High Confidence Association Mining Without Support Pruning

Ramkishore Bhattacharyya[1] and Balaram Bhattacharyya[2]

[1] Microsoft India (R&D) Pvt. Ltd., Gachibowli, Hyderabad – 500 032, India
[2] Dept of Computer and System Sciences, Visva-Bharati Universty,
Santiniketan – 731235, India
rk_ju@yahoo.com, balaramb@gmail.com

Abstract. High confidence associations are of utter importance in knowledge discovery in various domains and possibly exist even at lower threshold support. Established methods for generating such rules depend on mining itemsets that are frequent with respect to a pre-defined cut-off value of support, called support threshold. Such a framework, however, discards all itemsets below the threshold and thus, existence of confident rules below the cut-off is out of its purview. But, infrequent itemsets can potentially generate confident rules. In the present work we introduce a concept of *cohesion* among items and obtain a methodology for mining high confidence association rules from itemsets irrespective of their support. Experiments with real and synthetic datasets corroborate the concept of cohesion.

Keywords: Association rule, high-confidence, support threshold, cohesion.

1 Introduction

Association Rule Discovery (ARD) [1] is a foundational technique in exploring relationships between attributes. An association rule is an expression $A \xrightarrow{\sigma,\mu} C$, where A and C are itemsets and $A \cap C = \emptyset$, A and C are called antecedent and consequent respectively. Support σ of the rule is given by the percentage of transaction records containing the itemset $A \cup C$. Confidence μ of the rule is the conditional probability that a record, containing A, also contains C and is given as $\mu = \sigma(A \cup C)/\sigma(A)$. Classical ARD task concerns the discovery of every such rule with both support and confidence greater than or equal to pre-defined thresholds.

However, numerous applications such as medical diagnosis, bio-informatics, image categorization, weather forecasting, web mining, etc often contain potentially large number of low-support high-confidence rules, possibly with notable significance, but owing to the support constraint classical ARD misses them. Consider the rule {Down's syndrome} → { trisomy 21} (a chromosomal defect). Patients, with trisomy 21, suffer from Down's syndrome with almost 100% confidence. But Down's syndrome is found in about one in thousand cases. So a moderately low support threshold may miss it. Successively lowering the threshold cannot be a solution as the process is to capture unknown knowledge.

A. Ghosh, R.K. De, and S.K. Pal (Eds.): PReMI 2007, LNCS 4815, pp. 332 – 340, 2007.

Looking differently, high confidence of a rule indicates high probability of coexistence of the consequent with the antecedent. That is, we must look for a new property of an itemset that can reveal information on tendency of coexistence of items. We term this property as *cohesion* of an itemset.

The essence of mining support independent high-confidence rules has long been realized in the backdrop of classical Apriori algorithm [2]. Literature [4] employs dataset partitioning technique and border-based algorithm in mining high confidence rules. However, such an algorithm is restricted to discover all top rules (confidence = 100%) with the consequent being specified. Literature [3] proposes a measure called *similarity*, as a substitute for support, to explore low-support high confidence rules. However, the phenomenon has been studied only for two-itemsets for its computationally prohibitive nature in higher order itemsets. Similarity measure has been generalized for k-itemset in literature [5] with different nomenclature *togetherness*. The authors have discussed some positive sides of using togetherness in mining high-confidence rules. However, due to a less efficient Apriori-like algorithm, experimental opportunity with several real datasets has been missed. In fact, both the later two literatures do not establish the fact that due to some natural phenomenon of application domain, a set of items appear collectively which similarity or togetherness reflects. Therefore, we use the more spontaneous terminology cohesion. With the notion of cohesion of an itemset, the present work formalizes the problem of mining confident rules irrespective of their support information. We introduce a new algorithm CARDIAC[1] for mining cohesive itemsets which is the gateway of association mining. Theoretical as well as experimental studies prove its superiority over support threshold framework in mining confident associations. Since the problem is NP-Complete, we will not go into complexity analysis of our algorithm.

2 Formalism

Let $I = \{1, 2,\ldots, m\}$ be a set of items and $T = \{1, 2,\ldots,n\}$ be a set of transactions identifiers or tids. Assume $t(X)$ the set of all tids containing the itemset X as a subset.

