



Intelligente Systeme

Einführung

Prof. Dr. Rudolf Kruse Georg Ruß
Christian Moewes

`{kruse,russ,cmoewes}@iws.cs.uni-magdeburg.de`

Arbeitsgruppe Computational Intelligence
Institut für Wissens- und Sprachverarbeitung
Fakultät für Informatik
Otto-von-Guericke Universität Magdeburg



Maschinelles Lernen

(nach [Beierle and Kern-Isberner, 2008], Kapitel 5)

1. Maschinelles Lernen

Definitionen des Lernens

Klassifikation der Ansätze

Erlernen von Entscheidungsbäumen

Data Mining

Definitionen des Lernens (1)

Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time.

[Simon, 1983]

- umfaßt allerdings auch Veränderungen, die nichts mit Lernen zu tun haben
- die Verwendung eines schneller getakteten Prozessors wäre als schnellere Abarbeitung einer arithmetischen Berechnung bereits eine Lernleistung

Definitionen des Lernens (2)

The study and computer modeling of learning processes in their multiple manifestations constitutes the subject matter of machine learning.

[Michalski et al., 1986]

- Es wird direkt auf “Lernprozesse in verschiedenen Ausprägungen” angespielt.

Definitionen des Lernens (3)

Learning is constructing or modifying representations of what is being experienced. [Michalski and Michalski, 1986]

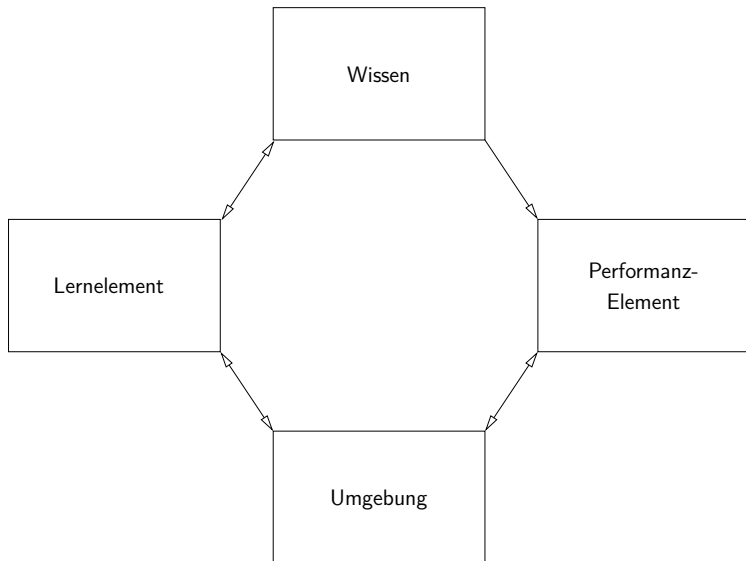
- Die Konstruktion einer Repräsentation wird als zentraler Aspekt gesehen.

Definitionen des Lernens (4)

Research in machine learning has been concerned with building computer programs able to construct new knowledge or to improve already possessed knowledge by using input information. [Michalski and Kodratoff, 1990]

- Ziel des ML: Computerprogramme sollen durch Erfahrung ihr eigenes Handeln verbessern können.

Schema eines allgemeinen Lernmodells



Performanzelement: interagiert mit der Umgebung, wird durch vorhandenes Wissen gesteuert

Lernelement: nimmt Erfahrungen und Beobachtungen aus der Umgebung auf, erzeugt/modifiziert Wissen

Zusätzlich meist:

Kritikelement: teilt dem Lernelement mit, wie erfolgreich es ist

Problemgenerator: erzeugt Aufgaben, die zu neuen und informativen Erfahrungen führen sollen

Klassifikation der Ansätze gemäß [Carbonell et al., 1984]:

- Klassifikation gemäß der zugrundeliegenden *Lernstrategie*: Dabei wird unterschieden, wieviel Information bereits vorgegeben wird und in welchem Maße das Lernsystem eigene Inferenzen durchführt;
- Klassifikation gemäß der benutzten *Repräsentation von Wissen*, welches das System erlernt;
- Klassifikation gemäß dem *Anwendungsbereich* des Lernsystems.

