Mining of Confidence-Closed Correlated Patterns Efficiently

R Hemalatha A Krishnan C Senthamarai R Hemamalini

Abstract

Correlated pattern mining has become increasingly important recently as an alternative or an augmentation of association rule mining. Though correlated pattern mining discloses the correlation relationships among data objects and reduces significantly the number of patterns produced by the association mining, it still generates quite a large number of patterns. This paper proposes closed correlated pattern mining to reduce the number of the correlated patterns produced without information loss. A new notion of the confidence-closed correlated patterns is proposed first, and then an efficient algorithm is present, called CCMine, for mining those patterns. Confidence closed pattern mining reduces the number of patterns by at least an order of magnitude. It also shows that CCMine outperforms a simple method making use of the traditional closed pattern miner. Confidence-closed pattern mining is a valuable approach to condensing correlated patterns.

Keywords : Data Mining, CC Mine, Database Systems.

0. Introduction

Association mining often generates a huge number of rules, but a majority of them either are redundant or do not reflect the true correlation relationship among data objects. To overcome this difficulty, interesting pattern mining has become increasingly important recently and many alternative interestingness measures have been proposed [1,2,3,415,16,17,18]. While there is still no universally accepted best measure for judging interesting patterns, all confidence is emerging as a measure that can disclose true correlation relationships among data objects [5,6,7,8].

One of important properties of all confidence is that it is not influenced by the co-absence of object pairs in the transactions—such an important property is called null-invariance [8]. The co-absence of a set of objects, which is normal in large databases, may have unexpected impact on the computation of many correlation measures. All confidence can disclose genuine correlation relationships without being influenced by object co-absence in a database while many other measures cannot. In addition, all confidence mining can be performed efficiently using its downward closure property [5].

Although the all confidence measure reduces significantly the number of patterns mined, it still generates quite a large number of patterns, some of which are redundant. This is because mining a long pattern may generate an exponential number of sub-patterns due to the downward closure property of the measure. For frequent itemset mining, there have been several studies proposed to reduce the number of items mined, including mining closed [9], maximal [10], and compressed (approximate) [11] itemsets. Among them, the closed itemset mining, which mines only those frequent itemsets having no proper superset with the same support, limits the number of patterns produced without information loss. It has been shown in [12] that the closed itemset mining generates orders of magnitude smaller result set than frequent itemset mining.

This paper introduces the concept of confidence closed correlated pattern, which plays the role of reducing the number of the correlated patterns produced without information loss. All confidence is used here is correlation measure. However, the result can be easily extended to several other correlation measures, such as coherence [6]. First, propose the notion of the confidence-closed correlated pattern.

Previously used concept is support-closed pattern, i.e., the closed pattern based on the notion of support. However, support-closed pattern mining fails to distinguish the patterns with different confidence values. In order to overcome this difficulty, this paper introduces confidence-closed correlated pattern, which encompasses both confidence and support. Then an efficient algorithm is proposed, called CCMine, for mining confidence-closed patterns. The experimental and performance study shows that confidence-closed pattern mining reduces the number of patterns by at least an order of magnitude. It also shows the superiority of the proposed algorithm over a simple method that mines the confidence-closed patterns using the patterns generated by the support-closed pattern miner.

1. Background

Let I = {i_1, i_2, . . . , i_m} be a set of items, and DB be a database that consists of a set of transactions. Each transaction T consists of a set of items such that T Í I. Each transaction is associated with an identifier, called TID. Let A be a set of items, referred to as an itemset. An itemset that contains k items is a k-itemset. A transaction T is said to contain A if and only if A Í T. The support of an itemset X in DB, denoted as sup(X), is the number of transactions in DB containing X. An itemset X is frequent if it occurs no less frequent than a user-defined minimum support threshold[15,16,118,17).

Generally in data mining, only the frequent itemsets are considered as significant and will be mined. The all confidence of an itemset X is the minimal confidence among the set of association rules i_j®X ® i_j, where i_j ® X. Its formal definition is given as follows. Here, the max_item_sup of an itemset X means the maximum (single) item support in DB of all the items in X.

