Gottfried Wilhelm Leibniz Universität Hannover
Institut für Praktische Informatik
Fachgebiet Datenbanken und Informationssysteme
Prof. Dr. Udo Lipeck/M.Sc. Oliver Pabst

13.-20.05.2015

Übung zur Vorlesung „Data Mining“ im Sommersemester 2015

Übungsblatt 4

Aufgabe 1 *(Apriori-Algorithmus mit mehrfachem Minimal-Support)*

Gegeben sei wieder die aus vorigen Übungsblättern bekannte Transaktionsdatenbank:

Nr.	Transaktion
1	{ E, M, O, R, S }
2	{ E, N, O, S }
3	{ A, B, D, E, N, O, R, S }
4	{ A, D, N, R, S, T }
5	{ E, M, R }

Diesmal gilt jedoch der minimale Support von minsup $= \frac{3}{5}$ nicht für alle Items der Datenbank: Für das Item D soll minsup$(D) = \frac{2}{5}$ und für R minsup$(R) = \frac{4}{5}$ gelten.

Wenden Sie die in der Vorlesung vorgestellte Modifikation des Apriori-Algorithmus für mehrfachen Minimal-Support an, um alle Frequent Itemsets zu finden.

Lösung:

Sortierung der Items nach ihrem Minimal-Support (und zweitrangig alphabetisch):

Nr.	Transaktion
1	{ E, M, O, S, R }
2	{ E, N, O, S }
3	{ D, A, B, E, N, O, S, R }
4	{ D, A, N, S, T, R }
5	{ E, M, R }

Schritte des Algorithmus:

F_1 :

Set	σ
{D}	2
{E}	4
{N}	3
{O}	3
{S}	4
{R}	4

C_2 :

Set	σ
{D, E}	1
{D, N}	2 ✓
{D, O}	1
{D, S}	2 ✓
{D, R}	2 ✓
{E, N}	2
{E, O}	3 ✓
{E, S}	3 ✓
{E, R}	3 ✓
{N, O}	2
{N, S}	3 ✓
{N, R}	2
{O, S}	3 ✓
{O, R}	2
{S, R}	3 ✓

F_2 :

set
{D, N}
{D, S}
{D, R}
{E, O}
{E, S}
{E, R}
{N, S}
{O, S}
{S, R}

C_3 :

Set	σ
{D, N, S}	2 ✓
{D, N, R}	2 ✓
{D, S, R}	2 ✓
{E, O, S}	3 ✓
{E, O, R}	2
{E, S, R}	2

F_3 :

set
{D, N, S}
{D, N, R}
{D, S, R}
{E, O, S}

$C_4 = F_4$:

Set	Anz
{D, N, S, R}	2 ✓

Im Vergleich zum unmodifizierten Apriori-Algorithmus mit minsup $= \frac{3}{5}$ (siehe Blatt 1) ist zusätzlich D ein Frequent Item (F_1). Damit ergibt sich C_2 aus C_2 des unmodifizierten Algorithmus zusammen mit den 5 möglichen Kombinationen von D und allen anderen Frequent Items.

Man beachte, dass das modifizierte Pruning nicht in der Lage ist, das Itemset { E, O, R } in C_3 auszuschließen. Dieses Pruning wirkt sich in diesem Beispiel gar nicht aus.

Aufgabe 2 *(Apriori-Algorithmus für Sequenzen – GSP)*

Die altbekannte Datenbank aus Aufgabe 1 sei hier in trivialer Weise als Sequenzdatenbank interpretiert: Jedes Itemset wird zu einer Sequenz der Länge 1. Außerdem sei wieder ein Minimal-Support (für alle Items) von minsup $= 0.6$ vorgegeben:

Nr.	Sequenz
1	< { E M O R S } >
2	< { E N O S } >
3	< { A B D E N O R S } >
4	< { A D N R S T } >
5	< { E M R } >

a) Wieviele 2-Sequenzen würde der GSP-Algorithmus mit dieser Eingabe als Kandidaten erzeugen?

Lösung:

Der GSP-Algorithmus erzeugt alle Kandidaten, die der Apriori-Algorithmus (im Fall der Transaktionsdatenbank) ebenfalls erzeugen würde (vgl. Blatt 1): D.h. alle zehn 2-Sequenzen bestehend aus einem Element der Form $< \{i\ j\} >$ mit zwei Frequent Items i und j, wobei $i < j$ gemäß der Item-Ordnung gilt.