Klassifikation gemäß der benutzten Lernstrategie

- **Direkte Eingabe neuen Wissens und Auswendiglernen:** keinerlei Inferenz oder eine andere Art der Wissenstransformation erforderlich; bspw. Speichern von Daten/Fakten, Lernen durch direkte Programmierung
- **Lernen durch Anweisungen:** aufbereitetes Wissen wird vorgegeben, was intern verarbeitet werden muß; Wissen soll effektiv verwendet werden; Anweisungen werden durch den Lehrenden aufgearbeitet, so daß das Wissen des Lernenden schrittweise erweitert werden kann

Klassifikation gemäß der benutzten Lernstrategie

- **Lernen durch Deduktion:** leitet aus vorhandenem Wissen mittels deduktiver Schlußweisen neues Wissen ab; neues Wissen kann zur Effizienz- oder Effektivitätssteigerung verwendet werden.
- **Lernen durch Analogie:** Erlernen neuer Fakten und Fähigkeiten durch Anpassung vorhandenen Wissens an neue Situationen

Klassifikation gemäß der benutzten Lernstrategie

- **Lernen aus Beispielen:** eine allgemeine Konzeptbeschreibung soll erstellt werden, die alle vorher gegebenen Beispiele umfaßt und evtl. vorhandene Gegenbeispiele ausschließt

Beispiele vom Lehrenden: Konzept ist dem Lehrer bekannt;
Beispiele können entsprechend ausgewählt werden;
schneller Lernerfolg möglich.

Beispiele vom Lernenden: Lernender hat Hypothese für das zu lernende Konzept und generiert Beispiele; von außerhalb kommt Feedback zu den Beispielen (positive oder negative Beispiele).

Beispiele aus der Umgebung: Zufallsbeobachtungen;
Notwendigkeit, dem Lernenden mitzuteilen, ob die Beobachtung ein positives oder ein Gegenbeispiel ist.

Klassifikation gemäß der benutzten Lernstrategie

Alternative Klassifizierung des “Lernens durch Beispiele”:

Nur positive Beispiele verfügbar: keine Informationen darüber verfügbar, ob das abgeleitete Konzept zu allgemein ist; dem wird oft durch Minimalitätskriterien entgegenzuwirken versucht.

Positive und negative Beispiele verfügbar: üblichste Situation beim Lernen; positive Beispiele sorgen dafür, daß das abgeleitete Konzept allgemein genug ist; negative Beispiele verhindern, daß das Konzept zu allgemein wird

Klassifikation gemäß der benutzten Lernstrategie

Weitere alternative Klassifizierung des “Lernens durch Beispiele”:

Alls Beispiele gleichzeitig: alle Informationen stehen in jedem Fall am Anfang zur Verfügung; Hypothesen können sofort auf Richtigkeit überprüft werden.

Beispiele sind inkrementell gegeben: Hypothese in Konsistenz mit den bisherigen Beispielen wird erstellt, die keines der Gegenbeispiele erfaßt; anhand nachfolgender Beispiele wird die Hypothese überprüft und ggf. verfeinert.

Klassifikation gemäß der benutzten Lernstrategie

- **Lernen aus Beobachtungen und durch Entdeckungen:** generelle Ausprägung des induktiven Lernens; keinerlei Steuerung durch Lehrenden; verschiedene Konzepte sind gleichzeitig zu erlernen

Passive Beobachtungen: Konzepte, die aufgrund der Beobachtungen der Umgebung durch den Lernenden entwickelt werden;

Aktive Experimente: Umgebung wird gezielt beeinflusst, um die Auswirkungen der Experimente beobachten zu können; Steuerung der Experimente per Zufall, nach allgemeinen Gesichtspunkten oder durch theoretische Überlegungen.