Definition 1: All-confidence of an itemset

Given an itemset X = {i1, i2, . . . , ik}, the all confidence of X is defined as,

Max_item_sup(X) = max{sup(i_j)|"i_j Î X} (1)

$$\text{All\_conf}(X) = \frac{\text{sup}(X)}{\text{max item sup}(X)} \quad (2)$$

Given a transaction database DB, a minimum support threshold min-_sup and a minimum all_confidence threshold min_a, a frequent itemset X is all_confidence(x) or correlated if all conf(X) ³ min_a and sup(X) ³ min_sup.

2. Confidence Closed Correlated Patterns

It is well known that closed pattern mining has served as an effective method to reduce the number of patterns produced without information loss in frequent itemset mining. Motivated by such practice, extend the notion of closed pattern so that it can be used in the domain of correlated pattern mining. The formal definitions of the original and extended ones are in Definitions 2 and 3, respectively. The former is called as support-closed and the latter is called as confidence-closed.

Definition 2. Support-Closed Itemset : An itemset Y is a support-closed (correlated) itemset if it is frequent and correlated and there exists no proper superset Y' É Y such that sup(Y') = sup(Y).

Since the support-closed itemset is based on support, it cannot retain the confidence information—notice that the confidence means the value of all confidence. In other words, support-closed causes information loss.

Example 1

Let itemset ABCDE be a correlated pattern with support 30% and confidence 30% and itemset CDE be one with support 30 and confidence 80%. How to get a set of non-redundant correlated patterns when min_sup = 20 and min_a = 20% ? Support-closed pattern mining generates ABCDE only eliminating CDE since ABCDE is superset of CDE with the same support. Thus lose the pattern CDE. However, CDE might be more interesting than ABCDE since the former has higher confidence that the latter. Thus extend the support-closed itemset to encompass the confidence so that it can retain the confidence information as well as support information.

Definition 3. Confidence-Closed Itemset : An itemset Y is a confidence-closed itemset if it is correlated and there exists no proper superset Y'ÉY such that sup(Y') = sup(Y) and all_conf(Y') = all_conf(Y).

By applying mining of confidence-closed itemsets to Example 1, obtain not only itemset ABCDE but also CDE as confidence-closed itemsets since they have different confidence values and therefore no information loss occurs. So, call the support-closed pattern as SCP and the confidence closed pattern as CCP, respectively.

3. Mining Confidence - Closed Correlated Patterns

In this section, two algorithms for mining CCPs named CCFilter and CCMine are introduced. CCFilter is a simple algorithm that makes use of the existing support-closed pattern generator. CCFilter consists of the following two steps:

First, get the complete set of SCPs using the previous proposed algorithms [13].

Second, check each itemset and its all-possible subsets in the resulting set whether it is confidence-closed or not.

If its confidence satisfies min_a and it has no proper superset with the same confidence, it is generated as a confidence-closed itemset. CCFilter is used as a baseline algorithm for comparison in Section 5. CCFilter has a shortcoming: It generates SCPs with less confidence than min_a during the mining process. At the end, these patterns are removed. In order to solve this problem, CCMine integrates the two steps of CCFilter into one. Since all confidence has the downward closure property, push down the confidence condition into the process of the confidence-closed pattern mining. CCMine adopts a pattern-growth methodology proposed in [14].

The CLOSET+ [13] and CHARM [15] for mining SCPs, two search space-pruning techniques, item merging and sub-itemset merging, have been mainly used. However, if these techniques are applied directly into confidence-closed pattern mining, a complete set of CCPs cannot be obtained. This is because if there exists a pattern, these techniques remove all of its sub-patterns with the same support without considering confidence. Modify these optimization techniques so that they can be used in confidence-closed pattern mining.

Lemma 1 : Confidence-closed item merging

Let X is a correlated itemset. If every transaction containing itemset X also contains itemset Y but not any proper superset of Y , and all_conf(XY) = all_conf(X), then XY forms a confidence closed itemset and there is no need to search any itemset containing X but no Y .