Definition 1. (Cohesion) Cohesion of an itemset X is the ratio of cardinalities of two sets viz. the set all of tids containing X as a subset and the set of all tids containing any non-empty subset of X as a subset and is given by,

$$\xi(X) = \left|\bigcap_{x\in X} t(x)\right| \Big/ \left|\bigcup_{x\in X} t(x)\right| . \tag{1}$$

For the sake of clarity, we choose $\lambda(X) = \left|\bigcap_{x\in X} t(x)\right|$ and $\rho(X) = \left|\bigcup_{x\in X} t(x)\right|$.

Definition 2. (Cohesive itemset) An itemset X is called a cohesive, if $\xi(X)$ is no less than pre-specified threshold.

As the name suggests, expression (**1**) gives the measure of a set of items to stick together. Clearly, every item of the dataset is cohesive (cohesion is unity). It is one of the major advantages achieved by shifting the domain from support to cohesion. So, cohesion of an itemset X needs both $\lambda(X)$ and $\rho(X)$ to be computed.

[1] CARDIAC stands for Confident Association Rule Discovery using Itemset Cohesion; 'A' is gratuitous.

Lemma 1. Cohesive itemsets retain downward closure property [2].

Proof. For all itemsets Y and X such that $Y \supseteq X$, $\lambda(Y) \leq \lambda(X)$ and $\rho(Y) \geq \rho(X)$. So, $\xi(Y) = \lambda(Y) / \rho(Y) \leq \lambda(X) / \rho(Y) \leq \lambda(X) / \rho(X) = \xi(X)$. Hence, all subsets of a cohesive itemset must also be cohesive.

3 Algorithm

The methodology, we develop for mining cohesive itemsets, incorporates the concept of vertical mining in [6]. Let $X = \{x_1, x_2, \ldots, x_n\}$ be an itemset. We follow the notation X' for $\{x_1', x_2', \ldots, x_n'\}$. If $t(X)$ is the set of tids containing X as a subset, $t(X')$ is the set of tids containing X' as a subset i.e. the set of tids containing none of $x \in X$ as a subset. So for an itemset X,

$$\bigcup_{x \in X} t(x) = T \setminus \bigcap_{x' \in X'} t(x') . \tag{2}$$

$$\left| \bigcup_{x \in X} t(x) \right| = n - \left| \bigcap_{x' \in X'} t(x') \right| , n = |T| . \tag{3}$$

Proof. Equation (**2**) follows directly from definition.

As $\bigcap_{x' \in X'} t(x') \subseteq T$, $T \setminus \bigcup_{x' \in X'} t(x') = |T| - \left| \bigcap_{x' \in X'} t(x') \right| = n - \left| \bigcap_{x' \in X'} t(x') \right|$.

Hence is the result.

Lemma 2. For all Y such that $Y \supseteq X$, $\bigcap_{y' \in Y'} t(y') \subseteq \bigcap_{x' \in X'} t(x')$.

Proof. For all Y such that $Y \supseteq X$, $\bigcup_{y \in Y} t(y) \supseteq \bigcup_{x \in X} t(x)$. So, $T \setminus \bigcup_{y \in Y} t(y) \subseteq T \setminus \bigcup_{x \in X} t(x)$. Hence, $\bigcap_{y' \in Y'} t(y') \subseteq \bigcap_{x' \in X'} t(x')$.

So, instead of maintaining the set $\bigcup_{x \in X} t(x)$ of monotonic increasing cardinality, it is wise to maintain $\bigcap_{x' \in X'} t(x')$ which is of monotonic decreasing cardinality.

```
CARDIAC([P],mincohesion)
//[P] is the set of cohesive prefixes
for all Xi ∈[P]do
  S = ∅;
  for all Xj ∈[P]| j > i do
    X = Xi ∪ Xj;
    t(X)= t(Xi)∩t(Xj);λ(X)= |t(X)|;
    t(X')= t(Xi')∩t(Xj');ρ(X)= n − |t(X')|;
    ξ(X)= λ(X)/ρ(X);
    if ξ(X) ≥ mincohesion
      S = S ∪ X;
  if S ≠ ∅ then call CARDIAC(S,mincohesion)
```

Fig. 1. CARDIAC algorithm for cohesive itemset generation

Lemma 3. Under the same threshold for cohesion and support, number of cohesive itemsets is greater or equal to that of frequent itemsets.

