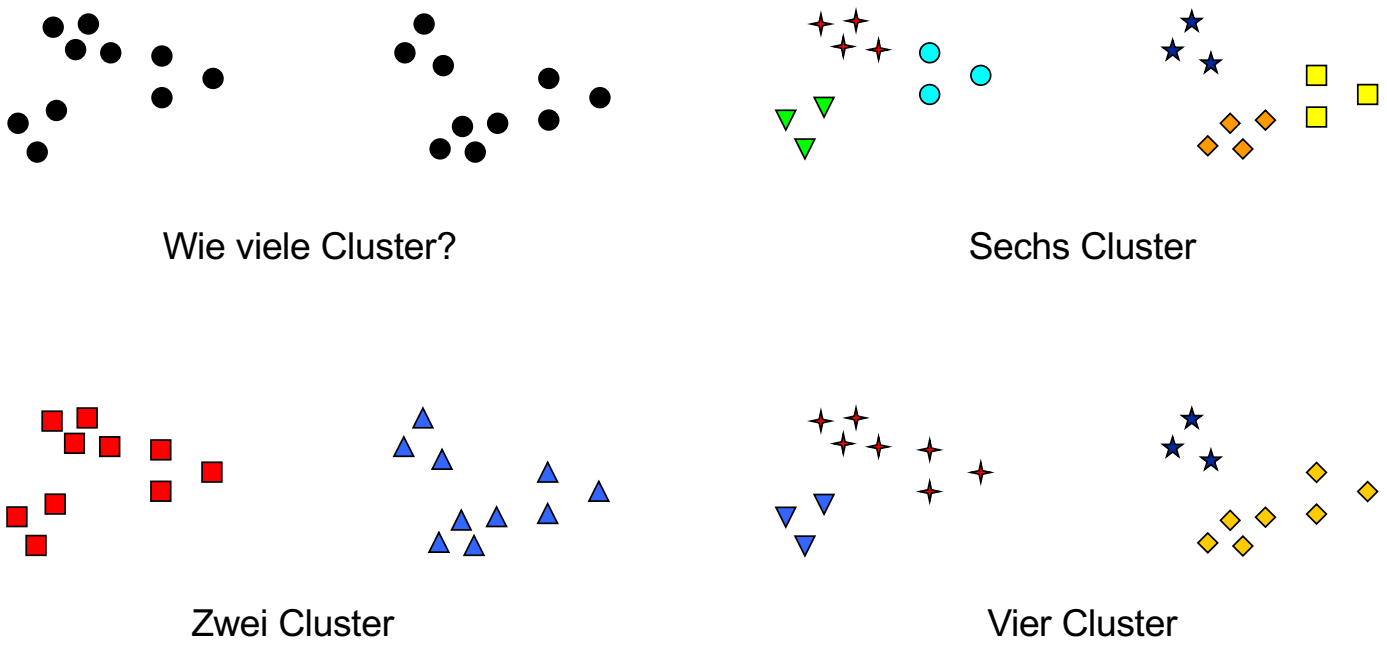
→ **Benötigt Ähnlichkeits- oder Distanzmaße!**

Clustering: Mehrdeutigkeit (1)



Wie viele Cluster?

Clustering: Mehrdeutigkeit (2)



Kategorien von Clustering-Verfahren

Hierarchisches Clustering

> Erzeugt baumartige Clusterstruktur (Dendrogram) durch rekursives Aufteilen (**divisive Methoden**) oder Zusammenfügen (**agglomerative Methoden**) bestehender Cluster

Nicht-hierarchisches (partitionierendes) Clustering

> Prototyp oder model-basiertes Clustering
  - Erzeugt alle Cluster (Partitionen) gleichzeitig
  - Meist sphärische Cluster werden durch Prototypen oder Modell kompakt repräsentiert

> Dichte-basiertes Clustering
  - Identifiziert dicht mit Objekten (Datenpunkten) besiedelte Regionen beliebiger Form

Eigenschaften von Clustering-Verfahren

Es gibt zahlreiche Clustering-Verfahren mit unterschiedlichen Stärken und Schwächen

Typische Unterscheidungsmerkmale

- Hierarchisch oder nicht-hierarchisch (Baumstruktur oder flache Partitionierung)
- Regelmäßige (z.B. sphärisch) oder beliebige Form der Cluster
- Variable oder feste Clusteranzahl
- Inkrementell oder nicht-inkrementell
- Geeignet für nominale, ordinale, metrische oder gemischt-skalierte Daten
- Geeignet für hochdimensionale Daten