Lemma 2 : Confidence-closed sub-itemset pruning

Let X is a correlated itemset currently under construction. If X is a proper subset of an already found confidence-closed itemset Y and all_conf(X) = all_conf(Y) then X and all of X's descendants in the set enumeration tree cannot be confidence-closed itemsets and thus can be pruned.

Lemma 1 means, the X-conditional database and the XY -conditional database separately have to mine if all_conf(X) ¹ all_conf(XY). However, though all_conf(X) and all_conf(XY) are different, the X and XY conditional databases are exactly the same if sup(X) = sup(XY). Using this property, avoid the overhead of building conditional databases for the prefix itemsets with the same support but different confidence. Maintain a list candidateList of the items that have the same support with the size of the X conditional database but are not included in the item merging because of their confidence. The list is constructed as follows. For X-conditional database, let Y be the set of items in f_list such that they appear in every transaction.

Do the following: Check that for each item Y_i in Y , if sup(Y_i)£ max_item_sup(X), X = XÈY_i; otherwise insert Y_i to candidateList. Check whether an itemset Z containing X(Z É X) is confidence-closed, also check whether the itemset Z È(Y' = Y1 . . .Yk, Y_i Î CandidateList) could be confidence-closed. Using this method, compute CCPs without generating the two conditional databases of X and of XY when all conf(X) > all conf(XY) and sup(X) = sup(XY).

Algorithm 4 shows the CCMine algorithm, which is based on the extension of CLOSET+ [13] and integrates the above discussions into the CLOSET+. Among a lot of studies for support-closed pattern mining, CLOSET+ is the fastest algorithm for a wide range of applications. CCMine uses another optimization technique to reduce the search space by taking advantage of the property of the all confidence measure. Lemma 3 describes the pruning rule.

Lemma 3 : Counting space pruning rule

Let a = i1i2 . . . ik. In the a-conditional database, for item x to be included in an all_confident pattern, the support of x should be less than sup(a)/min_a. Proof. In order for ax to be an all_confident pattern, max_item_sup(ax)sup(ax)/min_a.

Moreover, |sup(a)| ³ |sup(ax)|. Thus, max_item_sup(ax) ³sup(a)/min_a. Hence the lemma. With this pruning rule, reduce the set of items I_b to be counted and, thus, reduce the number of nodes visited when we traverse the FP-tree to count each item in I_b.

Example 2

Let us illustrate the confidence-closed mining process using an example. Figure 1 shows the running example of the transaction database DB. Let min sup = 2 and min_a = 40%. Scan DB once. Find and sort the list of frequent items in support descending order. This leads to f_list = (a:9, b:7, c:6, e:6, g:5, f:4, d:3, i:3, k:3, j:2, h:1). Figure 2 shows the global FP-tree. For lack of space, two representative cases: mining for prefix j:2 and eg:5. are shown after building the FP-tree mine the confidence-closed patterns with prefix j:2.

Computing counts :

Compute the counts for items a, c, e, f, and i to be included in the j-projected database by traversing the FP-tree shown in Fig. 2. First, use Lemma 3 to reduce items to be counted. The support of item z(z Î{a, c, e, f, i}) should be less than or equal to sup(j)/min_a = 2/0.4 = 5. With this pruning, items a, c and e are eliminated.

Now, compute counts of items f and i and construct j-projected database. They are 2 and 1, respectively. Pruning: We conduct pruning based on min sup and min_a. Item i is pruned since its support is less than min_sup. Item f is not pruned since and its confidence(2/4) is not less than min_a. Since f is the only item in j-conditional database, no need to build the corresponding FP-tree. And fj:2 is a CCP.

Algorithm CCMine: Mining confidence-closed correlated patterns

Input : A transaction database DB; a support threshold min sup a minimum all confidence threshold min_a

Output : The complete set of confidence-closed correlated patterns.

