Global Journal of Computer Sciences: Theory and Research

Volume 11, Issue 1, (2021) 01-11 www.gjcs.eu

Closed frequent item sets mining based on IT-Tree

Youssef Fakir[a*1] , Faculty of Sciences and Technics, Sultan Moulay Slimane University, Av Med V, BP 59 Beni-Mellal 23000, Morocco fakfad@yahoo.fr http://orcid.org/0000000267409594
Chaima Ahle Touateb[b], Faculty of Sciences and Technics, Sultan Moulay Slimane University, Av Med V, BP 59 Beni-Mellal 23000, Morocco. https://orcid.org/0000-0002-0068-1729
Rachid Elayachi[c], Faculty of Sciences and Technics, Sultan Moulay Slimane University, Av Med V, BP 59 Beni-Mellal 23000.Morocco http://orcid.org/0000-0001-9144-0316

Suggested Citation:
Fakir, Y., Touarate, C., A. & Elayachi, R.(2021). Closed frequent item sets mining based on IT- Tree. *Global Journal of Computer Sciences: Theory and Research*. *11*(1),01-11. https://doi.org/10.18844/gjcs.v11i1.4912

Received from December 9,2021; revised from Februaury 19,2021; accepted from April 22,2021.
Selection and peer review under responsibility of Asist. Prof. Dr. Ezgi Pelin YILDIZ, Kafkas University, Turkey.

Abstract

In the last decade, the amount of collected data, in various computer science applications, has grown considerably.These large volumes of data need to be analysed in order to extract useful hidden knowledge. This study focuses on association rule extraction. This technique is one of the most popular in data mining. Nevertheless, the numberof extracted association rules is often very high, and many of them are redundant. This paper aims to propose an algorithm for mining closed itemsets, with the construction of an Itemset-Tidset Search Tree (IT-Tree). This algorithm is compared with the Direct Counting & Intersect (DCI) algorithm based on min support and computingtime. CHARM needs to store all closed itemsets in the memory. The lower the min-sup is, the more frequent closed itemsets there are so that the amounts of memory used by CHARM are increasing.

Keywords: Data mining, Association rules, frequent closed itemset, CHARM, DCI.

[1] ADDRESS OF CORRESPONCE: **Youssef Fakir**[a*1] , Faculty of Sciences and Technics, Sultan Moulay Slimane University
Email Address: fakfad@yahoo.fr

1. INTRODUCTION

The field of data mining appeared with the promise of providing the tools and techniques to discover useful and previously unknown knowledge in the data fields. Data mining has been adopted for research dealing with the automatic discovery of implicit information or knowledge within the databases [1]. The implicit information contained in databases, principally the interesting association among sets of objects mayreveal useful patterns for decision support, marketing policies, financial forecast, medical diagnosis and manyother applications [2]. Figure 1 illustrates a flow chart of datamining techniques.