Hierarchisches Clustering

Agglomeratives Clustering

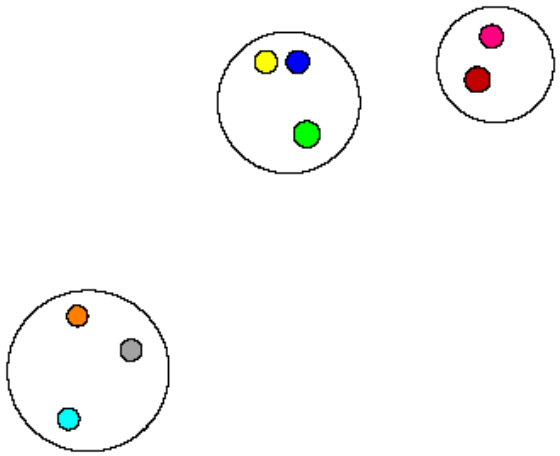
- Zu Beginn ist jedes Objekt sein eigenes kleines Cluster (Singleton-Cluster)
- Bis alle Objekte in einem großen Cluster vereint sind
  - Verbinde die beiden Cluster mit dem kleinsten Abstand zu einem neuen Cluster

Divisives Clustering

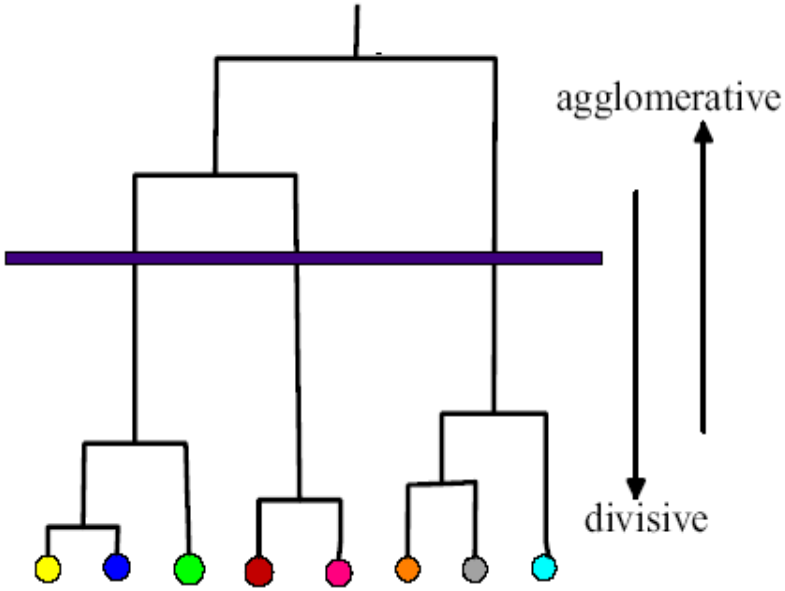
- Zu Beginn bilden alle Objekte ein großes Cluster
- Bis jedes Objekt sein eigenes Cluster ist
  - Teile ein Cluster mit Objekten, die sich sehr unähnlich sind, in verschiedene Cluster

Hierarchisches Clustering: Beispiel

Cluster

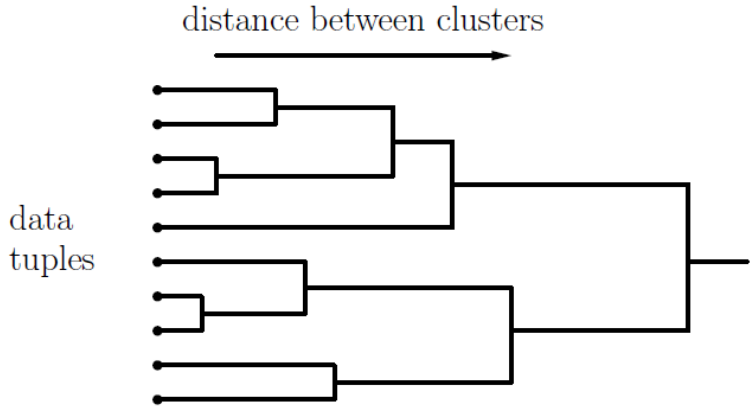


Dendrogramm



Dendrogramm

- Hierarchische Clusterverfahren erzeugen einen Binärbaum
- Zeichne die Objekte (Singleton-Clusters) unten oder links
- Verbinde die zwei Cluster, die in einer Clustering-Iteration zu zwei Clustern zusammengeschlossen werden
- Die Länge der Verbindungslinie ist proportional zum Abstand zwischen den beiden Clustern



Agglomerativer Clustering-Algorithmus

Idee / Voraussetzung

- Proximitätsmatrix speichert alle Ähnlichkeiten zwischen je zwei Clustern einer aktuellen Lösung
- Erweiterung der Ähnlichkeitsberechnung auf Cluster als Mengen von Objekten