Klassifikation gemäß dem gelernten Typ von Wissen

Parameter in algebraischen Ausdrücken: gegeben ist ein algebraischer Ausdruck; numerische Parameter oder Koeffizienten sind so zu optimieren, daß ein gewünschtes Verhalten erreicht wird

Entscheidungsbäume: zur Unterscheidung zwischen Elementen einer Klasse; Knoten: Attribute der Objekte; Blätter: Menge der Objekte, die der gleichen Unterklasse zugeordnet werden

Formale Grammatiken: zur Beschreibung einer formalen Sprache; ausgehend von Beispielausdrücken wird eine formale Grammatik erlernt

Regeln: **if C then A**; C ist Menge von Bedingungen, A ist eine Aussage

Vier Basisoperationen für Regeln:

- **Erzeugung:** eine neue Regel wird generiert oder aus der Umgebung aufgenommen
- **Verallgemeinerung:** Bedingungen aus dem Bedingungsteil werden entfernt, Regel wird allgemeiner
- **Spezialisierung:** zusätzliche Bedingungen werden dem Bedingungsteil hinzugefügt, Regel ist nur noch auf speziellere Situationen anwendbar
- **Komposition:** Regeln werden zusammengefaßt; nicht mehr notwendige Bedingungen und Folgerungen werden eliminiert

Klassifikation gemäß dem gelernten Typ von Wissen

Ausdrücke basierend auf formaler Logik: für die Beschreibung einzelner Objekte als auch für die Bildung des zu erlernenden Konzepts; Aussagen, Prädikate, Variablen, logische Ausdrücke

Begriffshierarchien: Begriffe, die in Beziehung zueinander stehen, werden hierarchischen Begriffskategorien zugeordnet; Begriffshierarchien bzw. Taxonomien sind zu lernen

- Hier: vereinfachte Entscheidungsbäume, die nur ja/nein-Knoten beinhalten.
 - Die Blätter sind mit dem Wahrheitswert markiert, der als Ergebnis der Funktion zurückgeliefert werden soll, wenn das Blatt erreicht wird.
 - Die inneren Knoten sind mit einem Attribut markiert. Eine solche Markierung a repräsentiert eine Abfrage, welchen Wert das betrachtete Objekt für das Attribut a hat.
 - Die von einem mit a markierten Knoten ausgehenden Kanten sind mit den zu a möglichen Attributwerten markiert.

Entscheidungsbäume: Beispiel

Beispiel: Kinobesuch, Definition der Attribute

Attraktivität des Films: hoch, mittel, gering

Preis: normal (n), mit Zuschlag (z)

Loge: noch verfügbar (ja), nicht verfügbar (nein)

Wetter: schön, mittel, schlecht

Warten: ja, nein

Besetzung: top, mittel

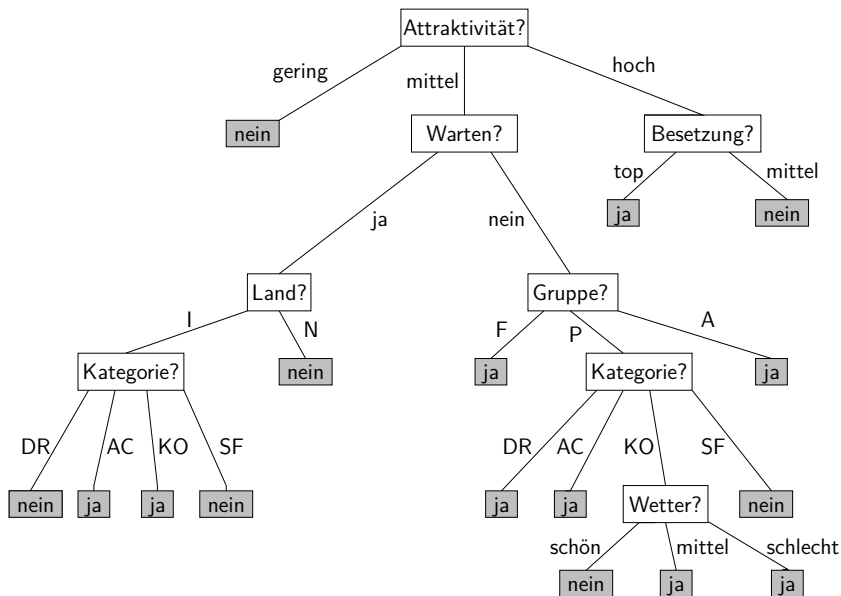
Kategorie: Action (AC), Komödie (KO), Drama (DR), Science Fiction (SF)

Reservierung: ja, nein

Land: national (N), international (I)

Gruppe: Freunde (F), Paar (P), allein (A)

Entscheidungsbäume: Beispiel



Entscheidungsbäume: Regeln

Aus Entscheidungsbäumen können sehr einfach Regeln abgelesen werden:

- **if** Attraktivität = hoch **and** Besetzung = top **then** Kinobesuch = ja.
- **if** Attraktivität = mittel **and** Warten = ja **and** Land = national **then** Kinobesuch = nein.

