File: homework3.pdf
Author: P.J. Leyden
Date: October 21st, 2019

Homework 3

1. {Bread, Milk, Coke}
   a. We exclude {} → {Bread, Milk, Coke} and {Bread, Milk, Coke} → {}
      1. {Bread} → {Milk, Coke}
      2. {Milk} → {Bread, Coke}
      3. {Coke} → {Bread, Milk}
      4. {Milk, Coke} → {Bread}
      5. {Bread, Coke} → {Milk}
      6. {Bread, Milk} → {Coke}

   b. s = support and c = confidence
      1. s = 0.4, c = 0.5
      2. s = 0.4, c = 0.5
      3. s = 0.4, c = 0.66
      4. s = 0.4, c = 0.66
      5. s = 0.4, c = 1
      6. s = 0.4, c = 0.66

2. Anti-monotonicity describes the property that, if an itemset X violates some constraint C so do all of X's super-sets. In terms of confidence, all rules that are derived from the same itemset have the monotonic property.
   This property can be used in the apriori algorithm by pruning the lattice. If we create a lattice of rules L, with levels starting at the top with L_0, and increasing, we can start checking the confidence of each rule in each level. We can prune however, by not checking subsets of rules who confidence falls below some *minimum confidence threshold.*

3. *Discovery of Multiple-Level Association Rules from Large Databases* [1]
   a. Motivation
      1. Han and Fu state that applications exist, that would greatly benefit from a deeper association. They provide a simple example described in Figure 1.

"80% of customers that purchase milk may also purchase bread."
"75% of people buy wheat bread if they buy 2% milk"

Table 1: Association examples

The first part of table 1 shows a single level association, while part 2 shows a multi-level association.

   b. Problem Definitions
      1. The problem that this paper seeks to solve is the lack of methods for mining multi-level association rules.
      2. They define a few basic terms:

1. A pattern ***A*** is one item $\boldsymbol{A_i}$ or a series of conjunctive items $\boldsymbol{A_i} \wedge \ldots \wedge \boldsymbol{A_j}$
2. The support of ***A*** in ***S*** is $\sigma(\boldsymbol{A/S})$, the confidence of $\boldsymbol{A \rightarrow B}$ is $\varphi(\boldsymbol{A \rightarrow B/S})$
3. Specifically, $\sigma(\boldsymbol{A/S})$ is the number of transactions in ***S*** that contain ***A*** versus the total number of transactions.
4. $\varphi(\boldsymbol{A \rightarrow B/S})$ is the number of transactions that contain ***A*** and ***B*** versus the number of transaction that contain ***A***.
5. A pattern ***A*** is large in a set ***S*** at level *l* if the support of ***A*** is no less than it's corresponding minimum support threshold.
6. A rule is strong if, for a set ***S,*** each ancestor of every item in ***A*** and ***B,*** if any, is large at its corresponding level.

c. Solutions
   1. The solution presented by the paper uses a hierarchy-encoded transaction table. What this means is that every item in a transaction is encoded to a sequence of numbers based on the levels in a taxonomy of relevant data items. What this means is that every relevant item in any of the transactions is categorized in a tree like structure where a read from the root to a leaf is a single relevant item. Figure 1 is an image taken from Han and Fu showing the taxonomy hierarchy [1].

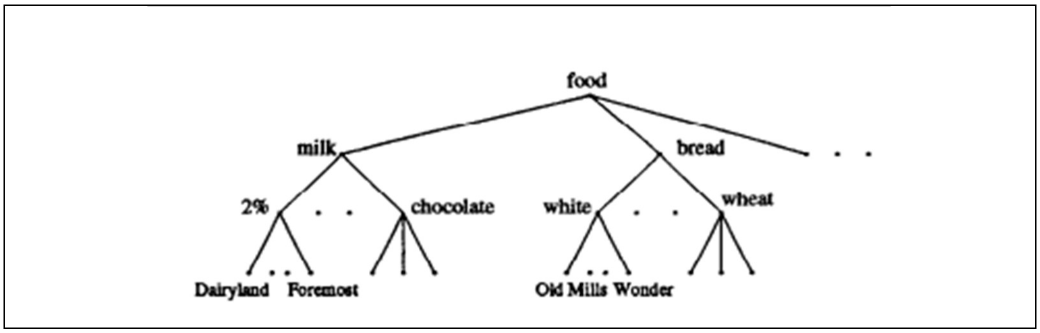

Figure 1 - Taken from Han and Fu paper, a tree structure representing the taxonomy of relevant data items.

   2. An example of an encoding
      1. Old Mills White Bread would be {2, 1, 1}. The 2 is for bread whish is the second leaf of the root. The first 1 is for white and the second 1 is Old Mills. Therefore, in a particular transaction Old Mills White Bread would be replaced with its encoded ID.
   3. The steps for applying this algorithm are as follows:
      1. Generate the encoded transaction table ***T1***.
      2. Use ***T1*** to determine the Level-1 Large 1-Itemset ***L[1, 1]***. This represents a list all the itemsets of size 1 who fit the definition of large above.
      3. ***L[1, 1]*** is then used to prune ***T1*** and generate ***T2***.
      4. We then generate all of the possible sets from ***L[1, 2] … L[L, |L[1, 1]|]***. These sets are again used to trim ***T2*** to ***T3*** etc…
      5. We repeat until done and start on ***L[2, *]***. Candidates for this table can only come form decedents of the large items at level-1.
      6. Once calculated, these tables represent multi-level associations, thus achieving a multi-level association mining algorithm.

References

[1] Han, J. and Fu, Y. (1995). Discovery of Multiple-Level Association Rules from Large Databases. In: *21th International Conference on Very Large Data Bases*. San Francisco, California: Morgan Kaufmann Publishers Inc., pp.420-431.