Algorithmus

1. Let each data point be a cluster
2. Compute the proximity matrix
3. **Repeat**
4. Merge the two closest clusters
5. Update the proximity matrix
6. **Until** only a single cluster remains

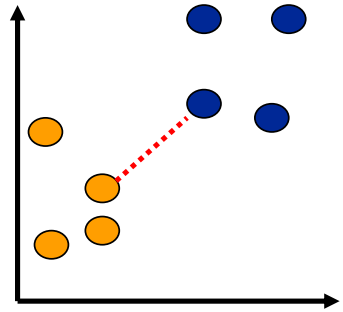
Die Proximitätsmatrix

- Proximitätsmatrix speichert alle Ähnlichkeiten zwischen je zwei Clustern einer aktuellen Lösung
- Die beiden Cluster mit dem kleinsten Wert in der Proximitätsmatrix sind am ähnlichsten und werden zu einem neuen Cluster zusammengefasst
- Anpassung der Proximitätsmatrix nach Zusammenfassen zweier Cluster zu einem neuen
  - Entferne die Zeile und die Spalte von einem der betroffenen Cluster
  - Aktualisiere alle Ähnlichkeitswerte, die sich auf den anderen der beiden Cluster beziehen
  - Der neue Ähnlichkeitswert lässt sich auf Basis der aktuellen Matrix berechnen
    - Single Linkage
    - Complete Linkage
    - Average Linkage

Single Linkage (Nearest Neighbor)

Ähnlichkeit zwischen zwei Clustern ist das Minimum aller paarweisen Ähnlichkeiten zwischen je einem Objekt aus den beiden Clustern:

$$d(C_1, C_2) = \min_{\mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2} d(\mathbf{x}_1, \mathbf{x}_2)$$



Eigenschaften

> Neigt zur Erzeugung länglicher, kettenartigen Cluster

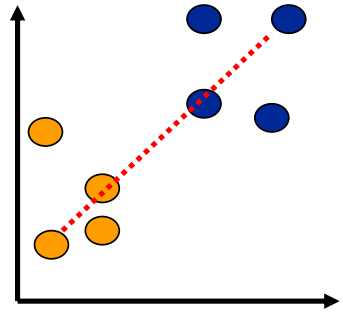
> Erzeugt oft wenige große und viele kleine Cluster

→ Ausreißererkennung

Complete Linkage (Furthest Neighbor)

Ähnlichkeit zwischen zwei Clustern ist das Maximum aller paarweisen Ähnlichkeiten zwischen je einem Objekt aus den beiden Clustern:

$$d(C_1, C_2) = \max_{\mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2} d(\mathbf{x}_1, \mathbf{x}_2)$$



Eigenschaften

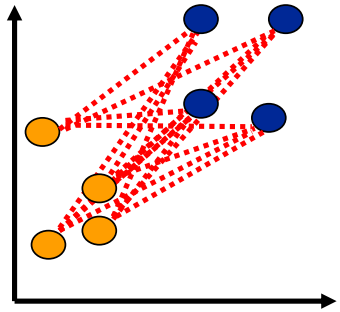
> Neigt zur Erzeugung kompakter, sphärischer Cluster

> Empfindlich gegenüber Ausreißern

Average Linkage

Ähnlichkeit zwischen zwei Clustern ist das Mittelwert aller paarweisen Ähnlichkeiten zwischen je einem Objekt aus den beiden Clustern:

$$d(C_1, C_2) = \frac{\sum_{\mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2} d(\mathbf{x}_1, \mathbf{x}_2)}{|C_1||C_2|}$$



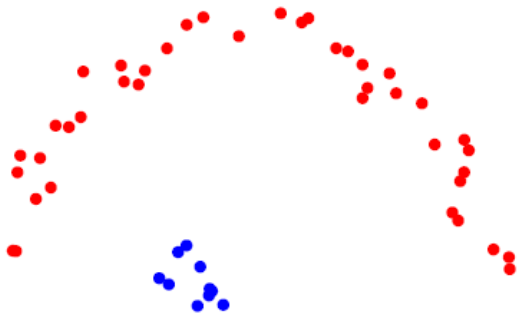
Eigenschaften

- Guter Kompromiss zwischen Single und Complete Linkage (weniger anfällig gegenüber Ausreißern als Complete Linkage)
- Neigt zu kompakten Clustern mit jeweils ähnlicher Varianz

