Volume 4, Issue 12, December 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Binary Matrix Approach for Mining Frequent Sequential Pattern in Large Databases

Sowjanya Pathi, Amarendra Kothalanka, Vasudevarao Addala
Department of CSE
DIET, JNTUK, Andhra Pradesh, India

Abstract- Data mining is the computational process of discovering patterns in large datasets involving methods at the intersection of artificial intelligence, machine_learning, statistics, and systems. Mining sequential patterns from inaccurate or uncertain data, such as those data arising from sensor readings and GPS trajectories, is important for discovering hidden knowledge in such applications. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Here this paper was proposed to find the frequent sequential patterns in a large uncertain database. For the finding of frequent pattern it is using binary matrix approach. The binary matrix approach is efficient and more flexible for finding the frequent pattern. By using this approach the time complexity was reduced for finding frequent patterns.

Keywords— Frequent patterns, uncertain databases, Apriori, Binary matrix, Sequential pattern

I. INTRODUCTION

Sequential pattern mining is the mining of frequently occurring ordered events or subsequences as patterns. An example of a sequential pattern is "Customers who buy a Canon digital camera are likely to buy an HP color printer within a month." For retail data, sequential patterns are useful for shelf placement and promotions. This industry, as well as telecommunications and other businesses, may also use sequential patterns for targeted marketing, customer retention, and many other tasks. Other areas in which sequential patterns can be applied include Web access pattern analysis, weather prediction, production processes, and network intrusion detection. Notice that most studies of sequential pattern mining concentrate on categorical (or symbolic) patterns, where as numerical curve analysis usually belongs to the scope of trend analysis and forecasting in statistical time-series analysis.

The problem of mining Frequent Sequential Patterns (FSPs) from deterministic databases has attracted a lot of attention in the research community due to its wide spectrum of real life applications [5]–[9]. For example, in mobile tracking systems, FSPs can be used to classify or cluster moving objects [3]; and in biological research, FSP mining helps discover correlations among gene sequences [4].

Consider a supermarket with a large collection of items. Typical business decisions that the management of the supermarket has to make include what to put on sale, how to design coupons, how to place merchandise on shelves in order to maximize the profit, etc. Analysis of past transaction data is a commonly used approach in order to improve the quality of such decisions. Until recently, however, only global data about the cumulative sales during some time period (a day, a week, a month, etc.) was available on the computer. Progress in bar-code technology has made it possible to store the so called basket data that stores items purchased on a per-transaction basis. Basket data type transactions do not necessarily consist of items bought together at the same point of time. It may consist of items bought by a customer over a period of time. Examples include monthly purchases by members of a book club or a music club.

Several organizations have collected massive amounts of such data. These data sets are usually stored on tertiary storage and are very slowly migrating to database systems. One of the main reasons for the limited success of database systems in this area is that current database systems do not provide necessary functionality for a user interested in taking advantage of this information. The problem of sequential pattern mining has been well studied in the literature in the context of deterministic data, and many algorithms have been proposed to solve this problem, including Apriori, PrefixSpan [5], SPADE [7], FreeSpan [8] and GSP [9].

Here the new approach is comparing with the Apriori algorithm. Apriori is as seminal algorithm proposed by R.Agrawal and R.Srikant in 1994 for mining frequent itemsets for Boolean association rules. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties. It needs candidate item generation. By using this technique it will take a lot of time for finding frequent pattern.

This paper proposing new technique for finding frequent patterns i.e. Binary matrix approach. By using this approach it will find the frequent patterns and also reduce the time complexity for finding frequent items.

II. RELATED WORK

A comprehensive survey of traditional data mining problems such as frequent pattern mining in the context of uncertain data can be found in [20]. Here only some concepts and issues arising from traditional sequential pattern mining are detailed.

© 2014, IJARCSSE All Rights Reserved

A. Traditional Sequential Pattern Mining

Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore(k+1)-itemsets. First, the set of frequent1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted L_1. Next, L_1 is used to find L_2, the set of frequent 2-itemsets, which is used to find L_3, and so on, until no more frequent k-itemsets can be found. The finding of each L_k requires one full scan of the database.

To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property, presented below, is used to reduce the search space.

Apriori property: All non empty subsets of a frequent itemset must also be frequent.

The Apriori property is based on the following observation. By definition, if an itemset I does not satisfy the minimum support threshold, min sup, then I is not frequent; that is, P(I)<min sup. If an item A is added to the itemset I, then the resulting itemset (i.e., I∪A) cannot occur more frequently than I. Therefore, I∪A is not frequent either; that is, P(I∪A)<min sup.

So in the Apriori algorithm it is finding the sequence of pattern and also find support count of each item. So that finding support count and frequent item sets in the iteration phase it will time consuming process.

III. PRELIMINARIES

In this section several fundamental concepts will be discussed.

A. Formal Model

Let I = I1; I2; . . .; Im be a set of binary attributes, called items. Let T be a database of transactions. Each transaction t is represented as a binary vector, with t[k]= 1 if t bought the item Ik, and t[k] = 0 otherwise. There is one tuple in the database for each transaction. Let X be a set of some items in I. So that a transaction t satisfies X if for all items Ik in X, t[k] =1.

B. Support Constraints

These constraints concern the number of transactions in T that support a rule. The support for a rule is defined to be the fraction of transactions in T that satisfy the union of items in the consequent and antecedent of the rule. Support should not be confused with confidence. While confidence is a measure of the rule's strength, support corresponds to statistical significance. Besides statistical significance, another motivation for support constraints comes from the fact that we are usually interested only in rules with support above some minimum threshold for business reasons. If the support is not large enough, it means that the rule is not worth consideration or that it is simply less preferred (may be considered later).

Generate all combinations of items that have fractional transaction support above a certain threshold, called minimum support. Call those combinations large itemsets, and all other combinations that do not meet the threshold small itemsets.

IV. BINARY MATRIX APPROACH

The proposed technique is used to find out sequential frequent patterns by using the data mining technique. This paper proposing new technique for finding frequent patterns i.e. binary matrix approach. By using this approach it will be find the frequent patterns and also reduce the time complexity for finding frequent items. Thus the research presented a new algorithm of mining maximum frequent itemsets first based on the Binary matrix of frequent length-1 itemsets.

The main idea of the algorithm is to create a Boolean matrix with frequent length-1 itemsets as row headings and transaction records' IDs as column headings(TABLE I). In the matrix, there are only two type of values, '1'and '0', which means that the transaction record contains or not the corresponding frequent length-1 itemset. Then it is necessary to calculate the number of value 1 in each column and the count of the columns with the same number of value 1.If the count of those columns is larger than the minimum support, in accordance, the number of value 1 in the column may be the size of maximum frequent itemset, vice versa.

Therefore, some values of which each may be the maximum frequent itemsets length will be calculated. Subsequently, a set of candidate itemsets used for extracting maximum frequent itemsets will be generated from frequent length-1 itemsets according to each maximum value and the support of each candidate itemset will be calculated based on the Binary matrix. If the support is larger than the minimum support, the candidate itemset is frequent, vice versa. Finally, all the frequent itemsets will be extracted from maximum frequent itemsets according to the nonempty sub-sets of frequent itemsets being still frequent.

Generally speaking, the main principles of the new algorithm include three aspects

A. Creating a Binary Matrix According to Frequent Length-1 Itemsets

All the frequent length-1 itemsets will be generated from transaction database when transaction database is scanned first time and for each frequent length-1 itemset, all the IDs of transaction records containing it need to be taken note in one array. Then the corresponding Binary array with the length being the number of the transaction records in database will be created for each frequent length-1 itemset. In each array, there are only two values, '0'and '1'. If transaction record contains frequent length-1 itemset, the value is 1 in the corresponding Binary n array, vice versa. At last, a Binary matrix will be constructed according to all the Boolean arrays of frequent length-1 itemsets.

© 2014, IJARCSSE All Rights Reserved

TABLE I A PART OF THE BOOLEAN MATRIX OF THE FREQUENT LENGTH-1 ITEMSETS

Frequent 1-itemsets	**The Transaction Records' ID**									
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
Mountain	1	1	1	0	1	1	0	0	1	1
Forest	0	1	0	1	1	1	1	1	1	1
Slope	1	1	0	0	0	1	0	1	1	1
Northwest	1	1	0	1	0	1	1	1	1	1

Definition 1: The corresponding Boolean array of each frequent length-1 itemset *Im*[*N*] is {BT_1, BT_2, ... , *BTn*} (1≤*n*≤*N*), where *Im* is the *mth* frequent length-1 itemset; *N* is the number of transaction records in database; *Tn* is ID of the *nth* transaction record respectively; and *BTn*'s value is 0 or 1 only.

Definition 2: The Boolean matrix of frequent length-1 itemsets ***IM*N*** is {*I*1[*N*], *I*2[*N*], ... , *Im*[*N*]} (1≤*m*≤*M*), where *Im*[*N*] is the Binary array with *N* dimensions of the *mth* frequent length-1 itemset; *M* is the number of frequent length-1itemsets.

Pseudo code

```
Input: Transaction Database D,
          Minimum support min_sup
Output: Frequent length-1 itemset L1
Begin
Find all the frequent length-1 itemset L1 from D
If L1 is not null
   for each Im in L1
      for each t in D
           if t contains(Im)   Im|t|=1
           else      Im|t|=0
         return Im|N|
         end for
    end for
 end if
IM*N={I1|N|, I2|N|, I3|N|,……,I1|N|}
End
```

Fig.1 Pseudo code of creating binary matrix

B. Extracting Maximum Frequent Itemsets from Boolean Matrix

Each column in the Binary matrix represents one transaction record. Value 0 in the column means the corresponding transaction record contains the corresponding frequent length-1 itemset, vice versa. Therefore, the number of value 1 in each column indicates the corresponding transaction record contains the number of frequent length-1 itemsets together. If there is the number of transaction records with the same number of value 1 being larger than the minimum support, the number of value 1 may be the size of maximum frequent itemset, vice versa.

As a result, a set of values in which each one may be maximum frequent itemset's length will be obtained. Then according to each of the values in descending order, a series of candidate itemsets will be generated from frequent length-1 itemsets and the support of each candidate itemset could be calculated according to the Binary matrix of frequent length-1 itemsets. If the support of each candidate itemset is larger than the minimum support, the candidate itemset is frequent, vice versa. At last, if the maximum frequent itemsets generated from the set of candidate itemsets are not empty, the size of candidate itemset is required, that is length of maximum frequent itemset. Otherwise, it is necessary to continue the previous operation to check the next value until maximum frequent itemsets are not empty. If all the maximum frequent itemsets are empty, the maximum length of frequent itemset is one.

Definition 3: *Max*[*n*] is an array used for storing some values of which each may be the length of maximum frequent itemset, where n is the size of *Max*[*n*].

Definition 4: The set of candidate itemsets of maximum frequent itemsets *C* is {*IM*1, *IM*2, ... , *IMn*}, therefore, the corresponding Binary matrix ***CMn*N*** is {*IM*1[*N*], *IM*2[*N*], ... ,*IMn*[*N*]}, where *IMn* is candidate itemset.

© 2014, IJARCSSE All Rights Reserved

Pseudo code

```
Input:  The binary matrix  frequent lengths-1 item sets Li
        Minimum support min_sup.
        Frequent length-1itemsets Li
Output: Maximum  frequent itemsets

Begin
For each column in the Binary Matrix
   Calculate  the number of value I in the current column
end for
return max[n]
sort( max[n])
for each one In the max[n]
      calculate number  of the columns with the same number value I
      if number >min_sup
            generate maximum length candidate itemsets from L,
            for each itemsets in the candidate itemsets
                  calculate Support(itemsets)
                  if support(itemsets)>min_ sup
                        itemset is frequent
                  end if
            end  for
       end if
       If maximum frequent itemsets is not null
      break;
      end if
end for
End
```

Fig.2 Pseudo code of extracting of maximum frequent items

Definition 5: The support of candidate itemset *C*, Support(*C*) = *IM1*[*N*] *And IM2*[*N*] *And ... IMn*[*N*]. Fig. 3 shows the example of the logical Binary operator "And" between the Binary arrays of candidate itemsets, where "*And*" is the logical Binary operator, if there exists value 0, then the calculation will be 0.

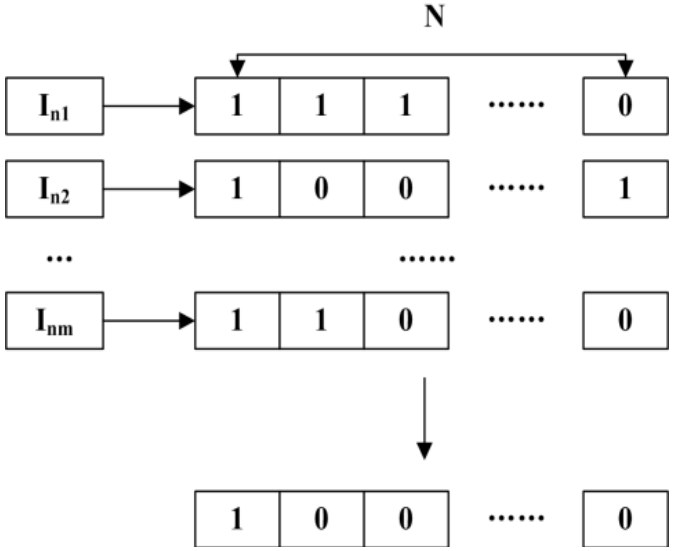

Fig. 3 The logical Binary operator of the Binary arrays of the set of Candidate itemset

C. Generating All the Frequent Itemsets from Maximum Frequent Itemsets

All the frequent itemsets could be extracted from all the maximum frequent itemsets according to the nonempty subsets of frequent itemsets being still frequent. And the support of each frequent itemset could be calculated by Definition 5. At last, all the strong association rules can be mined from all the Frequent itemsets.

© 2014, IJARCSSE All Rights Reserved

V. EXPERIMENTAL RESULT

The Binary matrix approach takes binary data for mining frequent patterns. The new approach will compared with old approach i.e. Apriori in terms of patterns count and time duration. Now here some patterns are given then based on minimum support count the frequent patterns are displayed for both algorithms. Then result will be analyzed in terms of pattern count and time duration. So the result as pattern count was equal but time duration was reduced for the new approach.

Then the given sequential pattern as

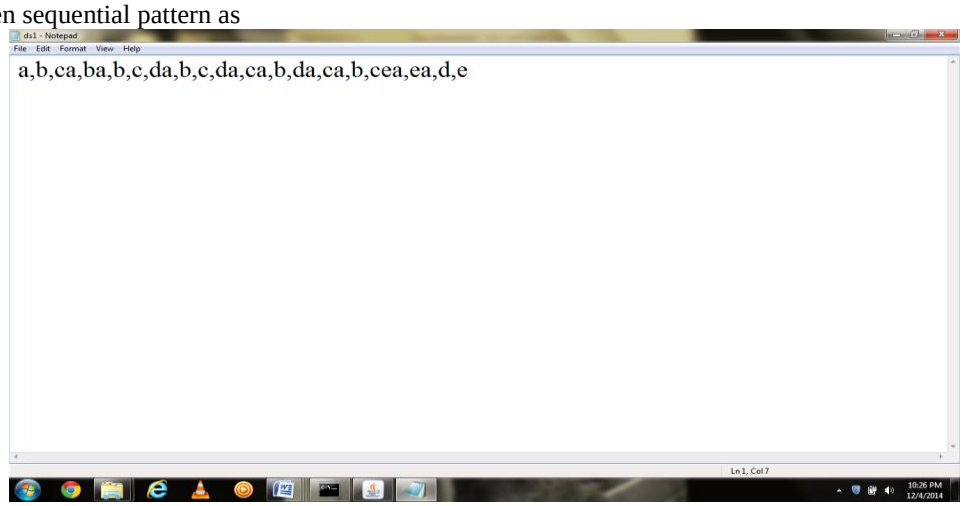

Fig.4 The given sequential pattern

Then by browsing this data to the experiment will be done. After that frequent patterns are mined for both algorithms. Then the analysis of those frequent patterns the pattern count and time duration will be displayed. This will be like this
The pattern count will be like in fig.4

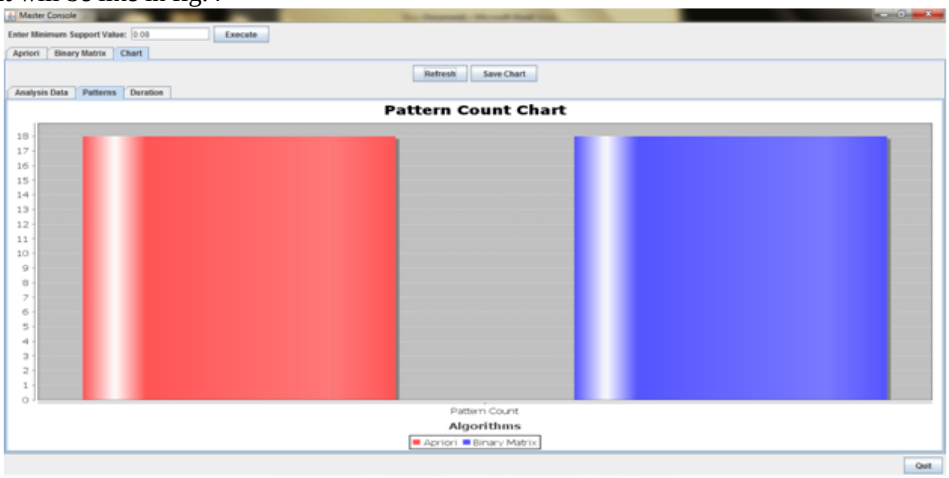

Fig.5 Pattern count chart

So the patterns count was equal for both the algorithms for same data
But the time duration will be like fig.5

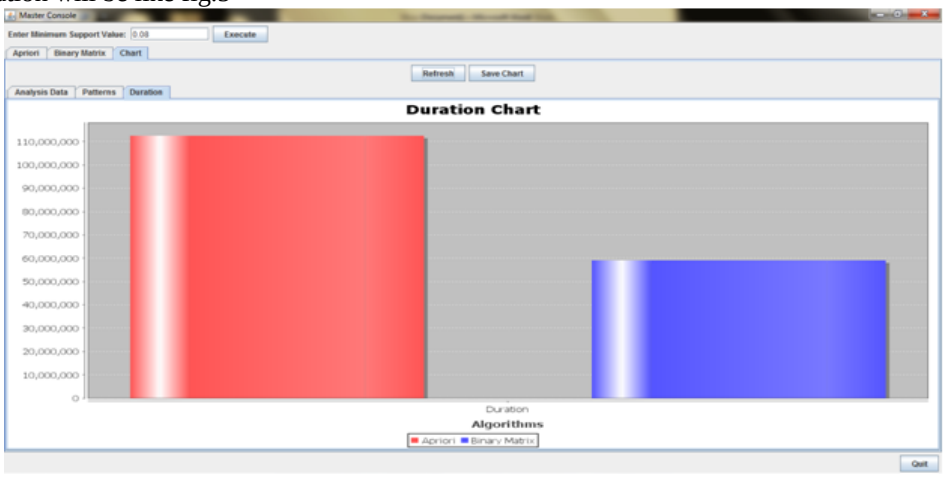

Fig.6 Time duration chart

© 2014, IJARCSSE All Rights Reserved

VI. CONCLUSION

In this paper, the new approach is proposed for identification of frequent itemsets for the given sequential patterns. Here the Binary matrix using binary data for mining frequent patterns. Then the frequent patterns are mined for both Apriori and Binary matrix algorithms. Experimental results shows the better performance of our new approach as compare to apriori. Thus concluding our research work with efficient frequent pattern mining approach with comparative analysis of traditional Apriori algorithm. This approach saves time duration for mining of frequent patterns. The experimental analysis of Binary matrix approach shows efficient results than Apriori in terms of time complexity.

REFERENCES

[1] Zhou Zhao, Da Yan, and Wilfred Ng, *"Mining Probabilistically Frequent Sequential Patterns in Large Uncertain Databases"*

[2] M. Muzammal and R. Raman, "*Mining sequential patterns from probabilistic databases*," in Proc. 15th PAKDD, Shenzhen, China, 2011.

