Maschinelles Lernen: Symbolische Ansätze

TECHNISCHE UNIVERSITÄT DARMSTADT

Wintersemester 2010/2011
Musterlösung für das 13. Übungsblatt

Aufgabe 1: Apriori

Gegeben seien folgende Beobachtungen vom Kaufverhalten von Kunden:

beer	chips	dip	pizza	wine
1	1	1	0	0
1	1	1	1	0
0	1	1	0	1
1	1	0	1	0
0	1	1	1	1
0	0	0	1	1
1	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	1	1	1	1

a) Wenden Sie den Apriori Algorithms (Foliensatz Clustering und Lernen von Assoziationsregeln, Folie 10) auf den obigen Datensatz an, um zunächst alle frequent Itemsets zu bestimmen. Benutzen Sie hierfür einen minimalen Support von 3 Beispielen.

Lösung: Im ersten Schritt des Algorithmus werden alle einelementigen Itemsets generiert und deren Support berechnet (die folgende Tabelle entspricht der Menge C_1 aus der Vorlesung):

itemset	support
beer	7
chips	7
dip	6
pizza	7
wine	5

Da keines der Itemsets einen Support kleiner als 3 hat, werden alle im nächsten Schritt weiterverwendet (und damit gilt $S_1 = C_1$). In diesem Schritt werden dann alle 2-elementigen Itemsets gebildet, indem man alle Kombinationen durchgeht. Es wird jeweils wieder der Support berechnet (C_2):

itemset	support
beer, chips	5
beer, dip	4
beer, pizza	5
beer, wine	2
chips, dip	6
chips, pizza	5
chips, wine	3
dip, pizza	4
dip, wine	3
pizza, wine	3

Das rot markierte Itemset {beer, wine} hat nur einen Support von 2 und ist deshalb nicht frequent. Daher gilt: $S_2 = C_2 \setminus$ {beer, wine}. Aus den 2-elementigen Itemsets werden nun die 3-elementigen generiert. Dazu werden nur diese Itemsets zusammengefügt, deren erster Wert gleich ist (Ordnung auf den items, hier: alphabetisch, siehe Foliensatz Clusterung und Lernen von Assoziationsregeln, Folie 11). Es ergeben sich daher folgende Itemsets (C_3):

itemset	support
beer, chips, dip	4
beer, chips, pizza	4
beer, dip, pizza	3
chips, dip, pizza	4
chips, dip, wine	3
chips, pizza, wine	2
dip, pizza, wine	2

Hier sind nun 2 Itemsets nicht mehr frequent. Daher sind in der Menge S_3 alle Itemsets außer {chips, pizza, wine} und {dip, pizza, wine} enthalten. Nun werden die 4-elementigen Itemsets generiert (C_4), indem die ersten beiden Stellen gleich sein müssen. Da {chips, pizza, wine} nicht frequent war, kann nur ein Itemset generiert werden:

itemset	support
beer, chips, dip, pizza	3

Dieses Itemset ist frequent. Daher ist $S_4 = \{\{\text{beer, chips, dip, pizza}\}\}$. Es ergeben sich demnach insgesamt folgende frequent Itemsets:

1-elementige (S_1):
{beer}, {chips}, {dip}, {pizza}, {wine}

2-elementige (S_2):
{beer, chips}, {beer, dip}, {beer, pizza}, {chips, dip}, {chips, pizza}, {chips, wine}, {dip, pizza}, {dip, wine}, {pizza, wine}

3-elementige (S_3):
{beer, chips, dip}, {beer, chips, pizza}, {beer, dip, pizza}, {chips, dip, pizza}, {chips, dip, wine}

4-elementige (S_4):
{beer, chips, dip, pizza}

b) Erstellen Sie für alle drei- und vierelementigen Itemsets die Assoziationsregeln mit einer minimalen Konfidenz von 0.8, wie im zweiten Teil des Algorithmus beschrieben (Foliensatz Clustering und Lernen von Assoziationsregeln, Folie 15). Um sich Arbeit zu sparen, können Sie die Antimonotonie der Konfidenz ausnutzen.

Lösung: Nun sollen alle Regeln aus den frequent Itemsets generiert werden, die mindestens eine Konfidenz von 0.8 haben. Wir machen uns hier die Antimonotonie der Konfidenz zu Nutze. Diese besagt: $conf(A \to B, C) \leq conf(A, B \to C)$. Wir generieren also erst die Regeln, die nur ein Element im head haben und können uns alle weiteren Regeln mit diesem head sparen, falls die mit einelementigen head bereits nicht konfident genug sind. Wir kürzen hierfür die items mit ihrem Anfangsbuchstaben ab und schreiben die Konfidenz der Regeln in Klammern hinter diese. Die Konfidenz erhält man aus den vorher berechneten Support-Werten.