Außerdem werden $5^2 = 25$ 2-Sequenzen mit zwei Elementen der Form $< \{i\}\ \{j\} >$ erzeugt; hier kann auch $i \geq j$ gelten!

Insgesamt werden also $10 + 25 = 35$ Kandidaten für 2-Sequenzen vom GSP-Algorithmus erzeugt.

Die letzten (25) Kandidaten haben allerdings keinen Support, da die Datenbank nur Sequenzen der Länge 1 enthält, die wiederum keine Teilsequenzen, die länger als 1 Element sind, enthalten können. Somit kann es höchstens häufige Teilsequenzen der Länge 1 geben.

b) Wie lauten die häufigen Teilsequenzen für diese Datenbank?

Lösung:

Alle häufigen Teilsequenzen ergeben sich auf triviale Weise (wie oben) aus dem Ergebnis des Apriori-Algorithmus für die Transaktionsdatenbank (siehe Blatt 1).

Aufgabe 3 *(Zeitliche Constraints für Sequenzen)*

Gegeben sei die Sequenz < {A} {B} {C} {A B} {A C} {B C} >. Nehmen Sie an, dass die Elemente der Sequenz an fortlaufenden Zeitpunkten auftreten.

Welche der folgenden Sequenzen sind Teilsequenzen der obigen Sequenz und erfüllen die zeitlichen Constraints max-gap = 2 bzw. max-span = 4 (mit windowsize=0) ?

Nr.	Sequenz
1	< {A} {C} {A C} {B C} >
2	< {B} {A C} >
3	< {A B} {A B C} {A B C} >

Was ändert sich mit windowsize = 1 ?

Lösung:

Mit windowsize = 0 (und min-gap = 0):

Sequenz Nr. 1:

ist Teilsequenz der Ausgangssequenz — Zuordnung:
{ A } → { A } (Position 1), { C } → { C } (Position 3),
{ A C } → { A C } (Position 5) und { B C } → { B C } (Position 6).

gaps(Seq. 1) = $\{(3-1),(5-3),(6-5)\} = \{1,2\} \leq 2$: max-gap erfüllt
span(Seq. 1) = $6-1 = 5 > 4$: max-span nicht erfüllt

Sequenz Nr. 2:

ist Teilsequenz der Ausgangssequenz – eine mögliche Zuordnung:
{ B } → { A B } (Position 4), { A C } → { A C } (Position 5).

Und damit gaps(Seq. 2) = $\{1\} \leq 2$ sowie span(Seq. 2) = $1 \leq 4$; Constraints erfüllt.
(Bei Nichterfüllung müssten alle möglichen Zuordnungen überprüft werden.)

Sequenz Nr. 3:

Keine Teilsequenz der Ausgangssequenz, da keine Zuordnung möglich.

Mit windowsize = 1 (und min-gap = 0):

Sequenzen Nr. 1 uns 2: wie vorher

Sequenz Nr. 3: neue Zuordnung möglich:
{ A B } → { A } ∪ { B } (Position 1 & 2), { A B C } → { C } ∪ { A B } (Position 3 & 4)
{ A B C } → { A C } ∪ { B C } (Position 5 & 6).

Und damit gaps(Seq. 2) = $\{1\} = 1 \leq 2$, aber maxspan = $6 - 1 = 5 > 4$.

Insgesamt (links windowsize=0, rechts windowsize=1):

Nr.	Teilseq.	max-gap≤ 2	max-span≤ 4	ws-Teilseq.	max-gap≤ 2	max-span≤ 4
1	ja	ja	nein	ja	ja	nein
2	ja	ja	ja	ja	ja	ja
3	nein	—	—	ja	ja	nein

Aufgabe 4 *(Generalized Sequential Pattern (GSP) Algorithmus)*

Gegeben sei nun eine (interessantere) Sequenzdatenbank und ein Minimal-Support von minsup = 0.25:

Nr.	Sequenz
1	$< \{1\}\ \{2\}\ \{3\} >$
2	$< \{1\}\ \{2\ 5\} >$
3	$< \{1\}\ \{5\}\ \{3\} >$
4	$< \{1\}\ \{2\ 5\}\ \{3\ 4\} >$
5	$< \{2\}\ \{3\}\ \{4\} >$
6	$< \{2\ 5\}\ \{3\} >$
7	$< \{3\}\ \{4\}\ \{5\} >$
8	$< \{5\}\ \{3\ 4\} >$

Führen Sie den GSP-Algorithmus gemäß Vorlesung durch, um alle häufigen Teilsequenzen zu finden. Geben Sie für jeden Iterationsschritt k die Menge der Frequent k-Subsequences F_k sowie die Menge der Kandidaten C_k an. (Sie dürfen C_2 weglassen.)