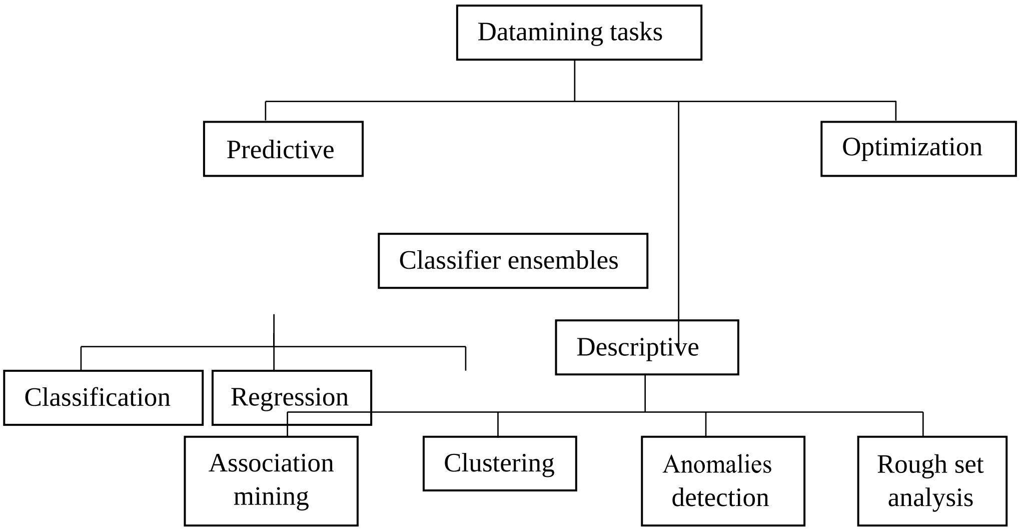

Figure 1: Datamining techniques

The issue of mining frequent itemsets emerges first as a sub problem of mining association rules. Frequent itemsets play a vital role in many data mining tasks that try to find compelling patterns from databases such as association rules, classifiers, correlations, clusters, sequences, and many more, of which the mining of association rules is one of the most common problems [3,4,5]. Mining frequent itemsets or patterns is a fundamental and indispensable problem in numerous data mining applications.

The original reason for searching association rules came from the need to analyse supermarket transaction data, that is, to examine client's behaviour in terms of the purchased products. Association rules describe how often items are bought together. For example, an association rule: {milk-oil (76%)} assert that 6 out of 7 clients that bought milk also bought oil. Such rules can be practical for decisions about product pricing, promotions, store arrangement and many others. Association rule mining (ARM)[1] is one of the most famous techniques of data mining and has received a wide attention in many areas. ARM technique has been first introduced by Agrawal et al. in 1993 [1] and they have described the formal model of association rule mining problem as follows.

Let I = {i1, i2, i3, i4, i5, …, im} be a finite set of items. An itemset is defined as a collection of zero or more items of I while k-itemset contains k items of I. Let D = {T1, T2, T3, …, Tn} be a finite set of transactions called datasets. Each transaction Ti in database D is an itemset such that Ti ⊆ I. Let X be a subset of set of items I, a transaction Ti contains X if X ⊆ Ti. The support of an itemset X is is given by

$$\text{supp}\ (X) = \text{supcount}\ (X) / n \qquad (1)$$

where n is total number of transaction in database and supcount (X) = $|\{T_i \mid X \subseteq T_i, T_i \in D\}|$.

An association rule is a conditional implication of the form $X \rightarrow Y$ where $X, Y \subset I$ are itemsets and $X \cap Y = \emptyset$. The strength of the rule is measured in terms of support and confidence denoted by supp (X ⍰ Y) and conf ($X \rightarrow Y$) respectively and defined as

$$\text{supp}\ (X \rightarrow Y) = \text{supcount}\ (X \cup Y) / n \qquad (2)$$

$$\text{conf}\ (X \rightarrow Y) = \text{supcount}\ (X \cup Y) / \text{supcount}\ (X) \qquad (3)$$

The search for association rules has been oriented towards two objectives:

a. Determine the set of frequent itemsets [2] that appear in the database with support greater than or identical to minsup. The problem of the extraction of frequent itemsets is of exponential complexity in the size m of the set of items as the number of potential frequent itemsets is 2^m.

b. Generate the set of associative rules, from these frequent itemsets, with a confidence measure greater than or identical to minconf. Indeed, the time of this phase is very small compared to the cost of extracting frequent itemsets because the generation of association rules is a problem that depends exponentially on the size of set in a frequent itemsets. Once all frequent itemsets and their support are known, the association rule generation is straightforward. Hence, the problem of mining association rules is reduced to the problem of determining frequent itemsets and their support.

In this paper, we show that it is not requisite to mine all frequent itemsets to guarantee that all non-redundant association rules will be found. Therefore, we are going to discuss two approaches. Before that, some definitions are given:

Definition 1 (Frequent closed itemset) An itemset X is a closed itemset if there exists no itemset X1 such that X1 is a proper superset of X, and every transaction containing X also contains X1. A closed itemset X is frequent if its support passes the given support threshold.

Thus, instead of mining association rules on all the itemsets, one can mine association rules on frequent closed itemsets only.

Definition 2 (Association rule on frequent closed itemsets) Rule $X \rightarrow Y$ is an association rule on frequent closed itemsets if (1) both X and XUY are frequent closed itemsets, (2) there does not exist frequent closed itemset Z such that $X \subset Z \subset (XUY)$, and (3) the confidence of the rule passes the given confident threshold.

Similar to mining association rules, the complete set of association rules on frequent closed itemsets can be mined in a two-step process: (1) mining the set of frequent closed itemsets with min sup, and (2) generating the complete set of association rules on the frequent closed itemsets with min conf.