3-elementige:

{b,c,d}: $b, c \to d$ ($conf = 0.8 = \frac{4}{5}$), $b, d \to c$ (1), $c, d \to b$ (0.67), $b \to c, d$ (0.57), es werden $c \to b, d$ und $d \to b, c$ geprunt, da $c, d \to b$ (0.67) bereits nicht konfident war

{b,c,p}: $b, c \to p$ (0.8), $b, p \to c$ (0.8), $c, p \to b$ (0.8), $b \to c, p$ (0.57), $c \to b, p$ (0.57), $p \to b, c$ (0.57)

{b,d,p}: $b, d \to p$ (0.75), $d, p \to b$ (0.75), $b, p \to d$ (0.6)

{c,d,p}: $c, d \to p$ (0.67), $c, p \to d$ (0.8), $d, p \to c$ (1), $p \to c, d$ (0.57)

{c,d,w}: $c, d \to w$ (0.5), $c, w \to d$ (1), $d, w \to c$ (1), $w \to c, d$ (0.6)

4-elementige:

{b,c,d,p}: $b, c, d \to p$ (0.75), $b, c, p \to d$ (0.75), $b, d, p \to c$ (1), $c, d, p \to b$ (0.75)

Nachtrag: Effizientes Generieren aller konfidenten Regeln eines Itemsets

Angenommen wir haben ein Itemset $I = \{i_1, \cdots, i_n\}$ gegeben, dann notieren wir eine Regel $B \implies H$ mit $B \subseteq I$ und $H = I \setminus B$ durch $[H]$, d.h. allein durch die Angabe der Items im Head ist die Regel definiert (Beispiel: Für $I = \{beer, chips, pizza, wine\}$ steht $[beer, pizza]_I$ für die Regel $chips, wine \implies beer, pizza$). Anhand dieser Notation können wir, wie bereits im Foliensatz Clustering und Lernen von Assoziationsregeln, Folie 15 erwähnt, den FreqSet-Algorithmus so modifizieren, daß er zur Generierung aller konfidenten Regeln eines Itemsets verwendet werden kann. Hierbei muß auch eine Sortierung der Itemsets vorgenommen bzw. die bereits bestehende Sortierung beibehalten werden.

Algorithm 1: ConfRule: Calculate all confident rules of itemset I

Data: Itemset I, min_conf
Result: R: all confident rules of itemset I
$R = \emptyset$
$k = 1$
$C_1 = \{[i] \,|\, i \in I\}$
while $C_k \neq \emptyset$ **do**
  //Store all confident rules in C_k:
  $R_k = \{[X] | [X] \in C_k, conf([X]) \geq min\_conf\}$
  //Join all rules in C_k whose first $k-1$ items in the head are equal:
  $C_{k+1} = \{[i_1, \cdots i_{k-1}, i_k, i_{k+1}] \mid [i_1, \cdots i_{k-1}, i_k] \in R_k, [i_1, \cdots i_{k-1}, i_{k+1}] \in R_k, i_k < i_{k+1}\}$
  //Remove all rules ([A,B]) which have an unconfident predecessor rule ([A])
  $C_{k+1} = C_{k+1} \setminus \{[i_1, \cdots i_k] \,| \exists X \subset \{i_1, \cdots i_k\}.conf([X]) < min\_conf\}$
  $R = R \cup R_k$
  $k{+}{+}$
return R

Betrachten wir noch einmal das 4-itemset $\{b, c, d, p\}$ und berechnen diesmal alle Regeln mit minimaler Konfidenz 0,6 mittels des obigen Algorithmus.

- Initialisierung:
  - $R = \emptyset$
  - $k = 1$
  - $C_1 = \{[b], [c], [d], [p]\}$
- 1. Iteration: $k = 1$
  - Entferne inkonfidente Regeln aus C_1
    Alle Regeln sind konfident:
    $conf([b]) = 0,75$,
    $conf([c]) = 1$,
    $conf([d]) = 0,75$,
    $conf([p]) = 0,75$
    $R_1 = C_1$
  - $C_2 = \{[b,c][b,d][b,p], [c,d][c,p], [d,p]\}$
  - Entferne alle Regeln aus C_2 mit inkonfidenten Vorgängern
    Alle Vorgängerregeln sind konfident
  - $R = R_1$
- 2. Iteration: $k = 2$
  - Entferne inkonfidente Regeln aus C_2: $[b,p]$
    $conf([b,c]) = 0,75$,
    $conf([b,d]) = 0,6$,
    $conf([b,p]) = 0,5$,
    $conf([c,d]) = 0,6$,
    $conf([c,p]) = 0,75$,
    $conf([d,p]) = 0,6$
    $R_2 = C_2 \setminus \{[b,p]\}$