Ein Lernverfahren für Entscheidungsbäume generiert aus einer Menge von Beispielen, der *Trainingsmenge* einen Entscheidungsbaum. Ein *Trainingsbeispiel* ist dabei eine Menge von Attribut/Wert-Paaren zusammen mit der Klassifikation.

Entscheidungsbäume: Generierung

- Für jedes Beispiel steht am Ende genau ein Pfad im Baum von der Wurzel zum Blattknoten.
- Diese Vorgehensweise liefert keine sinnvolle Generalisierung, der Baum paßt nur auf die vorhandenen Trainingsdaten, aber nicht auf neue Daten.
- **Occam's Razor:** Bevorzuge die einfachste Hypothese, die konsistent mit allen Beobachtungen ist.
- **Problem:** Welches Attribut wird ausgewählt, um in einem Knoten die Beispieldaten aufzuteilen? Welches Attribut ist das *wichtigste*?

Induktion von Entscheidungsbäumen

Wie werden “ziemlich gute” gegen “relativ nutzlose” Attribute abgewogen?

- mittlerer Informationsgehalt $H(P)$ einer Wahrscheinlichkeitsverteilung $P = (p_1, \dots, p_n)$:

$$H(P) = - \sum_{i=1}^n p_i \log_2 p_i$$

- Wahrscheinlichkeit, ein Beispiel e_i aus einer Menge $\{e_1, \dots, e_k\}$ auszuwählen, ist $\frac{1}{k}$
- Wahrscheinlichkeit, aus der Menge $\{e_1, \dots, e_k\}$ eines von j vorgegebenen Beispielen ($j \leq k$) auszuwählen:

$$\sum_{i=1}^j \frac{1}{k} = \frac{j}{k}$$

Induktion von Entscheidungsbäumen

- Wahrscheinlichkeit, aus einer Menge mit p positiven und n negativen Beispielen ein positives auszuwählen

$$\frac{p}{p+n}$$

- Informationsgehalt $I(E)$ der Antwort auf die Frage, ob es sich bei einem beliebigen Beispiel aus E mit p positiven und n negativen Beispielen um ein positives oder negatives Beispiel handelt:

$$I(E) := H\left(\frac{p}{p+n}; \frac{n}{p+n}\right) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n} \text{ bit}$$

Induktion von Entscheidungsbäumen

- Bei der Attributauswahl soll nun berücksichtigt werden, welchen Informationsgewinn man erhält, wenn man den Wert eines Attributs kennt.
- Jedes Attribut a teilt die Trainingsmenge E in Teilmengen E_1, \dots, E_k auf, wobei k die Anzahl der verschiedenen Werte w_1, \dots, w_k ist, die a annehmen kann.
- Jede Teilmenge E_i habe p_i positive und n_i negative Beispiele.
- Sei $a = w_i$ gegeben. Die noch benötigte Menge an Information ist:

$$I(E_i) = H\left(\frac{p_i}{p_i + n_i} + \frac{n_i}{p_i + n_i}\right) \text{bit}$$

Induktion von Entscheidungsbäumen

- mittlerer Informationsgehalt, nachdem Attribut a getestet wurde (*bedingte mittlere Information*):

$$\begin{aligned} I(E|a \text{ bekannt}) &= \sum_{i=1}^k P(a = w_i) \cdot I(E_i) \\ &= \sum_{i=1}^k \frac{p_i + n_i}{p + n} \cdot H\left(\frac{p_i}{p_i + n_i}; \frac{n_i}{p_i + n_i}\right) \text{bit} \end{aligned}$$