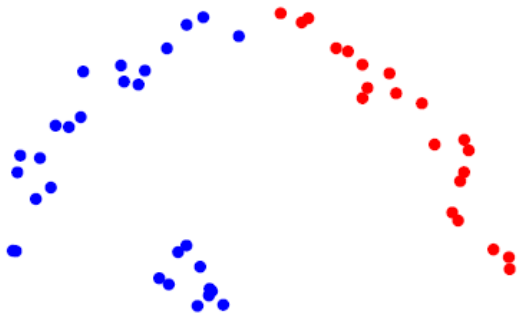
Single vs. Complete vs. Average Linkage

- Complete und Average Linkage neigen zur Erzeugung kompakter Cluster
- Single Linkage neigt zur Erzeugung länglicher, kettenartiger Cluster

  → Manchmal ist genau das gewünscht!



Single Linkage

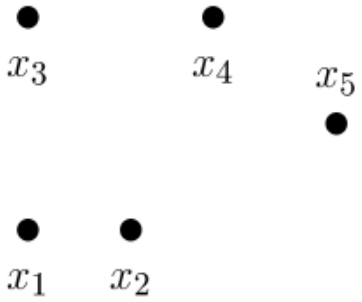


Complete Linkage

Agglomeratives Clustering: Beispiel (1)

1

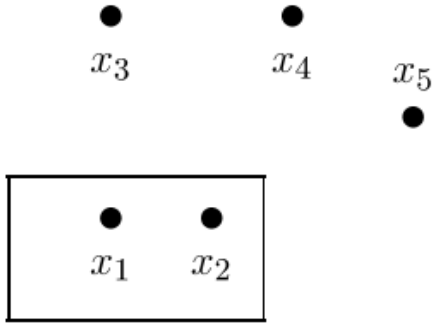
	x_1	x_2	x_3	x_4	x_5
x_1		1.00	4.00	7.24	10.00
x_2			5.00	4.64	5.00
x_3				3.24	10.00
x_4					2.44
x_5					



Aktualisiere Proximitätsmatrix basierend auf Single Linkage

2

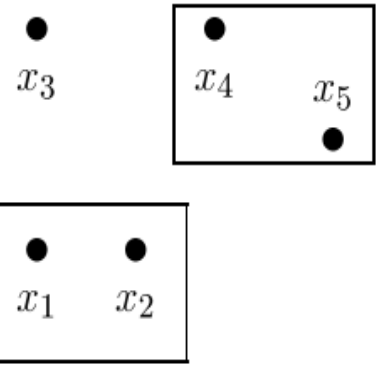
	x_{12}	x_3	x_4	x_5
x_{12}		4.00	4.64	5.00
x_3			3.24	10.00
x_4				2.44
x_5				



Agglomeratives Clustering: Beispiel (2)

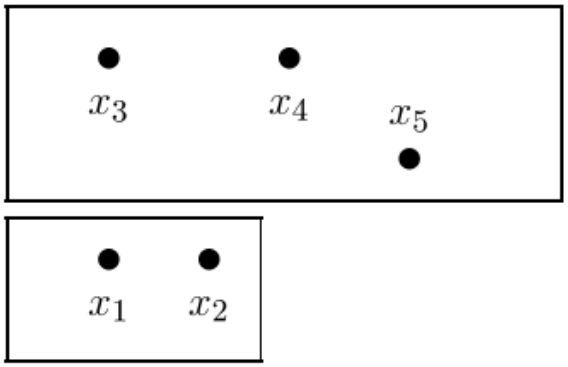
3

	x_{12}	x_3	x_{45}
x_{12}		4.00	4.64
x_3			3.24
x_{45}			



4

	x_{12}	x_{345}
x_{12}		4.00
x_{345}		



Bestimmung einer Partitionierung

Falls anstatt der vollständigen Baumstruktur eine bestimmte Einteilung in Cluster gewünscht ist

Einfacher Ansatz

> Vorgabe Schwellwert für Ähnlichkeit, der nicht unterschritten werden soll
> Abbruch der Zusammenführung von Clustern, sobald die Ähnlichkeit unterschritten wird

Visueller Ansatz

> Finde guten Schnittpunkt im Dendrogramm nach vollständiger Erzeugung der Baumstruktur
> Vorteil: Der Schnitt muss nicht strikt horizontal verlaufen

Anspruchsvollerer Ansatz

> Analysiere die Folge der zu überwindenden Ähnlichkeiten während des Zusammenführungsprozesses
> Finde Schritt bei dem die Ähnlichkeiten sprunghaft kleiner werden (→ Ellenbogen-Kriterium)

Hierarchisches Clustering: Vorteile und Nachteile

- ✓ Erzeugt nicht nur Cluster sondern eine vollständige Hierarchie (Baumstruktur)
- ✓ Leicht verständlich
- ✓ Clusteranzahl muss vorher nicht festgelegt werden
- ✓ Hierarchie gut als Dendrogramm visualisierbar
- ✓ Gute Partitionierung kann leicht vom Dendrogramm abgelesen werden
- ✘ Hohe Laufzeit-Komplexität: Nicht geeignet für große Datensätze
- ✘ Keine einfache (kompakte) Repräsentation der Cluster

Nicht-hierarchisches (partitionierendes) Clustering

- Prototyp- oder model-basiertes Clustering
- Dichte-basiertes Clustering
- Grid-basiertes Clustering

Prototypbasiertes partitionierendes Clustering

Die Einteilung erfolgt gleichzeitig → vollständige Einteilung verfügbar in jeder Iteration

Repräsentationen

> Zuordnung von Objekten zu Clustern (**Cluster-Zugehörigkeiten**)
  → Geben an ob (bzw. zu welchem Grad) ein Objekt zu einem Cluster gehört
> Cluster-Repräsentation (**Cluster-Zentren**)
  → Oft wird ein Cluster durch den Mittelwert aller ihm zugeordneten Objekte repräsentiert

Ziel

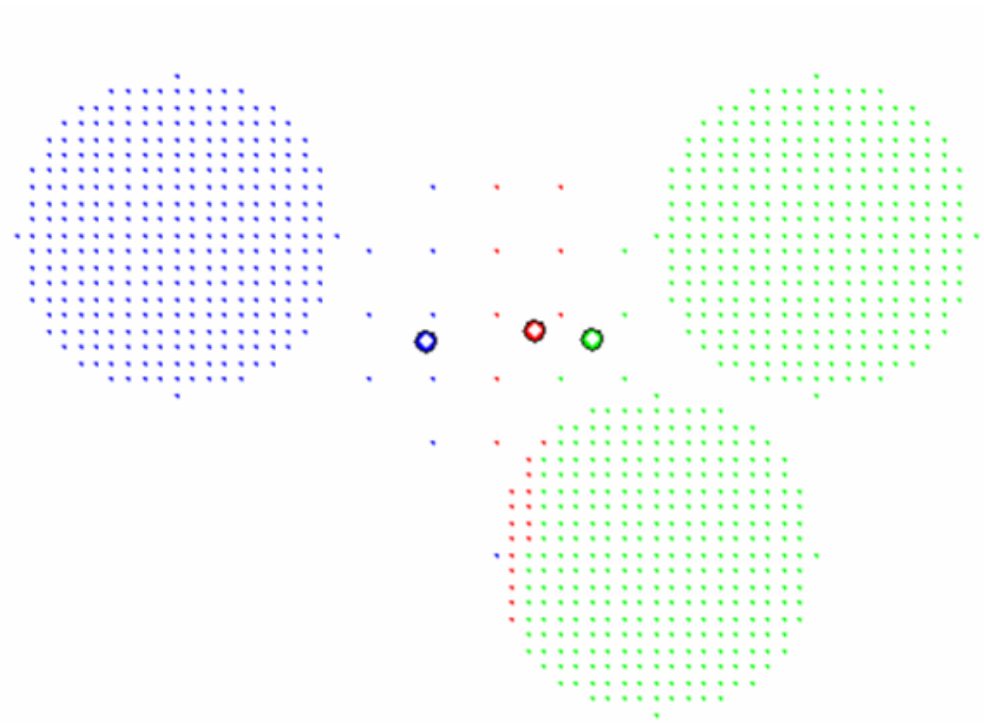
> Finde Cluster, die die Summe der Abstände aller Objekte zum nächsten Cluster minimieren

k-Means-Algorithmus

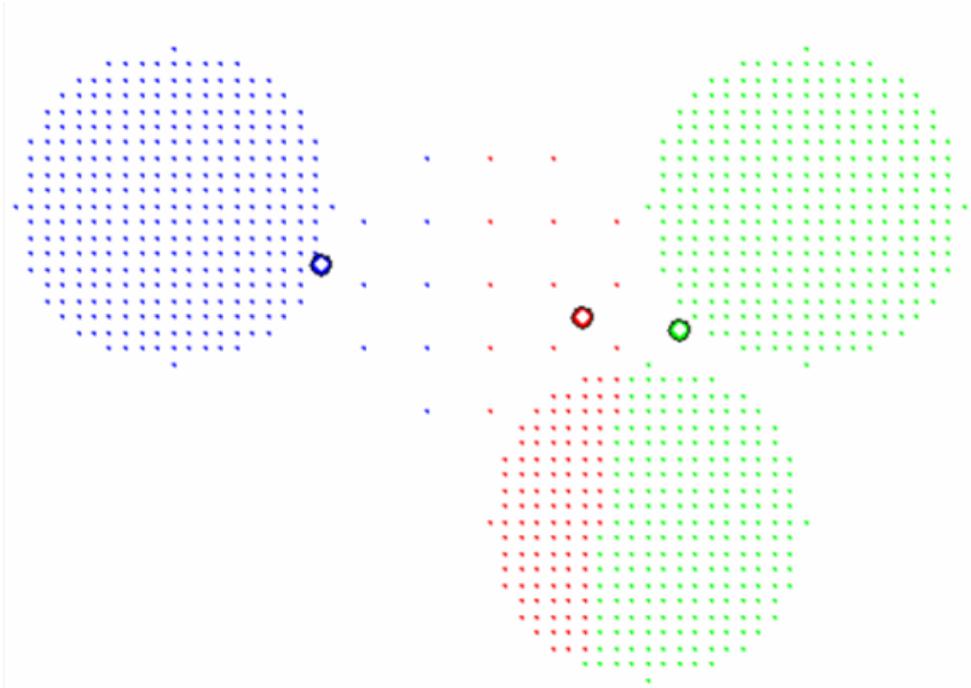
1. Choose the number of clusters, k
2. Randomly generate *k* clusters and determine the cluster centers, or directly generate *k* random points as cluster centers
3. Repeat the following steps until some convergence criterion is met (usually that the assignment did not change)
   - 3.1 Assign each point to the nearest cluster center
   - 3.2 Recompute the new cluster centers

→ Minimiert die Summe der Abstände von allen Objekten zum nächsten Cluster durch abwechselndes Anpassen der Cluster-Zentren und der Cluster-Zugehörigkeiten

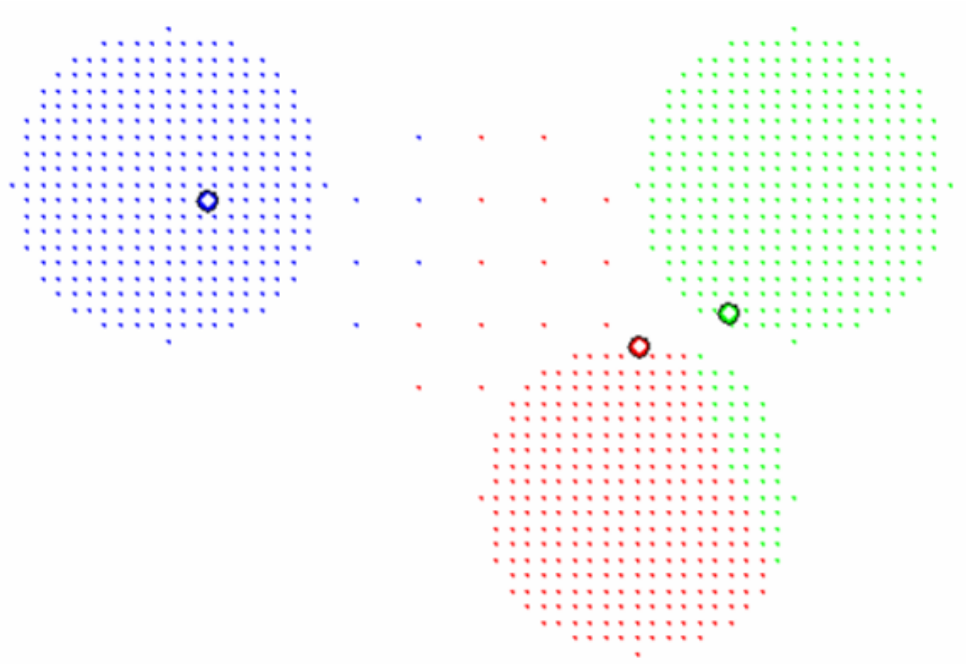
k-Means-Clustering: Beispiel (Iteration 1)



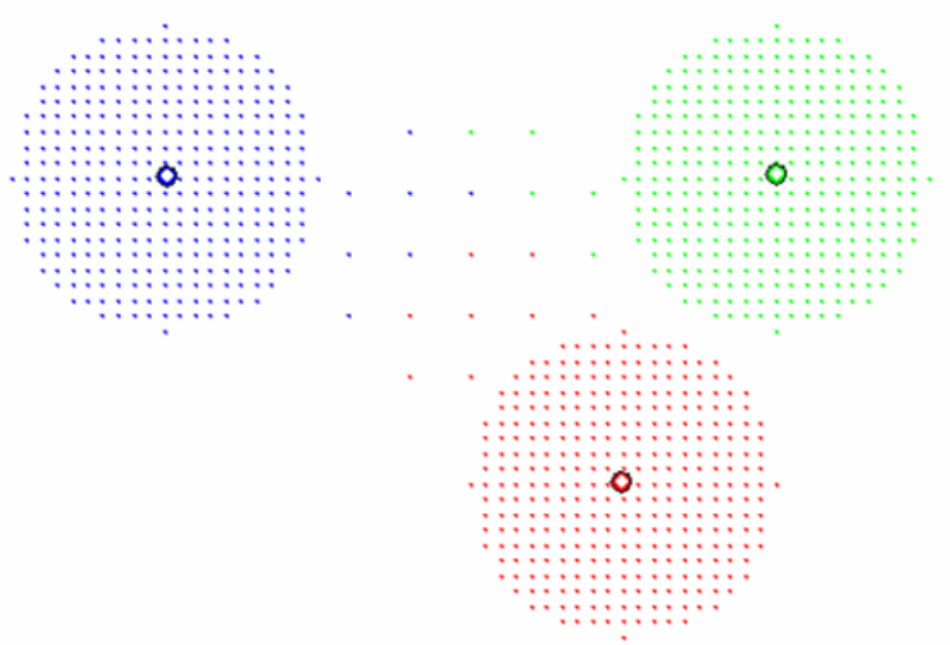
k-Means-Clustering: Beispiel (Iteration 2)



k-Means-Clustering: Beispiel (Iteration 3)



k-Means-Clustering: Beispiel (Iteration 12)

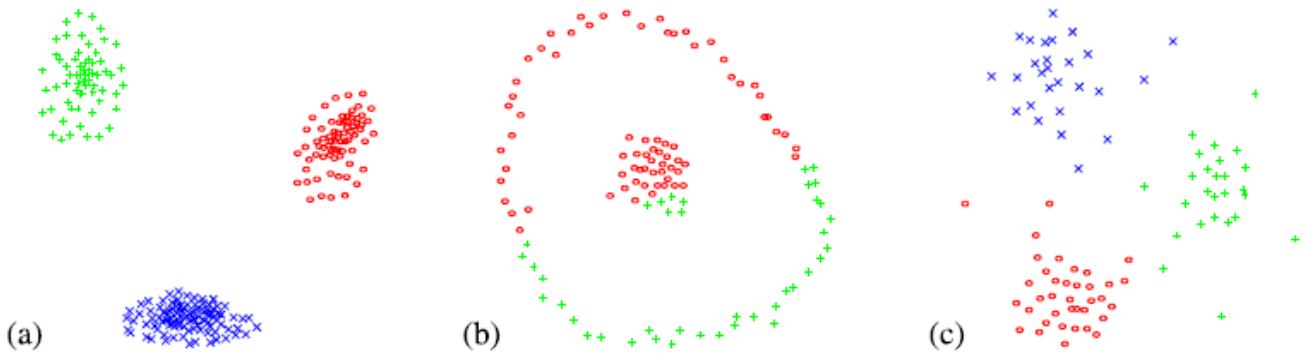


Entdeckbare Strukturen: Clustering-Ergebnisse

a) Gut-getrennte, kompakte (sphärische) Cluster sind einfach zu entdecken

b) Nicht-sphärische Cluster stellen oft ein Problem dar

c) Jedes Objekt gehört zu genau einem Cluster: Bei Ausreißern nicht sinnvoll



k-Means-Algorithmus: Vorteile und Nachteile

- ✓ Konvergenz ist garantiert
  → aber nur lokales Optimum!
- ✓ Relativ schnell
- ✓ Verständlich
- ✓ Erzeugt kompakte, sphärische Cluster

- ✘ Clusteranzahl k muss vorgegeben werden
- ✘ Wahl von k beeinflusst Qualität stark
  → Struktur ggf. nicht erkennbar
- ✘ Nicht-deterministisch (zufällige Initialisierung der Cluster-Zentren)
- ✘ Objekte gehören genau zu einem Cluster
- ✘ Bleibt oft in einem lokalen Optimum stecken
- ✘ Empfindliche gegenüber Ausreißern
- ✘ Standardmäßig nur für numerische Merkmale

k-Means-Algorithmus: Varianten

Cluster-Zugehörigkeiten

- $u_{ij} \in \{0, 1\}$ hard clustering (k-means)
- $u_{ij} \in [0, 1]$ probabilistic/fuzzy clustering (fuzzy c-means)

Abstands-/Ähnlichkeitsfunktion

- Bedeutung ausgewählter Merkmale hervorheben
  → unterstützt andere Clusterformen, z.B. elliptisch

Bestimmung der Cluster-Anzahl k

- Führe k-Means für alle relevanten Werte von k aus
- Bestimme beste Clustereinteilung
- Achtung: Zielfunktion nimmt mit wachsendem k stets ab → Overfitting

Bewertung der Cluster-Qualität

- Clustering-Verfahren entdecken immer eine Struktur—selbst wenn es keine gibt.
- Das Ergebnis ist weder richtig noch falsch (→ unüberwachtes Lernen)

- **Herausforderung**:
  - Definition von Qualitätsmaßen, die die Anforderungen der Anwendung reflektieren
  - Wie viele Cluster welcher Art sollen gefunden werden?
  - Beispiele für Qualitätskriterien (strukturelle Eigenschaften)
    - Kompaktheit (compactness)—Wie homogen sind die einzelnen Cluster?
    - Trennung (separation)—Wie heterogene sind verschiedene Cluster?
    - Anzahl Cluster—da durch Modellkomplexität die Verständlichkeit leidet

Custer-Qualität: Kompaktheit

Die Kompaktheit basiert auf der durchschnittlichen Distanz von Objekten eines jeden Clusters zu seinem Zentrum:

$$Q = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{\boldsymbol{x} \in C_i} d(\boldsymbol{x}, \mu_i)$$

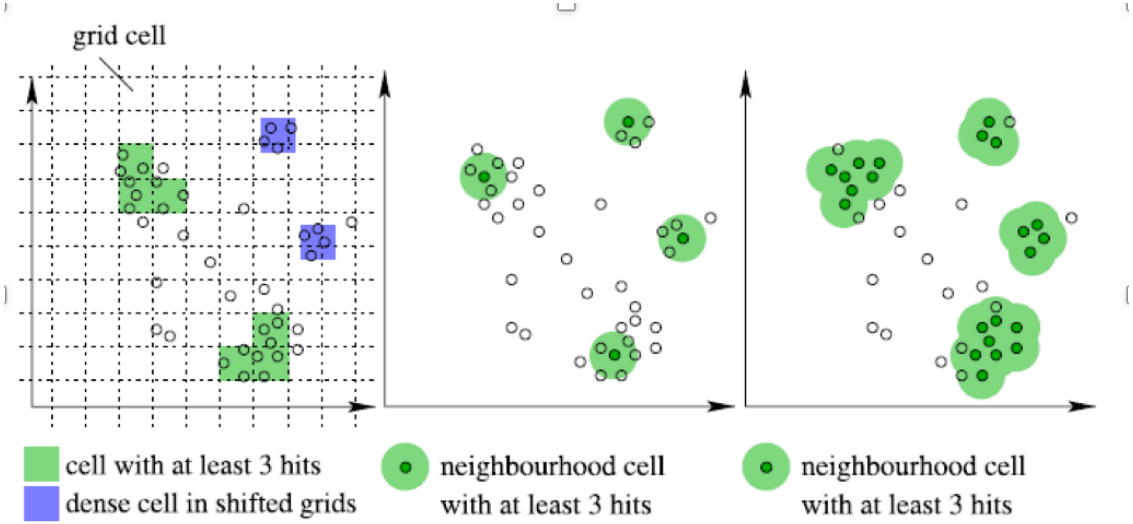
- $|C_i|$ Anzahl Objekte in Cluster i
- μ_i Mittelwert (Zentrum) von Cluster i
- Q ist klein, wenn bei jedem Cluster die Objekte im Durchschnitt dicht beim Zentrum liegen
- Trennung der Cluster wird nicht berücksichtigt

Dichte-basiertes Clustering

Idee

- Messe Dichte der Objekte an bestimmten Stellen oder Regionen des Datenraumes
- Definiere Regionen als Cluster, wenn die Dichte einen vorgegebenen Schwellwert überschreitet
- Verbinde benachbarte Cluster

- Kann Cluster beliebiger Form entdecken
- Cluster schwer zu beschreiben



Zusammenfassung

- Viele verschiedene Clustering-Verfahren mit unterschiedlichsten Eigenschaften
- Wahl des Clustering-Verfahrens bestimmt maßgeblich die Art von Struktur (Cluster), die entdeckt werden kann
- Wegen der Diversität: Einheitliche formale Beschreibung eines Clusters schwierig
- Hierarchische Cluster-Verfahren erzielen sehr gute Ergebnisse, sind aber nur für kleine Datensätze geeignet
- k-Means-Algorithmus gut geeignet für numerische Daten mit sphärischen Strukturen
- Evaluierung und Interpretation ist die größte Herausforderung bei der Clusteranalyse