The two approaches are as follows:

a. Approach based on the discovery of "closed" itemsets, coming from the theory of formal concepts propose to generate only a compact and generic subset of associative rules. This subset is much smaller than the size of the set of all rules. We show that it is sufficient to consider only the closed frequent itemsets. Moreover, all non-redundant rules are found by only considering rules among the closed frequent itemsets. The set of closed frequent itemsets is much smaller than the set of all frequent itemsets. This approach proposes to reduce the cost of extracting frequent itemsets based on the fact that the set of frequent closed itemsets is a generating set of the set of frequent itemsets. This approach makes it possible to decrease the number of extracted rules by keeping only the interesting ones to give the possibility to better visualize them and exploit them.

b. Approach that uses maximal frequent itemsets: A maximal set of elements is a frequent set of elements that is not included in an appropriate superset that is a common set of elements. The set of frequent maximal items is therefore a subset of the set of frequent closed items, which is a subset of frequent itemsets. That makes the set of frequent maximum items usually a lot smaller than the set of frequent items and smaller than the set of frequentclosed items.

This paper also gives the comparison of algorithms based on execution time and support value.

1.1. *Purpose*

In the last decade, the amount of collected data, in various computer science applications, has grown considerably. These large volumes of data need to be analysed in order to extract useful hidden knowledge. This study aims to focus on association rule extraction as one of the most used extraction methods. The research therefore conducted an experiment. In this paper, we propose an algorithm, for mining closed itemsets, with the construction of an Itemset-Tidset Search Tree (IT-Tree).

2. Methods

2.1. Charm Algorithm

After developing the main ideas behind closed association rule mining, we now present CHARM [4], an efficient algorithm for mining all the closed frequent itemsets. First, we will describe the algorithm in general terms, independent of the implementation details. Later we will show how the algorithm can be implemented successfully.

Developed by Zaki and al [6] CHARM Algorithm is an efficient algorithm for enumerating all closed elements.A number of innovative ideas are being used in the development of CHARM, which have made it the choice forever for the extraction of frequent closed itemsets among the benefits of CHARM:

- CHARM simultaneously explores the item space and the transaction space, above a new IT-tree [6,7] search space (tree of itemsets-tidset). On the other hand, most methods use only the item search space.

- CHARM uses a highly efficient hybrid search method that ignores multiple levels of the computer tree to quickly identify frequent closed-element sets, instead of having to enumerate many possible subsets.

- It uses a hash-based fast approach to remove non-closed items when checking for under-consumption.

- CHARM also uses a new vertical representation of data called diffset [7], for fast frequency calculations. Diffsets keep track of differences in the details of a candidate pattern from its prefix pattern. The diffsets significantly reduce (in order of magnitude) the memory size needed to store the intermediate data.

The CHARM algorithm goes through 3 phases:

a. Enumeration of closed sets using a double tree of itemset-tidset (itemset -transaction identification set) search.

b. Using the technique called diffsets to reduce the memory footprint of intermediate calculations.

c. Finally, uses a hash-based fast approach to remove all "unclosed" sets found during the calculation.The pseudo algorithm of CHARM is shown in Table 1.

Table 1: Charm algorithm

```
Input: K: extraction context, minsup

Output: FC: Set of frequent closed itemsets

$[P]= \{X_i \times (X_i)^J : X_i \in I$ A $support(X_i) \geq minsup\}$
CHARM-EXTEND ([P], FC=∅
return FC
```

Table 2: Charm-Extend

```
CHARM-EXTEND. Input : [P], FCOutput: FC

for all $X_i \times (X_i)^J \in [P]$ do
[P]=∅ and $X = X_i$
    for all $X_j \times (X_j)^J \in [P]$ A $X_j \geq X_i$ do
    $X = X \cup X_i$
    $Y = (X_i)^J \cap (X_j)^J$
CHARM PROPERTY ([P], [Pi])
  if $[P_i]f = \emptyset$ then
    CHARM-EXTEND ([Pi], FC)
    Delete ([Pi])
    FC = FC U X
  end if
```

```
    end for
end for
return FC
```

