Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Datenmanagement & -analyse

Unüberwachtes Lernen

Prof. Dr. Christoph M. Flath
Lehrstuhl für WI & BA
Julius-Maximilians-Universität Würzburg

Sommersemester 2021

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Agenda

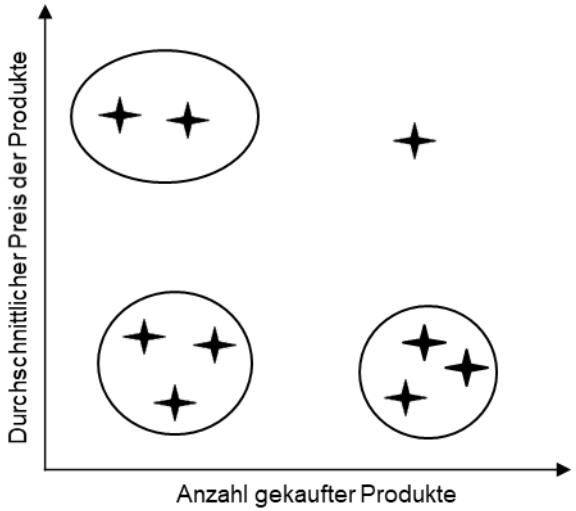
1 **Clusteranalyse**

- 1.1 **Partitionierende Verfahren**
- 1.2 **Hierarchische Verfahren**

2 **Assoziationasregeln**

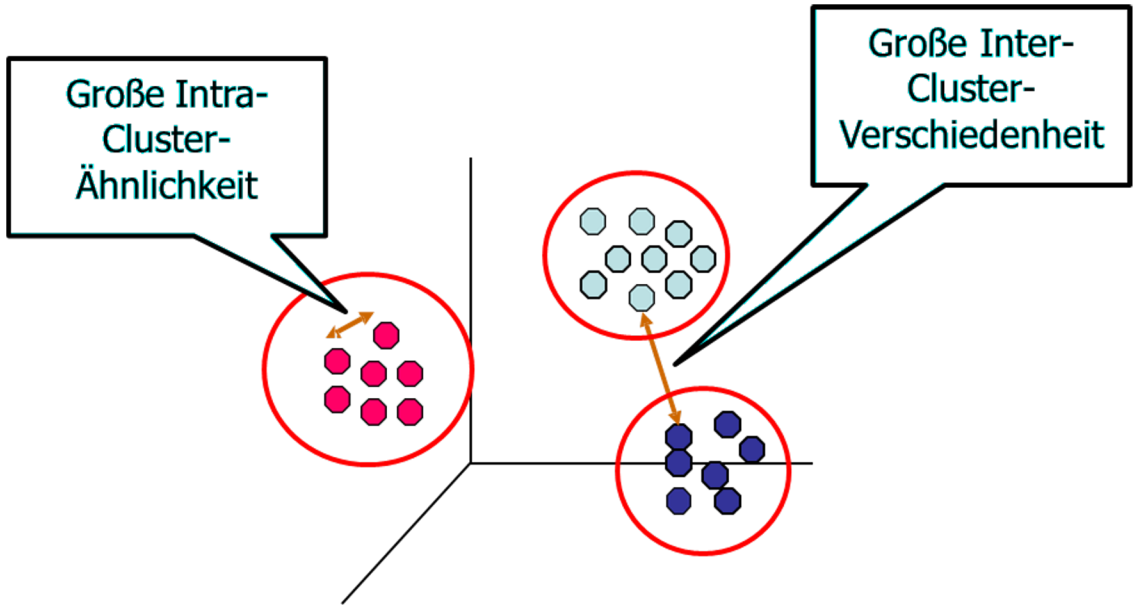
Clusteranalyse

- Idee
  - Bestimmung von Gruppen ähnlicher Tupel in multi-dimensionalen Datensätzen.
  - „Klassifizieren ohne die Klassen vorher zu kennen“ (nichtüberwachtes Lernen).
- Beispiel
  - Identifizieren von Kundengruppen zum Design passender Produkte.



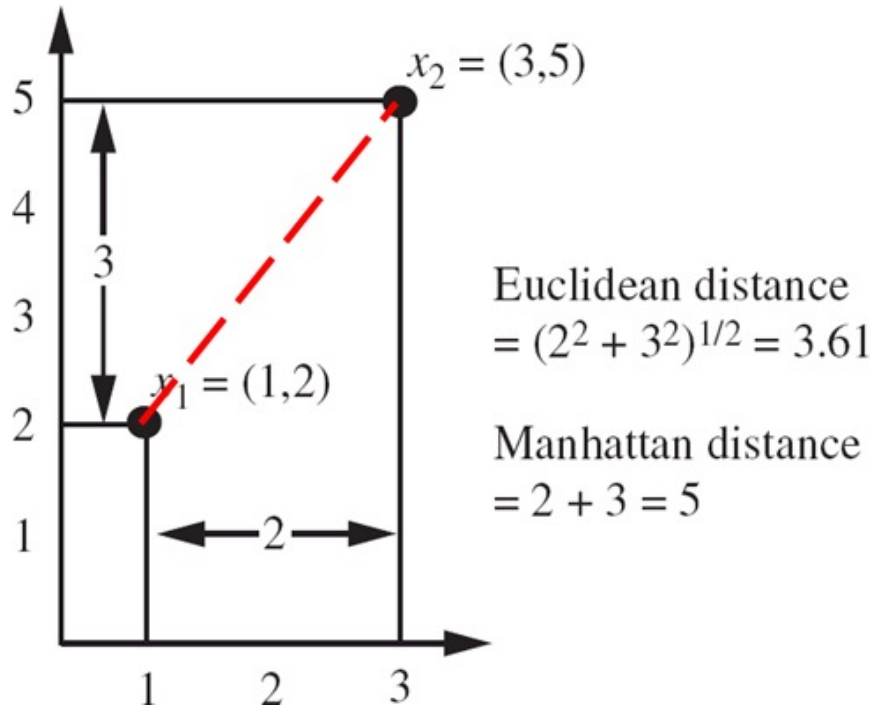
Eigenschaften „guter“ Cluster

- Als Ähnlichkeitsmaß wird oft Abstand verwendet
  - Geringer Abstand: Objekte sind ähnlich
  - Großer Abstand: Objekte sind unähnlich



