



Ecole Polytechnique de Louvain

LINGI2364: Mining Patterns in Data

Project 1: Implementing Apriori

Group 11

Alessandra Rossaro (01211800), Matteo Salvatore (01731800)

Version 1.0 - 21/10/2018

AY 2018-2019

1 Implementations

1.1 Apriori

To implement the Apriori algorithm we have created ad hoc implementation of an HashTree that is used to contain the internal structure of the Apriori search. Every node of the HashTree is characterized with an HashMap containig its children nodes and the frequency of the node. We have chosen this data structure in order to simplify the accesses of the algorithm to the elements of the SearchTree since the complexity of the access is $O(1)$.

Since the Apriori algorithm, in order to expand a new level, needs the previous one, during the Database reading process, we have saved in a TreeMap where the keys are all the patterns with a singular element and the values are the corresponding supports, computed incrementally during the reading process.

This is the only improvement that we have performed on the *DataSet* class that we have renamed into *AprioriDataSet*. For the next levels the search is performed accordin to the Apriori algorithm specifications.

We have then implemented the following optimizations:

- we generate candidates according to the *Merging itemsets technique*: when we expand nodes to generate new candidates we take the nodes with same father of the node that is being expanded and we select as candidates only those have keys with value higher w.r.t. the current node's key.
- after the nodes generation we prune the infrequent candidates by computing their frequency. Those nodes are not considered for the new levels' expansion.
- our HashTree data structure is implemented as a *Prefix Tree* to make the merging more efficient, the order of the items in the tree is the natural Integer order.

1.2 ECLAT

2 Performances