Table 3: Charm-Property

```
Input: [P], [Pi]
Output: [P]
if support(X) ≥ minsup then
    if (Xi)J = (Xj)J then
        delete Xj from [P]
        replace all Xi with X
    else
        if (Xi)J ⊂ (Xj)J then
            replace all Xi with X
        else
            if (Xi)J ⊃(Xj)J then
                replace Xj from [P]
                add X × Y to [Pi]
            else
                if (Xi)J ≠ (Xj)J then
                    add X × Y to [Pi]
                end if
            end if
        end if
    end if
end if
return [P]
```

CHARM begins by initializing the class of prefixes [P] of the nodes to be examined by the frequent 1-itemsets and their associated tidsets (transaction identification set). The two generic steps are instantiated as follows:

- **Pruning step**: This step is implemented via the CHARM-PROPERTY procedure (Table 3). This procedure canmodify the current class [P] by deleting IT-pairs or by inserting new ones in [Pi]. An IT pair is first pruned compared to minsup. Then, we check if it is maximum or not. To do this, just check that its Tidset is included in that of the pair that generated it. Once all the IT-pairs have been processed, the new class [Pi] is recursively explored in depth first, by calling the CHARM-EXTEND procedure (Table 2).

- **Construction step**: this stage is implemented via the CHARM-EXTEND procedure. It combines the IT-pairs,which appear in the class of prefixes [P]. For each IT pair Xi × (Xi)^J, it combines it with other IT pairs Xj × (Xj)^{J}following it in lexicographic order. Each Xi will generate a new class of prefixes [Pi], which would initially beempty. The two IT-pairs combined will produce a new pair X × Y, where X = Xi U Xj and Y= (Xi) $\cap^J$ (Xj)^J. Finally, the algorithm gives in its output FC (The Set of Frequent Closed Itemsets) that we seek.

We illustrate the CHARM algorithm on the following example that describes purchased products in an electronicsstore (Table 4 and Table 5) by choosing a minisupport =3.

Table 4: Itemsets

Scanner	PC	Notebook	Laser	Printer
A	C	D	T	W

Table 5: Transactions example

Transaction Id	Purchased items
1	A C T W
2	C D W
3	ACTW
4	ACDW
5	A C D T W
6	C D T

In Table 5 we describe the database in horizontal format, each record is a required set. A separate number namedtransaction ID is assigned to each record. Table 6 shows the database in vertical format, where each record is a transaction identifier set relating to an article. This item appears in these transactions. This format will help us during the process of making the IT-tree (itemset-tidset tree). Table 7 represents the items and their apparition in transaction of table4.

Table 6: Vertical format database (left), Binary representation (right)

A	C	D	T	W	TID	A	C	D	T	W
1	1	2	1	1	1	1	1		1	1
3	2	4	3	2	2		1	1		1

4	3	5	5	3	3	1	1		1	1
5	4	6	6	4	4	1	1	1		1
	5			5	5	1	1	1	1	1
	6				6		1	1	1	

Table 7 : Purchased items and their apparition in transaction.

Purchased items	Apparition in	count	Purchased items	Apparition in	Count
A	1345	4	DW	245	3
C	123456	6	TW	135	3
D	2456	4	ACD	45	2
T	1356	4	ACT	135	3
W	12345	5	ACW	1345	4
AC	1345	4	CDT	56	2
AD	45	2	CDW	245	3
AT	135	3	CTW	135	3
AW	1345	4	DTW	5	1
CD	2456	4	ACDT	5	1
CT	1356	4	ACDW	45	2
CW	12345	5	CDTW	5	1
DT	56	2	ACDTW	5	1

Let Itemset X, t (X) be the set of all tidset that contains X. CHARM searches for frequent closed sets on a new search space in the IT-tree where each node is a pair X × t (X), for example: AT × 135. All children in node X share the same X prefix and belong to the same equivalence classes. According to these, we can set our It-tree as illustrated in figure 2 by using Table 7.

Initially we have five branches, corresponding to the five items and their tidset from our example database. To generate the children of item *A* (or the pair *A 1345*) we need to combine it with all siblings that come after it. When we combine two pairs *X1 t(X1)* and *X2 t(X2)*, we need to perform the intersection of corresponding tidset whenever we combine two or more itemsets that is how we got the It-tree above. After sitting our new search space now, we proceed with the charm algorithm steps.