- $C_3 = \{[b,c,d],[c,d,p]\}$
- Entferne alle Regeln aus C_3 mit inkonfidenten Vorgängern
  Alle Vorgängerregeln sind konfident
- $R = R_1 \cup R_2$

- 3. Iteration: $k = 3$
  - Entferne inkonfidente Regeln aus C_3: $[b,c,d]$,$[c,d,p]$
    $conf([b,c,d]) = \frac{3}{7} = \approx 0,43$,
    $conf([c,d,p]) = \frac{3}{7}$
    $R_3 = \emptyset$
  - $C_4 = \emptyset$
  - Entferne alle Regeln aus C_4 mit inkonfidenten Vorgängern
    $C_4 = \emptyset$
  - $R = R_1 \cup R_2 \cup R_3 = R_1 \cup R_2$

- Alle konfidenten Regeln befinden sich in R:
  - $[b] : c,d,p \implies b$
  - $[c] : b,d,p \implies c$
  - $\cdots$
  - $[d,p] : b,c \implies d,p$

Aufgabe 2: Assoziationsregel-Maße

Ein online Buchgeschäft möchte eine Datenbank mit 10,000 Kunden analysieren, die jeweils eines oder mehrere von 500 verschiedenen Büchern gekauft haben. Zur Entdeckung von Assoziationsregeln wird der Algorithmus Apriori mit einem Minimum Support von 3% und einer minimalen Konfidenz von 75% verwendet.

a) Es wird festgestellt, daß die beiden häufigsten Verkäufe “Harry Potter und der Stein der Weisen” (HP1) und “Harry Potter und die Kammer des Schreckens” (HP2) sind. HP1 wurde von 6,000 Kunden und HP2 von 8,000 Kunden gekauft. 4,000 Kunden kauften beide Bücher.

Welche der beiden Assoziationsregeln findet sich im Output des Assoziations-regel-Lerners?

- HP1 → HP2
- HP2 → HP1
- beide
- keine von beiden

Geben Sie Support und Konfidenz für beide Regeln an.

Lösung: Bevor wir den Support der beiden Regeln berechnen, betrachten wir zunächst, welche absoluten Häufigkeiten hierfür benötigt werden:

$$\begin{aligned} support(HP1 \rightarrow HP2) &= support(HP1 \cup HP2) \\ &= support(HP2 \cup HP1) \\ &= support(HP2 \rightarrow HP1) \end{aligned}$$

$$\Rightarrow support(HP1 \rightarrow HP2) = support(HP2 \rightarrow HP1) = \frac{n(HP1 \cup HP2)}{n}$$

$n(HP1 \cup HP2)$ ist die Anzahl der Kunden, die beide Bücher gekauft haben, und n ist die Gesamtanzahl von Kunden.

Demnach gilt:

$$\Rightarrow support(HP1 \rightarrow HP2) = support(HP2 \rightarrow HP1) = \frac{4.000}{10.000} = 0,4 > 0,03$$

Da beide Regeln frequent sind, müssen wir für beide jeweils die Konfidenz berechnen.

$$\begin{aligned} confidence(HP1 \rightarrow HP2) &= \frac{n(HP1 \cup HP2)}{n(HP1)} \\ &\neq \frac{n(HP1 \cup HP2)}{n(HP2)} \\ &= confidence(HP2 \rightarrow HP1) \end{aligned}$$

$n(HP1)$ bzw. $n(HP2)$ ist die Anzahl der Kunden, die HP1 bzw. HP2 gekauft haben:

$$\begin{aligned} confidence(HP1 \rightarrow HP2) &= \frac{4.000}{6.000} < 0,67 < 0,75 \\ confidence(HP2 \rightarrow HP1) &= \frac{4.000}{8.000} = 0,5 < 0,75 \end{aligned}$$

Beide Regeln erfüllen nicht die Mindestanforderung an Konfidenz und sind deshalb nicht im Output des Regellerners.

b) Wenn man annimmt, daß alle Kunden, die beide Bücher gekauft haben, zuerst HP1 und später HP2 gekauft haben: Wie interpretieren Sie den Einfluß des Kaufs von HP1 auf den Kauf von HP2?

