An Efficient Algorithm for Mining Coherent Association Rules

Sharada Narra
Assistant Professor
Dept. of Information Technology
ANITS
Visakhapatnam

Siva Ponugoti
Dept. of Information Technology
ANITS
Visakhapatnam

Suresh Mullapudi
Dept. of Information Technology
ANITS
Visakhapatnam

Madhavi Dabbiru
Professor
Dept. of Computer science Engineering
Dr.L.B.College of Engineering
Visakhapatnam

ABSTRACT

There are many data mining techniques for finding association rules with the predefined minimum support from transaction databases. However, common problems with existing approaches are that an appropriate minimum support is difficult to determine and that the derived rules convey common-sense knowledge. Mining association rules without minimum support threshold is an important approximation of the association rule mining problem, which has been recently proposed[1].In this approach rules are generated based on logical implications and hence fewer and more meaningful rules called coherent rules are reported. An algorithm named *ChSearch,* is proposed to mine coherent rules for the user specified itemset. In this paper, the same problem is studied with two additional improvements. First, Apriori-based algorithm, namely ECARM for mining all coherent rules is proposed. Second to speed up time consuming candidate generation process an efficient pruning strategy is introduced. The proposed method generates coherent rules in shorter average execution times and fewer database scans, and thereby reduce great amount of I/O time per database scan and hence scaled to large databases. The method has been evaluated using both synthetic and real datasets, and the experimental results demonstrate its effectiveness and efficiency.

General Terms

Data Mining

Keywords

Association Rules, Minimum Support threshold, Coherent Rules, Preposition logic

1. INTRODUCTION

Association mining one of the well researched techniques of data mining is used to increase product sales and provide convenience to supermarket customers. The association patterns among items represent domain knowledge which is known only to experts with years of experience. Thus association mining technique extracts domain knowledge which is internally hidden in large databases.

The association rule is an implication of the form X→Y where X and Y are sets of items and $X \cap Y = \emptyset$ and is defined with two interestingness measures support and confidence. The first algorithm proposed in association mining is Apriori [2, 3] and many other algorithms were derived from it. All these algorithms discover association rules based on the support and confidence thresholds. The minimum support thresholds help in reducing the exponential search space and prevent the algorithm from running out of memory. The search for less frequent and rare items [4, 5] reveals valuable information. To extract such rare and useful association rules it is necessary to specify small support count threshold. Unfortunately the lower the support the larger the volume of rules obtained which are difficult to analyze. However when the support count is increased, the algorithms become more efficient but generate obvious uninteresting rules. Hence, association rule mining technique discovers association rules based on the accuracy in defining minimum support threshold [6].While positive association rules are useful in decision making, negative association rules [7, 8, 9] are also useful in market-basket analysis to identify products that conflict with each other. Sometimes self-contradictory and redundant association rules would occur when we consider both the positive and negative association rules simultaneously. For example to analyze the purchase of printer and scanner and the minimum support threshold is set to 25%.The positive rule printer→scanner, with 30% support and negative rule ~printer → scanner, with 29% support are discovered.

The number of association rules grows exponentially with the number of items, especially when the minimum support thresholds are small. Several strategies have been proposed to reduce the number of association rules such as generating only "non-redundant" rules or generating only those rules satisfying interesting measures such as all-confidence, coverage, leverage, lift, interestingness, chi-squared test. These approaches need to explicitly check all candidate rules; this would require a long processing time and I/O time if the number of candidates is huge. To address the issues related to traditional support and confidence threshold Sim et al., proposed logic based discovery of association rules. In this approach rules are discovered based on the properties of preposition logic and hence called coherent rules. A coherent rule is reported by considering both presence and absence of items during mining, for example the rule printer → scanner will only be reported if there are fewer occurrences of the rules printer → scanner and printer → ~scanner and more of ~printer → ~scanner. Search for coherent rules is more expensive compared to Apriori because it needs to consider both the presence and absence of item sets.

In this paper, an Apriori like algorithm namely Efficient Mining of Coherent Association Rules (ECARM) for mining coherent rules efficiently by using the properties of preposition logic is proposed. Since finding the support count of all the candidate item sets is time consuming, we prune the candidate itemsets which cannot become coherent itemsets by means of anti-monotone and prequalifying criteria. The resulting qualifying candidate itemsets are tested for logical equivalence. Experiments are made with multiple data sets to show the performance of the proposed approach.