When we try to extend *A* with *C*, we find that *t(A)=1345 ⊂ t(C)=123456*. According to CHARM-PROPERTY we can thus remove *A* and replace it with *AC* combining *A* with *D* produces an infrequent set *ACD*, which is pruned. Combination with *T* produces the pair *ACT 135*. When we try to combine *A* with *W,we* find that *t(A) ⊂ t (W)*, we replace all unpruned occurrences of *A* with *AW.* Thus, *AC* becomes *ACW* and *ACT* becomes *ACT W.* At this point there is nothing further to be processed from the *A* branch of the root.

We now start processing the *C* branch. When we combine *C* with *D,* we observe that *t(C) ⊃ t(D)*. This means that wherever *D* occurs *C* always occur. Thus, *D* can be removed from further consideration, and the entire *D* branchis pruned, the child *CD* replaces *D*. Exactly the same scenario occurs with *T* and *W.* Both the branches are pruned and are replaced by *CT* and *CW* as children of *C*. Continuing in a depth-first manner, we next process the node *CD*. Combining it with *CT* produces an infrequent itemset *CDT,* which is pruned. Combination with *CW* produces *CDW*. Similarly, the combination of *CT* and *CW* produces *CT W.* At this point all branches have been processed. Finally, we remove *CTW 135* since it is contained in *ACT W 135*. As we can see, in just 10 steps we haveidentified all 7 closed frequent itemsets. The routine CHARM-PROPERTY simply tests if a new pair is frequent,discarding it if it is not. It then tests each of the four basic properties of itemset-tidset pairs, extending existing itemsets, removing some subsumed branches from the current set of nodes, or inserting new pairs in the node set for the next (depth-first) step. At the end, we get our new It-tree which now holds only closed frequent itemsets as illustrated in figure 6.