Proof. For an itemset X, $\xi(X) = \lambda(X) / \rho(X) \geq \lambda(X) / |T| = \sigma(X)$ as $\rho(X)$ is upper bounded by $|T|$. So, under the same threshold for both cohesion and support, all frequent itemsets are cohesive. But there possibly exists cohesive itemsets that are not frequent. For a dataset, count of such itemsets can be large enough as will be shown in experimental section.

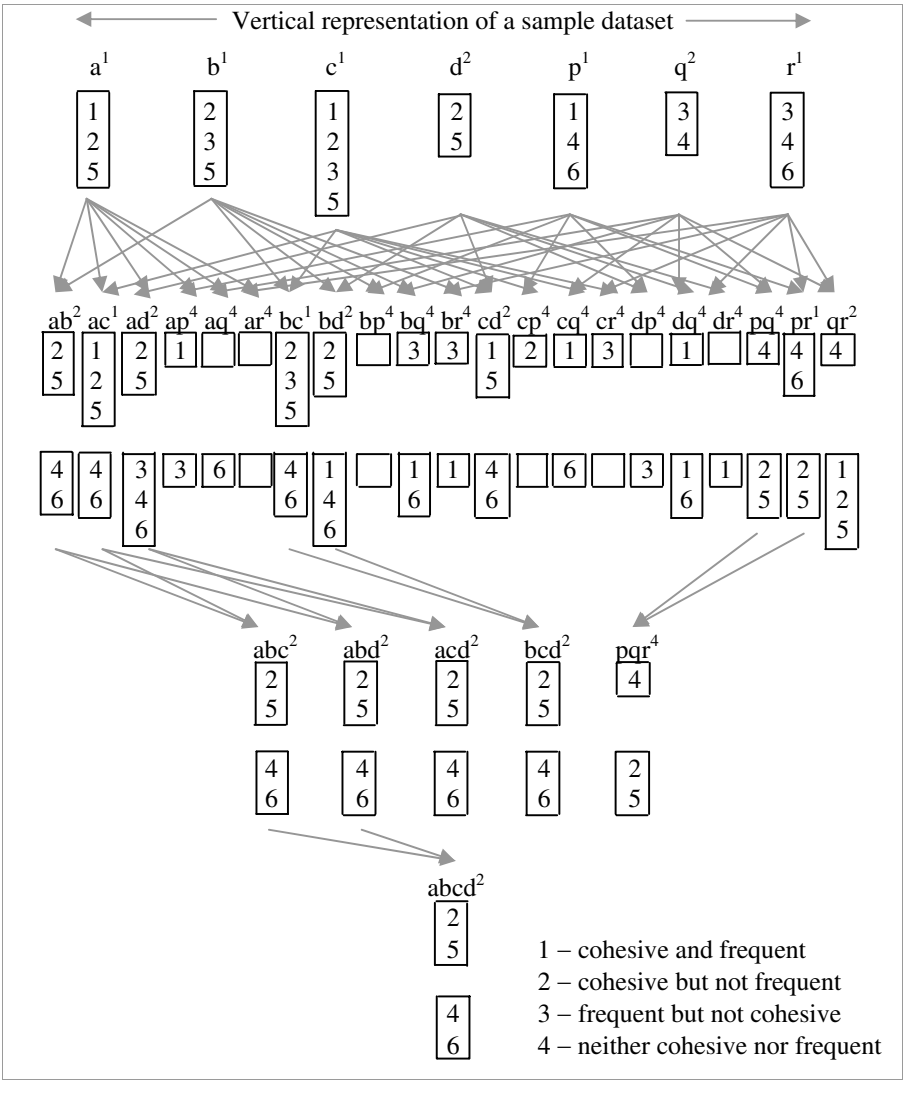

Fig. 2. Illustration of algorithm

Example. Fig. 2 illustrates the algorithm with a vertical sample dataset in vertical representation, threshold for cohesion or support being set to 50%. The superscript for itemsets indicates its status which is defined at right-bottom corner inside Fig. 2.