Method :

1. Let CCP be the set of confidence-closed patterns. Initialize CCP ¬ F

2. Scan DB once to find frequent items and compute frequent list f_list(=(f0, f1, . . .)).

3. Call CCMine(F, DB, f_list, CCP,F).

4. Procedure CCMine(á, CDB, f list, CCP, candidate List)

```
1.  For each item Y in f_list such that it appears in every transaction of CDB, delete Y from f_list
    and set á ¬ YÈa if all_conf(Ya)³all_conf(á), otherwise insert Y into candidateList in the
    support increasing order; {confidence-closed item merging}
2.  call GenerateCCP(á, candidate List, CCP);
3.  build FP-tree for CDB using f list, which excludes all the items Y s in the previous step;
4.  for each $a_i$ in f_list (in reverse descending support order) do
5.  set â = á È $a_i$;
6.  call Generate CCP(â, candidate List, CCP);
7.  get a set $I_b$ of items to be included in â-projected database; {counting space pruning rule}
8.  for each item in $I_b$, compute its count in â-projected database;
9.  for each bj in $I_b$ do
10. if sup(â$b_j$)< min_sup, delete $b_j$ from $I_b$; {pruning based on min sup}
11. if all conf(â$b_j$ )< min_ao, delete $b_j$ from $I_b$;{pruning based on min a}
12. end for
13. call FP-mine(â, CDB, f list, CCP, candidate List);
14. delete the items that was inserted in step 1 from candidate List;
15. end for
```

5. Procedure Generate CCP(á, candidate List, CCP)

for k-itemset Y = Y1 . . . Yk(Y$_i$Î candidate List) **do**

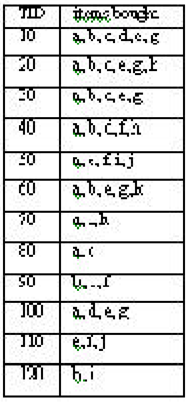
add áÈY into CCP if all_conf(á ÈY) ¡Ý min a if áÈY is not a subset of X (in CCP) with the same support and confidence; {confidence-closed sub-itemset pruning}

end for

2. After building conditional FP-tree for prefix g:5 and we mine g:5-conditional FPtree with f_list = (a:5, e:5, b:4, c:3).

6. Confidence Item Merging

Try confidence-closed item merging of a and e. Delete a and e from f_list. Since all_conf(ag) < all_conf(g), insert a into candidateList. Then, extend the prefix from g to eg by the confidence-closed item merging. Generate CCP: generate eg:5 as a CCP. In addition, also generate aeg:5, in which item a comes from candidateList. Now, in f_list, only two items :4 and c:3 are left. Mine the CCPs with prefix ceg:3. First, we generate ceg as a CCP. However, we cannot generate aceg as CCP since all_conf(aceg) < min_ a. Since item b is the only item in f_list, bceg is a CCP. Again, abceg cannot be CCP, since it also does not satisfy min_a. In this way, mine the beg:4- conditional database and generate beg and abeg as a CCP. After returning mining beg:4-conditonal FP-tree, item a is removed from candidateList.



TID	items bought
10	a,b,c,d,e,g
20	a,b,c,e,g,f
30	a,b,c,e,g
40	a,b,c,f,h
50	a,e,f,i,j
60	a,b,e,g,k
70	a,-,h
80	a,c
90	b,..,f
100	a,d,e,g
110	e,i,j
120	b,i

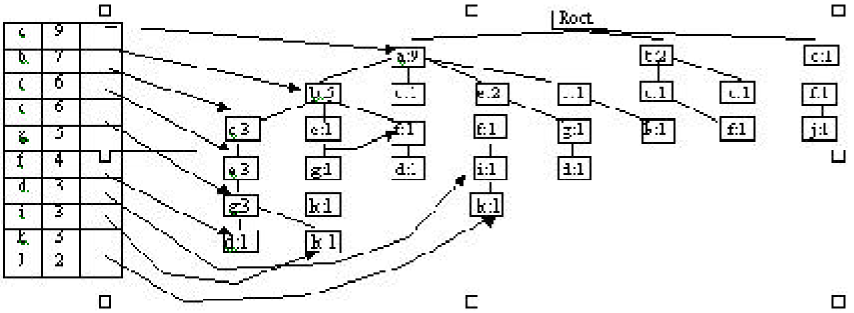


Fig. 1. A Transaction Database DB *Fig. 2. FP-Tree for the Transaction Database DB*

7. Experiments

In this section, we report out experimental results on the performance of CCMine in comparison with CCFilter algorithm. The result shows that CCMine always outperforms CCFilter especially at low min_sup. Experiments were performed on a 2.2GHz Pentium IV PC with 512MB of memory, running Windows 2000. Algorithms were coded with Visual C++.

