Maschinelles Lernen: Symbolische Ansätze

Musterlösung für das 13. Übungsblatt

Aufgabe 1

Gegeben sei folgender String: `ADFASASDASFASDFSDFFAS`

1. Finden Sie mit einen Apriori-ähnlichen Algorithmus alle Teilstrings, die mindestens 2x vorkommen.

   Lösung: Zuerst sucht man die Teilstrings, die aus einem Buchstaben bestehen (ähnlich der Vorgehensweise, die auf Folie 12 skizziert ist), bildet also die Menge C_1. In Klammern steht die Anzahl der Teilstrings, wobei man theoretisch nur nachschauen müsste, ob der Teilstring 2 mal vorkommt und nicht mehr weiterzählen müsste. Da aber in der nächsten Aufgabe unter anderem der Support berechnet werden soll, steht die Gesamtanzahl in Klammern.

   C_1={A(6), D(4), F(5), S(6)}

   Nun schauen wir nach, welche der Items im Itemset frequent sind (also einen "Support" größer 2 haben - natürlich ist hier der Support keine Prozentzahl wie üblich, sondern resultiert aus der Forderung, dass die Teilstrings mindestens 2 mal vorkommen müssen). Da alle frequent sind folgt $S_1 = C_1$

   Nun bilden wir alle möglichen Kombinationen aus den Frequent Items der Länge 1 und zählen die Vorkommnisse:

   C_2={AA(0), AS(5), AF(0), AD(1), SA(1), SS(0), SF(1), SD(0), FA(3), FS(1), FF(1), FD(0), DA(1), DS(0), DF(3), DD(0)}

   Wie im vorherigen Schritt löschen wir alle Items, deren Support < 2 ist:

   S_2={AS(5), SD(3), FA(3), DF(3)}

   Beim Standard *Apriori-Algorithmus* würde man nun so vorgehen, dass man immer die Teilstrings miteinander verbindet, deren erster Wert gleich ist und nachschaut, ob der Teilstring, der aus den beiden zweiten Werten entsteht in der S-Menge enthalten ist. Hätte man beispielsweise folgende Situation: S = {AS, AD, FA, DF}, so würde man den Teilstring 'ASD' bilden. Da aber der Teilstring 'SD' nicht in S enthalten ist, wäre 'ASD' nicht zulässig (vergleiche Folie 12), obwohl 'ASD' ja im String vorkommt. In der Aufgabe hier geht es aber darum Teilstrings aus einem gesamten String herauszusuchen. Aus diesem Grund muss man ein leicht verändertes

Verfahren anwenden, da man sonst Teilstrings verpassen würde (aus der Menge S_2 würde sich kein einziger Teilstring der Länge 3 bilden lassen, obwohl natürlich welche vorhanden sind). Man geht aus diesem Grund folgendermaßen vor:

Um weitere mögliche Teilstrings der Länge 3 zu finden, schiebt man die bisherigen Teilstrings 1 Position ineinander. Man würde also nach Teilstrings suchen, die einen gewissen zweiten Buchstaben haben und überprüfen, ob dieser Buchstabe in einem anderen Teilstring an der ersten Position steht. Dann erhält man zB aus 'AS' und 'SD' den Teilstring 'ASD' der Länge 3, wobei man hier keine weitere Überprüfung wie beim Standard *Apriori-Algorithmus* machen kann, sondern höchstens eine Abschätzung wie oft der Teilstring der Länge 3 maximal vorkommen kann (höchstens so häufig, wie das Minimum der Vorkommnisse der beiden Teilstrings der Länge 2).

C_3={ASD(2), SDF(2), DFA(1), FAS(3)}

Nun löscht man wieder alle deren Support nicht ausreicht:

S_3={FAS(3), ASD(2), SDF(2)}

Die Teilstrings der Länge 4 bildet man nun indem man die Strings um 2 Positionen ineinander verschiebt:

C_4={FASD(1), ASDF(1)}

Da beide Teilstrings den minimalen Support nicht mehr erfüllen folgt:

S_4={}

Nun sind wir fertig, da S leer ist. Wir haben folgende Teilstrings gefunden, die mindestens 2 mal vorkommen:

$\mathcal{S} = S_1 \cup S_2 \cup S_3 \cup S_4$={A, S, F, D, AS, SD, FA, DF, ASD, SDF, FAS}

2. Bilden Sie aus den gefundenen Subsequenzen mit minimum support = 2 alle Regeln mit minimum confidence = 0.5

   Lösung: Aus S_1 können wir keine Regeln erstellen, da die Menge nur aus einelementigen Teilstrings besteht. Wir errechnen die Konfidenz der Elemente aus S_2 und S_3, indem wir die Support-Werte aus der vorherigen Aufgabe verwenden und das Konfidenzmaß $confidence(A \rightarrow B) = \frac{support(A \cup B)}{support(A)}$:

   Für S_2: $A \rightarrow S : 5/6$, $S \rightarrow D : 3/6 = 0.5$, $F \rightarrow A : 3/5$ und $D \rightarrow F : 3/4$ wie man sieht haben alle Regeln eine Konfidenz ≥ 0.5

   Für S_3: $F \rightarrow AS : 3/5$, $FA \rightarrow S : 3/3 = 1$, $A \rightarrow SD : 2/6 = 1/3$, $AS \rightarrow D : 2/5$, $S \rightarrow DF : 2/6 = 1/3$ und $SD \rightarrow F : 2/3$ wie man sieht haben hier die Regeln $A \rightarrow SD$, $AS \rightarrow D$ und $S \rightarrow DF$ eine zu geringe Konfidenz und werden deshalb gelöscht.

   Es resultiert die Regelmenge:

   $\mathcal{R} = \{A \rightarrow S, S \rightarrow D, F \rightarrow A, D \rightarrow F, F \rightarrow AS, FA \rightarrow S, SD \rightarrow F\}$

Aufgabe 2

Ein on-line 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.

1. 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 Assoziationsregel-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.

2. 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:

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

  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.

3. 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.