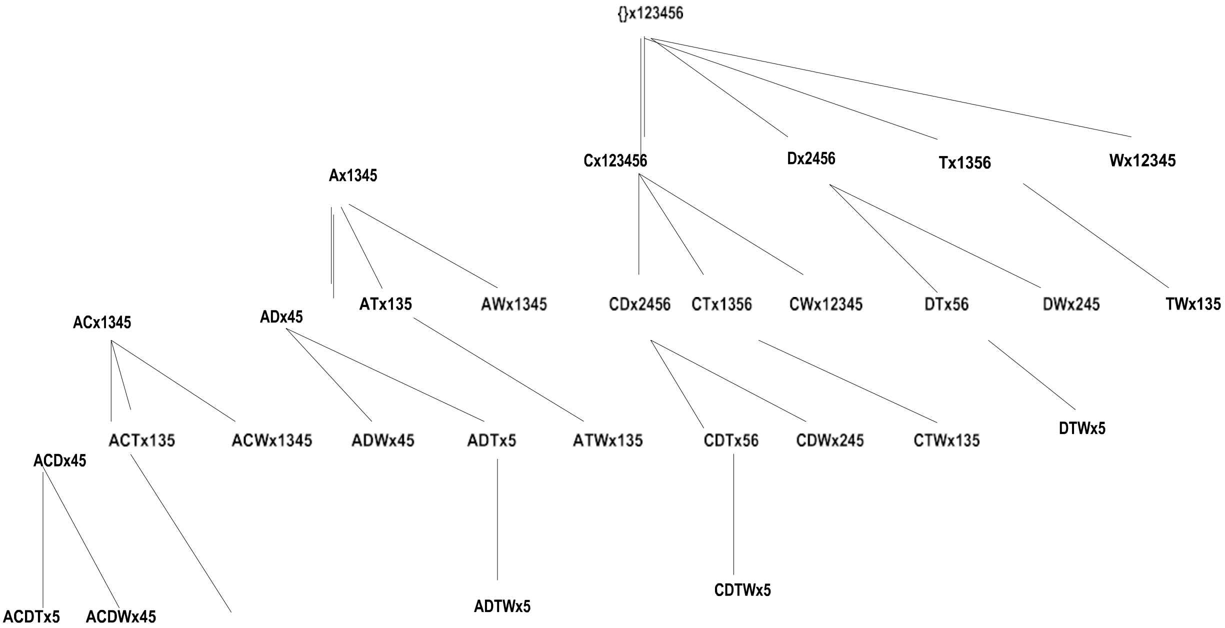

Figure 2: Itemset-Tidset Search Tree

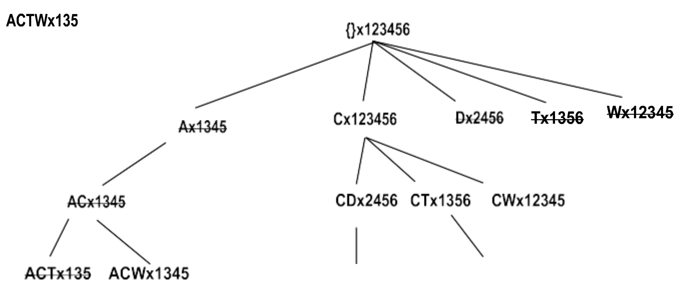

Figure 3: Final IT-Tree

3. Result And Discussion

The Algorithms was coded in Java using the dataset collected from electronics stores. For the comparison we usedDCI-closed (Direct Count & Intersect)-closed [8], a famous algorithm for mining frequent closed itemsets to be compared with CHARM Algorithm. For performance comparison, we used the original source or object code forDCI-closed provided to us by [9].

Figure 4 illustrates the execution time in the data of both algorithms with different minsup. Comparing with DCI-closed, we find that both CHARM and DCI-closed have similar performance for lower minimum support values. However, as we increase the minimum support, the performance gap between CHARM and DCI-closed widens. For example, at the highest support value plotted, CHARM is faster than DCI- closed in execution time, which makes CHARM, outperforms better on higher support than the DCI-closed for our database.

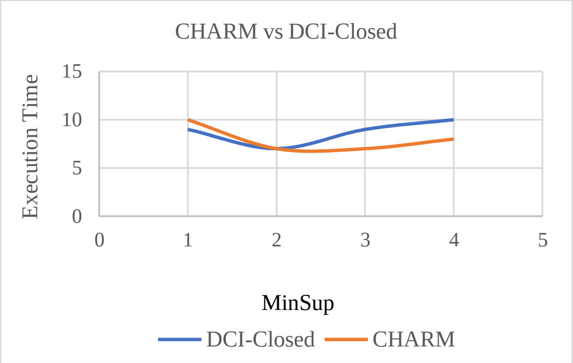

Figure 4: Execution time

4. Conclusion

In this paper, an efficient algorithm (called CHARM) for mining closed frequent itemsets is presented. Using a new IT-Tree framework, this algorithm explores simultaneously the itemset space and tidset space. The IT-Tree skips many levels to identify quickly the closed frequent itemsets. According to the experiment, CHARM perform better on higher minsup compared to the algorithm DCI-Closed for mining closed patterns.

CHARM faces a memory-inefficient challenge since it needs to maintain all closet itemsets in the memory to check if an itemset isclosed or not. For this reason, an improvement of CHARM is necessary. As future work, the researchers intend to optimiseCHARM, in order to solve the issue of memory-inefficiency.

REFERENCES

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules.In U. Fayyad and et al, editors, Advances in Knowledge Discovery and Data Mining, pages 307–328. AAAIPress, Menlo Park, CA, 1996.

[2] M. J. Zaki. Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering, 12(3):372-390, May-June 2000.

[3] M. J. Zaki and K. Gouda. Fast vertical mining using Diffsets. Technical Report 01-1, Computer Science Dept., Rensselaer Polytechnic Institute, March 2001.

[4] Mohamed El far, Lahcen Moumoun, Taoufiq Gadi , An Efficient CHARM Algorithm for indexation 2D/3D and Selection of characteristic views, 5th International Symposium On I/V Communications and Mobile Network, 2010

[5] D. Burdick, M. Calimlim, and J. Gehrke. MAFIA: a maximal frequent itemset algorithm for transactional databases. In Intl. Conf. on Data Engineering, April 2001.

[6] M. J. Zaki and C.-J. Hsiao. CHARM: An efficient algorithm for closed association rule mining. Technical Report 99-10, Computer Science Dept., Rensselaer Polytechnic Institute, October 1999.

[7] Xin Ye, Feng Wei, Fan Jiang and Shaoyin Cheng , An Optimization to CHARM Algorithm for Mining Frequent Closed Itemsets, 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing