

UNIVERSITÄT
LEIPZIG

DATA WAREHOUSING

Kapitel 6: Überblick Data Mining/ML-Verfahren

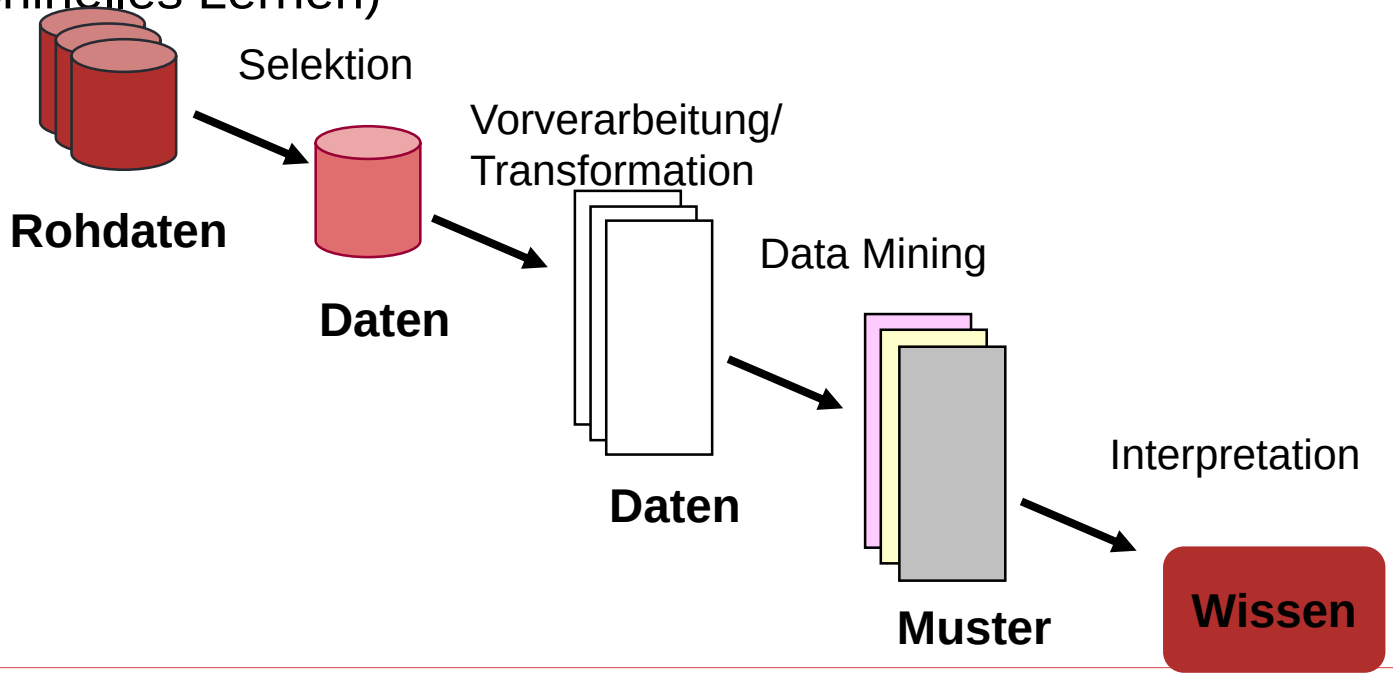
Victor Christen
christen@informatik.uni-leipzig.de
Universität Leipzig
Institut für Informatik
Abteilung Datenbanken

AGENDA

- Einführung Data Mining / maschinelles Lernen
  - KDD-Prozess
  - Anwendungsbeispiele
- Assoziationsregeln / Warenkorbanalyse
  - Support und Konfidenz
  - A Priori-Algorithmus
  - Frequent Pattern (FP)-Trees
- Clusteranalyse
  - k-Means-Algorithmus
  - Canopy Clustering
- Klassifikation
  - Klassifikationsprozess
  - Entscheidungsbäume
  - Neuronale Netze

KNOWLEDGE DISCOVERY IN DATABASES (KDD)

- (semi-)automatische Extraktion von Wissen aus Datenbanken, das
  - gültig (im statistischen Sinn)
  - bisher unbekannt
  - und potenziell nützlich ist
- Kombination von Verfahren zu Datenbanken, Statistik und KI (maschinelles Lernen)

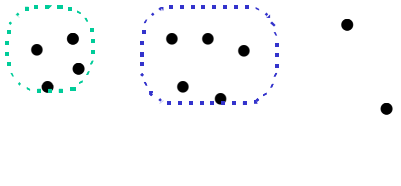


DATA MINING / MACHINE LEARNING

- Data Mining: Anwendung effizienter Algorithmen zur Erkennung von Mustern in großen Datenmengen
- Machine Learning: lernbasierte (KI-) Ansätze für Vorhersagen / Wissensgenerierung
- Data Mining/ML für Big Data / Datenbanken bzw. Data Warehouses
  - Skalierbarkeit auf große Datenmengen (nicht nur im Hauptspeicher)
  - parallele Realisierungen
- umfassende Datenaufbereitung und Vorverarbeitung erforderlich
  - Diskretisierung numerischer Attribute (Aufteilung von Wertebereichen in Intervalle, z.B. Altersgruppen)
  - Erzeugen abgeleiteter Attribute (z.B. Aggregationen für bestimmte Dimensionen, Umsatzänderungen)
  - Extraktion von analyserelevanten Merkmalen / Features aus unstrukturierten Daten

ANALYSETECHNIKEN

- Assoziationsregeln
  - Warenkorbanalyse (z.B. Kunde kauft A und B => Kunde kauft C)
  - Sonderformen zur Berücksichtigung von Dimensionshierarchien (z.B. Produktgruppen), quantitativen Attributen, zeitlichen Beziehungen (sequence mining)
- Clusteranalyse
  - Objekte werden aufgrund von Ähnlichkeiten in Klassen eingeteilt (Segmentierung)



- Klassifikation
  - Zuordnung von Objekten zu Gruppen/Klassen mit gemeinsamen Eigenschaften bzw. Vorhersage von Attributwerten
  - Verwendung von Stichproben (Trainingsdaten)
  - Ansätze: Entscheidungsbaum-Verfahren, neuronale Netze, statistische Auswertungen (z.B. Maximum Likelihood-Schätzung / Bayes-Schätzer)
- weitere Ansätze:
  - genetische Algorithmen (multivariate Optimierungsprobleme, z.B. Identifikation der besten Bankkunden)
  - Regressionsanalyse zur Vorhersage numerischer Attribute . . .

DATA MINING/ML: ANWENDUNGSBEISPIELE

- Warenkorbanalyse: Produkt-Platzierung im Supermarkt, Preisoptimierung, ...
- Kundensegmentierung für Marketing
  - Gruppierung von Kunden mit ähnlichem Kaufverhalten / ähnlichen Interessen
  - Nutzung für gruppenspezifische Empfehlungen, Product Bundling, ...
- Bestimmung der Kreditwürdigkeit von Kunden
  - elektronische Vergabe von Kreditkarten
  - schnelle Entscheidung über Versicherungsanträge, ...
- Entdeckung wechselbereiter Kunden
- Entdeckung von Kreditkarten-Missbrauch
- Erkennung von Spam-Emails
- Auswahl personalisierter Behandlungsstrategien für Krebspatienten

…

EVALUATION/INTERPRETATION

- Ablauf
  - Präsentation der gefundenen Muster, z.B. über Visualisierungen
  - Bewertung der Muster durch den Benutzer
  - falls schlechte Bewertung: erneute Analyse mit anderen Parametern, anderem Verfahren oder anderen Daten
  - falls gute Bewertung: Integration des gefundenen Wissens in die Wissensbasis / Metadaten und Nutzung für zukünftige KDD-Prozesse
- Bewertung der gefundenen Muster: Interessantheit, Vorhersagekraft
  - sind Muster schon bekannt oder überraschend?
  - wie gut lassen sich mit „Trainingsdaten“ (Stichproben) gefundene Muster auf zukünftige Daten verallgemeinern?
  - Vorhersagekraft wächst mit Größe und Repräsentativität der Stichprobe

ASSOZIATIONSREGELN

- Warenkorbanalyse auf Transaktions-Datenbank
  - Transaktion umfasst alle gemeinsam getätigten Einkäufe, innerhalb eines Dokuments vorkommenden Worte, innerhalb einer Web-Sitzung referenzierten Seiten, ...
- Regeln der Form “Rumpf → Kopf [support, confidence]”
- Beispiele
  - kauft(“Papier”) → kauft(“Toner”) [0.5%, 60%]
  - 80% aller Kunden, die Reifen und Autozubehör kaufen, bringen ihr Auto auch zum Service
- relevante Größen
  - **Support** einer Regel X → Y: Anteil der Transaktionen, in denen alle Objekte X und Y vorkommen
  - **Konfidenz** einer Regel X → Y: Anteil der Transaktionen mit Rumpf-Objekten X, für die Regel erfüllt ist (d.h. für die auch Objekte Y vorliegen)
  - **Interessantheit**: hoher Wahrscheinlichkeitsunterschied für Y gegenüber zufälliger Verteilung