- Informationsgewinn durch Attribut a als Differenz definiert:

$$\text{gain}(a) = I(E) - I(E|a \text{ bekannt})$$

ID3: Induktion von Entscheidungsbäumen

- ID3 ist mit dieser Heuristik, das Attribut mit dem höchsten Informationsgewinn als “Split-Attribut” zu verwenden, sehr erfolgreich
- Allerdings bevorzugt ID3 Werte mit sehr vielen Attributen: man nehme beispielsweise bei einer Einkommensteuererklärung die jedem Bürger neuerdings zugeordnete eineindeutige Steuernummer.
 - Diese hat so viele Ausprägungen, wie es Bürger (n) gibt.
 - Die Beispielmenge E wird in n Teilmengen partitioniert.
 - Die bedingte mittlere Information

$$I(E|StNr \text{ bekannt}) = \sum_{i=1}^n \frac{1}{n} H(0; 1) = 0 \text{ bit}$$

- Der Informationsgewinn ist also maximal, das Attribut allerdings hier nutzlos.

C4.5: Induktion von Entscheidungsbäumen

- Verbesserung: C4.5
- Statt des absoluten Informationsgewinns wird ein normierter Informationsgewinn genutzt.

$$\textit{gain ratio}(a) = \frac{\textit{gain}(a)}{\textit{split info}(a)}$$

- *split info*(*a*) ist hierbei die Entropie des Attributes *a*:

$$\textit{split info}(a) = H(a) = - \sum_{i=1}^k P(a = w_i) \log_2 P(a = w_i)$$

- Im Beispiel Steuernummer würde eine Gleichverteilung induziert, der Normierungsfaktor würde maximal. Als nächstes Attribut würde dasjenige mit maximalem *gain ratio* gewählt.

- Oberbegriffe für die Automatisierung der Analyse von Daten, Knowledge Discovery in Databases (KDD)
- zentrales Forschungsthema in der Künstlichen Intelligenz
- KDD: Prozeß, neues, nützliches und interessantes Wissen aus Daten herauszufiltern und in verständlicher Form zu präsentieren

1. **Hintergrundwissen und Zielsetzung:** Relevantes, bereichsspezifisches Wissen wird zur Verfügung gestellt. Die Ziele des durchzuführenden KDD sollten definiert werden.
2. **Datenauswahl:** Eine Menge von DATen wird als Untersuchungsobjekt festgelegt. Darüberhinaus erfolgt gegebenenfalls eine Vorauswahl der betrachteten Variablen.
3. **Datenbereinigung:** Ausreißer müssen aus der Datenbasis entfernt, Rauscheffekte herausgefiltert werden. Datentypen werden festgelegt und die Behandlung fehlender Daten wird geklärt.
4. **Datenreduktion und -projektion:** Die vorbehandelte Datenmenge wird noch einmal komprimiert durch Reduktion oder Transformation der behandelten Variablen.

5. **Modellfunktionalität:** Welchem Zweck dient das Data Mining? U.a. gibt es Klassifikation, Clustering, Regressionsanalyse.
6. **Verfahrenswahl:** Bestimmung eines Data-Mining-Verfahrens, das zu den untersuchten Daten und der Zielvorgabe des gesmaten KDD-Prozesses paßt.
7. **Data Mining:** der eigentliche Data-Mining-Prozeß, bei dem das ausgewählte Verfahren auf die behandelte Datenmenge angewandt wird, um interessante Informationen z.B. in Form von Klassifikationsregeln oder Clustern zu extrahieren
8. **Interpretation:** Die im Data-Mining-Schritt gewonnene Information wird aufbereitet, indem z.B. redundante Information entfernt wird, und schließlich dem Benutzer in verständlicher Form (Visualisierung!) präsentiert.

Einsatzgebiete für Data Mining:

- *Klassifikation*: Ein Objekt wird einer oder mehreren vordefinierten Kategorien zugeordnet
- *Clustering*: Ein Objekt wird einer oder mehreren Klassen bzw. Clustern zugeordnet, wobei diese im Unterschied zur Klassifikation nicht vorgegeben sind, sondern erst bestimmt werden müssen. Natürliche Gruppierungen von Clustern sollen gefunden werden.
- *Modellierung von Abhängigkeiten*: Lokale Abhängigkeiten zwischen Variablen werden etabliert. Die Stärke der Abhängigkeiten wird bei quantitativen Methoden numerisch angegeben.