The rest of this paper is organized as follows. Related works are described in Section 2. The preliminary definitions and derivation of the prequalifying criteria are described in Section 3. The proposed mining algorithm is presented in Sections 4. Experimental results are shown in Section 5. The conclusions and suggestions for future work are given in Section 6.

2. RELATED WORK

This section reviews the related works including association rule mining approach and coherent rule mining approach.

2.1 Association Rule Mining Approaches

There is an increasing focus on data mining which is often used to discover association rules from large datasets [10, 11, 12, 13, 14, and 15]. To improve the performance of mining process [16] and apply to different domains many different methods are proposed such as FP-Tree, sampling, and parallelization. The performance of most of the techniques is based on the user specified minimum support threshold and confidence threshold. This can be limiting as the support is not directly relating to the potential value of the rule. For example Ketel vodka and Beluga caviar [17] are low sales volume products and hence the purchase of them is likely to have low support. However such rules may be of considerable interest as they are high profit items. The rules that involve infrequent or rare items will be found with low support, but low support will typically generate many rules that are of no interest. However using a high support reduces the number of rules mined but will eliminate the rules with rare items. This is called rare item problem. The authors [18, 19] solve the rare item problem by allowing users to specify multiple minimum supports to reflect the varied frequencies of items in the database. Another serious problem in association rule discovery algorithms is they generating many hundreds and often thousands of patterns as the number if items increase in the dataset. A large number of these rules may be redundant. It is therefore necessary to filter out those patterns which are interesting through the use of some measures. The interesting measures which measure the deviation from independence have been considered such as conviction, proposed in [20], collective strength in [21], and Pearson's correlation coefficient and conviction [22].Authors in [23] consider both the presence and absence of items as a basis for generating rules.

Table 1: Conditional mapping of association rules to equivalence.

Equivalences	P ≡ Q	~P ≡ ~Q
Association Rules	X → Y	~X → ~Y
True or False On Association Rules	Required Conditions	
T	X → Y	~X → ~Y
F	X → ~Y	~X → Y
F	~X → Y	X → ~Y
T	~X → ~Y	X → Y

The significance of association rule is measured based on the chi-squared test for correlation from classical statistics. When new interesting measures are used along with minimum support threshold some interesting association rules could still be lost.

2.2 Coherent Rule Concept

Traditional association mining techniques discover the rules subjected to the accuracy in the user specified minimum support and minimum confidence. To address this problem Sim et al., proposed preposition logic based pattern discovery. In this technique the association rules are mapped to the equivalences subjected to the conditions specified in Table 1. X and Y are item sets in Table 1. In multiple transactions, association rules are mapped to implications as follows: X→Y is mapped to an implication p ≡ q if and only if (1) Sup(X, Y) > Sup (~X, Y); (2) Sup(X, Y) > Sup(X, ~Y); and (3) Sup(X, Y) > Sup (~X, ~Y). The name pseudoimplications are given to association rules when they are mapped to implications based on comparison between supports. Two pseudoimplication of equivalence such as (X,Y), (~X, ~Y) when mapped to two logical equivalences p ≡ q and ~p≡~q, the result is a coherent rules which meets the following conditions : (1) Sup(X, Y) > Sup(~X, Y); (2) Sup(X, Y) > Sup(X, ~Y); (3) Sup(~X, ~Y) > Sup(~X, Y); and (4) Sup(~X, ~Y) > Sup(X, ~Y).These four conditions can also be represented as a contingency table, as shown in Table 2.

Table 2: Four conditions represented in contingency table

Frequency of Co-occurrences	Consequence Y	
	Y	~Y
Antecedent X		
X	Q1=Sup(X, Y)	Q2=Sup(X, ~Y)
~X	Q3=Sup (~X, Y)	Q4=Sup(~X, ~Y)

In coherent rule mining approach when a rule is reported it has the strongest value in the data since it is reported by comparing the presence and absence of item during the mining process. Sim et al., proposed coherent algorithm can be stated as follows: the user specifies the consequent itemset Y. The power sets of the user specified item set are in P(C) and the power sets of all other item sets except the user specified item set are in P(A).To generate coherent rule between the two item sets X and Y, logic based four conditions are tested with the contingency table. The search space is reduced with anti-monotone property. The limitations of this approach are to construct the contingency table for a given item set, a scan over the entire database is required and to discover coherent rules for all possible k items, in the given database then **3^k-2^k+1** candidate coherent rules need to be considered. Because the mining of coherent association rules is time consuming, we need an efficient approach when the item sets and transactions are large.