ASSOZIATIONSREGELN (2)

- Aufgabe: Bestimmung aller Assoziationsregeln, deren Support und Konfidenz über bestimmten Grenzwerten liegen
- Gegeben:
  - R eine Menge von Items/Objekten (z.B. Produkte, Webseiten)
  - t eine Transaktion, $t \subseteq R$
  - r eine Menge von Transaktionen
  - $s_{min} \in [0,1]$ die minimale Unterstützung,
  - $conf_{min} \in [0,1]$ die minimale Konfidenz
- Aufgabe: Finde alle Regeln c:

$$X \rightarrow Y \text{ mit } X \subseteq R,\ Y \subseteq R,\ X \cap Y = \{\}$$

$$supp(r,c) = \frac{\left|\left\{t \in r \middle| X \cup Y \in t\right\}\right|}{|r|} \geq s_{\min} \qquad conf(r,c) = \frac{\left|\left\{t \in r \middle| X \cup Y \in t\right\}\right|}{\left|\left\{t \in r \middle| X \in t\right\}\right|} \geq conf_{\min}$$

BEISPIEL: WARENKORBANALYSE

EinkaufsID	Aftershave	Bier	Chips
1	0	1	1
2	1	1	0
3	0	1	1
4	1	0	1
5	1	1	1

{Aftershave} → {Bier} s = 2/5, conf = 2/3
{Bier} → {Chips} s= 3/5, conf = 3/4
{Aftershave} → {Chips}
{Chips} → {Aftershave}
{Aftershave} → {Bier,Chips} s= conf=

WEITERE BEISPIELE

- Assoziationsregeln:
  - $A \rightarrow C$ s=2/4=0,5
  - c=2/3
  - $C \rightarrow A$ s= 0,5
    c=

TransaktionsID	Items
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

s_{min} = 0,5

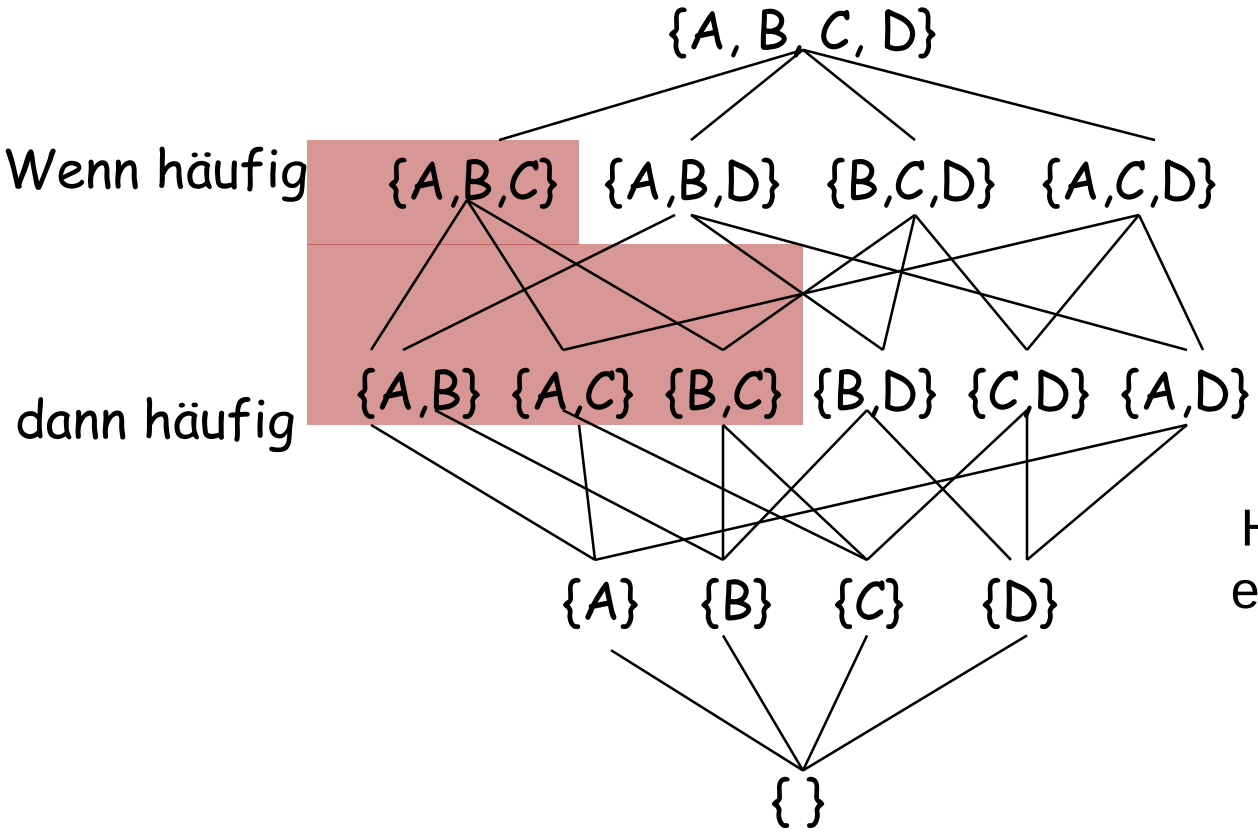
$conf_{min}$= 0,5

- 5 Warenkörbe:
  - Drucker, Papier, PC, Toner
  - PC, Scanner
  - Drucker, Papier, Toner
  - Drucker, PC
  - Drucker, Papier, PC, Scanner, Toner

Assoziationsregel: PC $\rightarrow$ Drucker
s=
c=

FREQUENT ITEMSETS

- Frequent Item-Set (häufige Mengen):
  - Menge von Items/Objekten, deren Support Schranke s_{min} übersteigt
  - Bestimmung der Frequent-Itemsets wesentlicher Schritt zur Bestimmung von Assoziationsregeln
- effiziente Realisierung über A-Priori-Algorithmus
- Nutzung der sog. A-Priori-Eigenschaft:
  - wenn eine Menge häufig ist, so auch all ihre Teilmengen (Anti-Monotonie)
  - wenn eine Menge selten ist, so auch all ihre Obermengen (Monotonie)
- Support jeder Teilmenge und damit jedes einzelnen Items muss über Schranke s_{min} liegen
- iterative Realisierung beginnend mit 1-elementigen Itemsets
  - schrittweise Auswertung von k-Itemsets (k Elemente, k >=1),
  - Ausklammern von Kombinationen mit Teilmengen, die Support s_{min} nicht erreichen („Pruning“)
  - wird „A Priori“ getestet, bevor Support bestimmt wird



Häufige Mengen L_k ergeben Kandidaten C_{k+1}

A-PRIORI-ALGORITHMUS

```
Apriori(R, r, s_min, conf_min)
    L:= Häufige-Mengen(R, r, s_min)
    c:= Regeln (L, conf_min)
    Return c
```

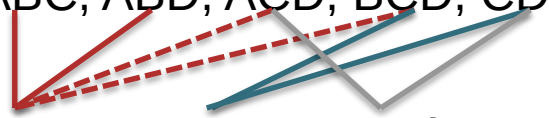
```
Häufige-Mengen(R, r, s_min)
    k=1
    L1:= {i ∈ R, supp(i)>= s_min}//häufige Items
    while L_k ≠ {}
        C_k+1 := Erzeuge-Kandidaten(Lk)
        L_k+1 := Prune(C_k+1, r) // eliminiere Itemsets, die s_min nicht
        erreichen
        k:= k+1
    Return ⋃_{j=2}^{k} L_j
```

A-PRIORI-ALGORITHMUS (2)

```
Erzeuge-Kandidaten(Lk)
    Ck+1 := {}
    Forall l1, l2 in Lk,
     so dass l1 = {i1,...,ik-1 , ik}
              l2 = {i1,...,ik-1,i'k} ik < i'k
       l := {i1, ..., ik-1 , ik, i'k}// sortierte Items pro
    Itemset
       if (alle k-elementigen Teilmengen von l in Lk) then
          Ck+1 := Ck+1  ∪ {l}
    Return Ck+1
```

Beispiel:

L_3 ={ABC, ABD, ACD, BCD, CDE}



L_4 ={ABCD, BCDE, ACDE}

A-PRIORI-ALGORITHMUS: BEISPIEL

- 5 PC-Warenkörbe (Forderung: minimaler Support s_{min} 60%)
  - Drucker, Papier, PC, Toner
  - PC, Scanner
  - Drucker, Papier, Toner
  - Drucker, PC
  - Drucker, Papier, PC, Scanner, Toner
- Algorithmus-Ausführung
  - k=1: Drucker 4, Papier 3, PC 4, Toner 3, Scanner 2
  - L1= C1 = {Drucker, Papier, PC, Toner}
  - k=2: Drucker-Papier: 3, Drucker-PC: 3, Drucker-Toner: 3, Papier-PC: 2, Papier-Toner: 3, PC-Toner: 2
  - L2= {Drucker-Papier, Drucker-PC, Drucker-Toner, Papier-Toner}
  - k=3: Drucker-Papier-PC? Drucker-Papier-Toner? Drucker-PC-Toner?
  - k=4:

REGELGENERIERUNG

- Itemset I = $\{i_1, ..., i_{k-1}, i_k\}$ erlaubt viele Regeln der Form $X \rightarrow Y$, mit $X \cup Y = I$, $X \cap Y = \{\}$
- Konfidenz: conf $(X \rightarrow Y)$= supp (I) / supp (X)
- Generierung der Assoziationsregeln aus Itemset I
  - wenn Konklusion (rechte Seite) länger und somit die linke Seite kürzer wird, kann Konfidenz sinken
  - die Ordnung der Attribute kann ausgenutzt werden
  - c1 = $\{i_1, ..., i_{k-1}\} \rightarrow \{i_k\}$ conf 1
  - c2 = $\{i_1, ...\} \rightarrow \{i_{k-1}, i_k\}$ conf 2 ...
  - ck = $\{i_1\} \rightarrow \{..., i_{k-1}, i_k\}$ conf k
- Es gilt dann: conf 1 $\geq$ conf 2 $\geq ... \geq$ conf k
  - Elimination aller Kombinationen, deren Konfidenz Minimalwert unterschreitet

REGELGENERIERUNG: BEISPIEL

- Regeln für ({Drucker, Papier, Toner}, minimaler Support s_{min} 60%
- Bestimmung aller Regen mit minimaler Konfidenz $conf_{min}$ 75%

  - conf ({Drucker, Papier} →{Toner}) = supp(Dr.,Pap.,Toner)/supp(Dr.,Pap.)= 3/3=1
  - conf ({Drucker, Toner} →{Papier}) = 3/3 = 1
  - conf ({Papier, Toner} →{Drucker}) = 3/3= 1
  - conf ({Drucker} →{Papier, Toner}) =
  - conf ({Papier} →{Drucker, Toner}) =
  - conf ({Toner} →{Drucker, Papier}) =

ASSOZIATIONSREGELN: WEITERE ASPEKTE

- Nutzbarkeit u.a. für Cross-Selling, Produkt-Platzierung ...
  - Amazon: Kunden die dieses Buch gekauft haben, kauften auch ...
- Sonderfall: Sequenzanalyse (Erkennung sequenzieller Muster)
  - Berücksichtigung der Zeit-Dimension
  - Bsp. 1: in 30% der Fälle, wenn Produkt A gekauft wurde, wird bei einem späteren Besuch Produkt B gekauft
  - Bsp. 2: in 20% aller Fälle, in denen ein Nutzer über Werbung auf die Web-Site gelangt und die Site vorher schon besucht hat, kauft er einen Artikel (betrifft 10% der Sessions)
- Probleme
  - sehr viele Produkte / Web-Seiten / Werbeaktionen / Besucher etc. erfordern Bildung gröberer Bezugseinheiten
  - es können sinnlose Korrelationen ohne kausalen Zusammenhang ermittelt werden
    - z.B. Schmutzeffekte aufgrund transitiver Abhängigkeiten (Bsp.: Haarlänge korreliert negativ mit Körpergröße)

Data Warehousing

Assoziationsregeln

Zeit zum Abstimmen: 0:20

Wählen Sie alle richtigen Antwortmöglichkeiten aus:

- ☐ **Support (X -> Y) >= Support (X)**
- ☐ **Wenn Itemset X häufig ist, dann ist dies auch jedes Item in X**
- ☐ **Konfidenz(X -> Y) = Support (X) / Support (X -> Y)**
- ☐ **Konfidenz (XY -> Z) >= Konfidenz (X -> YZ)**

Abstimmen!

FREQUENT PATTERN-BAUM (FP-BAUM)

- Probleme des A-Priori-Algorithmus
  - exponenziell wachsende Anzahl zu prüfender Kandidaten
  - Bsp.: 104 häufige Objekte;
    > 107 2-Itemsets; ca. 1030 Kandidaten für 100-Itemsets
- FP-Tree: Berechnung von Assoziationsregeln ohne Kandidatengenerierung
  - komprimierte Repräsentation aller Transaktionen durch Frequent-Pattern Tree (FP-Baum)
  - effiziente Erkennung von Frequent Itemsets (Pattern Mining) mit Divide-and-Conquer-Suche auf Teilbäumen
  - oft eine Größenordnung schneller als A-Priori

FP-BAUM: GENERIERUNG (1)

- Input: Transaktionsdatenbank D, Schwellwert min-support s_{min}
- Erstellung Objekttabelle / FP-Baum
  - Scan von D, Erstellen der Menge F häufiger Objekte und ihrer Häufigkeiten, Ordnen von F in absteigender Häufigkeit (Objekttabelle)
- Wurzel des FP-Baums ist leere Menge
  - pro Transaktion in D: ordne Objekte gemäß F (absteigende Häufigkeit); füge Pfad in FP-Baum ein (Knotenformat: Objekt-Id, Zähler für Verwendungshäufigkeit in Transaktionen mit gleichem Präfix)
  - Knoten mit gleicher Objekt-ID werden untereinander verkettet (ausgehend von Objekttabelle F)

min_support = 0.5

TID	*Objekte*
10	Drucker, Papier, PC, Toner
20	PC, Scanner
30	Drucker, Papier, Toner
40	Drucker, PC
50	Drucker, Papier, PC,Scanner, Toner

Objekttabelle F

Objekt	*count*
Drucker	*4*
PC	*4*
Papier	*3*
Toner	*3*

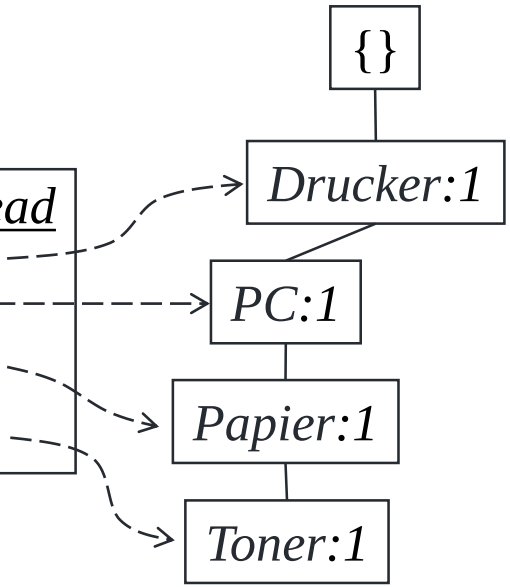
FP-BAUM: GENERIERUNG (2)

TID	*Objekte*	*gemäß F-Ordnung*
10	Drucker, Papier, PC, Toner	Drucker, PC, Papier, Toner
20	PC, Scanner	PC
30	Drucker, Papier, Toner	Drucker, Papier, Toner
40	Drucker, PC	Drucker, PC
50	Drucker, Papier, PC, Scanner, Toner	Drucker,

min_support = 0.5

Objekt-Tabelle F

Objekt	*count*	*Head*
Drucker	*4*	
PC	*4*	
Papier	*3*	
Toner	*3*	



pro Transaktion gibt es Pfad, wobei gemeinsame Präfixe nur einmal repräsentiert werden (Komprimierungseffekt)

FP-BAUM: ERKENNUNG HÄUFIGER MENGEN

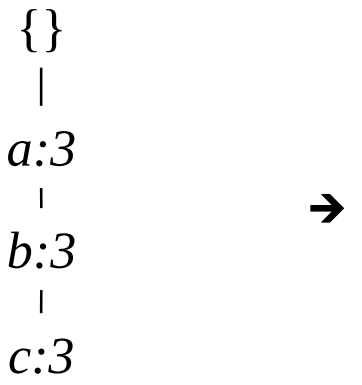
- Erkennung von Frequent Itemsets durch rekursive Suche auf Teilbäumen (Divide and Conquer)
- für jeden Knoten im FP-Baum erzeuge Musterbasis (conditional pattern base)
  - gehe Objekt-Tabelle von unten (selten) nach oben durch. Die Verweise führen zu den Pfaden, in denen das Objekt vorkommt.
  - das Objekt wird als Suffix betrachtet und alle Präfixe davon als Bedingungen für dieses Suffix. Die Präfixpfade eines Suffixes bilden seine Musterbasis

FP-BAUM: ERKENNUNG HÄUFIGER MENGEN

- für jede Musterbasis erzeuge reduzierten FP-Baum (conditional FP tree)
  - Häufigkeiten der Präfixe (für das betrachtete Suffix) werden von unten nach oben propagiert.
  - Gleiche Präfixpfade zu einem Suffix (vom Anfang bis zu einer bestimmten Stelle) werden zusammengelegt und die ursprünglichen Häufigkeiten addiert.
  - nur Präfixpfade, die min_support erfüllen, verbleiben im reduzierten FP-Baum
- rekursives Ableiten von Frequent Itemsets aus reduzierten FP-Trees
  - bei einzigem Pfad im Baum: Aufzählen aller Knoten-Kombinationen