4 Experiment

All experiments are conducted on a 2GHz AMD Athlon PC with 1GB primary memory. Algorithm CARDIAC (Fig. 1) is implemented in C++ and run on Debian Linux platform. Characteristics of the real and synthetic datasets, used for the purpose of performance evaluation, are tabulated in Fig. 3.

Dataset	# Items	Avg. Transaction Size	# Transactions
Chess	76	37	3,196
Pumsb*	2087	50	49,046
T10I4D100K	869	10	101,290
T40I10D100K	941	40	100,000

Fig. 3. Dataset Characteristics

Fig. 4 depicts execution time of CARDIAC with respect to decreasing cohesion threshold, results being plotted in logarithmic scale. Execution time has drastically been reduced by computing the set $\cap_{x' \in X'} t(x')$ which quickly converges to null set but still equally serves the purpose. Datasets, that are sparse with fewer items, contain less number of tidsets with lesser cardinality and hence require less execution time compared to other dense datasets.

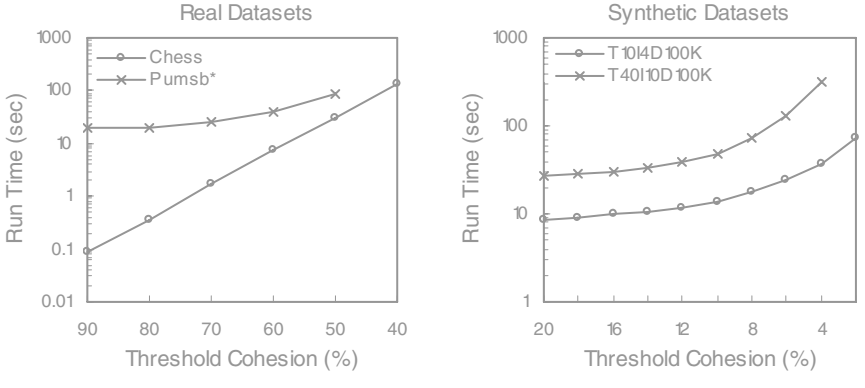

Fig. 4. Performance of CARDIAC algorithm with decreasing threshold cohesion

Fig. 5 presents a comparative study on count of frequent and cohesive itemsets under fixed thresholds. The chess dataset contains only 76 items with average transaction length 37. Hence it exhibits minimum variation from one record to

other. So number of cohesive itemsets is slightly more than the frequent itemsets. Remaining datasets, however, contain lots of items and their variations in record to record. Here, number of cohesive itemsets is orders of magnitude higher than corresponding frequent itemsets. These itemsets actually contribute to the low support high confidence rules.

