Technische Universität Dresden

Prof. Dr. Sebastian Rudolph

Formal Concept Analysis

Exercise Sheet 7, Winter Semester 2014/15

Exercise 1 (frequent concept intents and closure systems)

Definition (frequent concept intent)**.** *Let* $\mathbb{K} = (G, M, I)$ *be a formal context.*

(a) The support *of a set* $B \subseteq M$ *of attributes in* $\mathbb{K}$ *is given by*

$$\text{supp}(B) := \frac{|B'|}{|G|}.$$

(b) For a given minimal support minsupp the set of frequent concept intents *is given by*

$$\{B \subseteq M \mid \exists A \subseteq G : (A, B) \in \mathfrak{B}(G, M, I) \wedge \text{supp}(B) \geq \textit{minsupp}\}.$$

Show that the set of frequent concept intents together with the set M forms a closure system.

Exercise 2 (support)

Show the validity of the properties of the support function that are employed by the Titanic algorithm:

Let (G, M, I) be a formal context $X, Y \subseteq M$. Then it holds:

1) $X \subseteq Y \implies \text{supp}(X) \geq \text{supp}(Y)$

2) $X'' = Y'' \implies \text{supp}(X) = \text{supp}(Y)$

3) $X \subseteq Y \wedge \text{supp}(X) = \text{supp}(Y) \implies X'' = Y''$

Exercise 3 (computing concept intents with Titanic)

The following context contains transactions in a supermarket. Compute the closure system of all concept intents using the Titanic algorithm. (hint: use the table structure from the example computation in the lecture slides)

	apples (a)	beer (b)	chips (c)	tv magazine (d)	toothpaste (e)
t_1	×	×	×		
t_2			×	×	
t_3		×	×	×	
t_4	×	×			×
t_5			×		×
t_6		×	×	×	
t_7	×	×			
t_8			×	×	

Exercise 4 (optimizing TITANIC for iceberg concept lattices)

In the lecture the following steps to optimize TITANIC for the computation of iceberg concept lattices have been mentioned:

1. Stop, as soon as only *non-frequent* minimal generators are computed.
2. Return only the closures of *frequent* minimal generators.
3. Generate candidates only from the *frequent* minimal generators.
4. All subsets of candidates with $k-1$ elements must be *frequent*.

Implement the corresponding modifications in TITANIC. Utilize the fact that for a formal context $\mathbb{K} = (G, M, I)$ and a minimal support constraint *minsupp* the set of frequent concept intents together with M form a closure system. The corresponding closure operator h and the support function *support* are given by

$$h(X) := \begin{cases} X'', & \text{if } \operatorname{supp}(X) \geq \mathit{minsupp} \\ M, & \text{otherwise} \end{cases} \qquad \mathit{support}(X) := \begin{cases} \operatorname{supp}(X), & \text{if } \operatorname{supp}(X) \geq \mathit{minsupp} \\ -1, & \text{otherwise} \end{cases}$$

Insert the corresponding changes directly into the algorithms attached to this exercise sheet.

Exercise 5 (computing iceberg concept lattices)

We are regarding the following excerpt from the mushroom database:

	edible (e)	poisonous (p)	cap shape: convex (c)	cap shape: flat (l)	cap surface: fibrous (i)
Mushroom 1	×		×		
Mushroom 2	×		×		×
Mushroom 3	×			×	×
Mushroom 4	×			×	×
Mushroom 5	×		×		×
Mushroom 6	×		×		
Mushroom 7		×		×	×
Mushroom 8		×		×	×
Mushroom 9		×		×	×
Mushroom 10		×		×	

a) Compute the corresponding iceberg concept lattice for $\mathit{minsupp} = 30\%$ using the modified algorithm from the previous exercise.

b) Compute the corresponding iceberg concept lattices and label each (frequent) concept with its corresponding support value.

Algorithm 1 TITANIC

1) SUPPORT($\{\emptyset\}$);
2) $\mathcal{K}_0 \leftarrow \{\emptyset\}$;
3) $k \leftarrow 1$;
4) **forall** $m \in M$ **do** $\{m\}.p\_s \leftarrow \emptyset.s$;
5) $\mathcal{C} \leftarrow \{\{m\} \mid m \in M\}$;
6) **loop begin**
7) SUPPORT($\mathcal{C}$);
8) **forall** $X \in \mathcal{K}_{k-1}$ **do** X.closure $\leftarrow$ CLOSURE(X);
9) $\mathcal{K}_k \leftarrow \{X \in \mathcal{C} \mid X.s \neq X.p\_s\}$;
10) **if** $\mathcal{K}_k = \emptyset$ **then exit loop** ;
11) $k++$;
12) $\mathcal{C} \leftarrow$ TITANIC-GEN($\mathcal{K}_{k-1}$);
13) **end loop** ;
14) **return** $\bigcup_{i=0}^{k-1}\{X.\text{closure} \mid X \in \mathcal{K}_i\}$.

Algorithm 2 TITANIC-GEN

Input: $\mathcal{K}_{k-1}$, the set of key $(k-1)$-sets K with their weight $K.s$.

Output: $\mathcal{C}$, the set of candidate k-sets C
with the values $C.p\_s := \min\{s(C \setminus \{m\} \mid m \in C\}$.

The variables p_s assigned to the sets $\{m_1, \ldots, m_k\}$ which are generated in step 1 are initialized by $\{m_1, \ldots, m_k\}.p\_s \leftarrow s_{\max}$.

1) $\mathcal{C} \leftarrow \{\{m_1 < m_2 < \cdots < m_k\} \mid \{m_1, \ldots, m_{k-2}, m_{k-1}\}, \{m_1, \ldots, m_{k-2}, m_k\} \in \mathcal{K}_{k-1}\}$;
2) **forall** $X \in \mathcal{C}$ **do begin**
3) **forall** $(k-1)$-subsets S of X **do begin**
4) **if** $S \notin \mathcal{K}_{k-1}$ **then begin** $\mathcal{C} \leftarrow \mathcal{C} \setminus \{X\}$; **exit forall** ; **end**;
5) $X.p\_s \leftarrow \min(X.p\_s, S.s)$;
6) **end**;
7) **end**;
8) **return** $\mathcal{C}$.

Algorithm 3 CLOSURE(X) for $X \in \mathcal{K}_{k-1}$

1) $Y \leftarrow X$;
2) **forall** $m \in X$ **do** $Y \leftarrow Y \cup (X \setminus \{m\})$.closure; **forall** $m \in M \setminus Y$ **do begin**
3) **if** $X \cup \{m\} \in \mathcal{C}$ **then** $s \leftarrow (X \cup \{m\}).s$
4) **else** $s \leftarrow \min\{K.s \mid K \in \mathcal{K},\ K \subseteq X \cup \{m\}\}$;
5) **if** $s = X.s$ **then** $Y \leftarrow Y \cup \{m\}$
6) **end**;
7) **return** Y.