REDUZIERTE FP-BÄUME MIT EINEM PFAD

- bei nur einem Pfad T kann weitere (rekursive) Auswertung entfallen
- Bestimmung der Frequent Itemsets durch Aufzählung aller Kombinationen von Knoten / Teil-Pfaden



reduzierter FP-Baum
für Suffix d (d-reduzierter FP-Baum)

➔

Frequent Itemsets bezüglich *d*

d,

ad, *bd*, *cd*,

abd, *acd*, *bcd*,

abcd

ALGORITHMUS *FP-GROWTH (TREE, I)*

–Finden häufiger Muster durch sukzessives Wachstum von Pattern-Fragmenten initialer Aufruf mit `FP-Growth (FP-Baum, {})`

Input: FP-Baum *tree*, Frequent Itemsets *I*

Output: vollständige Menge häufiger Itemsets

if *tree* hat nur einen Pfad *P* **then**

    foreach Kombination *K* von Knoten in *P* **do**

        return $I \cup K$ mit support = minimaler Support der Items in *K*

else

    foreach Item *i* in Tabelle F (in umgekehrter Häufigkeitsreihenfolge) **do**

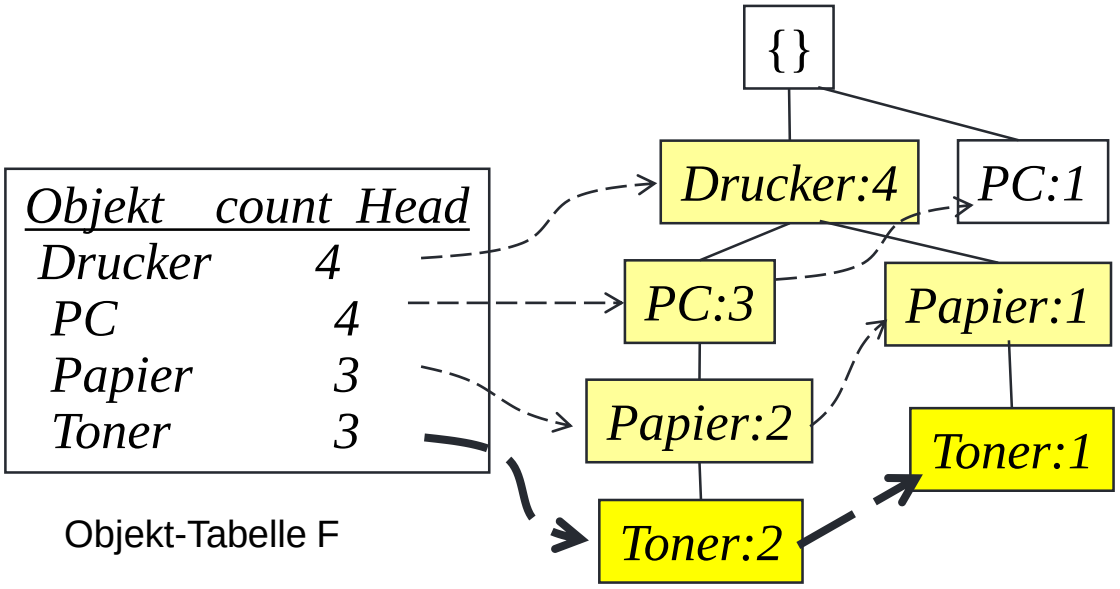
        $K = I \cup i$ mit support = Support von i;

        Erstelle Musterbasis von K und reduzierten FP-Baum FP_K ;

        if FP_K *nicht leer* **then** FP-Growth (FP_K ,K) **end**

SCHRITT 1: MUSTERBASIS-ERSTELLUNG

- Gehe Objekt-Tabelle von unten (selten) nach oben durch. Die Verweise führen zu den Pfaden, in denen das Objekt vorkommt.
- Objekt wird als Suffix betrachtet und alle Präfixe davon als Bedingungen für dieses Suffix. Die transformierten Präfixpfade eines Suffixes bilden seine Musterbasis



Objekt	count	Head
Drucker	4	
PC	4	
Papier	3	
Toner	3	

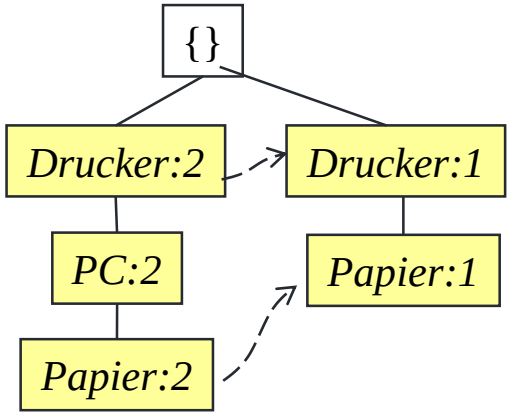
Objekt-Tabelle F

Objekt	Musterbasis
Toner	Drucker,PC,Papier:2, Drucker, Papier:1
Papier	Drucker, PC: 2 Drucker: 1
PC	
Drucker	

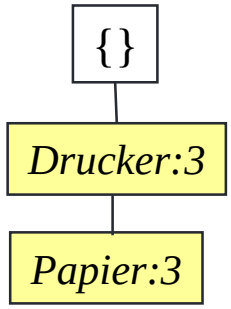
SCHRITT 2: ERZEUGE REDUZIERTE FP-BÄUME

- propagiere Counts pro Objekt der Musterbasis von unten nach oben.
- nur Knoten/Präfix-Pfade, die min_support erfüllen, verbleiben im reduzierten FP-Baum

Musterbasis für Toner



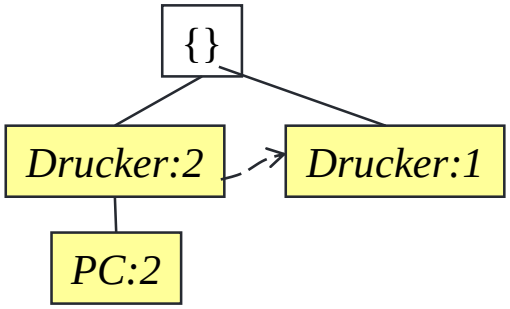
***reduzierter** FP-Baum für Toner*



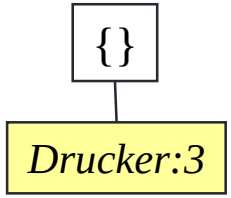
Frequent Itemsets:

Toner
Papier, Toner
Drucker, Toner
Drucker, Papier, Toner

Musterbasis für Papier



reduzierter FP-Baum für Papier



Frequent Itemsets:

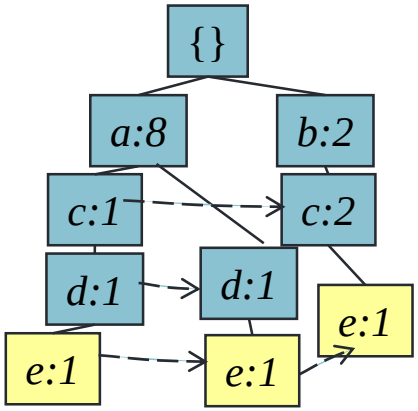
Papier
Drucker, Papier

ERGEBNISSE IM BEISPIEL

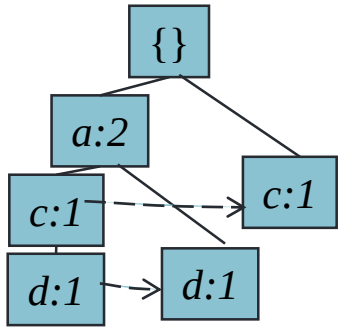
Objekt	Musterbasis	reduzierter FP-Baum	Frequent Itemsets
Toner	{(Drucker,PC,Papier:2, Drucker, Papier:1}	{ (Drucker:3, Papier:3)} | Toner	Toner Papier, Toner Drucker, Toner Drucker, Papier, Toner
Papier	{(Drucker,PC:2, Drucker:1}	{ (Drucker:3)} | Papier	Papier Drucker, Papier
PC	{(Drucker:3) }	{ (Drucker:3)} | PC	PC Drucker, PC
Drucker	{}	{ }	Drucker

ALTERNATIVBEISPIEL ZUR REKURSIVEN AUSWERTUNG

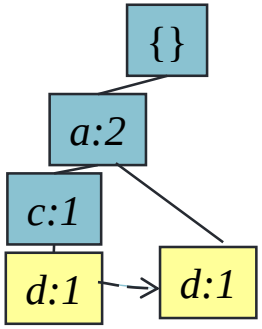
- reduzierter FP-Baum bezüglich Suffixobjekt e, min-support=2



