Universidad de La Sabana
Facultad de Ingenería

Felix Mohr
2020.2

Business Intelligence

- Sheet 10 -

Exercise 1 (4 Points)

1. Show that $\mathcal{M} \subseteq \mathcal{C} \subseteq \mathcal{F}$
2. Prove that the closure operator $c = \mathbf{i} \circ \mathbf{t}$ satisfies the following properties (X and Y are some itemsets): (a) Extensive: $X \subseteq c(X)$, (b) Monotonic: If $X \subseteq Y$ then $c(X) \subseteq c(Y)$ (c) Idempotent: $c(X) = c(c(X))$
3. Prove that

   a) $\mathbf{c}(X_i) = \mathbf{c}(X_j) = \mathbf{c}(X_i \cup X_j)$ if $\mathbf{t}(X_i) = \mathbf{t}(X_j)$ and that

   b) $\mathbf{c}(X_i) = \mathbf{c}(X_i \cup X_j)$ if $\mathbf{t}(X_i) \subseteq \mathbf{t}(X_j)$
4. Show that every rule produced with AssociationRules when being invoked with $\mathcal{F}$ is also produced when being invoked with $\mathcal{C}$ or is confidence-obsolete. Here, a rule $X \to Y$ is said to be confidence-obsolete if it has the same confidence like another stronger rule $X \to YZ$ (with better conclusion) where $Z \neq \emptyset$.

   What do you conclude from this? Do you think that there can be interesting rules based on non-closed itemsets? If yes, give an example. If not, argue why not.

Exercise 2 (4 Points)

1. Write functions `genMax(db, minsup)` and `charm(db, minsup)` that compute all maximal/closed frequent sets. Run `charm` on the normal and the extended shop dataset and compare the runtime to eclat.
2. Compare the runtime and the numbers of produced rules when invoking AssociationRules with the results of charm and eclat. When using the output of charm to produce rules, can you find rules that were not produced but interesting?
3. Write a function `getTransactionsFromWeblog(file)` that takes the weblog file and transforms it into a file `weblog.dat`, which is a file only consisting of transactions with numbers. Merge website visits into one transaction if they come frome the same IP and if the timestamp of the i-th visit is at most 3 minutes later than the $i-1$-th visit. Maybe you need to clean the data a bit before processing it. Apply your algorithms, which are the sets of websites (at least two) which are most often visited together? What is their support?