Einsatzgebiete für Data Mining:

- *Sequenzanalyse*: beschreibt Muster in sequentiellen Daten, um Regelmäßigkeiten und Trends transparent zu machen, beispielsweise in der Zeitreihenanalyse
- *Assoziationen*: sind Zusammenhänge zwischen mehreren Merkmalen und werden meist durch *Assoziationsregeln* repräsentiert.

Im Folgenden werden Assoziationen in Form von Assoziationsregeln eingehender behandelt.

Assoziationsregeln

- beschreiben gewissen Zusammenhänge und Regelmäßigkeiten zwischen verschiedenen Dingen wie z.B. den Artikeln eines Warenhauses oder sozio-ökonomischen Merkmalen
- Zusammenhänge sind allgemeiner Art, nicht notwendigerweise kausaler Natur
- Annahme: in diesen Assoziationen manifestieren sich implizite strukturelle Abhängigkeiten

Assoziationsregeln, Formales

- behandelte Dinge: *Items*, $\mathcal{I} = \{i_1, i_2, \dots\}$
- $\mathcal{X} \subseteq \mathcal{I}$: Itemmenge
- k-Itemmenge: Itemmenge mit k Elementen
- Transaktion $t \subseteq \mathcal{I}$ ist eine Itemmenge
- $\mathcal{D} = \{t_1, t_2, \dots\}$ Menge von Transaktionen als Datenbasis
- Relativer Anteil aller Transaktionen, die X enthalten:

$$\text{support}(X) = \frac{|\{t \in \mathcal{D} | X \subseteq t\}|}{|\mathcal{D}|}$$

Assoziationsregeln, Formales

- Assoziationsregel: $X \rightarrow Y$
 - $X, Y \subseteq \mathcal{I}$
 - $X \cap Y = \emptyset$
- $support(X \rightarrow Y) = support(X \cup Y)$
- Relativer Anteil derjenigen X enthaltenden Transaktionen, die auch Y enthalten:

$$confidence(X \rightarrow Y) = \frac{|\{t \in \mathcal{D} | (X \cup Y) \subseteq t\}|}{|\{t \in \mathcal{D} | X \subseteq t\}|} \quad (1)$$

$$= \frac{support(X \rightarrow Y)}{support(X)} \quad (2)$$

Assoziationsregeln, Algorithmus

- **Aufgabe:** Finde alle Assoziationsregeln, die in der betrachteten Datenbasis mit einem Support von mindestens *minsupp* und einer Konfidenz von mindestens *minconf* gelten, wobei *minsupp* und *minconf* benutzerdefinierte Werte sind.
- **Teilaufgabe 1:** Finde alle Itemmengen, deren Support über der *minsupp*-Schwelle liegt. Diese Mengen werden *häufige Itemmengen* (frequent itemsets) genannt.
- **Teilaufgabe 2:** Finde in jeder häufigen Itemmenge I alle Assoziationsregeln $I' \rightarrow (I - I')$ mit $I' \subset I$, deren Konfidenz mindestens *minconf* beträgt.
- Nützliche Tatsache für den folgenden *apriori*-Algorithmus: Alle Teilmengen einer häufigen Itemmenge sind ebenfalls häufig. Alle Obermengen einer nicht häufigen Itemmenge sind ebenfalls nicht häufig.

Apriori-Algorithmus

Algorithmus: Apriori

Eingabe: Datenbasis \mathcal{D}

Ausgabe: Menge häufiger Itemmengen

1. $L_1 := \{\text{häufige 1-Itemmengen}\}$
2. $k := 2$
3. **while** $L_{k-1} \neq \emptyset$ **do**
4. $C_k := \text{AprioriGen}(L_{k-1})$
5. **for all** Transaktionen $t \in \mathcal{D}$ **do**
6. $C_t := \{c \in C_k \mid c \subseteq t\}$
7. **for all** Kandidaten $c \in C_t$ **do**
8. $c.\text{count} := c.\text{count} + 1$
9. **end for**
10. **end for**
11. $L_k := \{c \in C_k \mid c.\text{count} \geq |\mathcal{D}| \cdot \text{minsupp}\}$
12. $k := k + 1$
13. **end while**
14. **return** $\bigcup_k L_k$