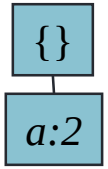
***e**-reduzierter FP-Baum*



Nutzung dieses Baums zur Bestimmung der Frequent Itemsets für Suffixe *de*, *ce* und *ae*



Pfade mit Suffix *de*



***de**-reduzierter FP-Baum*

-> Frequent Itemsets: ***de***, ***ade***

Quelle: Han/Kamber



- Einführung Data Mining / maschinelles Lernen
  - KDD-Prozess
  - Anwendungsbeispiele
- Assoziationsregeln / Warenkorbanalyse
  - Support und Konfidenz
  - A Priori-Algorithmus
  - Frequent Pattern (FP)-Trees
- **Clusteranalyse**
  - k-Means-Algorithmus
  - Canopy Clustering
- **Klassifikation**
  - Klassifikationsprozess
  - Entscheidungsbäume
  - Neuronale Netze

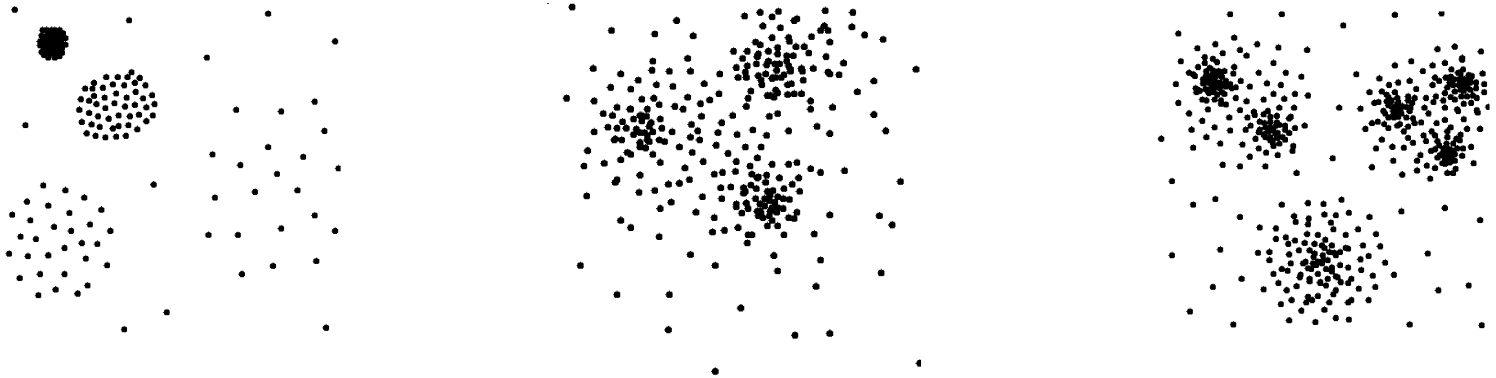
CLUSTERANALYSE

- Ziele
- automatische Identifikation einer endlichen Menge von Kategorien, Klassen oder Gruppen (Cluster) in den Daten
- Objekte im gleichen Cluster sollen möglichst ähnlich sein
- Objekte aus verschiedenen Clustern sollen möglichst unähnlich zueinander sein

$$dist(x,y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

CLUSTERANALYSE

- Ähnlichkeitsbestimmung
- meist: Distanzfunktion dist(o1,o2) für Paare von Objekten o1 und o2
  - z.B. Euklidische Distanz für numerische Attribute: $dist(x,y) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$
  - spezielle Funktionen für kategorische Attribute oder Textdokumente
- Clustering-Ansätze: hierarchisch, partitionierend, dichtebasiert, ...



CLUSTERANALYSE (2)

- oft nicht 2, sondern 10 oder 10.000 Dimensionen
  - fast alle Objektpaare sind ähnlich weit voneinander entfernt
  - Methoden zur Dimensionsreduzierung erforderlich

- Beispiel: ähnliche Musik-Alben
  - einige wenige Kategorien vs. käuferspezifische Präferenzen
  - jeder Musikkunde bildet potenziell eigene Dimension
  - ähnliche Alben haben ähnliche Kunden und umgekehrt

- ähnliche Text-Dokumente / News-Meldungen
  - ähnliche Mengen von Keywords

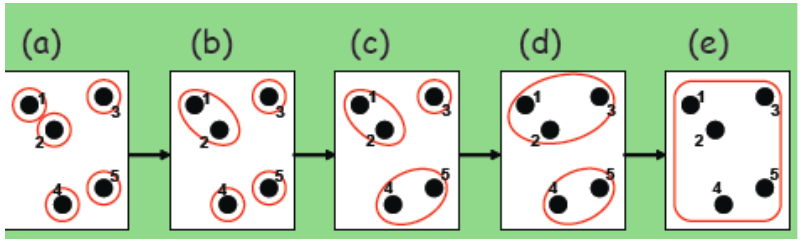
HIERARCHISCHES CLUSTERING

Top-Down (aufteilend)

- beginne mit 1 Cluster
- rekursives Splitten in Teil-Cluster

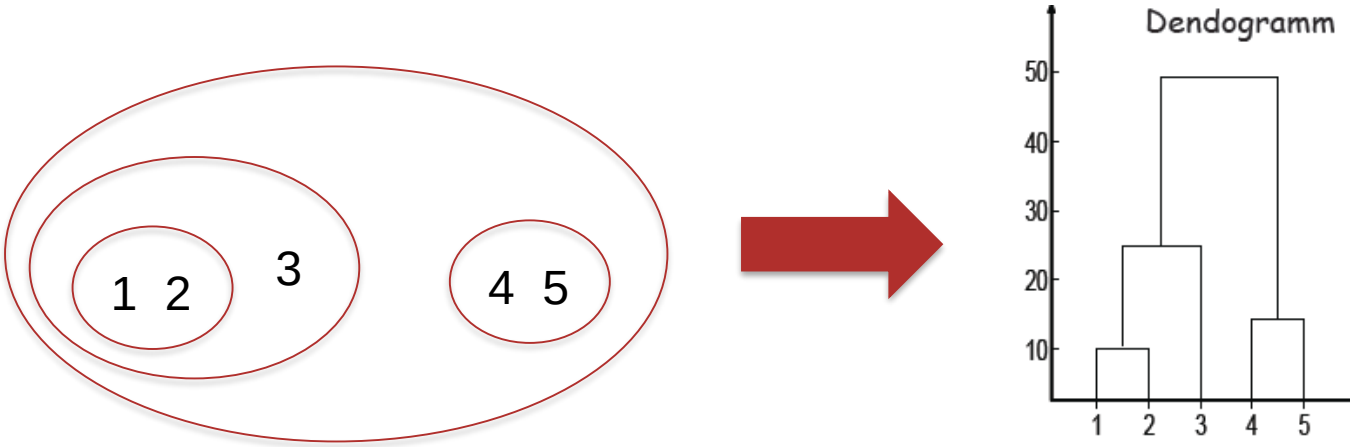
Agglomerativ (Bottom-Up-Ansatz)

- zunächst ist jeder Punkt eigenes Cluster
- wiederholte Vereinigung der “ähnlichsten” Cluster
- Kernproblem: Bestimmung der Merge-Kandidaten
- naiv: kubische Komplexität



HIERARCHISCHES CLUSTERING

– Repräsentation durch Dendogramme



Bestimmung von Cluster-Repräsentanten

– z.B. Cendroid (bei Euklidischer Distanz)

– „Clustroid“: ex. Punkt, der am nächsten zu allen anderen im Cluster liegt

K-MEANS ALGORITHMUS

Ausgangssituation

- Objekte besitzen Distanzfunktion (meist Euklidische Distanz)
- für jedes Cluster wird damit Clusterzentrum bestimmt („Mittelwert“)
- Anzahl k der Cluster wird vorgegeben

Basis-Algorithmus

Schritt 1 (Initialisierung): k Clusterzentren werden (zufällig) gewählt

Schritt 2 (Zuordnung): Jedes Objekt wird dem nächst gelegenen Clusterzentrum zugeordnet

Schritt 3 (Clusterzentren): Für jedes Cluster wird Clusterzentrum neu berechnet

Schritt 4 (Wiederholung): Abbruch, wenn sich Zuordnung nicht mehr ändert, sonst zu Schritt 2

Probleme: Konvergenz zu lokalem Minimum, d.h. Clustering muss nicht optimal sein; relativ hoher Aufwand für Abstandsberechnungen

K-MEANS ALGORITHMUS: BEISPIEL

– Clustering der Zahlen 1, 3, 6, 14, 17, 24, 26, 31 in 3 Cluster (k=3)

(1) Zentren: 10, 21, 29 (zufällig gewählt, müssen nicht existieren)