Chen et al., [24] proposed an Apriori like approach namely highly coherent Association Rule mining algorithm for mining coherent rules. This approach calculates lower and upper bound interval for the consequent part of a candidate coherent itemset to speed up the mining process. However this approach mines highly coherent rules subjected to the user specified minimum support threshold.

In this work an Apriori based coherent association rule mining algorithm that uses logic based four conditions instead of support threshold in the mining process and generate coherent rules in shorter average execution times and fewer database scans, and thereby reduce great amount of I/O time per database scan than the previous known methods in coherent rules.

3. PRELIMINARY DEFINITIONS

A coherent rule is between the two itemsets X and Y such that $X \in I_{A,}$ $Y \in I_C$, $X \neq \emptyset$ and $Y \neq \emptyset$, I set of unique items. The pair of rules $\{X \rightarrow Y\}$, $\{\sim X \rightarrow \sim Y\}$ is a coherent rule when the following conditions are met:

1) Sup(X, Y) > Sup (~X, Y)
2) Sup(X, Y) > Sup(X, ~Y)
3) Sup (~X, ~Y) > Sup (~X, Y)
4) Sup (~X, ~Y) > Sup(X, ~Y).

In the figure 1 support of itemsets are represented in set cardinality.

|X| - Support of itemset X
|Y| - Support of itemset Y
|X∩Y| -Support of itemset(X, Y)
|X-Y| - Support of itemset(X, ~Y)
|Y-X| - Support of itemset (~X, Y)

μ - |X∪Y| indicates Support of itemset (~X, ~Y), where μ is the number of transactions.

From set cardinality $|X\cup Y| = |X| + |Y| - |X\cap Y|$

From the set properties $|X\text{-}Y| = |X| - |X\cap Y|$ and $|Y\text{-}X| = |Y| - |X\cap Y|$

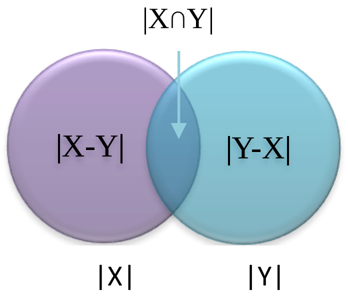

Figure 1: Set Cardinality Representation

3.1 Pruning Strategies

Since there could be a huge number of candidate itemsets in a database, and only some of them are useful for generating coherent association rules, pruning is critical to efficiently search for interesting candidate itemsets. In this section we design a pruning strategy. The candidate itemsets are pruned by exploiting anti-monotone property and prequalification criteria.

Definition 1:
Prequalification criteria (Minimum (|X|, |Y|) > Abs (|X|-|Y|) is formulated from the first two coherent conditions:
S(X, Y) > S (~X, Y) and S(X, Y)>S(X, ~Y)
Combining these two conditions the following condition is obtained.
S(X, Y) > S (~X, Y) - S(X, ~Y)
S(X, Y) > |X| - |X∩Y| - (|Y|-|X∩Y|)
S(X, Y) > Abs (|X| - |Y|) and from anti-monotone property maximum support of the itemset(X, Y) is Min (|X|, |Y|)

The candidate itemsets are tested against this criterion, before their support is determined by scanning the database. This filters the spurious candidates from further consideration and thereby increases the efficiency in generation of coherent itemsets.

4. PROPOSED ALGORITHM

The time efficient coherent mining algorithm is an Apriori like approach which generates coherent rules in multiple passes. In the first pass coherent 1-itemsets along with their support count is determined and stored in C_1. A subsequent pass, say k consists of two phases. First, the coherent itemsets CI_{k-1} found in the (k-1) pass are used to generate the candidate item sets C_k there by reducing the search space with the help of antimonotone property. To further enhance the efficiency in the generation of coherent itemsets, each candidate itemset is tested with the prequalification condition (discussed in Section 3) to determine highly coherent candidate itemsets.

The candidate itemset that do not satisfy the condition is removed, from further consideration which reduces database scans. The database is scanned to determine the support of qualified candidate itemsets and is given by $C_{k.}$ Then these candidate item sets are tested against the four set cardinality conditions. If the conditions, are met then candidate coherent k-item set is added into coherent k-item set CI_k.

The process is repeated, until no more coherent k-itemsets are found. The coherent rules are then generated from coherent itemsets.

Algorithm: Efficient mining of Coherent Association Rules

Input: An Input Database D

Output: The complete set of Coherent Rules CR
C_k: Candidate itemsets of size k
//C_k is generated by joining $CI_{k-1}$with itself
QC_k : Qualified itemsets of size k
$CI_{k:}$ Coherent itemsets of size k

```
Begin
CI ← ∅;  CR ← ∅;
C_1 and CI_1 = {items with support count};
For (k = 1; CI_k != ∅; k++) do
    C_{k+1} = candidate coherent itemsets generated from CI_k;
    QC_{k+1} ← ∅;
    For each candidate itemset (X, Y) in C_{k+1}  do
        If (Minimum (|X|, |Y|) > Abs (|X|-|Y|) then
            QC_{k+1} = QC_{k+1} ∪ (X,Y);
        end if
    end for
    For each qualified itemset (X,Y) in QC_{k+1} do
        If (|X∩Y| > |X-Y|) and ( |X∩Y| > |Y-X|)
          and (μ - |X∪Y| > |X-Y|) and (μ - |X∪Y|) > |Y-X|) then
            CI_{k+1} = CI_k ∪ (X, Y);
        end if
    end for
returns ∪_k CI_k;
For each coherent itemset S in CI_k generate
    all coherent association Rules (X→Y)
    where X and Y are subsets of S and X∩Y = ∅
End
```

5. EXPERIMENTAL RESULTS

The efficient coherent association rule mining algorithm (ECARM) and Sim et al., proposed basic algorithm (*ChSearch*) are implemented with java. The proposed efficient coherent association rule mining algorithm (ECARM) is evaluated as opposed to basic coherent association algorithm (*ChSearch*).The experiments are performed on an Intel i3 processor with 4GB RAM, running windows 7 operating system. The evaluation is examined from two aspects: the candidate itemsets that are effective in generating coherent rules, and the execution time.

Data sets from different domains are used for experiments. The data sets used in the evaluation were taken from the sequential pattern mining framework (www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php).The mushroom data contains characteristics of various mushroom species, chess data contain game state information, Accidents data and T10I4D100K synthetic data from IBM data set generator are used.

The description of the data sets is given in Table 3. Experimental results are shown in Table 4, which reports the number of coherent itemsets found, execution time in seconds. These results bring to several important observations. First of all, the number of evaluated itemsets is much lower and closer to the number of coherent itemsets. This entails that ECARM is able to accomplish the task efficiently and effectively. The scale-up experiments are performed with two synthetic data sets generated from IBM synthetic data generator tool(Agrawal & Srikant, 1994).This tool is widely used in literature to generate experimental data sets.

Table 3: Details of databases

Database	# of transactions	# of items
Chess	3,196	37
Mushrooms	8,124	23
T10I4D100k	100000	10
Accidents	340183	462

Table 4: Execution details of different data sets

Database	# of coherent itemsets	Execution time(s) ECARM	Execution time(s) ChSearch
Chess	109	0.6	1
Mushrooms	761	1	5
Accidents	415	28	591
T10I4D100k	940	378	736

Table 5: Description of parameters

Parameter name	Description
D	Number of transactions
I	Average length of maximal pattern
T	Average length of transaction
N	Number of distinct items
K	Thousands

Table 5 shows the parameter descriptions, the data sets differ in number of items, average length of transactions and number of transactions.For the experiments in this section, the data sets that are shown in Table 6 will be used.