Apriori-Algorithmus

Algorithmus: AprioriGen(L_{k-1})

Eingabe: Menge häufiger ($k-1$)-Itemmengen L_{k-1}

Ausgabe: Obermenge der Menge häufiger k -Itemmengen

1. $C_k := \emptyset$
2. **for all** $p, q \in L_{k-1}$ mit $p \neq q$ **do**
3. **if** p und q haben $k - 2$ gleiche Elemente
4. $p = \{e_1, \dots, e_{k-2}, e_p\}$
5. $q = \{e_1, \dots, e_{k-2}, e_q\}$
6. **und** $e_p < e_q$ **then**
7. $C_k := C_k \cup \{\{e_1, \dots, e_{k-2}, e_p, e_q\}\}$
8. **end if**
9. **end for**
10. **for all** $c \in C_k$ **do**
11. **for all** ($k - 1$)-Teilmengen s von c **do**
12. **if** $s \notin L_{k-1}$ **then**
13. $C_k := C_k \setminus \{c\}$
14. **end if**
15. **end for**
16. **end for**
17. **return** C_k

- ideales Einsatzszenario für Assoziationsregeln
 - Modellbildung ist nicht nötig
 - Regeln können isoliert betrachtet werden
 - Daten stehen in der Regel bereits zur Verfügung

Beispiel: Warenkorbanalyse

Label	Artikel	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	support
A	Seife	•				•		•		•		0,4
B	Shampoo	•	•	•	•		•		•	•	•	0,8
C	Haarspülung		•	•	•		•		•	•		0,6
D	Duschgel	•			•		•	•		•	•	0,6
E	Zahnpasta	•		•		•		•				0,4
F	Zahnbürste			•		•						0,2
G	Haarfärbung		•		•				•			0,3
H	Haargel		•									0,1
J	Deodorant			•	•	•	•	•	•			0,6
K	Parfüm						•		•			0,2
L	Kosmetikartikel		•		•		•		•		•	0,5

Einkaufstransaktionen in einem Drogeriemarkt

Beispiel: Warenkorbanalyse

- gesucht: alle Assoziationsregeln mit:
 - $\text{minsupp} = 0,4$
 - $\text{minconf} = 0,7$
- in realen Anwendung wird *minsupp* in der Regel sehr viel kleiner gewählt ($< 0,01$)
- häufige 1-Itemmengen:

$$L_1 = \{\{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{J\}, \{L\}\}$$

Beispiel: Warenkorbanalyse

Berechnung der Menge C_2 : alle paarweisen Kombinationen von Mengen in L_1 bilden und deren Support bestimmen.

C_2 -Menge	Support	C_2 -Menge	Support	C_2 -Menge	Support
{A,B}	0,2	{B,D}	0,5	{C,L}	0,4
{A,C}	0,1	{B,E}	0,2	{D,E}	0,2
{A,D}	0,2	{B,J}	0,4	{D,J}	0,3
{A,E}	0,3	{B,L}	0,5	{D,L}	0,3
{A,J}	0,2	{C,D}	0,3	{E,J}	0,3
{A,L}	0,0	{C,E}	0,1	{E,L}	0,0
{B,C}	0,6	{C,J}	0,4	{J,L}	0,3

Beispiel: Warenkorbanalyse

- häufigste 2-Itemmengen:

$$L_2 = \{\{B, C\}, \{B, D\}, \{B, J\}, \{B, L\}, \{C, J\}, \{C, L\}\}$$

- Berechnung von C_3 :

C_3 vor Teilmengencheck	C_3 nach Teilmengencheck	Support
$\{B, C, D\}$	$\{B, C, J\}$	0,4
$\{B, C, J\}$	$\{B, C, L\}$	0,4
$\{B, C, L\}$		
$\{B, D, J\}$		
$\{B, D, L\}$		
$\{B, J, L\}$		
$\{C, J, L\}$		

