Exercises Frequent Pattern Mining

Exercise 1: Frequent Item Set Mining

Given are the following six transactions on items $\{A, B, C, D, E\}$:

tid	items
1	ABC
2	ABC
3	BC
4	BD
5	$BCDE$
6	E

(a) Use the Apriori algorithm to compute all frequent item sets, and their support, with minimum support 2. Clearly indicate the steps of the algorithm, and the pruning that is performed.

(b) Use the A-close algorithm to compute all *closed* frequent item sets, and their support, with minimum support 2. Clearly indicate the steps of the algorithm, and the pruning that is performed.

(c) Compute the confidence and the lift of the rule $B \rightarrow C$.

Exercise 2: Constraints

In frequent item set mining, suppose that in addition to the transactions, we also have information about the price of each item. Consider constraints of the type

$$\text{sum}(I.\text{price}) \leq c$$

where sum(I.price) denotes the sum of the prices of the items in item set I, and c is some positive constant. Suppose that we want to find all frequent item sets that also satisfy a constraint of this type.

(a) Can we use this constraint in an Apriori style level wise search to further prune the search space, while still guaranteeing completeness of the results? Explain.

(b) Answer the same question for constraints of type

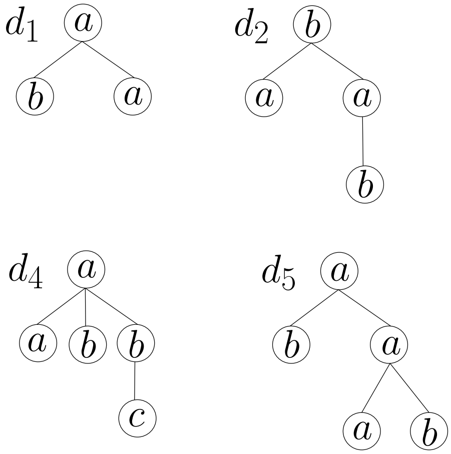
$$\text{avg}(I.\text{price}) \le c$$

where avg(I.price) denotes the average price of the items in item set I, and c is some positive constant.

(c) If your answer to (a) or (b) was "No", could you turn it into "Yes" by imposing a certain order on the items?

Exercise 3: Frequent Tree Mining

Consider the following database of ordered labeled trees:



We use the following string representation of an ordered labeled tree: list the labels according to the pre-order traversal of the tree, and use the special symbol ↑ to indicate we go up one level in the tree. For example, the string representation of d_4 is: $aa \uparrow b \uparrow bc$.

Answer the following questions:

(a) Is $aa \uparrow c$ an induced subtree of d_4? An embedded subtree? If yes, give the corresponding matching function.

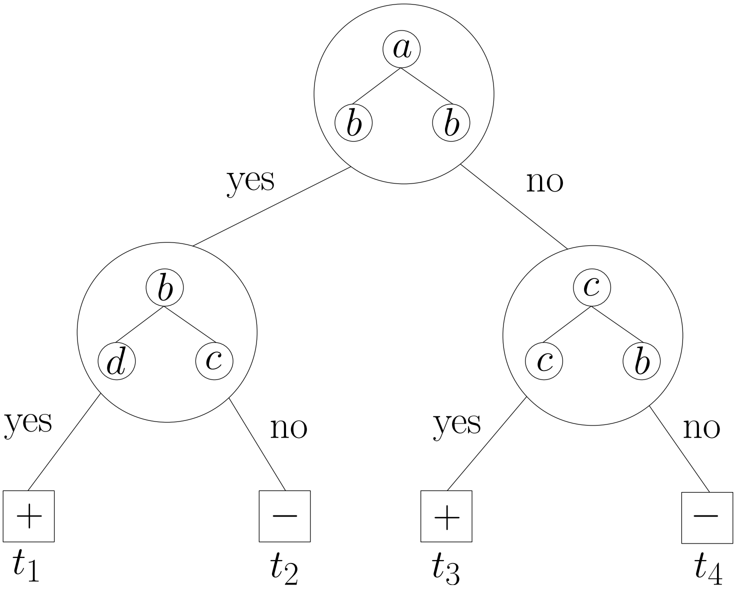
(b) Is d_1 an induced subtree of d_4? If yes, give the corresponding matching function.

Consider the ordered labeled tree $ab \uparrow bb \uparrow\uparrow bb$.

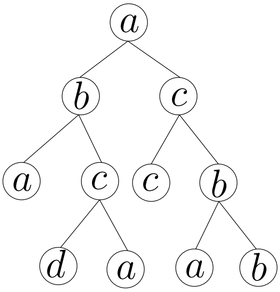
(c) How many times does $ab \uparrow b$ occur as an embedded subtree? As an induced subtree?

Exercise 4: Trees with Trees as Features

Consider the following classification tree:



Inside each node of the classification tree we test whether the pattern tree depicted inside the node is a subtree of the data tree to be classified. Classify the data tree



into the positive or negative class (also indicate the leaf node where the data tree ends up). Do it once using the induced subtree relation, and once using the embedded subtree relation.

Exercise 5: An Interestingness Measure

Consider the association rule interestingness measure

$$\text{novelty}(X \to Y) = s(x \wedge y)s(\neg x \wedge \neg y) - s(x \wedge \neg y)s(\neg x \wedge y)$$

Here x denotes the event that all items in X are bought in a transaction, and $s(E)$ denotes the support (expressed as a fraction) of event E, that is, the number of transactions in which event E occurs divided by the total number of transactions. The symbol $\neg$ denotes negation ("not"), and the symbol $\wedge$ denotes conjunction ("and").

(a) Suppose we have computed the collection of all frequent item sets and their support. We also know the total number of transactions contained in the database. Is this information sufficient to compute the novelty of every association rule that can be generated from these frequent item sets? If your answer is "Yes", explain how it can be done. If your answer is "No", explain why it is impossible.

(b) What is the value of novelty($X \to Y$) if the events x and y are independent in the database?

Exercise 6: Frequent Item Sets: Two Proofs

Prove the following:

(a) The set of maximal frequent item sets is a subset of the set of closed frequent item sets.

(b) For any association rule $X \to Y$, if we move an item from X to Y, then the confidence can never go up.

Could this property be used as a basis for pruning the search space when generating all association rules with confidence $\geq t_2$?