(2) Cluster: C1={1, 3, 6, 14}, C2={17, 24}, C3={26, 31}

(3) Zentren (arithmetisches Mittel): C1: 6 C2: 20,5 C3: 28,5

(2) Cluster: C1={1, 3, 6}, C2={14, 17, 24}, C3={26, 31}

(3) Zentren: C1: 3,3 C2: 18,3 C3: 28,5

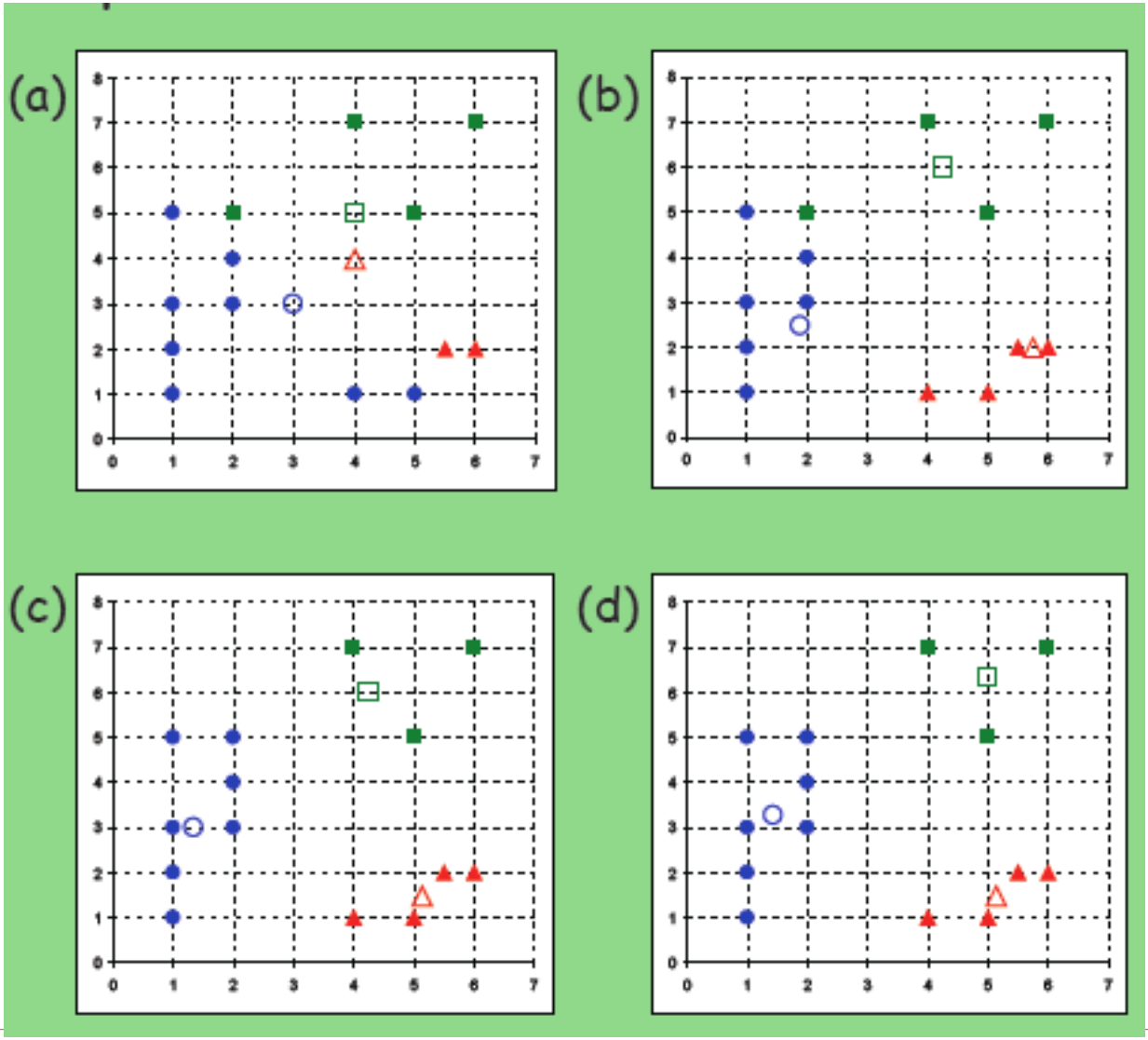
(2) Cluster: C1={1, 3, 6}, C2={14, 17}, C3={24, 26, 31}

(3) Zentren: C1: 3,3 C2: 15,5 C3: 27

(2) Cluster: C1={1, 3, 6}, C2={14, 17}, C3={24, 26, 31}

Abbruch, da sich das Clustering nicht mehr geändert hat.

K-MEANS: 2-D-BEISPIEL (K=3)



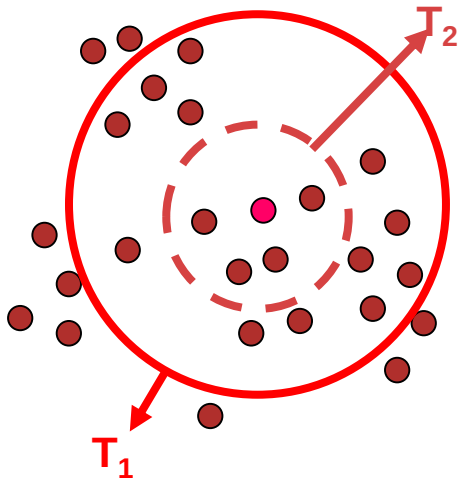
CANOPY CLUSTERING*

Bildung von überlappenden Clustern (Canopies)

- oft Nutzung als erster Schritt in mehrstufigem Analyseverfahren (Pre-Clustering)
- skalierbar auf sehr große Datenmengen
- auf String-Daten einsetzbar (mit String-Ähnlichkeits/Distanzfunktionen)
- Clusteranzahl nicht vorzugeben

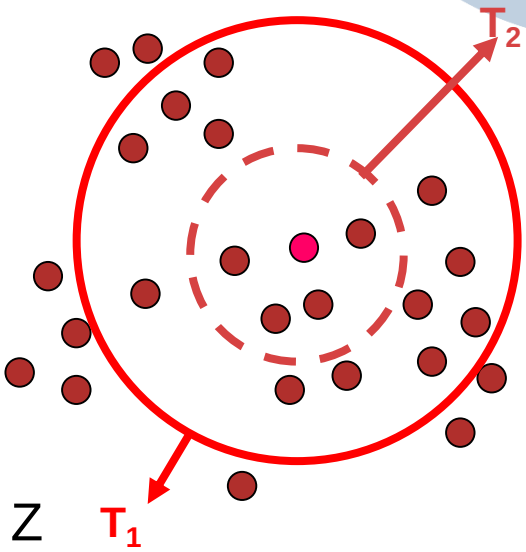
Eingabe

- Punktmenge
- Distanzfunktion
- 2 Schwellwerte T1 und T2 (T1 > T2)



* *McCallum, A. et al.:**Efficient clustering of high-dimensional data sets with application to reference matching** Proc. ACM KDD, 2000*

CANOPY CLUSTERING (2)



Algorithmus

Schritt 1 (Initialisierung): Kandidatenliste für Wahl der Canopy-Zentren wird mit allen Objekten initialisiert

Schritt 2 (Canopy-Zentrum): Zentrum Z wird (zufällig) aus Kandidatenliste gewählt

Schritt 3 (Zuordnung): alle Objekte, deren Abstand zu Z geringer ist als Schwellwert T_1, werden (Canopy) Z zugeordnet

Schritt 4 (Kandidatenliste): alle Objekte, die innerhalb des Schwellwertes T_2 zu Z liegen, werden aus Kandidatenliste gelöscht

Schritt 5 (Ende/Wiederholung): Abbruch, wenn Kandidatenliste leer ist, sonst zu Schritt 2

CANOPY CLUSTERING ALGORITHMUS: BEISPIEL

Clustering der Zahlen 1, 2, 3, 8, 9, 15

Abstandsfunktion absolute Differenz, $T_1 = 6$, $T_2 = 4$

(1) Kandidatenliste = {1, 2, 3, 8, 9, 15}
(2) Canopyzentrum sei 1
(3) Canopy bilden
(4) Entferne aus Kandidatenliste

(5) Kandidatenliste = { }
(2) Canopyzentrum:
(3) Canopy bilden
(4) Entferne aus Kandidatenliste

(5) Kandidatenliste = { }
(2) Canopyzentrum:
(3) Canopy bilden
(4) Entferne aus Kandidatenliste

(5) Abbruch, da Kandidatenliste leer

KLASSIFIKATION

Klassifikationsproblem

- gegeben sei Stichprobe (Trainingsmenge) O von Objekten des Formats (a1, . . ., ad) mit Attributen A_i, $1 \leq i \leq d$, und Klassenzugehörigkeit c_i, $c_i \in C = \{c_1, \ldots, c_k\}$
- gesucht: die Klassenzugehörigkeit für Objekte aus D \ O
  d.h. ein Klassifikator $K : D \rightarrow C$

weiteres Ziel:

- Generierung (Lernen) des expliziten Klassifikationswissens (Klassifikationsmodell, z.B. Klassifikationsregeln oder Entscheidungsbaum)

Abgrenzung zum Clustering

- Klassifikation: Klassen vorab bekannt (überwachte Verfahren, „supervised“)
- Clustering: Klassen werden erst gesucht („unsupervised“)

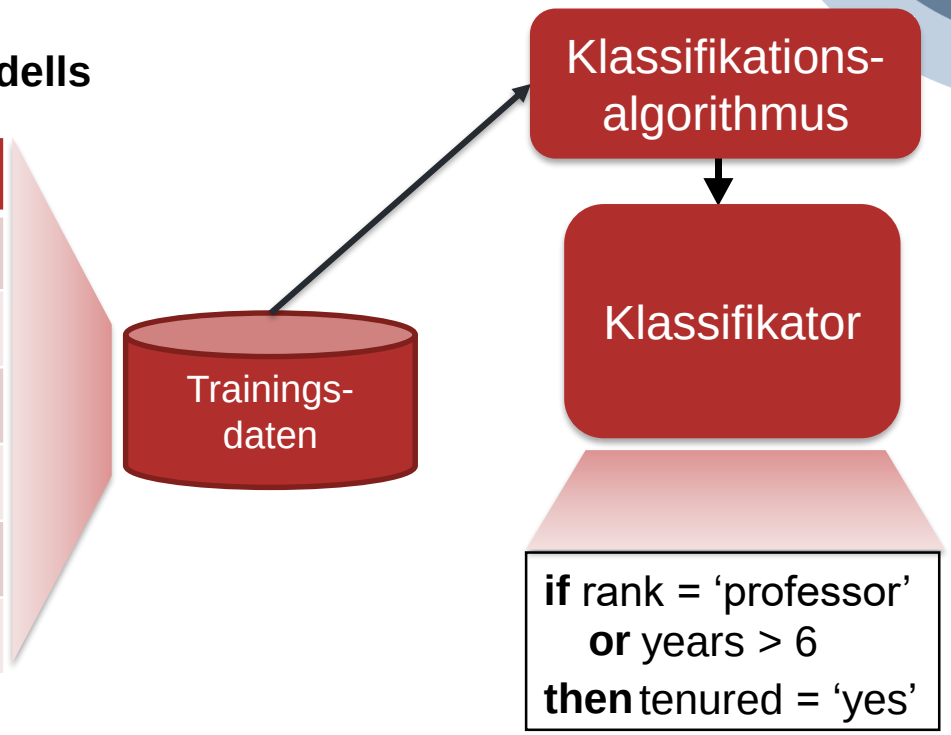
Klassifikationsansätze

- Entscheidungsbaum-Klassifikatoren, Bayes-Klassifikatoren (Auswertung der bedingten Wahrscheinlichkeiten von Attributwerten)
- neuronale Netze

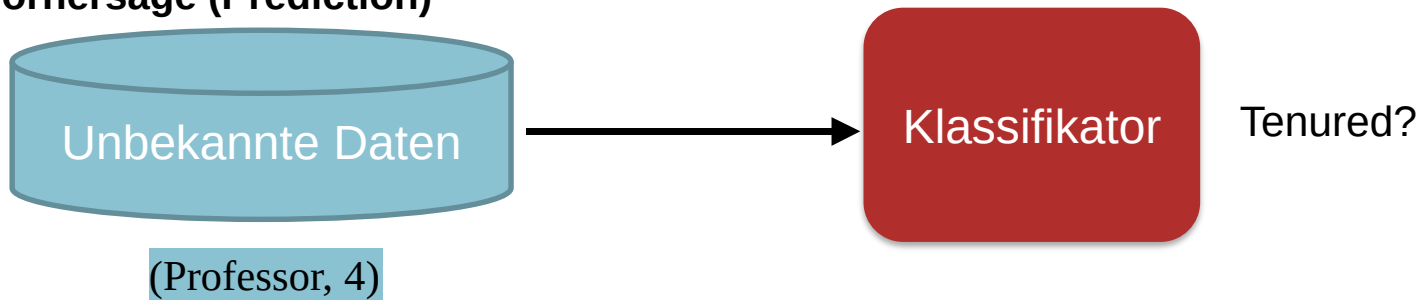
KLASSIFIKATIONSPROZESS

1. Konstruktion des Klassifikationsmodells

Rank	Years	Tenured
Assistant Prof	3	No
Assistant Prof	7	Yes
Professor	2	Yes
Associate Prof	7	Yes
Assistant Prof	6	No
Associate Prof	3	No



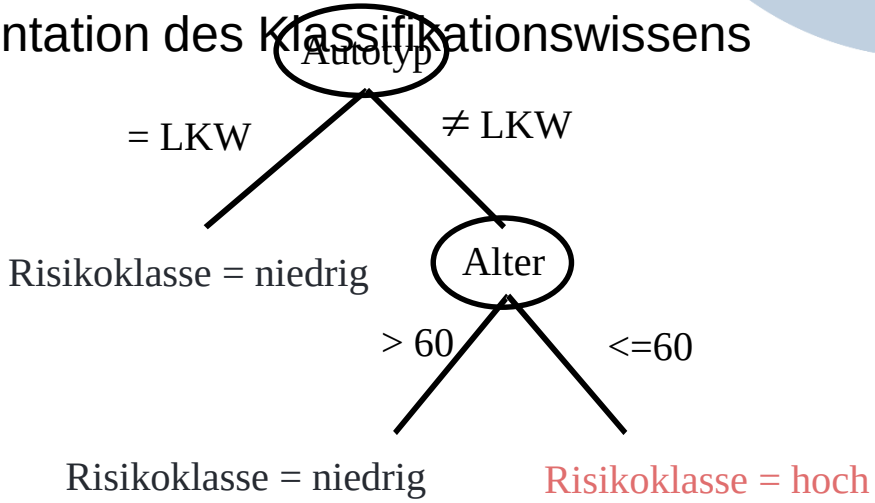
2. Anwendung des Modells zur Vorhersage (Prediction)



ENTSCHEIDUNGSBÄUME

– explizite, leicht verständliche Repräsentation des Klassifikationswissens

ID	Alter	Autotyp	Risiko
1	23	Familie	hoch
2	17	Sport	hoch
3	43	Sport	hoch
4	68	Familie	niedrig
5	32	LKW	niedrig
	40	Familie	



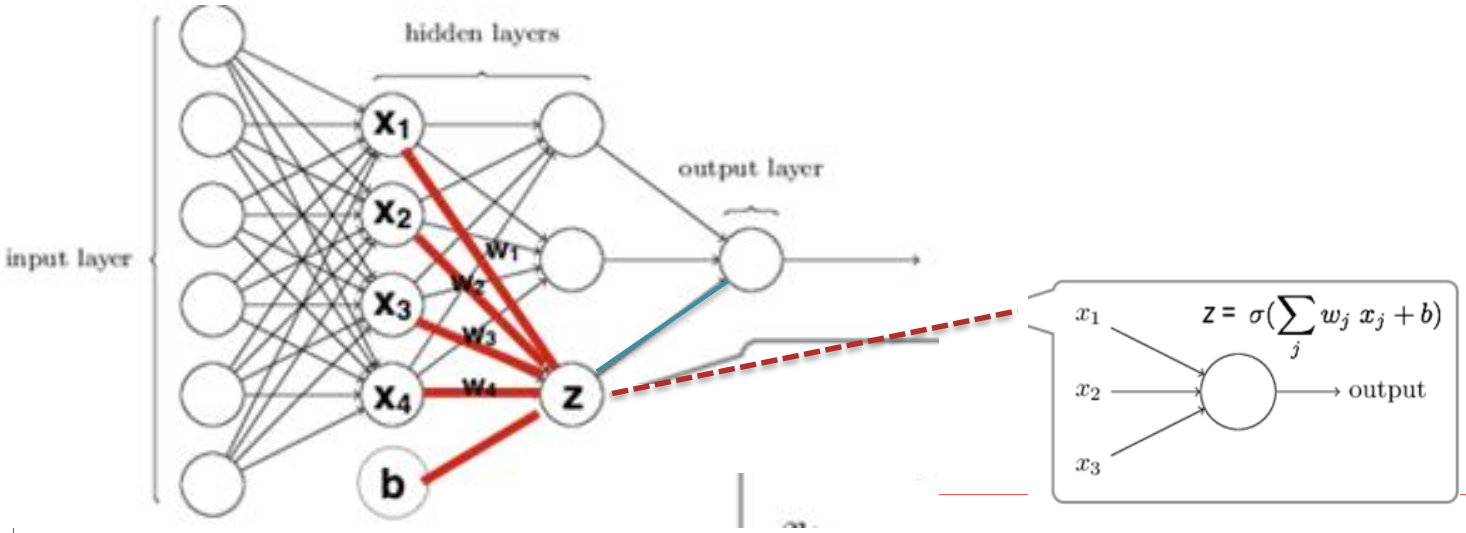
- Entscheidungsbaum ist Baum mit folgenden Eigenschaften:
  - ein innerer Knoten repräsentiert ein Attribut
  - eine Kante repräsentiert einen Test auf dem Attribut des Vorgängerknotens
  - ein Blatt repräsentiert eine der Klassen
- jeder Pfad von Wurzel zu Blatt entspricht einer Klassifikationsregel
  - Bsp: Autotyp ≠ LKW AND Alter > 60 -> Risiko=niedrig
- Anwendung zur Vorhersage:
  - Top-Down-Durchlauf des Entscheidungsbaums von der Wurzel zu einem der Blätter
  - eindeutige Zuordnung des Objekts zur Klasse des erreichten Blatts

KONSTRUKTION EINES ENTSCHEIDUNGSBAUMS

- Basis-Algorithmus (divide and conquer)
  - Anfangs gehören alle Trainingsdatensätze zur Wurzel
  - Auswahl des nächsten Attributs (Split-Strategie): Maximierung des Informationsgewinns (meßbar über Entropie o.ä.)
  - Partitionierung der Trainingsdatensätze mit Split-Attribut
  - Verfahren wird rekursiv für die Partitionen fortgesetzt
- Abbruchbedingungen
  - keine weiteren Split-Attribute
  - alle Trainingsdatensätze eines Knotens gehören zur selben Klasse
- Typen von Splits
  - *kategorische Attribute*: Split-Bedingungen der Form „attribut = a“ oder „attribut $\in$ set“ (viele mögliche Teilmengen)
  - *numerische Attribute*: Split-Bedingungen der Form „attribut < a“ (viele mögliche Split-Punkte)

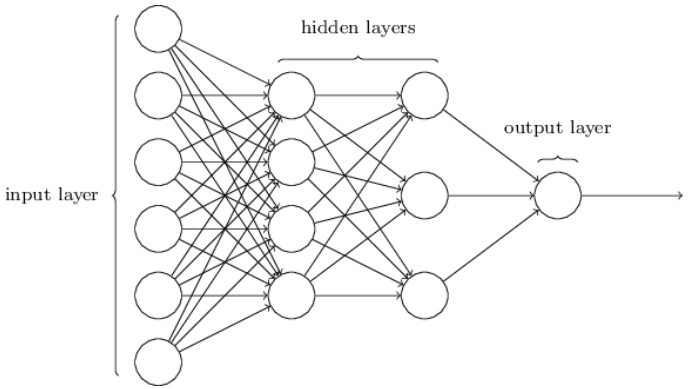
NEURONALE NETZE

- neuronales Netz (NN) besteht aus mehreren Schichten
  - Eingabe-/Ausgabeschicht
  - mind. einer verdeckten (hidden) Schicht
- jede Schicht besteht aus Neuronen, die mit anderen Neuronen verbunden sind
  - Verbindungen / Kanten verwenden Zahlen, z.B. Gewichte ($\boldsymbol{w}_i \in \mathbb{R}$)
- Deep Learning: mehrere hidden layers



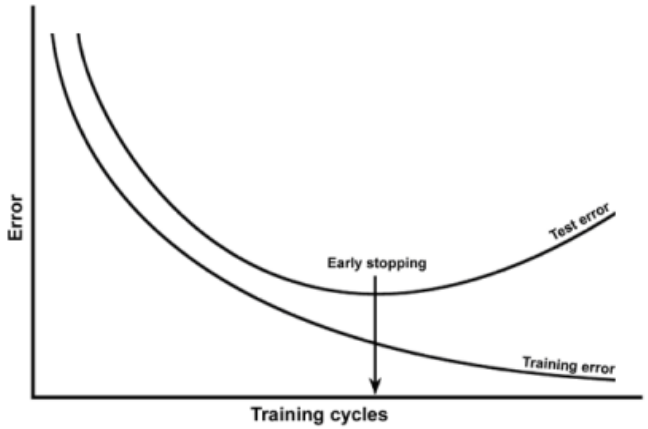
DEEP LEARNING

- Lernen einer Datenrepräsentation(Embeddings) auf großen Mengen an Trainingsdaten
- Nutzen des gelernten Wissens für Klassifikation, Vorhersagen etc. vor allem für unstrukturierte Daten (Bilder, Sprache, Text) mit verborgener Struktur
- zahlreiche Anwendungsfälle
  - Erkennung von Bildern
  - Erkennung von Handschriften
  - Spracherkennung
  - Verarbeitung von Texten …
- verschiedene Varianten von Netzen
  - Autoencoder networks (Erzeugung verbesserter Repräsentationen)
  - Convolutional deep Neural Networks (CNN), v.a. für Bildklassifizierung
  - Recurrent Neural Networks (RNN), u.a. LSTM (Long short-term memory),
    - u.a. für Text/Sprachverarbeitung oft unter Nutzung von Word Embeddings



BEWERTUNG VON KLASSIFIKATOREN

- Klassifikator ist für die Trainingsdaten optimiert
  - liefert auf der Grundgesamtheit der Daten evtl. schlechtere Ergebnisse (-> Overfitting-Problem)
- Bewertung mit von Trainingsmengen unabhängigen Testmengen
- Klassifikationsgenauigkeit/Accuracy:
  - Anteil der korrekten Klassenzuordnungen in Testmenge
- Klassifikationsfehler:
  - Anteil der falschen Klassenzuordnungen



BEWERTUNG (2)

Klassifikationsgenauigkeit (Accuracy)

- n Sätze/Objekte, nr korrekt zugeordnet

$$Accuracy = \frac{nr}{n}$$

- alle Kategorien gehen gleichberechtigt ein
- bei 2 Kategorien (positive/negative) gilt
- $Accuracy = \frac{TP+TN}{n}$
- Accuracy problematisch, falls eine der Kategorien stark dominiert (z.B. non-Matches)

	real: class C	**real:** not class C
predicted: class C	True Positive TP	False Positive FP
predicted: not class C	False Negative FN	True Negative TN

BEWERTUNG (2)

Precision / Recall / F-Measure

- fokussieren auf eine der Kategorien C (z.B. Risiko, Krankheit, Match) in binären Entscheidungsproblemen
- seien nc der n Objekte von dieser Kategorie
- Klassifikationsmodell ordne mc Objekte dieser Klasse zu, davon mcr (TP) richtig
- $Recall = \frac{mcr}{nc}$ $Precision = \frac{mcr}{mc}$

$$F - Measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

- Beispiel binäres Entscheidungsproblem
  - n=100, nc = 20, mc = 25, mcr = 15
  - Recall=
  - Precision= F-Measure=
  - Accuracy=

BEWERTUNG (3)

- Konfusionsmatrix

	A	B	C	*Summe*
a	**37**	12	4	*53*
b	5	**23**	11	*39*
c	12	8	**42**	*62*
Summe	*54*	*43*	*57*	*154*

- Accuracy: (37+23+42)/154=102/154 = 66,2%
- kategoriebezogene Bewertungen
  - Recall für Kat. C: 42/57=0,74
  - Precision für Kat. C: 42/62=0,68

ZUSAMMENFASSUNG (1)

- Data Mining/ML-Verfahrensklassen: Assoziationsregeln, Clusteranalyse, Klassifikation,
- zahlreiche Nutzungsmöglichkeiten: Kundensegmentierung, Vorhersage des Kundenverhaltens, Warenkorbanalyse etc.
- Assoziationsregeln, u.a. zur Warenkorbanalyse
  - Berechnung der Frequent Itemsets mit minimalem Support
  - Ableitung der Regeln mit ausreichender Konfidenz
  - Nutzung der Anti-Monotonie: Itemset ist nur häufig, wenn es alle Teilmengen davon sind
  - A-Priori-Algorithmus: Bottom-Up-Auswertung mit Kandidatengenerierung für häufige Itemsets
  - FP-Baum: schnellere Alternative mit kompakter Repräsentation von Transaktionen und rekursivem Pattern Mining auf Teilbäumen (Divide-and-Conquer)

ZUSAMMENFASSUNG (2)

- Clusteranalyse: Segmentierung über Distanzmaß
  - K-Means: vorgegebene Anzahl von Clustern
  - Canopy Clustering: schnelle Berechnung überlappender Cluster; nützlich als Vorbereitung für K-Means
- Klassifikation, z.B. über Entscheidungsbäume
  - erfordert Trainingsdaten mit bekannter Klassenzuordnung
  - Bestimmung der Split-Attribute eines Entscheidungsbaums gemäß Informationsgewinn
- Bewertung von Klassifikationsergebnissen
  - Accuracy
  - für 2 Klassenprobleme: Precision/Recall/F-Measure
- neuronale Netze / Deep Learning
  - vielseitige Nutzungsmöglichkeiten, u.a. für Klassifikationsprobleme für unstrukturierte Daten (Bilder, Texte)
  - hoher Ressourcenbedarf