The experiments were performed on two real datasets, as shown in Table 1. Pumsb dataset contains census data for population and housing and is obtained from http://www.almaden.ibm.com/software/

quest .Gazelle, a transactional data set comes from click-stream data from Gazelle.com. In the table, ATL/ MTL represents average/maximum transaction length. The gazelle dataset is rather sparse in comparison with pumsb dataset, which is very dense so that it produces many long frequent itemsets even for very high values of support.

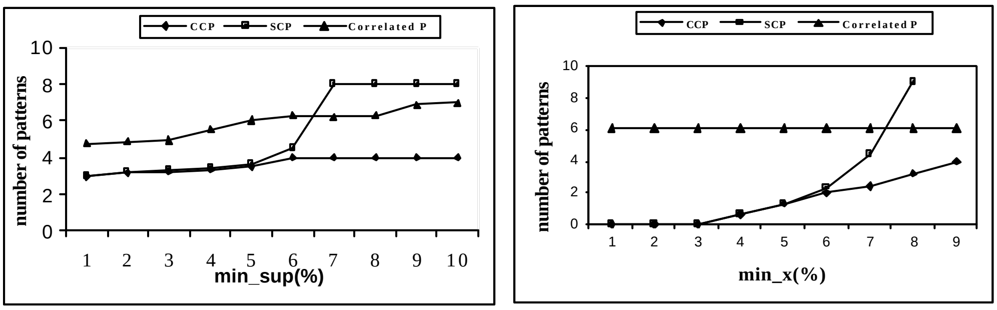
Table 1. Characteristics of Real Datasets.

Dataset	#Tuples	#Items	ATL/MTL
Gazelle	59602	497	2.5/267
Pumb	49046	2113	74/74

First show that the complete set of CCPs is much smaller in comparison with both that of correlated patterns and that of SCPs. Figure.3 shows the number of CCPs, correlated patterns, and SCPs generated from the gazelle data set. In this figure, the number of patterns is plotted on a log scale.

Figure 3(a) shows the number of patterns generated when min_sup varies and min_a is fixed while Figure 3(b) shows those when min_a varies and min_sup is fixed. First describe how many , can reduce the number of correlated patterns with the notion of CCPs. Figures 3(a) and 3(b) show that CCP mining generates a much smaller set than that of correlated patterns as the support threshold or the confidence threshold decreases, respectively. It is a desirable phenomenon since the number of correlated patterns increases dramatically as either of the thresholds decreases.

These figures also show that the number of SCPs is quite bigger than that of CCPs over the entire range of the support and confidence threshold. These results indicate that CCP mining generates quite a smaller set of patterns even at the low minimum support threshold and low minimum confidence threshold.



(a) (b)

Fig. 3. Number of patterns generated from the gazelle data set.

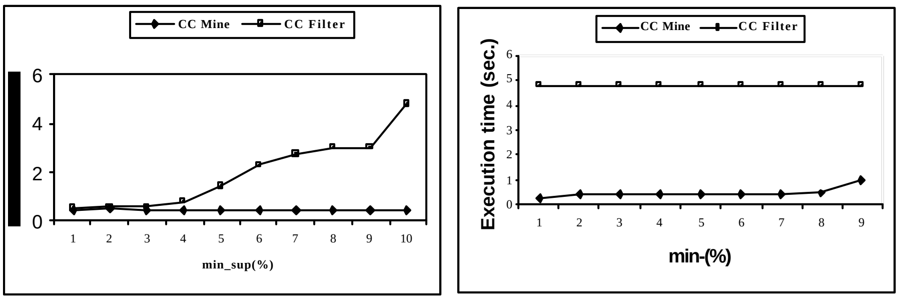
Let us then compare the relative efficiency and effectiveness of the CCMine and CCFilter methods. Figure 4 (a) shows the execution time of the two methods on the gazelle dataset using different minimum support threshold while min_a is fixed at 25%. Figure 4(a) shows that CCMine always outperforms CCFilter over the entire supports of experiments. When the support threshold is low, CCMine is faster more than 100 times compared with CCFilter, e.g., with min sup 0.05%, CCFilter uses 20 seconds to finish while CCMine only uses 0.2. The reason why CCMine is superior to CCFilter is that CCFilter has to find all of the support closed patterns although many of them do not satisfy the minimum confidence threshold and the number of these patterns increases a lot as the minimum support threshold decreases. Figure 4(b) shows the performance on the gazelle dataset when min sup is fixed at 0.01% and min_a varies. As shown in the figure, CCMine always outperforms CCFilter and the execution times of CCMine increases very slowly while min_a decreases.