Beispiel: Warenkorbanalyse

- Damit ist

$$L_3 = \{\{B, C, J\}, \{B, C, L\}\}$$

- einzig mögliche weitere Kombination: $\{B, C, J, L\}$
- allerdings nicht häufig, daher ist $C_4 = L_4 = \emptyset$

Beispiel: Warenkorbanalyse

- Bildung der Assoziationsregeln aus den häufigen Itemmengen:

Regel	Konfidenz	Regel	Konfidenz
B→C	0,75	C→B	1,00
B→D	0,63	D→B	0,83
B→J	0,50	J→B	0,67
B→L	0,63	L→B	1,00
C→J	0,67	J→C	0,67
C→L	0,67	L→C	0,80

- fünf der Regeln erfüllen die Konfidenzbedingung (minconf= 0,7)

Beispiel: Warenkorbanalyse

- L_3 enthält $l_{3.1} = \{B, C, J\}$ und $l_{3.2} = \{B, C, L\}$
- $l_{3.1}$ (in \square die Konfidenz der Regel)
 - $H_1 = \{B, C, J\}$
 - Regeln: $BC \rightarrow J$ [0,67], $BJ \rightarrow C$ [1,00], $CJ \rightarrow B$ [1,00]
 - $H_2 = \text{AprioriGen}(H_1) = \{B, C\}$
 - Regel: $J \rightarrow BC$ [0,67]
- $l_{3.2}$
 - Regeln: $BC \rightarrow L$ [0,67], $BL \rightarrow C$ [0,8], $CL \rightarrow B$ [1,00]
 - durch Erweiterung der Konklusion noch: $L \rightarrow BC$ [0,8]

Beispiel: Warenkorbanalyse

Regel		Support	Konfidenz	
Shampoo	→	Haarspülung	0,6	0,75
Haarspülung	→	Shampoo	0,6	1,00
Duschgel	→	Shampoo	0,5	0,83
Kosmetik	→	Shampoo	0,5	1,00
Kosmetik	→	Haarspülung	0,4	0,80
Shampoo, Deodorant	→	Haarspülung	0,4	1,00
Haarspülung, Deodorant	→	Shampoo	0,4	1,00
Shampoo, Kosmetik	→	Haarspülung	0,4	0,80
Haarspülung, Kosmetik	→	Shampoo	0,4	1,00
Kosmetik	→	Shampoo, Haarspülung	0,4	0,80

Werbung in eigener Sache: im **Sommersemester** wird unsere reguläre Vorlesung **“Intelligent Data Analysis”** stattfinden. Es geht dort unter anderem um klassische Statistik, Assoziationsregeln, Bayes'sche Klassifikation, Entscheidungs- und Regressionsbäume, Fuzzy-Datenanalyse und Clustering-Techniken.

Weitere Informationen werden bald verfügbar sein unter:

<http://fuzzy.cs.ovgu.de/wiki/pmwiki.php?n=Lehre.IDA2010>



Beierle, C. and Kern-Isberner, G. (2008).

Methoden wissensbasierter Systeme : Grundlagen, Algorithmen, Anwendungen.

Vieweg+Teubner, Wiesbaden, 4., überarbeitete und erweiterte Auflage edition.



Carbonell, J. G., Michalski, R. S., and Mitchell, T. M. (1984).

An overview of machine learning.

In Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., editors, *Machine Learning: An Artificial Intelligence Approach*, pages 3–23. Springer, Berlin, Heidelberg.



Michalski, R. S. and Kodratoff, Y. (1990).

Research in machine learning: recent progress, classification of methods, and future directions.
pages 3–30.



Michalski, R. S. and Michalski, R. S. (1986).

Understanding the nature of learning: Issues and research directions.

In *Machine Learning: An Artificial Intelligence Approach*, pages 3–25. Morgan Kaufmann.



Michalski, S. R., Carbonell, G. J., and Mitchell, M. T., editors (1986).

Machine learning an artificial intelligence approach volume II.

Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.



Simon, H. A. (1983).

Why should machines learn?

In *Machine Learning, An Artificial Intelligence Approach*. Tioga, Palo Alto, California.