Hinweis: Die angegebene Datenbank ist sehr ähnlich zum GSP-Beispiel aus der Vorlesung.

Lösung:

F_1 :

Seq	σ
$< \{1\} >$	4
$< \{2\} >$	5
$< \{3\} >$	7
$< \{4\} >$	4
$< \{5\} >$	6

C_2 :

Seq	σ
$< \{1\}\ \{1\} >$	0
$< \{1\}\ \{2\} >$	3
$< \{1\}\ \{3\} >$	3
$< \{1\}\ \{4\} >$	1
$< \{1\}\ \{5\} >$	3
$< \{1\ ...\} >$	0
$< \{2\}\ \{1\} >$	0
$< \{2\}\ \{2\} >$	0
$< \{2\}\ \{3\} >$	4
$< \{2\}\ \{4\} >$	2
$< \{2\}\ \{5\} >$	0
$< \{2\ \neq 5\} >$	0
$< \{2\ 5\} >$	3
$< \{3\}\ \{1\} >$	0
$< \{3\}\ \{2\} >$	0
$< \{3\}\ \{3\} >$	0
$< \{3\}\ \{4\} >$	2
$< \{3\}\ \{5\} >$	1
$< \{3\ 4\} >$	2
$< \{3\ 5\} >$	0
$< \{4\}\ \{1\} >$	0
$< \{4\}\ \{2\} >$	0
$< \{4\}\ \{3\} >$	0
$< \{4\}\ \{4\} >$	0
$< \{4\}\ \{5\} >$	1
$< \{4\ 5\} >$	0
$< \{5\}\ \{1\} >$	0
$< \{5\}\ \{2\} >$	0
$< \{5\}\ \{3\} >$	4
$< \{5\}\ \{4\} >$	2
$< \{5\}\ \{5\} >$	0

F_2 :

Seq
$< \{1\}\ \{2\} >$
$< \{1\}\ \{3\} >$
$< \{1\}\ \{5\} >$
$< \{2\}\ \{3\} >$
$< \{2\}\ \{4\} >$
$< \{2\ 5\} >$
$< \{3\}\ \{4\} >$
$< \{3\ 4\} >$
$< \{5\}\ \{3\} >$
$< \{5\}\ \{4\} >$

C_3 :

Set	σ
$< \{1\}\ \{2\}\ \{3\} >$	2
~~$< \{1\}\ \{2\}\ \{4\} >$~~	
$< \{1\}\ \{2\ 5\} >$	2
~~$< \{1\}\ \{3\}\ \{4\} >$~~	
~~$< \{1\}\ \{3\ 4\} >$~~	
$< \{1\}\ \{5\}\ \{3\} >$	2
~~$< \{1\}\ \{5\}\ \{4\} >$~~	
$< \{2\}\ \{3\}\ \{4\} >$	1
$< \{2\}\ \{3\ 4\} >$	1
$< \{2\ 5\}\ \{3\} >$	2
$< \{2\ 5\}\ \{4\} >$	1
$< \{5\}\ \{3\}\ \{4\} >$	0
$< \{5\}\ \{3\ 4\} >$	2

F_3 :

Seq
$< \{1\}\ \{2\}\ \{3\} >$
$< \{1\}\ \{2\ 5\} >$
$< \{1\}\ \{5\}\ \{3\} >$
$< \{2\ 5\}\ \{3\} >$
$< \{5\}\ \{3\ 4\} >$

C_4 :

Set	Anz
$< \{1\}\ \{2\ 5\}\ \{3\} >$	1
~~$< \{1\}\ \{5\}\ \{3\ 4\} >$~~	
~~$< \{2\ 5\}\ \{3\ 4\} >$~~	

$F_4 = \emptyset$

Durchgestrichene Sequenzen enthalten Teilsequenzen, die nicht häufig sind, und fallen somit durch das Pruning weg.