CCFilter almost does not change while min_a varies, which means it does not take any advantage from min_a. This is because it spends most of processing time on mining SCP. Now, conduct the experiments on the pumsb dataset, which is a dense dataset. Figure 5(a) shows the execution time on the pumsb dataset when min_a varies while min_sup is fixed at 60%. Figure 5(a) shows that CCMine method outperforms CCFilter method when min_sup is less than 60%. When min_sup becomes less then 50%, CCFilter run out of memory and cannot finish. Figure 5(b) shows that CCMine method always outperforms CCFilter method over entire range of min_a. In summary, experimental results show that the number of confidence closed correlated patterns are quite small in comparison with that of the support-closed patterns. The CCMine method outperforms CCFilter especially when the support threshold is low or the confidence threshold is high.

8. Conclusions

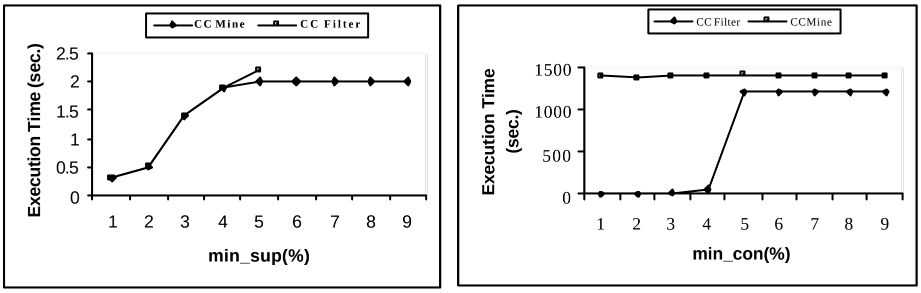
This paper presented an approach that can effectively reduce the number of correlated patterns to be mined without information loss. A new notion of confidence-closed correlated patterns is proposed. Confidence-closed correlated patterns are those that have no proper superset with the same support and the same confidence. For efficient mining of those patterns, we presented the CCMine algorithm. Several pruning methods have been developed that reduce the search space.

The performance study shows that confidence-closed, correlated pattern mining reduces the number of patterns by at least an order of magnitude in comparison with correlated (non-closed) pattern mining. It also shows that CCMine outperforms CCFilter in terms of runtime and scalability. Overall, it indicates that confidence-closed pattern mining is a valuable approach to condensing correlated patterns.



(a) *min á* = 25% (b) *min sup* = 0.01%.

Fig. 4. Execution time on gazelle data set



(a) when min á = 60% *(b) when min sup = 50%.*

Fig. 5. Execution time on the pumsb dataset.

All confidence is one of several favorable correlation measures, with null in variance property. Based on the examination, CCMine can be easily extended to mining some correlation measures, such as coherence or bond [6, 5, 8]. It is an interesting research issue to systematically develop other mining methodologies, such as constraint-based mining, approximate pattern mining, etc. under the framework of mining confidence-closed correlated patterns.