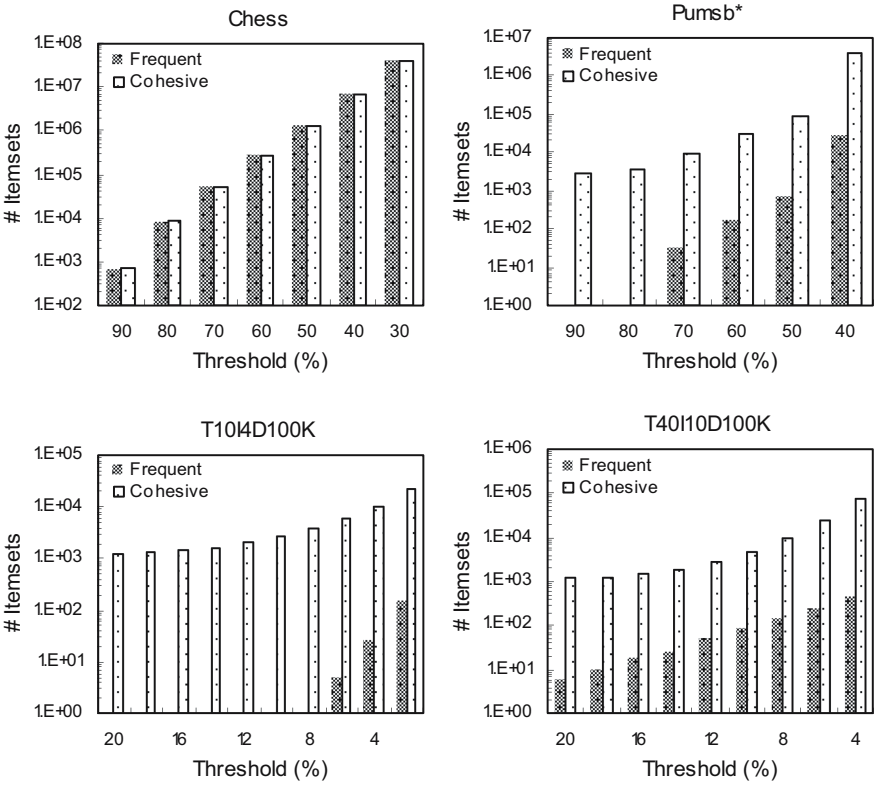

Fig. 5. Comparison on count of frequent and cohesive itemsets

Fig. 6 shows a comparative study on count of association rules under support and cohesion measure, threshold confidence being set to 90% in each. Again, in chess dataset, there is no much difference in the count of rules generated by both the measures. But for other datasets, count of rules is incomparably greater for cohesion as expected. The excess rules obtained are actually the low support high confidence ones which classical ARD misses. If we stick to the support measure and try to mine those rules, threshold support has to be set to such a low value that mining would rather be intractable. In addition, it would incur the cost of pruning huge low confidence rules. A higher choice for cohesion successfully mines the high confidence rules with lesser number of rules to be pruned.