Lösung: Man kann diese Aufgabe auf verschiedene Weisen betrachten:

- **Wahrscheinlichkeitstheorie**: Wenn HP1 und HP2 Zufallsereignisse sind, können wir deren Wahrscheinlichkeiten verwenden, um zu testen, ob ihr Auftreten unabhängig voneinander ist. Wir kennen die folgenden geschätzten Wahrscheinlichkeiten:

  $$\begin{aligned} \Pr(HP1) &= 0,6 \\ \Pr(HP2) &= 0,8 \\ \Pr(HP1 \cap HP2) &= 0,4 \end{aligned}$$

  Wären die Ereignisse unabhängig voneinander, dann müßte eigentlich gelten:

  $$\Pr(HP1 \cap HP2) \stackrel{!}{=} \Pr(HP1) \cdot \Pr(HP2) = 0,48 > 0,4$$

  Hieraus können wir schließen, daß der Kauf der beiden Bücher nicht unabhängig ist. Der Kauf eines der Bücher hat einen leicht negativen Effekt auf den Kauf des anderen. D.h. für jemanden, der HP1/2 gekauft hat, ist der Kauf von HP2/1 unwahrscheinlicher als für jemanden, der noch keines der Bücher besitzt.

- **Leverage**: Übertragen wir diese Überlegung wieder auf den Support des Bodies und des Head, können wir die Differenz zwischen dem tatsächlichen Auftreten und dem erwartenden Auftreten von Body und Head berechnen. Dies entspricht dem Leverage der Assoziationsregel.

  $$\begin{aligned} & leverage(HP1 \rightarrow HP2) \\ = \; & support(HP1 \rightarrow HP2) - support(HP1) \cdot support(HP2) \\ = \; & 0,4 - 0,48 = -0,08 < 0 \end{aligned}$$

  Ein negativer Leverage bedeutet, daß der Kauf einen negativen Einfluß auf den Kauf des anderen Buches (s.o.). Dies gilt auch für die umgekehrte Assoziation, da Leverage symmetrisch ist.

- **Lift**: Mit dem Maß Lift können wir den Einfluß ähnlich begründen:

  $$\begin{aligned} lift(HP1 \rightarrow HP2) &= \frac{support(HP1 \rightarrow HP2)}{support(HP1) \cdot support(HP2)} \\ &= \frac{0,4}{0,8 \cdot 0,6} = 0,8\overline{3} < 1 \end{aligned}$$

  Ein Lift kleiner eins bedeutet, daß Body und Head gemeinsam seltener vorkommen als zu erwarten wäre. D.h. das Auftreten des Bodies hat einen negativen Effekt auf das Auftreten des Heads. Da das Maß Lift symmetrisch ist, gilt diese Aussage auch analog umgekehrt. Bezogen auf unsere Aufgabe bedeutet, daß der Kauf eines der Bücher wie schon oben erwähnt einen negativen Einfluß auf den Kauf des anderen Buches hat.

c) Die längste Assoziationsregel, die gefunden wurde, wurde aus einem Itemset der Größe 20 konstruiert. Geben Sie eine möglichst große untere Schranke für die Anzahl der gefundenen Frequent Itemsets an.

Lösung: Wir wissen, daß wir mindestens ein Itemset der Größe 20 gefunden haben. Aus diesem Grund können wir die minimale Anzahl von Itemsets, die dann auch frequent sein müßten, abschätzen, indem wir von einem einzigen Itemset der Größe 20 (kurz: I) ausgehen. Da diese Itemset frequent, müssen alle seine Teilmengen der Größe 19 auch frequent sein. **Erinnerung:** Monotonie des Supports:

$$\begin{aligned} & support(\{Item_1, Item_2, ..., Item_n\}) \\ \leq \quad & support(\{Item_1, Item_2, ..., Item_n\} \backslash \{Item_i\}) \text{ für alle } i \in \{1, 2, ..., n\} \end{aligned}$$

Dasselbe gilt wiederum auch für alle Teilmengen des Itemsets der Größe 18 bis 1. Demnach müssen wir einfach nur alle Kombinationen von Teilmengen der Größe 1 bis 19 bestimmen. Dies entspricht der Potenzmenge von I. Diese hat 2^{20} Elemente, da I selbst und die leere Menge jedoch nicht relevant sind, erhalten wir die folgende Abschätzung: es sind mindestens $2^{20} - 2$ weitere Itemsets frequent.