CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets

by Jian Pei, Jiawei Han and Runying Mao

-Presentation-
Cmput 695
by Luiza Antonie

Introduction

Association mining often derives a large set of frequent itemsets and association rules.

This paper presents a new algorithm,called CLOSET for mining closed itemsets.

CLOSET proves to be efficient, scalable over large databases and faster than other proposed methods.

Some Definitions

- Itemset
- Transaction
- Transaction database
- Support
- Association rule
- Support and confidence of a rule

Some Definitions

- Frequent itemset
- Frequent closed itemset
- Association rules on frequent closed itemsets

Example

Let's consider the following transaction database:

Transaction ID	Items in transaction
10	a,c,d,e,f
20	a,b,e
30	c,e,f
40	a,c,d,f
50	c,e,f

Given the min_sup=2 and min_conf=50% let's try to mine this small database.

According to the Apriori algorithm the frequent itemsets that are found in this database are the following:

1-itemsets: a, c, d, e, f,

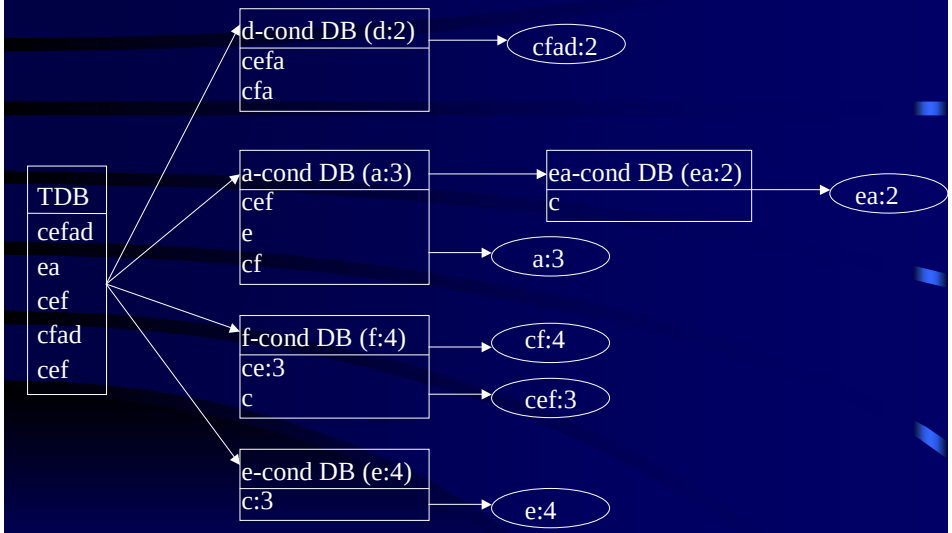
2-itemsets: ac, ad, ae, af, cd, ce, cf, df, ef,

3-itemsets: acd, acf, adf, cef, cdf,

4-itemsets: acdf

Among all these 20 frequent itemsets there are only 6 frequent closed itemsets.

f_list < c:4, e:4, f:4, a:3, d:2 >



Optimization 1

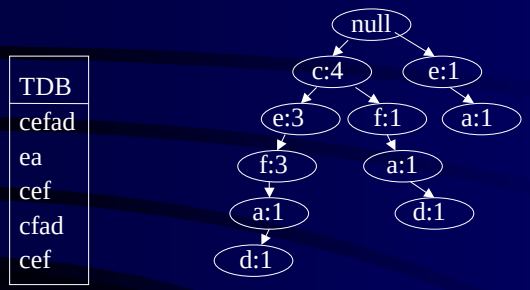
Compress transactional and conditional databases using an FP tree structure.

Advantages:

• FP tree compresses databases for frequent itemset mining;

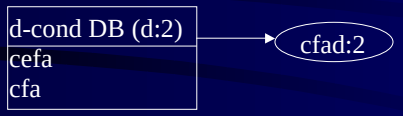
•conditional databases can be derived efficiently from FP tree.

Optimization 1



Optimization 2

Extract items appearing in every transaction of conditional database. It takes effect when forming the conditional databases.



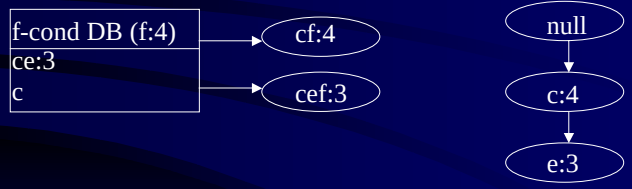
Advantages:

•it reduces the size of FP tree because the conditional database contains less number of items after such extraction;

•it may reduce the level of recursions since it combines a few items into one.

Optimization 3

Directly extract frequent closed itemsets from FP tree.

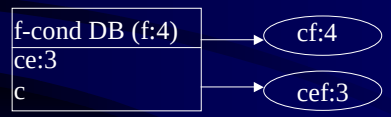


Optimization 4

Prune search branches.

Let X and Y be 2 frequent itemsets with the same support. If X is included in Y and Y is a FCI, then there is no need to search the X-cond DB because there is no hope to generate FIC from there.

Ex:



Because of ‘cf’ which is a CFI c-cond DB is no more searched.

Scaling up CLOSET in large databases

FP tree improves substantially the CLOSET algorithm.

If the database is large it is unrealistic to construct a main memory based FP tree.

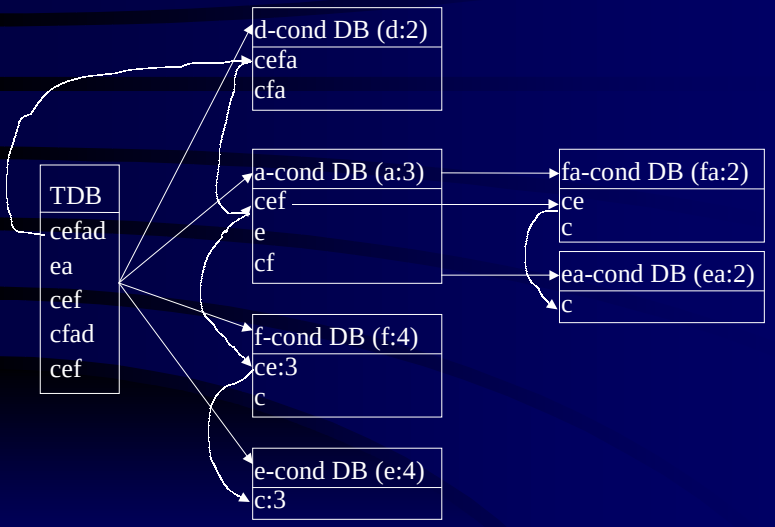
In such a case there are 2 possibilities:

•construct conditional databases without FP tree

•construct disk-based FP tree

This article presents a method of constructing conditional databases using a partitioned-based approach.

Example



Performance Study

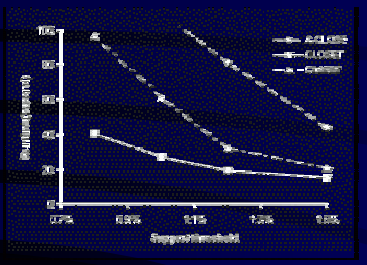
CLOSET was compared to CHARM and A-close.

There were tested on a synthetic dataset and on 2 real datasets.

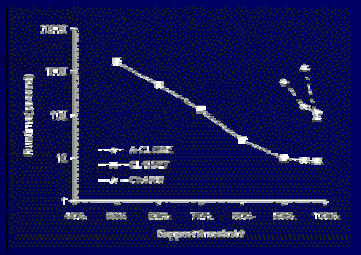
The results showed that the number of frequent itemsets was reduced by an order of magnitude using CLOSET.

Comparison of A-close, CHARM and CLOSET

As it can be seen in the following graphs CLOSET performs better than the other 2 algorithms.



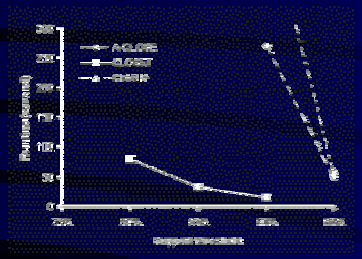
Scalability with support threshold on sparse dataset T25I20D100K



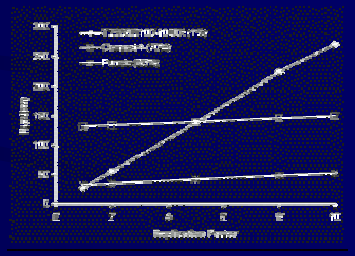
Scalability with support threshold on dense dataset Connect-4

Comparison of A-close, CHARM and CLOSET

As it can be seen in the following graphs CLOSET performs better than the other 2 algorithms.



Scalability with support threshold on dense dataset pumsb



Size scaleup on dataset

Conclusions

Mining a complete set of items can often generate a very large number of frequent items and association rules.

CLOSET performs better than other algorithms.

The results obtained are the same as those using frequent itemsets for mining the whole dataset, but the amount of data is greatly reduced.