Table 6: Synthetic test data sets

Dataset	T	N	I	D
T10N100I4D500k	10	100	4	50000
T10N200I4D500k	10	200	4	50000
T10N1200I4D500k	10	1200	4	50000
T10N500I4D100k	10	500	4	10000
T10N500I4D200k	10	500	4	20000
T10N500I4D1000k	10	500	4	100000

For both BCARM and ECARM the execution times are plotted for the dataset T10.N500.I4 in Figure 2.For the ten datasets with transaction sizes of 10k, 20k, 30k, 40k, 50k, 60k, 70k, 80k, 90k, and 100k *ChSearch* took too long to execute. As the transactions increase, the execution times of both the algorithms increase because of the increase in the total number of candidate and coherent itemsets. However, ECARM beats *ChSearch* by more than an order of magnitude for large datasets. The problem with *ChSearch* is that it generates too many candidates that later turn out to be small, causing it to waste too much effort.

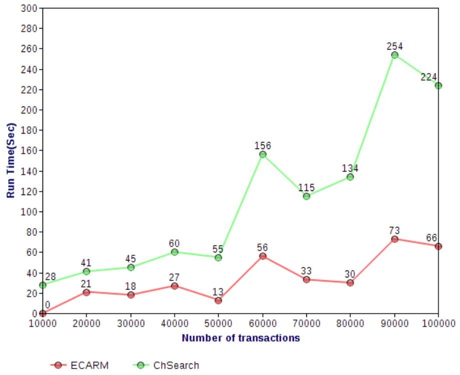

Figure 2: Performance comparison on database T10N500I4D (10K, 20K, 30K…100K).

Next scale up performance of ECARM with the number of items is examined. The number of items is increased from 100 to 1200 for the parameter settings T10.I4.D50K. All other parameters were the same as for the data in Table 6.The experiments are performed and obtained the results shown in Figure 3. The execution times decreased a little since the average support for an item decreased as the number of items is increased. This resulted in fewer large itemsets and hence faster execution times. As shown, the execution times of the proposed ECARM increase with the number of items, but only gradually.

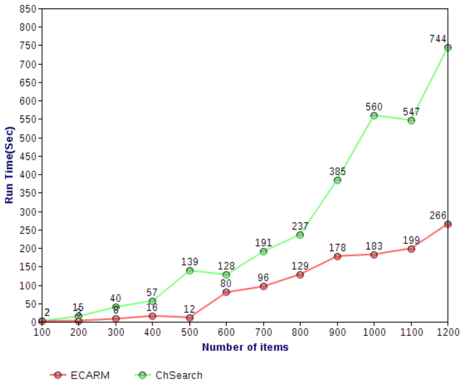

Figure 3: Performance comparison on database T10I4D50K (N100, N200…N1200)

6. CONCLUSION

In traditional support and confidence based association rule mining approach the rules below the threshold are not mined and plenty of rules above the threshold are generated which may be common sense knowledge. The proposed approach eliminates the need for minimum support threshold and generates the rules with high statistical value as rules are generated based on the presence and absence of items during the mining process. The search for coherent rules is more expensive compare to Apriori because it needs to consider both the positive and negative association rules.

The proposed algorithm efficiently discovers all significant coherent rules between items in a large database of transactions. A direct application of coherent rule mining technique may generate too many candidate rules. The problem is handled by using an efficient pruning technique. The proposed algorithm is compared to the previously known coherent rule mining algorithm and the experimental results, shows that the proposed algorithm always out performs.

The experiments performed on real and synthetic datasets indicate that the algorithm scales linearly with the number of records. These experiments demonstrate the feasibility of using ECARM in real applications involving very large databases. Based on this study we conclude that mining coherent rules without minimum support threshold should be more preferable than the traditional minimum support based mining for frequent itemsets. More detailed study along this direction is needed, including further improvement of the performance by using parallel processing, projection technique, or sampling techniques and flexibility at mining top-k coherent itemsets.

7. ACKNOWLEDGEMENTS

Our thanks to Prof.Ch.Suresh and Director (R&D) Dr.K.V.S.V.N Raju for many helpful suggestions and comments in this work.

8. REFERENCES

[1] A. T. H. Sim, M. Indrawan, S. Zutshi, and B. Srinivasan. 2010. Logic-Based Pattern Discovery. IEEE Transactions on Knowledge and Data Engineering, Vol. 22, No. 6, pp. 798-811

[2] Agrawal, R., Imielinski, T., and Swami, A. 1993a. Database mining: A performance perspective. IEEE Trans. Knowledge and Data Eng. Volume: 5, Issue: 6 (Nov.), 914-925.

[3] Agrawal, R., Imielinski, T., and Swami, A.1993b. Mining association rules between sets if items in massive databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data.ACM, Washington D.C., 207-216.

[4] B. Padmanabhan, A Tuzhilin 1999 Unexpectedness as a measure of interestingness in knowledge discovery. Decision Support Systems, 27(3):303-318

[5] B. Liu, W. Hsu, and Y. Ma 1999.Mining Association Rules with Multiple Minimum Supports. In Proceedings of ACM SIGKDD, pp. 337-341

[6] G.I. Webb and S. Zhang.2005. K-Optimal Rule Discovery. Data Mining and Knowledge Discovery, vol. 10, no. 1, pp. 39-79.

[7] Brin, S., Motwani, R., and Silverstein, C 1997: Beyond market baskets: Generalizing association rules to correlations. In Proceedings of the ACM SIGMOD International Conference on Management of Data.ACM, Tucson, Arizona, 265-276

[8] Savasere, A., Omiecinski, E., Navathe, and S.B. 1998: Mining for strong negative associations in a large database of customer transactions. In: ICDE, pp. 494–502

[9] Wu, X., Zhang, C., Zhang, S. 2004 Efficient Mining of Both Positive and Negative Association Rules, ACM Transactions on Information Systems, Vol. 22, No. 3, Pages 381–405.

[10] Aggrawal, C., Yu, P. 1998. A new framework for itemset generation. In Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. ACM, Seattle, Washington, 18-24.

[11] Bayardo, B. 1998. Efficiently mining long patterns from databases. In Proceedings of the ACM SIGMOD International Conference on Management of Data.ACM, Seattle, Washington, 85-93.

[12] Chen, M., Han, J., and Yu, P. 1996. Data mining: An overview from a database perspective .IEEE Trans. Knowledge and Data Eng. 8, 6(Nov), 866-881.

[13] Han, Jian., Pei, J., and Yin, Y. 2000. Mining frequent patterns without candidate generation. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, ACM, Dallas, Texas, 1-12.

[14] Srikant, R and Agrawal, R. 1996. Mining quantitative association rules in large relational tables. In Proceedings of the ACM SIGMOD International Conference on Management of Data. ACM, Montreal, Quebec, Canada.

[15] Srikant, R. and Agrawal, R. 1997. Mining generalized association rules. Future Generation Computer Systems 13, 2-3(Nov.), 161-180.

[16] Sotiris Kotsiantis, Dimitris Kanellopoulos.2006. Association Rules Mining: A Recent Overview**.** GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), pp. 71-82.

[17] Cohen, E., M. Datar, S. Fujiwara, A. Gionis, R. Indyk, P. Motwani, J. Ullman, and C. Yang: 2000.Finding Interesting Associations without Support Pruning. In Proceedings International Conference on Data Engineering.

[18] H. Mannila.1998. Database Methods for Data Mining. In Proceedings of Fourth Int'l Conf. Knowledge Discovery and Data Mining.

[19] Y.-H. Hu and Y.-L. Chen.2006.Mining Association Rules with Multiple Minimum Supports: A New Mining Algorithm and a Support Tuning Mechanism. Decision Support Systems, vol. 42, pp. 1-24.

[20] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur.1997.Dynamic Itemset Counting and Implication Rules for Market Basket Data. In proceedings of SIGMOD Record, vol. 26, pp. 255-264.

[21] C.C. Aggarwal and P.S. Yu, 1998.A New Framework for Itemset Generation. In Proceedings of 17th ACM SIGACT-SIGMOD-SIGART Symp. Principles of Database Systems (PODS '98), pp. 18-24.

[22] J. Blanchard, F. Guillet, H. Briand, and R. Gras.2005.Assessing Rule Interestingness with a Probabilistic Measure of Deviation from Equilibrium. In Proceedings of 11th Int'l Symp. Applied Stochastic Models and Data Analysis (ASMDA '05), pp. 191-200.

[23] Brin, S., Motwani, R. and Silverstein, C. 1999. Beyond Market Baskets: Generalizing Association Rules to Correlations. In Proceedings of the ACM SIGMOD Conference, pp. 265-276.

[24] Chun-Hao Chen, Guo-Cheng Lan, Tzung-Pei Hong, Yui-kai Lin. 2013. Mining high coherent association rules with consideration of support measure Expert Systems with Applications. Elsevier 6531-6537.