Abstandsmasse

- Viele Clustering-Verfahren benötigen ein geeignetes Abstandsmaß.
- Übliche Maße (in der Ebene und in hochdimensionalen Räumen):
  - Euklidisch
  - Manhattan
  - Maximumsnorm



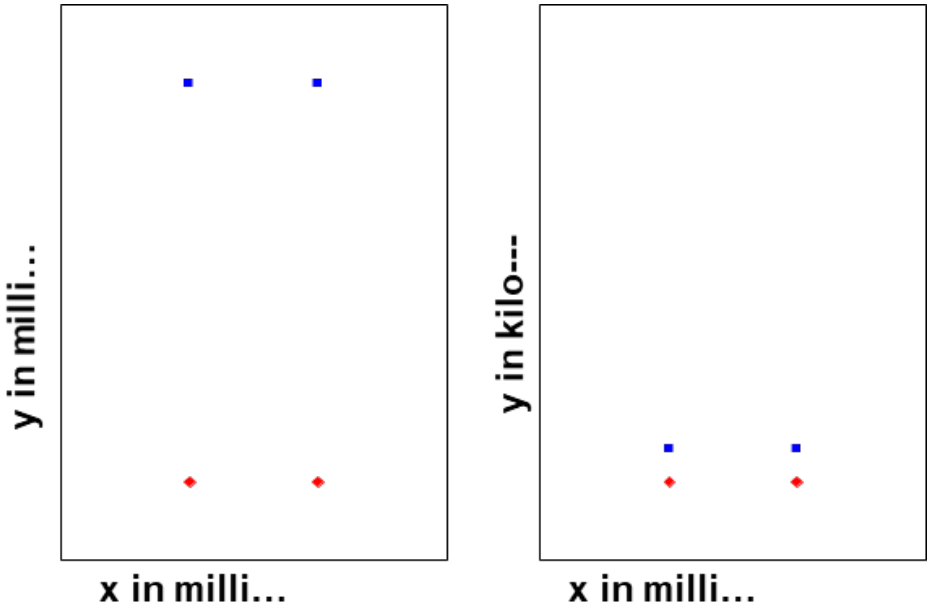
Diskrete Werte und Normalisierung

- Abstände von binären, kategorischen und ordinalen Werten sind nicht intuitiv klar.
- Beispiel ordinale Werte
  - Nur Testen auf =, ≠, <, >, ≤ und ≥ möglich.
  - Reihenfolge der möglichen Werte von r_0 bis $r_n, r_0 \leq r_i \leq r_n$
  - Lineares Mapping des Platzes in der Reihenfolge r_i auf 0…1: Wert = $\frac{r_0 - 1}{r_n - 1}$
- Beispiel kategorische Werte:
  - Es kann nur auf Gleichheit getestet werden.
  - Abstandsmaß bei p kategorischen Variablen:

$$d = \frac{p - m}{p}$$

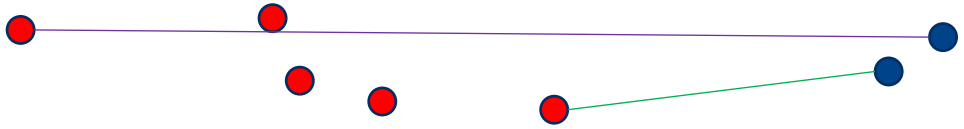
    m: Anzahl Übereinstimmungen
  - Vermutlich werden unterschiedliche Gewichtungen nötig sein

- Um Abstände zu berechnen, müssen die Daten oft normalisiert werden.
  - Sonst: Höheres Gewicht bestimmter Attribute.
- Einfache Technik: Lineare Skalierung auf 0…1
  - Achtung: Ausreißer vorher behandeln!

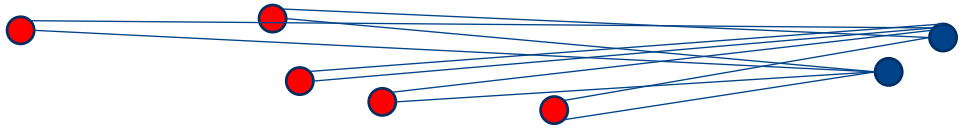


Berechnung des Abstands zwischen Clustern

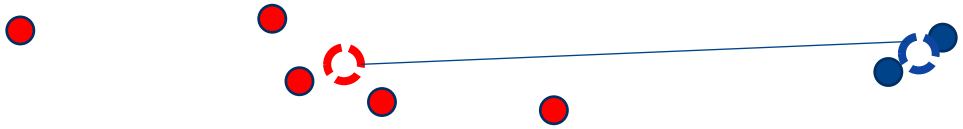
- **Single (complete) linkage**
  - Kleinster (größter) Abstand zweier Punkte von zwei Clustern. Sehr anfällig bei Ausreißern/Rauschen.



- **Average linkage**
  - Mittlerer Abstand zwischen allen Punkten von zwei Clustern. Sehr rechenaufwändig.



- **Centroid**
  - Abstand der Cluster-Zentren zueinander. Ähnlich wie Average, aber weniger aufwändig.