9. References

1. C. C. Aggarwal and P. S. Yu. A new framework for itemset generation. In Proc. 1998 ACM Symp. Principles of Database Systems (PODS'98), pages 18–24, Seattle, WA, June 1999
2. S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. In Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'97), pages 265–276, Tucson, Arizona, May 1997.
3. S. Morishita and J. Sese. Traversing itemset lattice with statistical metric pruning. In Proc. 2000 ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS'00), pages 226–236, Dallas, TX, May 2001.
4. P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In Proc. 2002 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'02), pages 32–41, Edmonton, Canada, July 2002.
5. E. Omiecinski. Alternative interest measures for mining associations. IEEE Trans. Knowledge and Data Engineering, 15:57–69, 2003.
6. Y.-K. Lee, W.-Y. Kim, D. Cai, and J. Han. CofiMine: Efficient Mining of Correlated Patterns. In Proc. 2003 Int. Conf. Data Mining (ICDM'03), pages 581–584, Melbourne, FL, Nov. 2003.
7. S. Ma and J. L. Hellerstein. Mining mutually dependent patterns. In Proc. 2001 Int. Conf. Data Mining (ICDM'01), pages 409–416, San Jose, CA, Nov. 2001.
8. H. Xiong, P.-N. Tan, and V. Kumar. Mining Strong Affinity Association Patterns in Data Sets with Skewed Support Distribution. In Proc. 2003 Int. Conf. Data Mining (ICDM'03), pages 387–394, Melbourne, FL, Nov. 2003.
9. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In Proc. 7th Int. Conf. Database Theory (ICDT'99), pages 398–416, Jerusalem, Israel, Jan. 1999.
10. R. J. Bayardo. Efficiently mining long patterns from databases. In Proc. 1998 ACMSIGMOD Int. Conf. Management of Data (SIGMOD'98), pages 85–93, Seattle,WA, June 1998.
11. J. Pei, G. Dong, W. Zou, and J. Han. On computing condensed frequent pattern bases. In Proc. 2002 Int. Conf. on Data Mining (ICDM'02), pages 378–385, Maebashi, Japan, Dec. 2002.
12. M. Zaki. Generating Non-redundant Association Rules. In Proc. 2000 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'00), Aug. 2000.
13. J. Wang, J. Han, and J. Pei. Closet+: Searching for the best strategies for mining frequent closed itemsets. In Proc. 2003 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'03), Washington, D.C., Aug. 2003.
14. J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In Proc. 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'00), pages 1–12, Dallas, TX, May 2000.
15. M. Zaki and C. Hsiao. CHARM: An Efficient Algorithm for Closed Itemset Mining, In Proc. SDM'02, Apr. 2002.

16. A. Krishnan, R. Hemalatha, C. Senthamari, “Efficient Mining of Association Rules using Partition Algorithm”, National Conference on Mathematical and Computational Models (NCMCM 2003), December 2003

17. A. Krishnan, R. Hemalatha, C. Senthamarai, “Mining of Association Rules in Distributed Database Using Partition Algorithm”, International Conference on Systemic, Cybernetics & Informatics (ICSCI 2004), February, 2004

18. A. Krishnan, R. Hemalatha, “Parallel Association Rule Mining – Finding Frequent Patterns Without Candidate Generation”, National Level Conference, Tech Fete 2004 on “Intelligence Techniques”, February 2004

19. A. Krishnan, R. Hemalatha, R. Hemamalini, C. Senthamarai, “An Efficient Data Mining Approach to Extract High Frequency Rules from Different Data Sources”, National Conference on Recent Trends in Computational Mathematics, March 2004

About Authors

Mrs. R Hemalatha is a Lecturer in K. S. R. College of Technology, Tiruchengode. Namakkal Dt., Tamil Nadu. She holds M.Sc. (I.T).
E-mail : hemaa_msc@yahoo.com

Dr. A Krishnan is a Principal in R. R. Engineering College, Tiruchengode. Namakkal Dt., Tamil Nadu.
E-mail : a_krishnan26@hotmail.com

Mrs. C Senthamarai is heading the Computer Science Department in K.S.R.College of Technology, Tiruchengode. Namakkal Dt., Tamil Nadu. She holds MCA.
E-mail : senthukumaran20023@yahoo.com

Ms. R. Hemamalini is a Lecturer in Computer Science Department in R. R. Engineering. College, Tiruchengode. Namakkal Dt., Tamil Nadu. She holds MCA.
E-mail : rk_hema2000@yahoo.com