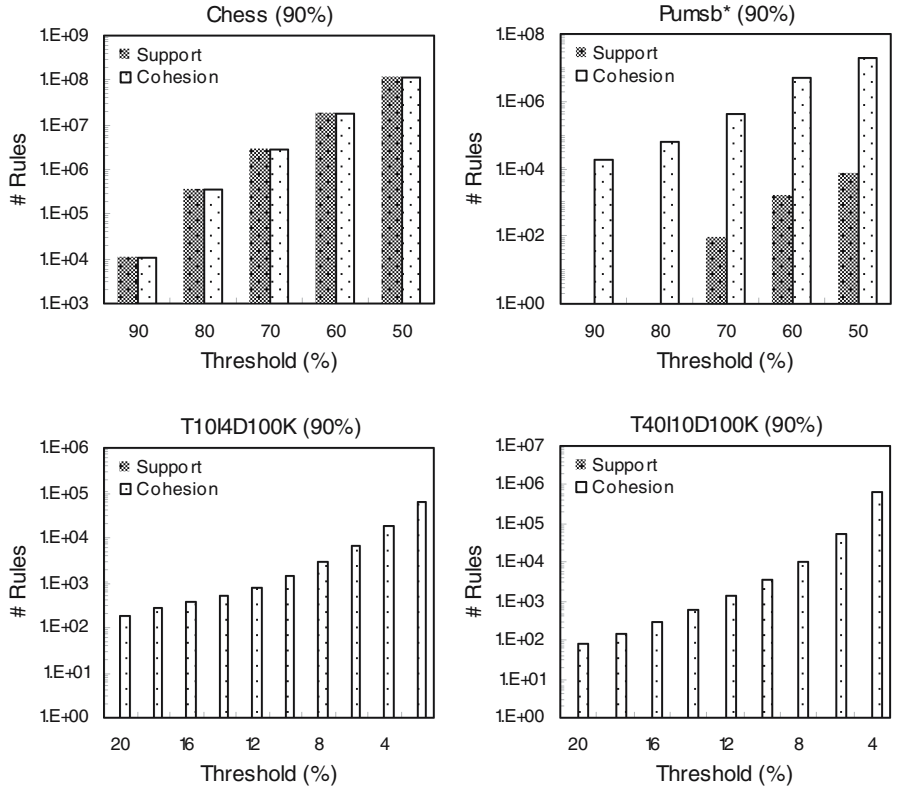

Fig. 6. Comparison on count of rules under support and cohesion measure

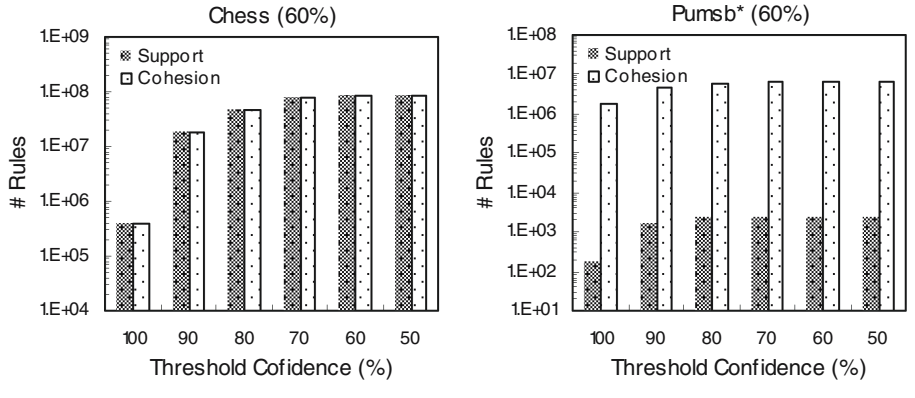

Fig. 7. (*continued*)

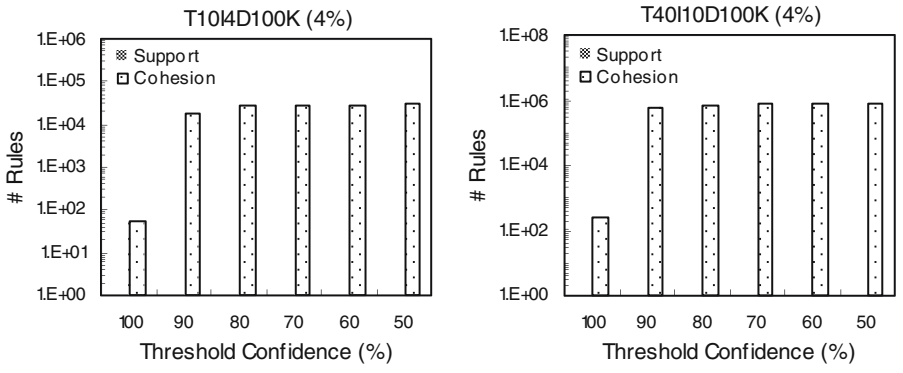

Fig. 7. Comparison on count of rules with decreasing threshold confidence

Fig. 7 shows a comparative study on count of rules versus confidence under fixed support and cohesion measures. In all datasets, except chess, cohesion mines significant number of confident rules which support measure fails to achieve. For the two synthetic datasets, count of rules under respective threshold support is nil down to 50% threshold of confidence. But cohesion successfully mines them.

5 Conclusion

Associations, explored using cohesion, are indicative of natural properties prevalent in the application domain, no matter whatever are their support counts. Mining such rules using support constraint is well-nigh infeasible as it would incur huge cost of dealing with intractable number of itemsets. Additionally, cohesion explores comparatively lesser number of itemsets than support to discover same number of rules, reducing burden on the rule miner process.

Acknowledgment. The authors wish to acknowledge Dr. Raghunath Bhattacharyya for his valuable suggestions on applicability of cohesion in medical domain.

References

1. Aggarwal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of ACM SIGMOD 1993, Washington DC, pp. 207–216 (1993)
2. Aggarwal, R., Srikant, R.: Fast Algorithm for Mining Association Rules. In: VLDB. Proceedings of 20th Very Large Database Conf., pp. 487–499 (1994)
3. Cohen, E., Datar, M., Fujiwara, S., Gionis, A., Indyk, P., Motwani, R., Ullman, J.D., Yang, C.: Finding Interesting Associations without Support Pruning. In: Proceedings of IEEE ICDE, pp. 489–500 (2000)

4. Jinyan, L., Xiuzhen, Z., Guozhu, D., Kotagiri, R., Qun, S.: Efficient Mining of High Confidence Association Rules without Support Thresholds. In: Żytkow, J.M., Rauch, J. (eds.) PKDD 1999. LNCS (LNAI), vol. 1704, pp. 406–411. Springer, Heidelberg (1999)
5. Pal, S., Bagchi, A.: Association against Dissociation: some pragmatic considerations for Frequent Itemset generation under Fixed and Variable Thresholds. SIGKDD Explorations 7(2), 151–159 (2005)
6. Zaki, M.J.: Scalable Algorithms for Association Rule Mining. IEEE Trans. On Knowledge and Data Engg 12(3), 372–390 (2000)