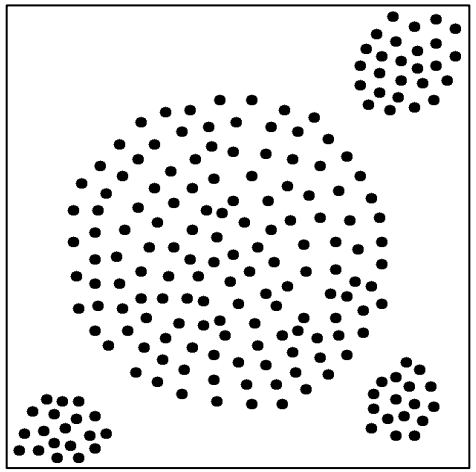


- **Medoid**
  - Abstand der zentralsten Repräsentanten der Cluster zueinander.
  - Ähnlich wie Centroid, aber weniger Anfällig für Ausreißer/Rauschen.

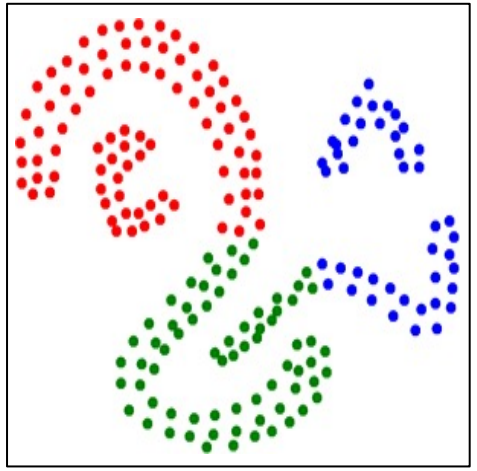
Probleme mit abstandsbasiertem Clustering

- Oft werden nur konvexe Cluster gefunden
  - Abhängig von Berechnungsmethode zum Abstand zweier Cluster (letzte Folie).
  - Single link findet am ehesten konkave Cluster.
  - Mögliche Lösung: dichtebasierte Maße

- Hohe Dimensionen / dünn besetzte Attribute verringern die Bedeutung des Abstands.
  - Ein Ansatz: Subspace-Clustering
  - Suche Cluster unter Verwendung eines Subsets von Attributen.
  - Subsets schwer zu bestimmen, wenn Bedeutung der Attribute unbekannt.
  - Achtung: Nicht unbedingt globale Lösung!

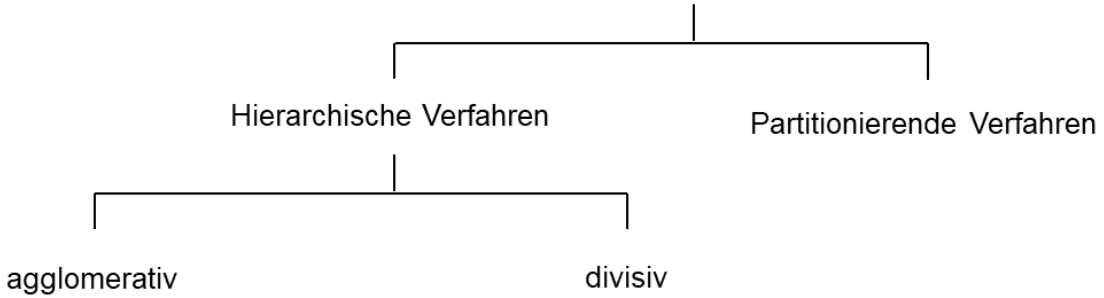


Konvexe Punktwolken



Konkave Punktwolken

Kategorisierung von Clustering-Methoden



- Partitionierende Verfahren (# Cluster fest)
  - Zunächst: Zufällige Clustereinteilung
  - Dann: Sukzessiver Wechsel der Clusterzugehörigkeit der einzelnen Tupel
- Hierarchische Verfahren (# Cluster variabel)
  - Agglomerative Verfahren
    - Zunächst: Jedes Tupel ein Cluster
    - Dann: Schrittweise Zusammenfassung ähnlicher Cluster
  - Divisive Verfahren
    - Zunächst: Alle Tupel in selbem Cluster
    - Dann: Schrittweise Aufteilung in möglichst unähnliche Teilcluster

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Agenda

Partitionierendes Clustering: k-means

- Gegeben:
  - Lerndatensatz
  - Anzahl der zu findenden Cluster k
  - Abstandsmaß

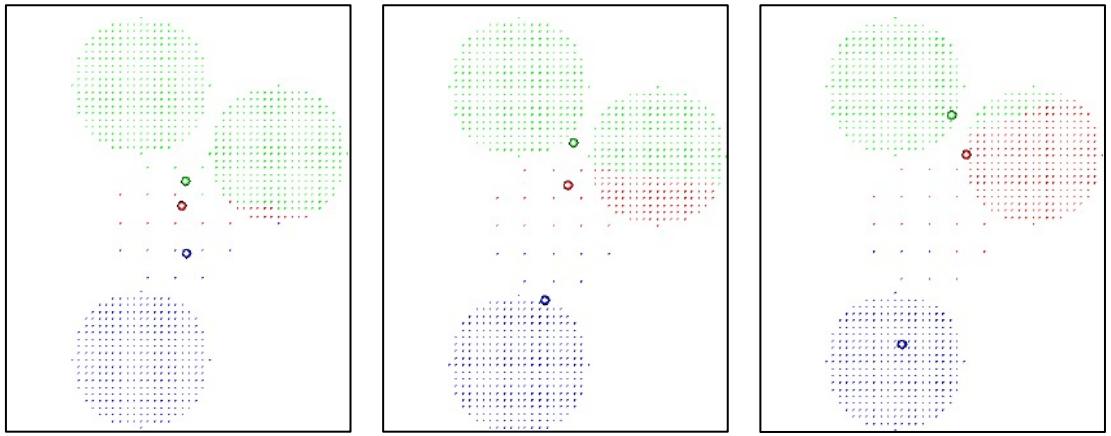
- Gesucht:
  - Partitionierung des Datensatz in k Cluster

Algorithmus

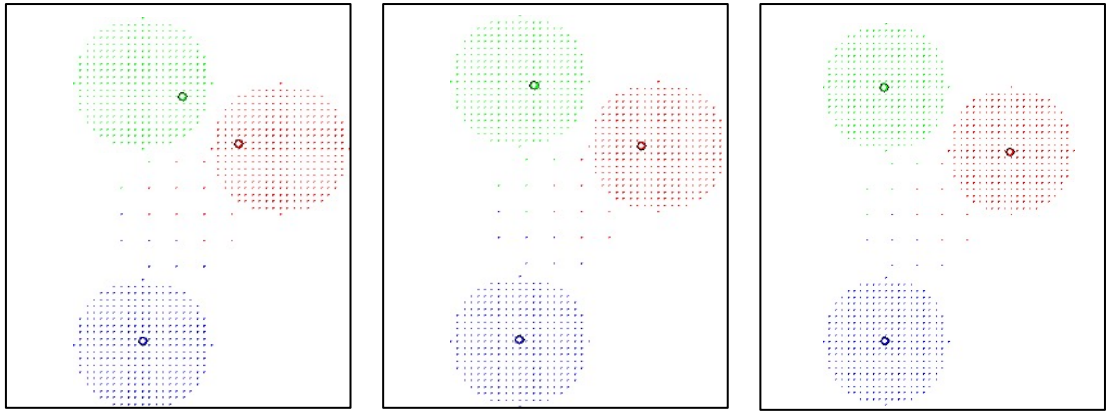
1. Wähle zufällig *k* Cluster-Zentren
2. Weise allen Objekten Cluster zu, basierend auf den Abständen zu den Cluster-Zentren
3. Berechne die neuen Zentren der Cluster
4. Wiederhole Schritt 2 und 3 bis Zentren stabil

Illustration k-means

Cluster nach der 1., 2. und 3. Iteration



Cluster nach der 4., 5. und 12. Iteration



Bemerkungen zu k-means

- Sehr unterschiedliche Ergebnisse möglich.
  - Ggf. lokales Minimum bei “schlechtem” Seed.
  - Abhilfe: Mehrfache Ausführung, unterschiedliche Seeds.

- k wird nicht automatisch ermittelt
  - 1.) Mehrfache Ausführung mit verschiedenen k
  - 2.) Bestimmung der Ähnlichkeit von jedem Cluster
  - 3.) Entscheidung für k mit homogensten Clustern

- Anfällig für Outlier und Rauschen, da Verwendung von Mittelwerten.
  - *k*-medoids anstatt von *k*-means (reale Repräsentanten für die Cluster anstatt rechnerischer Mittelpunkte).

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Agenda

1 Clusteranalyse

1.1 Partitionierende Verfahren

1.2 Hierarchische Verfahren

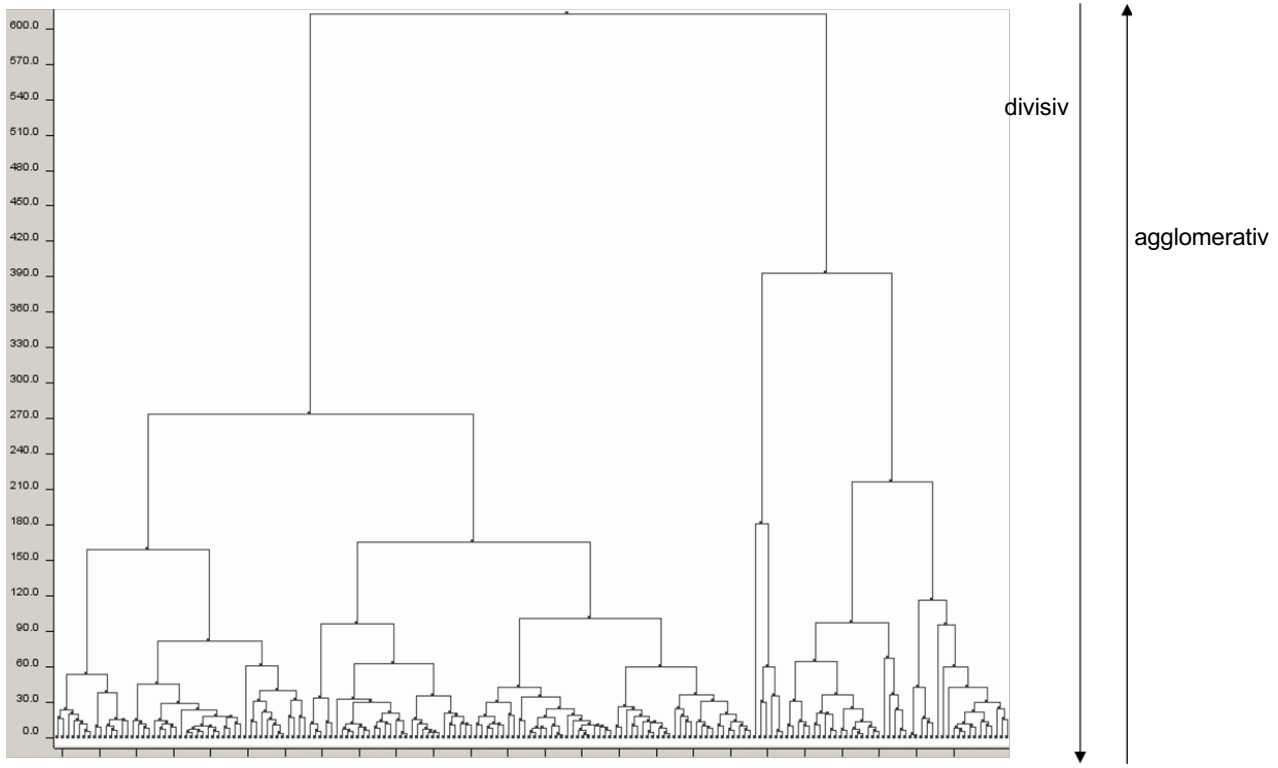
2 Assoziationsregeln

Hierarchisches Clustering: Dendrogramme

- Ein Dendrogramm ist ein Binärbaum.
  - Wurzel repräsentiert alle Tupel.
  - Blätter repräsentieren die einzelnen Tupel.

- Nachfolgerknoten repräsentieren die zwei Cluster, aus denen der Vorgängerknoten entstanden ist bzw. in die er aufgespalten wurde.
  - Jede horizontale Ebene ist eine Clustereinteilung.

- Die Kantenlänge repräsentiert die Unähnlichkeit der Cluster.
  - Vorteil: Anzahl der Cluster kann mit Hilfe eines Dendrogramms bestimmt werden.
  - Die Ebene mit der längsten Kante wird als Clustereinteilung genommen (Maximierung der Unähnlichkeit).



Hierarchisches Clustering

- Es wird ein geeignetes Abbruchkriterium benötigt.
  - k ist nur eine, einfache Möglichkeit.
  - Besser: Dann abbrechen, wenn Unterschied zwischen Supercluster und Kindclustern besonders groß.
  - Es kann auch die gesamte Hierarchie berechnet werden um anschließend die Anzahl der Cluster zu bestimmen.

- Agglomeratives Clustering
  - Gegeben: Lerndatensatz, Abstandsmaß,
  - Strategie zur Cluster-Abstand-Berechnung
  - Wichtige Datenstruktur: Matrix mit Distanzen zwischen den Clustern
  - Gesucht: Gesamte Cluster-Hierarchie (Dendrogramm)

Algorithmus Agglomeratives Clustering

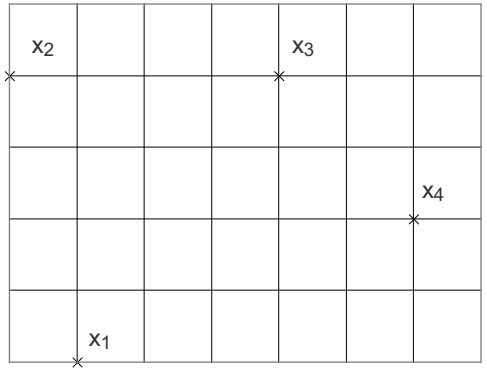
1. Berechne die Distanz-Matrix
2. Betrachte jeden Datenpunkt als Cluster
3. Vereinige die zwei Cluster mit minimalem Abstand
4. Aktualisiere die Distanz-Matrix
5. Wiederhole Schritt 3 und 4 solange mehr als ein Cluster

Anschließend: Bestimmung der Anzahl der Cluster mittels Dendrogramm.

Illustration agglomeratives Clustering

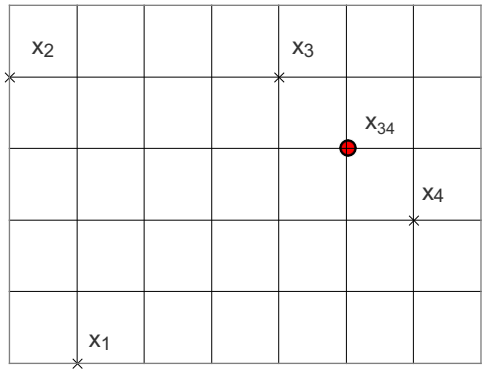
- Entfernungsmatrix mit Datenpunkten $x_1 \ldots x_4$ (euklidische Centroid-Abstände)

	x_1	x_2	x_3	x_4
x_1	0	4,12	5,00	5,39
x_2		0	4,00	6,32
x_3			0	2,83
x_4				0



- Verschmelzen von x_3 und x_4 zu neuem Cluster x_{34}.

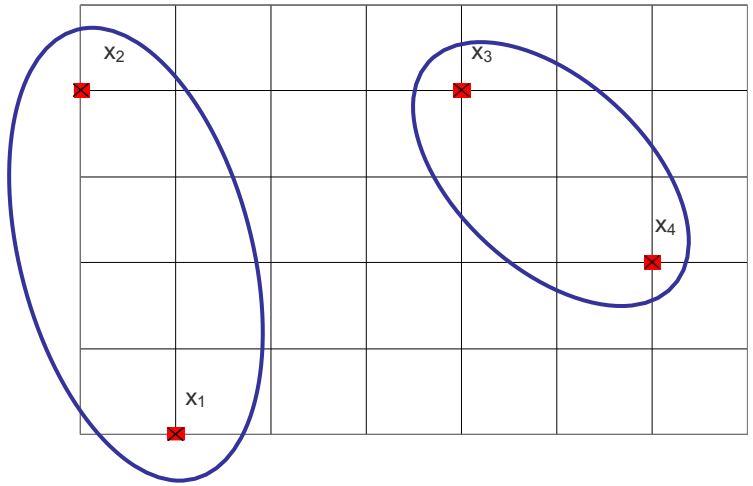
	x_1	x_2	x_{34}
x_1	0	4,12	5,00
x_2		0	5,10
x_{34}			0

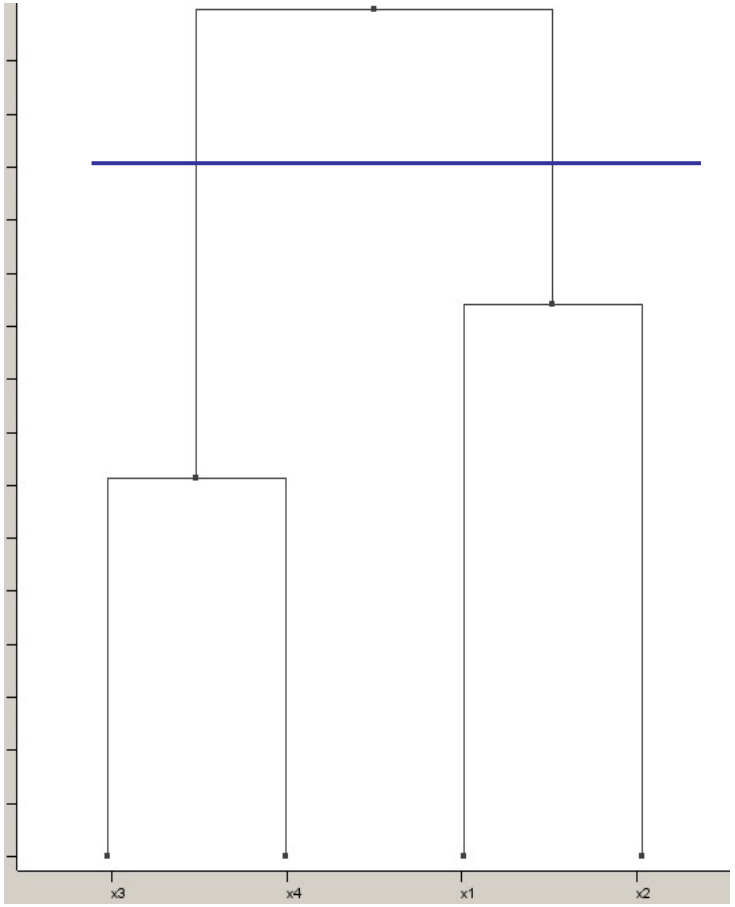


- Welche Tupel/Cluster werden als nächstes verschmolzen?

x_{34} repräsentiert Cluster-Zentrum (Centroid).

Dendrogramm zum Beispiel





Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Agenda

1 **Clusteranalyse**

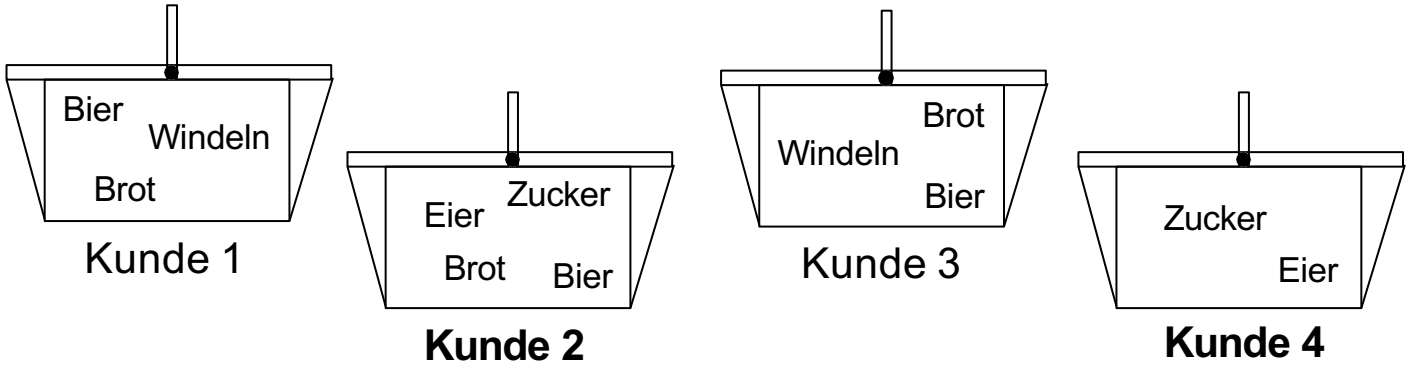
2 **Assoziationasregeln**

Motivation - Warenkorbanalyse

- Gesucht: Einkaufsgewohnheiten
  - Höhere Kundenzufriedenheit durch günstige Anordnung
  - Höherer Absatz durch ungünstige Anordnung

- Fragestellung: Welche Kombinationen werden häufig gekauft (Frequent Itemsets)?

- Warenkörbe (Beispiel)



Assoziationsregeln

- Darstellung von Assoziationsregeln
  - Antecedent X → Consequent Y

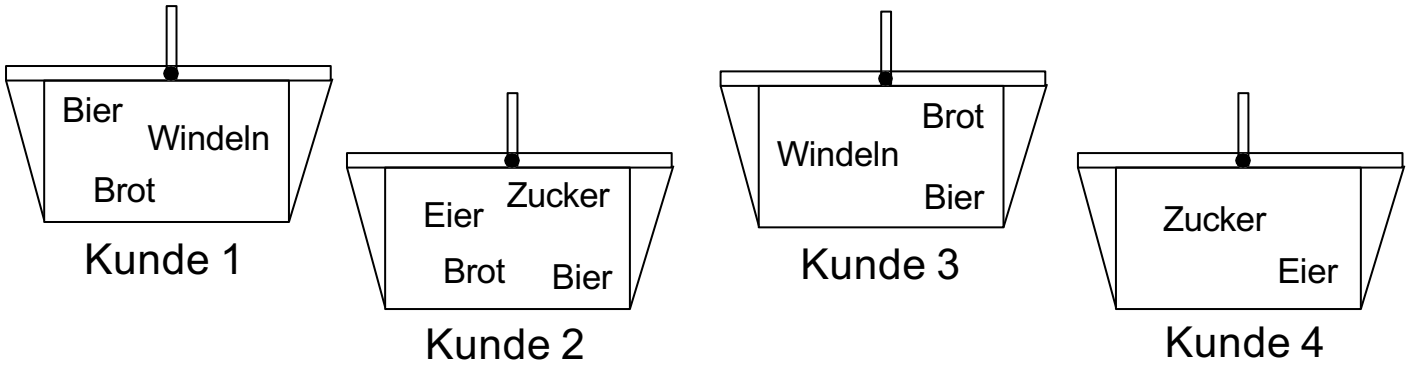
- Wahrscheinlichkeitsbasierter Charakter
  - Consequent Y ist mit der Wahrscheinlichkeit P wahr,
  - … wenn der Antecedent X wahr ist
  - Bedingte Wahrscheinlichkeit P(Y | X)!

- Zugelassene Wertebereiche
  - Besonders geeignet für kategorische Daten
  - Möglichkeit Grenzwerte für kontinuierliche Werte zu setzen

Motivation – Warenkorbanalyse II

- Frequent Itemsets (mit mind. 2 Items)
  - {Brot, Bier}, {Brot, Bier, Windeln}, {Zucker, Eier}, {Bier, Windeln}, {Brot, Windeln}

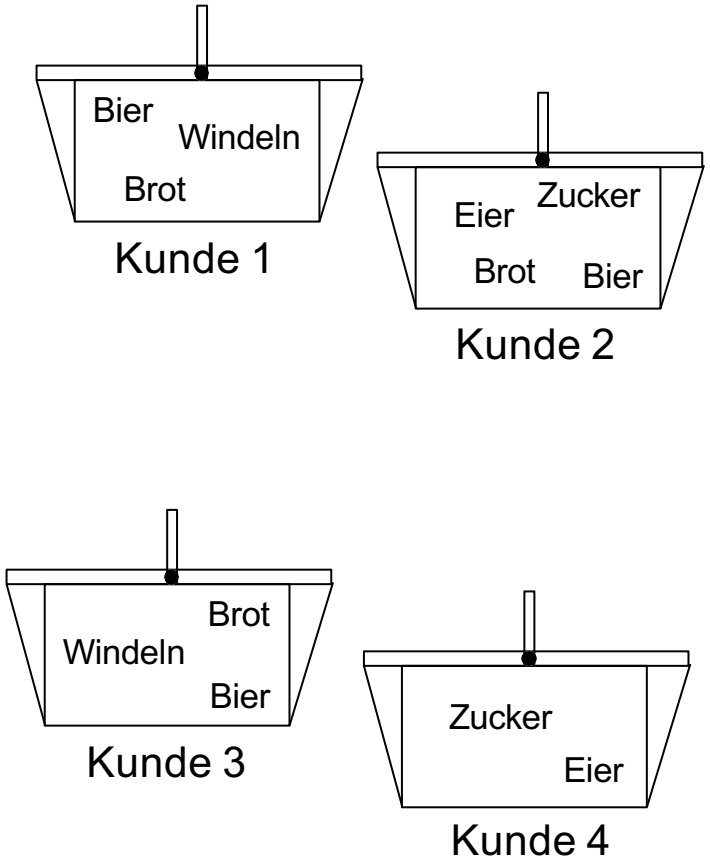
- Wie lassen sich aus Frequent Itemsets Assoziationsregeln ableiten?
  - Beispiel: Wer Windeln kauft, kauft auch Brot



Support

- Alternative Namen
  - Häufigkeit, Abdeckung
- Angabe bezüglich der Häufigkeit eines Portfolios
- Anzahl / Anteil der Transaktionen, die $X \cup Y$ enthalten
- Formal: $P(X \cup Y)$

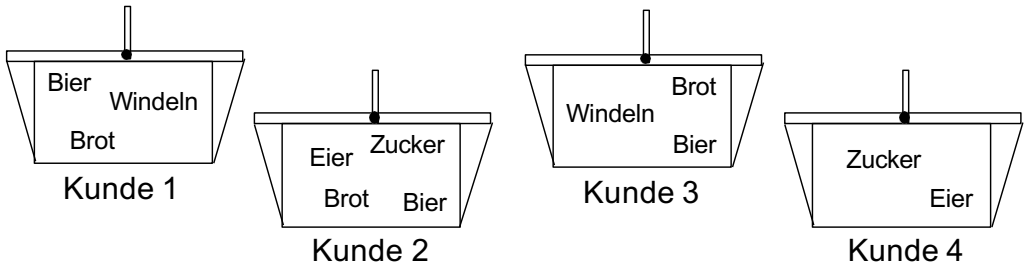
- Beispiel:
  - Die Kombination <Windeln, Bier> tritt in 50% der Warenkörbe auf.
  - Support = 50%



Confidence

- Alternative Namen
  - Genauigkeit
  - „Überraschungsmaß“
- Wenn eine Transaktion X enthält, dann auch Y (mit gegebener Genauigkeit)
  - Formal: $P(Y \mid X) = \frac{|X \cup Y|}{|X|}$
- Beispiel:
  - Wenn Windeln gekauft wurden, wurde in 100% aller Fälle auch Bier gekauft
  - Confidence = 100%
- Ziel: Finden von Regeln mit
  - … hohem Support (support > minSup) und …
  - … hoher Confidence (confidence > minConf)

- Frequent Itemsets (minSup = ½)
  - {Brot, Bier} (Support = ¾);
  - {Brot, Bier, Windeln} (Support = ½);
  - {Zucker, Eier} (Support = ½);
  - {Bier, Windeln} (Support = ½);
  - {Brot, Windeln} (Support = ½)
- Assoziationsregeln (Auswahl)
  - Brot → Bier (Confidence = 100%)
  - Brot, Bier → Windeln (Confidence = 67%)
  - Zucker → Eier (Confidence = 100%)



A-Priori Algorithmus für Frequent Itemsets

A-Priori Eigenschaft

- Itemset häufig, wenn Supermenge häufig
- Beispiel:
  - {Bier, Windeln, Brot} häufig
    → {Bier, Windeln}, {Bier, Brot}, {Windeln, Brot} und {Bier}, {Windeln}, {Brot} häufig
- Itemset kann nur häufig sein, …
  - … wenn alle Teilmengen häufig
- Dadurch:
  - Bestimmung von Frequent Itemset Kandidaten mit n Elementen aus solchen mit (n - 1) Elementen möglich

A-Priori Algorithmus – Frequent Itemsets

- Finden aller Itemsets mit ausreichendem Support:
  - Beginn mit ein-elementigen Sets (1)-Sets: einfaches Abzählen
  - Berechnung der k-Sets aus den (k-1)-Sets: Join-Step: Ermittlung von Kandidaten; Aus A-Priori Eigenschaft:
    - Alle (k-1)-elementigen Teilmengen eines k-Sets sind (k-1)-Sets,
- Prune-Step: Löschen aller Kandidaten, die eine „unzulässige“ (k-1)-elementige Teilmenge haben. (Support Counting, d. h. Abzählen wie häufig die Kandidaten wirklich sind.)

A-Priori – Frequent Itemset (Beispiel)

- Beispieltupel:
  - {A, B, E}
  - {B, D}
  - {B, C}
  - {A, B, D}
  - {A, C, D}
  - {B, C}
  - {A, C}
  - {A, B, C, E}
  - {A, B, C}

- MinSup: 2/9,
  - d.h. Itemset ist häufig, wenn 2 Tupel es enthalten

- Ein-elementige Frequent Itemsets
  - {A}: 6
  - {B}: 7
  - {C}: 6
  - {D}: 3
  - {E}: 2
- Alle Items sind häufig!

A-Priori – Frequent Itemset (Beispiel)

- Beispieltupel:
  - {A, B, E}
  - {B, D}
  - {B, C}
  - {A, B, D}
  - {A, C, D}
  - {B, C}
  - {A, C}
  - {A, B, C, E}
  - {A, B, C}

- Ein-elementige Frequent Itemsets
  - {A}: 6
  - {B}: 7
  - {C}: 6
  - {D}: 3
  - {E}: 2

- Zwei-elementige Frequent Itemsets
- {A, B}: 4
- {A, C}: 4
- {A, D}: 2
- {A, E}: 2
- {B, C}: 4
- {B, D}: 2
- {B, E}: 2
- ~~{C, D}: 1~~
- ~~{C, E}: 1~~
- ~~{D, E}: 0~~

A-Priori – Frequent Itemset (Beispiel)

- Beispieltupel:
  - {A, B, E}
  - {B, D}
  - {B, C}
  - {A, B, D}
  - {A, C, D}
  - {B, C}
  - {A, C}
  - {A, B, C, E}
  - {A, B, C}

- Zweielementige Frequent Itemsets
  - {A, B}: 4, {A, C}: 4, {A, D}: 2, {A, E}: 2, {B, C}: 4, {B, D}: 2, {B, E}: 2

- Dreielementige Frequent Itemsets
  - {A, B, C}: 2
  - ~~{A, B, D}: 1~~
  - {A, B, E}: 2
  - ~~{A, C, D} 0~~
  - ~~{A, C, E} 0~~
  - ~~{A, D, E} 0~~
  - ~~{B, C, D} 0~~
  - ~~{B, C, E} 0~~
  - ~~{B, D, E} 0~~

A-Priori – Frequent Itemset (Beispiel)

- Beispieltupel:
  - {A, B, E}
  - {B, D}
  - {B, C}
  - {A, B, D}
  - {A, C, D}
  - {B, C}
  - {A, C}
  - {A, B, C, E}
  - {A, B, C}

- Dreielmentige Frequent Itemsets
  - {A, B, C}: 2, {A, B, E}: 2

- Vierelementige Frequent Itemsets
  - ~~{A, B, C, E} 0~~

A-Priori – Assoziationsregeln

- Assoziationsregel-Kandidaten
  - Aufteilen der Frequent Itemsets in Antecedents und Consequents
- Berechnung der Confidence pro Kandidat
  - Erfüllt Kandidat gegebene minimal Confidence → Association Rule, sonst verwerfen
- Modifikation: Andere Evaluierungsmasse
  - Chi-Quadrat-Maß, Informationsgewinn, …

- Gesucht Assoziationsregeln mit mind. 2 Antecedents
  MinConf = 5/9
- Frequent Itemsets mit 3 oder mehr Elementen:
  - {A, B, C}
  - {A, B, E}

- Mögliche Assoziationsregeln
  - ~~A, B ⇒ C; Confidence = 2/4~~
  - ~~A, C ⇒ B; Confidence = 2/4~~
  - ~~B, C ⇒ A; Confidence = 2/4~~
  - ~~A, B ⇒ E; Confidence = 2/4~~
  - A, E ⇒ B; Confidence = 2/2
  - B, E ⇒ A; Confidence = 2/2

Beispieltupel:
{A, B, E}
{B, D}
{B, C}
{A, B, D}
{A, C, D}
{B, C}
{A, C}
{A, B, C, E}
{A, B, C}

A-Priori – Nachteile

- Generate & Test:
  - Schon das Generieren kann teuer sein
  - Eventuell viele falsche Kandidaten (trotz A-Priori!)
  - Überprüfen vieler Kandidaten ist teuer
    - Ein DB-Scan pro Iteration