

# Transition in Time Series Data Mining on Correlated Items

D.Sujatha  
Aurora's Technological and  
Research Institute  
Parvathapur,Uppal  
Hyderabad

Priti Chandra  
Scientist,Advanced System  
Laboratory  
Hyderabad

B.L.Deekshatulu  
Distinguished Fellow  
IDRBT  
Hyderabad

## ABSTRACT

We are given a large database of customer transactions, where each transaction consists of transaction-id, the items bought in the transaction and the transaction time. The whole set of transaction is divided into a number of segments called durations (intervals) based on transaction time. And the dividing standard can be monthly, quarterly or yearly. We introduce the problem of mining strong association rules between consecutive durations using FP-tree and correlation coefficient, which is used to quantitatively describe the strength and sign of a relationship between two variables. This paper deals with the changes in the correlation between any two itemsets at the transition of the consecutive duration. Milestone is a change over point between durations. The transition may be positive or negative which are time points at which the pattern is either positively or negatively correlated. Also the method provides rare items, whose support is poor but are highly correlated.

## General Terms

Data Mining

## Keywords

Association Rule mining; support; Itemsets; Frequent Patterns; FP-Tree; Correlation; Correlation Coefficient ( $\phi$ )

## 1. INTRODUCTION

The process of extracting useful information from large quantities of data is known as Data mining. Such useful information is hidden and various techniques are applied to the data to obtain it. Association rule mining is one such method of discovering knowledge hidden in databases.

In general, an association rule denotes a relationship between two items in the same database. It is written in the form  $X \rightarrow Y$ , where X and Y are items and  $X \rightarrow Y = \phi$  It means that item Y is purchased or taken along with item X. The strength or validity of such rule is shown by two terms namely Support and Confidence. Support specifies the frequency with which items X and Y occur together in the database, while confidence is a measure of strength of the association rule which is basically the number of times purchase of item X has resulted in purchase of item Y. Rules mined from the database are pruned using these two attributes.

Association Rule Mining has been under study for a long time and still efforts to determine association rules or frequent patterns in a flexible, efficient and with minimum mathematical assumptions keep coming. In its early day's association rules were determined by using Apriori algorithm

[1] where rules were generated by finding candidates and verifying that their support and confidence meet a predefined minimum support and confidence. This approach was greatly limited because of its redundancy in generating candidates and multiple database scans hence its performance was greatly affected by the size of the database. FP-Growth algorithm [2] followed the Apriori, and it overcame the drawback by eliminating the candidate generation phase and multiple database scans. As time went on many methods and approaches have come up to improve association rule mining. However, association rules generated by such logical means could not prove to be strong. So mathematical approaches have come up such as correlation and principal component analysis to analyze or validate association rules [3].

Correlation mainly, promises to find meaningful association rules by providing information on how an association between two items behaves, i.e. whether the two items are positively correlated (purchase of one items results mostly in the purchase of the other) or negatively correlated (purchase of one items hinders the possibility of purchase of the other) or not correlated (items are independent).

## 2. RULE EFFECTIVENESS

Finding association rules using Apriori, FP-Growth utilizes support mainly to remove unimportant or less frequent itemsets. The minimum support is predefined and its assumed value greatly affects the number of rules generated. However, as already said, mere rules which qualify minimum support are not meaningful and moreover we might need to find items that do not go together.

Coming to correlation, the limitation here is that, we find the correlation coefficient of a pair of items and can say whether they are positively correlated or negatively correlated or not correlated, but the soundness of such itemsets are questionable, i.e. we have to make sure that the items in the itemset have a considerable share in the database to be considered.

The notions of support and correlation do not always agree with each other [4]. For instance, suppose we have an item pair {A, B}, where support (A) = support (B) = 0.8 and support (A, B) = 0.64. Both items are not correlated because support (A, B) = support (A) \* support (B). In contrast, the item pair {A, B} with support (A) = support (B) = support (A, B) = 0.001 is perfectly correlated regardless of its low support.

In order to ensure that a rule is both meaningful and strong enough to be considered, we can make use of support as well as correlation as factors to identify sound pairs of items. We can identify highly correlated pairs that have poor support, a

way to find rare events and also identify low correlated pairs that have high support, a way to prune a less useful frequent itemset. Identifying itemsets with fair correlation and support will be more helpful than just finding highly correlated items or itemsets with high support [5].

#### MILESTONES

A frequent pattern is not always static throughout the database. Every pattern in a real-world dataset has a dynamic behavior. The frequencies of the items in the pattern increase/decrease dramatically at some points in the database. There may be many milestones throughout the time period of the database, identifying a significant milestone which is a point in the database where there is a significant change in the frequencies of the items in the itemset. Statistically, plotting the frequencies of an item can give an overview of how it appears throughout the database. But this is not efficient as it requires a massive data interval which is not feasible to analyze.

In order to observe how the two items in a pattern vary with each other at different time points in the database, in [9] method to find lightly supported Boolean association rule is proposed by dividing the data into subsets according to time and then apply Apriori to mine these subsets and get the initialized rule set. Then finds support and confidence of every rule in other time periods and forms the rule matrix. In [6] a method to find significant milestones for a transitional pattern, which are time points at which the frequency of the pattern changes most significantly. In this paper we employ correlation between the two items to identify their behavior. A pattern that is found to be positively correlated after finding its correlation coefficient using the entire database might not always be positively correlated as we proceed through the database, the items can be negatively correlated during some time interval of the database. Hence finding time points in the database where such changes occur can be quite helpful data.

Milestones provide a wide range of applications. For instance, in the market basket scenario, business owners can identify what combinations of products have become more popular and what combinations of products have lost popularity, identifying this they can plan their business accordingly and position products to in their retail environment. Another application in the medical domains, where data collected from a group of patients with similar disease administered with a new drug, can help identify at what point of time certain symptoms occur and the drug responsible. This way the side effects of a new drug can be indicated.

#### CORRELATION COEFFICIENT

Correlation coefficient measures the strength of a linear relationship between two variables.

The correlation coefficient for two variables X and Y is given by the formula [4],

$$\left[ \rho = \frac{\text{Cov}(x, y)}{\sigma_x \sigma_y} \right] \quad [1]$$

In Equation 1, Cov( X ,Y) denotes the covariance of the two variables while  $\sigma_x$  stands for standard deviation. The values of  $\rho$  range between -1 and +1. The correlation coefficient is 0 if the two variables are independent. When  $\rho = +1$  the variables are perfectly positively correlated. Similarly, when

$\rho = -1$  the variables are considered to be perfectly negatively correlated. Positive correlation indicates that when X increases or decreases so does Y. Negative correlation occurs when an increase in X causes a decrease in Y and vice versa.

Correlation can be tested in many ways using statistics; one such method is based on Chi-square test as proposed in [7].

Let us consider two variables X and Y. Table 1 gives us the contingency table of X and Y, i.e. the frequency associated with all the possible combinations of X and Y. N is the size of the dataset considered. For such binary variables, Pearson introduced the correlation coefficient as in equation 2:

TABLE 1. X and Y CONTINGENCY TABLE

	Y	-Y	$\Sigma_{\text{row}}$
X	$f_{11}$	$f_{10}$	$f_{1+}$
-X	$f_{01}$	$f_{00}$	$f_{0+}$
$\Sigma_{\text{col}}$	$f_{+1}$	$f_{+0}$	N

$$\phi = \frac{f_{11}f_{00} - f_{10}f_{01}}{\sqrt{f_{+0}f_{0+}f_{+1}f_{1+}}} \quad [2]$$

We use the above formula to determine the correlation between two items in our approach [8].

### 3. EXISTING APPROACHES

In this section, we discuss existing work related to transitional pattern mining, milestone detection and work related to determining frequent itemsets through correlation analysis.

Itemsets whose support increases significantly from one data set to another are called as emerging patterns, this was proposed in [10]. Emerging patterns are used to capture the significant differences between two datasets by contrasting the growth rate of a pattern. Growth rate is the ratio of the support of the itemset in the two datasets. This only tells us how the frequency of an itemset varies at different time points, without addressing the relationship between the items, even if the support in the second dataset is more, it cannot answer the relative increase or decrease between the items in the itemset although their overall support satisfies the threshold.

A temporal association rule is an association rule that holds during specific time intervals. This was an extension to the problem of association rule mining that started over a decade ago. There are several kinds of temporal association rules having meaningful implication such as cyclic association rules, rules that periodically over time [11] and temporal association rules over items' lifespan (i.e., the period between the first and the last time the item appears in transactions of a database) [12]. Even in these approaches of mining temporal association rules, although itemsets are analyzed over a period of time, it is not concerned about how the relation between the items in the pattern varies from time to time, it just informs about the pattern's behavior as a whole.

Other previous studies that considered time stamps and periodicity in mining frequent patterns are sequential pattern

mining [13][14][15]. A sequential pattern, defined first by Agrawal and Srikant [13], is a sequence of elements whose occurrence frequency in a set of sequences (called a sequence database) is not less than a minimal support threshold. Early sequential pattern mining algorithms (e.g., in [13], [14]) are based on a level-wise candidate generation and testing process, in which length-k candidates are generated from the frequent sequences of length k - 1 and then tested by scanning the database to compute the frequency of each candidate. Some later algorithms improve the efficiency of sequential pattern mining by, e.g., using a recursive divide-and-conquer procedure that generates the complete set of frequent sequences without candidate generation and testing [15]. These methods have the same drawback of not dealing with the nature of the items in a pattern from time to time.

An efficient method to determine the relationship between items in an itemset is by using the statistical measure of correlation. In [4][5][7][8], this method has been utilized to mine frequent itemsets and also show their correlation (i.e., saying whether the items in an itemset are positively correlated or negatively correlated). Approached using correlation did not consider time as a factor in analysis of association rules.

#### 4. PROPOSED APPROACH

The goal of our paper is to suggest a method to find strong meaningful rules (with correlation and support) with means to provide information on transitional patterns basing on their correlation.

We present an approach that comprises efficient mining, correlation based relationship between items in a frequent itemset and also significant milestone detection to identify when items in a pattern behave differently.

To achieve this we will make use of an FP-Tree generated from frequent items, then find correlation of adjacent nodes in the tree, itemsets are formed by considering pair of items from each path of the tree. Each path starts from the root and ends at a leaf node. Items in a path are associated; hence we find the Pearson correlation coefficient of adjacent items and also their support making the discovered patterns more meaningful.

To determine how the itemset behaves, for a particular pair we find its Pearson correlation coefficient at different time intervals of the database. We plot these coefficients on a line graph and we can observe how the behavior between the items in the pattern varies at different time points in the database.

We use FP-Tree approach here because it eliminates the generation of massive candidates and also prevents unnecessary database scans. Storing Transaction ID's for the frequent items also reduces the effort as we need not scan the dataset to find support again and also we need not worry about storing the support for each pattern, thus reducing memory usage.

#### Algorithm

Inputs: Dataset, number of transactions, number of items, minimum support.

Output: Highly positive and negative correlated items.

Method:

- 1) Scan dataset line by line and store for each item, all the TIDs in which it occurs.
- 2) Determine frequent items by calculating frequency of each item
- 3) Prune based on minimum support
- 4) Construct FP-Tree with the frequent items
- 5) Determine all the paths in the FP-Tree
- 6) Take one path at a time and form pairs of items in the path.
- 7) For each pair formed, determine support and calculate Pearson correlation coefficient.
- 8) Prune itemsets based on their support and correlation coefficient values to obtain highly positive and negative correlated itemsets with optimal support.
- 9) Partition the transactions in the database into intervals based on the timestamps.
- 10) In each interval find the Pearson correlation coefficient for the itemsets obtained in step 8.
- 11) For a particular itemset plot the values of the correlation coefficient.

Example: For an example we assume the transactional dataset as shown in Table 2 which has been adopted from [9]. This dataset consists of thirty transactions and consists of six items. The dataset has been partitioned into four blocks namely time periods I, II, III and IV in order to explain our approach.

To this we apply our algorithm,

**TABLE 2. THE TRANSACTION DATASET**

Transaction ID	Items	Time Period
1	I <sub>1</sub> , I <sub>2</sub> , I <sub>5</sub>	I
2	I <sub>2</sub> , I <sub>4</sub> , I <sub>6</sub>	
3	I <sub>2</sub> , I <sub>3</sub>	
4	I <sub>1</sub> , I <sub>2</sub> , I <sub>4</sub>	
5	I <sub>1</sub> , I <sub>3</sub>	
6	I <sub>2</sub> , I <sub>3</sub>	
7	I <sub>1</sub> , I <sub>3</sub>	II
8	I <sub>1</sub> , I <sub>2</sub> , I <sub>3</sub> , I <sub>5</sub>	
9	I <sub>1</sub> , I <sub>2</sub> , I <sub>3</sub>	
10	I <sub>1</sub> , I <sub>5</sub>	
11	I <sub>3</sub> , I <sub>4</sub>	
12	I <sub>2</sub> , I <sub>3</sub>	

13	I <sub>1</sub> , I <sub>2</sub> , I <sub>5</sub> , I <sub>6</sub>	III	
14	I <sub>1</sub> , I <sub>5</sub>		
15	I <sub>2</sub> , I <sub>4</sub>		
16	I <sub>1</sub> , I <sub>3</sub>		
17	I <sub>1</sub> , I <sub>2</sub> , I <sub>4</sub> , I <sub>5</sub>		
18	I <sub>4</sub> , I <sub>6</sub>		
19	I <sub>1</sub> , I <sub>5</sub>		
20	I <sub>4</sub> , I <sub>5</sub>		
21	I <sub>2</sub> , I <sub>3</sub> , I <sub>4</sub> , I <sub>7</sub>		
22	I <sub>2</sub> , I <sub>5</sub>		
23	I <sub>3</sub>		
24	I <sub>3</sub> , I <sub>4</sub> , I <sub>5</sub>		IV
25	I <sub>1</sub> , I <sub>2</sub> , I <sub>3</sub> , I <sub>4</sub>		
26	I <sub>1</sub> , I <sub>3</sub>		
27	I <sub>1</sub> , I <sub>3</sub> , I <sub>5</sub>		
28	I <sub>1</sub> , I <sub>2</sub> , I <sub>4</sub>		
29	I <sub>1</sub> , I <sub>2</sub> , I <sub>7</sub>		
30	I <sub>1</sub> , I <sub>2</sub> , I <sub>3</sub>		

Initially, we find the frequencies of all the items and determine frequent items as shown in Table 3.

**TABLE 3. FREQUENT ITEMS**

Items	Support Count
I <sub>1</sub>	18
I <sub>2</sub>	17
I <sub>3</sub>	15
I <sub>4</sub>	11
I <sub>5</sub>	11

As we scan the dataset, we store for each item the transaction ID's in which it occurs as in Table 4.

**TABLE 4. STORED TRANSACTION ID'S OF FREQUENT ITEMS**

Item	Transaction ID's
I <sub>1</sub>	1, 4, 5, 7, 8, 9, 10, 13, 14, 16, 17, 19, 25, 26, 27, 28, 29, 30
I <sub>2</sub>	1, 2, 3, 4, 6, 8, 9, 12, 13, 15, 17, 21, 22, 25, 28, 29, 30
I <sub>3</sub>	3, 5, 6, 7, 8, 9, 11, 12, 16, 21, 23, 24, 25, 26, 27, 30
I <sub>4</sub>	2, 4, 11, 15, 17, 18, 20, 21, 24, 25, 28
I <sub>5</sub>	1, 8, 10, 13, 14, 17, 19, 20, 22, 24, 27

Once the frequent items are identified, we construct an FP-Tree by using the traditional approach, we donot store support for each item, we simply form a tree. Fig. 1 shows the FP-Tree for the example dataset.

Once the tree is formed, we scan each path from the root to the leaf-node.

For example

Root→I<sub>1</sub>→I<sub>2</sub>→I<sub>5</sub> forms one such path in the FP-Tree. This has been marked in Fig. 1.

The items in the path form a pattern i.e. {I<sub>1</sub>,I<sub>2</sub>}, {I<sub>1</sub>,I<sub>5</sub>}.

P{I<sub>2</sub>,I<sub>5</sub>} are valid itemsets. To find the support of pattern {I<sub>1</sub>,I<sub>2</sub>}, we make use of Table 4 and perform intersection of I<sub>1</sub> and I<sub>2</sub> transaction ID's. Similarly, we construct a contingency table for this pattern as in Table 5 to determine the Pearson correlation coefficient using equation (1).

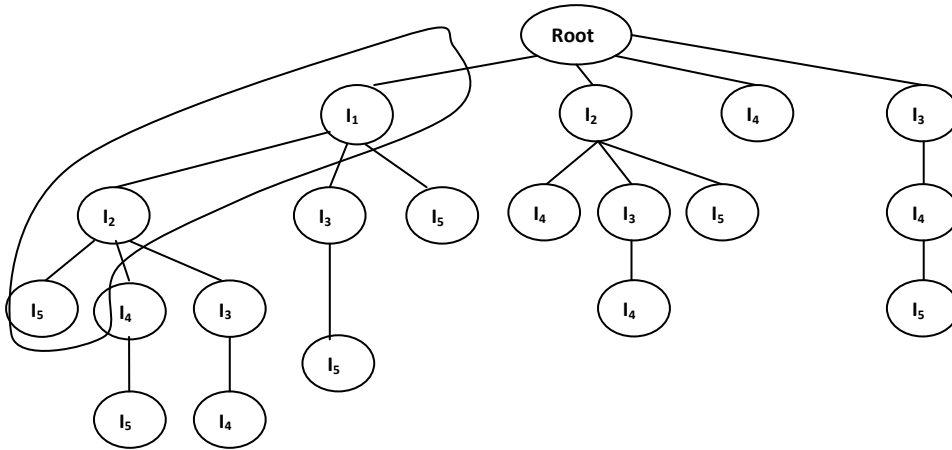


FIG. 1. FP-TREE

TABLE 5. CONTINGENCY TABLE FOR I<sub>1</sub> AND I<sub>2</sub>  
DURING TIME PERIOD I-IV

	I <sub>2</sub>	-I <sub>2</sub>	Σ <sub>row</sub>
I <sub>1</sub>	10	8	18
-I <sub>1</sub>	7	5	12
Σ <sub>Col</sub>	17	13	30

We get the Pearson correlation coefficient for pattern {I<sub>1</sub>,I<sub>2</sub>} as  $\phi = -0.0274$ , this means that the pattern is less negatively correlated and has a high support of 18. The coefficient is calculated considering the entire dataset.

Inorder to observe the behaviour of this itemset, we find  $\phi$  of it during different time periods of the dataset, we obtain Table 6.

TABLE 6.  $\phi$  FOR PATTERN I<sub>1</sub> AND I<sub>2</sub> AT DIFFERENT INTERVALS

Time Period	$\phi ( I_1 \text{ and } I_2 )$	$\phi ( I_2 \text{ and } I_5 )$	$\phi ( I_1 \text{ and } I_5 )$
I	-0.4472	0.2	0.4472
II	0	0	0.5774
III	-0.1581	0.1	0.3162
IV	0.4714	-0.7303	-0.6455
I-IV	-0.0274	-0.1722	0.1977

In Fig. 2. The graph shows us how the correlation between items I<sub>1</sub> and I<sub>2</sub> vary through out the dataset at different time intervals. The items are negatively correlated during time periods I and III, positively correlated during interval IV and not correlated in interval II. Similarly patterns {I<sub>2</sub>, I<sub>5</sub>} and {I<sub>1</sub>, I<sub>5</sub>} are shown.

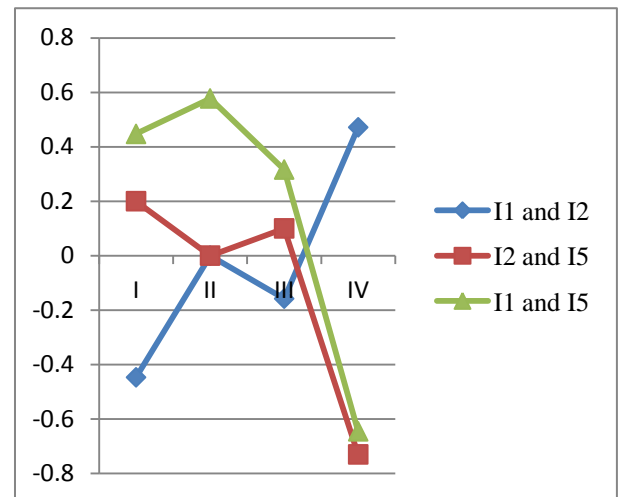


FIG. 2.VARIATIONS IN  $\phi$  FOR PATTERN {I<sub>1</sub>,I<sub>2</sub>}

Similarly, we can find itemsets that have good support as well as correlation and observe how the correlation between them varies throughout the database.

## 5. RESULTS

In this section, we report the results of testing our method on mushroom dataset provided by fimi repository, which has 8124 transactions with 119 distinct items.

The experiment has been performed with minimum support as 3000 and fixed block sizes specified next. The resulting itemsets shown in the following tables are determined from the FP-Tree constructed for the entire dataset.

Table 7 shows highly positively correlated items (i.e whose correlation coefficient is greater than 0.5) which have support greater than minimum support which is 3000 transactions. For these item pairs we have found the correlation coefficients at different time intervals as in Table 8, by dividing the dataset into the following blocks, each block having transactions as shown below.

BLOCK1 (B1): 1-2500  
BLOCK2 (B2): 2501-4300  
BLOCK3 (B3): 4301-6900  
BLOCK4 (B4): 6901-8124

**TABLE 7. POSITIVELY CORRELATED ITEMS**

Itemset		Correlation ( $\phi$ )	Support
Item	Item		
1	24	0.5015	3292
2	28	0.7856	3408
2	39	0.5400	3920
2	93	0.5405	3152
23	59	0.5616	3232
23	63	0.5058	3040
23	93	0.7640	3184
34	86	0.9352	7906
59	63	0.5301	4156
59	93	0.5817	3664
63	93	0.5351	3470
67	76	0.5515	3520

**TABLE 8. BLOCK WISE CORRELATION OF ITEMSETS SHOWN IN TABLE 7**

Itemset		Correlation coefficient ( $\phi$ )			
Item	Item	Block 1	Block 2	Block 3	Block 4
1	24	-0.2226	0.9643	0.3862	0.1970
2	28	0.4002	0.9988	0.8487	0.9865
2	39	0.7143	0.4313	0.1450	0.9460
2	93	-0.2111	0.7528	0.0227	1
23	59	0.5865	0.7755	0.3893	0.1397
23	63	0.3706	0.7787	0.4079	0.1399
23	93	0.9127	0.7929	0.7211	0.1970
34	86	NaN	NaN	0.9485	0.9243
59	63	0.2804	0.9747	0.3980	0.2247
59	93	0.6426	0.9720	0.2237	0.2209
63	93	0.4302	0.9748	0.2326	0.2290
67	76	0.5941	0.3799	0.4496	0.4271

From Table 8, we observe that the items 2 and 93 are negatively correlated in Block 1 (i.e., purchase of item 2 results in less chance of purchase of item 93), the same pair is highly positively correlated in Block 2 (i.e., purchase of item 2 results greatly in the purchase of item 93), in Block 3 the pair the correlation coefficient is nearly zero (i.e., the two

items do not go together most of the time), finally in Block 4, the correlation for the pair is one. (i.e., purchase of item 2 definitely results in purchase of item 93). Hence we can see the usefulness and meaningfulness of determining the information regarding item relationships at different time periods. We can conclude here that the transition from Block 1 to Block 2 is a significant milestone for pattern {2, 93} as the correlation changes from negative to highly positive. Similarly, we can analyze for the other patterns shown in the table.

NaN denotes that it can't be determined, it occurs in cases where the pattern occurs in all the transactions of a dataset block.

Table 9 shows highly negatively correlated items (i.e whose correlation coefficient is smaller than 0.5) which have support greater than minimum support which is 3000 transactions. For these item pairs we have found the correlation coefficients at different time intervals as in Table 10, by dividing the dataset into the following blocks, each block having transactions as shown below.

**TABLE 9. NEGATIVELY CORRELATED ITEMS**

Itemset		Correlation ( $\phi$ )	Support
Item	Item		
2	36	-0.34839	3008
24	36	-0.29947	3540
36	67	-0.392	3160
36	76	-0.36777	3128

**TABLE 10. BLOCK WISE CORRELATION OF ITEMSETS SHOWN IN TABLE 9**

Itemset		Correlation coefficient ( $\phi$ )			
Item	Item	Block 1	Block 2	Block 3	Block 4
2	36	-0.2401	-0.2489	-0.3952	-0.6163
24	36	-0.8233	-0.3256	-0.0552	-0.0856
36	67	-0.3117	-0.3610	-0.2076	-0.4754
36	76	-0.3117	-0.3536	-0.0159	-0.4787

Coming to negatively correlated pairs, our approach does not concerns about perfectly negatively correlated itemsets (i.e., items with correlation as -1), because such itemsets won't help much in product positioning in a market. Such item pairs do not appear together in a single path in the FP-Tree.

We have also discovered item pairs with high correlation but poor support, such itemsets are referred to as rare items in the context of association rule mining. This was done by not setting any minimum support and looking for highly correlated and low support pairs. Hence with our approach mining rare association rules is also possible. Table 11 shows the rare itemsets and their correlation and support values.

**TABLE 11. RARE ITEMSETS WITH HIGH POSITIVE CORRELATION AND LOW SUPPORT**

Itemset		Correlation ( $\phi$ )	Support
Item	Item		
1	48	0.53881	1728
29	61	0.58419	1584
29	66	0.60045	1584
29	95	0.72393	1296
29	101	0.79975	1584
37	54	0.56966	768
37	114	0.53784	1056
38	48	0.7769	1728
38	58	0.60215	1808
38	94	0.53777	1816
38	102	0.63468	1824
48	58	0.78413	1728
48	94	0.72145	1728
48	102	0.80557	1728
48	110	0.5226	1728
56	101	0.50845	1584
58	94	0.6225	1952
61	66	0.67707	1800
61	95	0.67843	1296
61	101	0.55373	1296
66	95	0.69243	1296
66	101	0.56787	1296
95	101	0.86893	1296

Hence our results support the claims of our approach.

Observations: assuming the mushroom dataset from fimi repository. It was observed that finding highly correlated pairs in a large dataset is quite small compared to the number of pairs that exist when only support and confidence are considered.

## 6. CONCLUSION

This paper highlights the importance of correlation over the transactions by grouping the transactions into different time intervals. These intervals need not be uniform and will depend on occasions, seasons etc. An interesting offshoot of this method is to find correlation that exist even in rare transactions in a large database.

## 7. FUTURE WORK

In the future, we would like to extend this work in the following directions. First, computing correlation coefficients for a large number of pairs can be expensive. To address the efficiency, improved quantities that measure associations or correlation can be found. Second, we would like to explore other types of patterns (such as sequential, periodical etc).

## 8. REFERENCES

- [1] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proc. 1993 ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '93), pp. 207-216, 1993.
- [2] J.Han, J. Pei and Y. Yin. "Mining frequent patterns without candidate generation". In ACM SIGMOD 2000.
- [3] Wen-Chi Hou. "Extraction and applications of statistical relationships in relational databases." IEEE Transactions on knowledge and data engineering, 1996.
- [4] H. Xiong, S. Shekhal, P. N. Tan, V. Kumar, "Exploiting a support-based Upper Bound of Pearson's Correlation Coefficient for Efficiently Identifying stringly correlated pairs.
- [5] J. Zhang, J. Feigenbaum, "Finding highly correlated pairs efficiently with powerful pruning."
- [6] Q. Wan and A. An, "Transitional patterns and their significant milestones." Proc. Seventh IEEE International conference Data Mining, 2007.
- [7] S. Brin, R. Motwani, and C. Silverstein. "Beyond market baskets: Generalizing association rules to correlations." In ACM SIGMOD, 1997.
- [8] Wu, X. Zhang, C. Zhang, "Mining both positive and negative association rules." In Proc. ICML, 2002.
- [9] Z. Zhang, Z. Chen, "Time series data mining method based on lightly-supported boolean association rules." Proc. CIKM'06.
- [10] G.-Z. Dong and J.-Y. Li, "Efficient Mining of Emerging Patterns: Discovering Trends and Differences," Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '99), pp. 43-52, 1999.
- [11] B. Ozden, S. Ramaswamy, and A. Silberschatz, "Cyclic Association Rules," Proc. 14th Int'l Conf. Data Eng. (ICDE '98), pp. 412-421, 1998.
- [12] J.M. Ale and G.H. Rossi, "An Approach to Discovering Temporal Association Rules," Proc. 2000 ACM Symp. Applied Computing (SAC'00), pp. 294-300, 2000.
- [13] R. Agrawal and R. Srikant, "Mining Sequential Patterns," Proc. 11th Int'l Conf. Data Eng., pp. 3-14, 1995.
- [14] R. Agrawal and R. Srikant, "Mining Sequential Patterns: Generalizations and Performance Improvements," Proc. Fifth Int'l Conf. Extending Database Technology (EDBT '96), 1996.
- [15] J. Pei, J. Han, B. Mortazavi-Asl, J.-Y. Wang, H. Pinto, Q.-M. Chen, U. Dayal, and M.-C. Hsu, "Mining Sequential Patterns by Pattern-Growth: The Prefixspan Approach," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 11, pp. 1424-1440, Nov. 2004.