[3] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "*Trajectory pattern mining,*" in Proc. 13th ACM SIGKDD, San Jose, CA, USA, 2007.

[4] D. Tanasa, J. A. López, and B. Trousse, "*Extracting sequential patterns for gene regulatory expressions profiles,*" in Proc. KELSI, Milan, Italy, 2004.

[5] J. Pei et al., "PrefixSpan*: Mining sequential patterns efficiently by prefix-projected pattern growth,*" in Proc. 17th ICDE, Berlin, Germany, 2001.

[6] R. Agrawal and R. Srikant, "*Mining sequential patterns*," in Proc. 11th ICDE, Taipei, Taiwan, 1995.

[7] M.J.Zaki,"*SPADE:An efficient algorithm for mining frequent sequences,*" Mach. Learn., vol. 42, no. 1–2, pp. 31–60, 2001.

[8] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M. C. Hsu, "*FreeSpan: Frequent pattern-projected sequential pattern mining*," in Proc. 6th SIGKDD, New York, NY, USA, 2000.

[9] R. Srikant and R. Agrawal, "*Mining sequential patterns: Generalizations and performance improvements,*" inProc.5thInt. Conf. EDBT, Avignon, France, 1996.

[10] Z. Zhao, D. Yan, and W. Ng, "*Mining probabilistically frequent sequential patterns in uncertain databases,*" inProc.15thInt.Conf. EDBT, New York, NY, USA, 2012.

[11] C. Gao and J. Wang, "*Direct mining of discriminative patterns for classifying uncertain data*," in Proc. 16th ACM SIGKDD, Washington, DC, USA, 2010.

[12] N. Pelekis, I. Kopanakis, E. E. Kotsifakos, E. Frentzos, and Y. Theodoridis, "*Clustering uncertain trajectories*," Knowl. Inform. Syst., vol. 28, no. 1, pp. 117–147, 2010.

[13] H. Chen, W. S. Ku, H. Wang, and M. T. Sun, "*Leveraging spatiotemporal redundancy for RFID data cleansing*," in Proc. ACMSIGMOD, Indianapolis, IN, USA, 2010.

[14] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "*Model-driven data acquisition in sensor networks*," in Proc. 13th Int. Conf. VLDB, Toronto, ON, Canada, 2004.

[15] L. Sun, R. Cheng, D. W. Cheung, and J. Cheng, "*Mining uncertain data with probabilistic guarantees,*" in Proc. 16th ACM SIGKDD, Washington, DC, USA, 2010.

[16] C. C. Aggarwal, Y. Li, J. Wang, and J. Wang, "*Frequent pattern mining with uncertain data*," in Proc. 15th ACM SIGKDD, Paris, France, 2009.

[17] Q. Zhang, F. Li, and K. Yi, "Finding *frequent items in probabilistic data,*" in Proc. ACM SIGMOD, Vancouver, BC, Canada, 2008.

[18] T. Bernecker, H. P. Kriegel, M. Renz, F. Verhein, and A. Zuefle, "*Probabilistic frequent itemset mining in uncertain databases*," in Proc. 15th ACM SIGKDD, Paris, France, 2009.

[19] C. K. Chui, B. Kao, and E. Hung "*Mining frequent itemsets from uncertain data*," in Proc. 11th PAKDD, Yichang, China, 2007.

[20] C. C. Aggarwal, and P. S. Yu, "A *survey of uncertain data algorithms and applications*," IEEE Trans. Knowl. Data Eng., vol. 21, no. 5, pp. 609–623, May 2008.

[21] J. Yang, W. Wang, P. S. Yu, and J. Han, "*Mining long sequential patterns in a noisy environment*," in Proc. ACM SIGMOD, Madison, WI, USA, 2002.

[22] P. Agrawal et al., "*Trio: A system for data, uncertainty, and lineage*," in Proc. VLDB, Seoul, Korea, 2006.

[23] X. Lian and L. Chen, "*Set similarity join on probabilistic data,*" in Proc. VLDB, Singapore, 2010.

[24] J. Jestes, F. Li, Z. Yan, and K. Yi, "*Probabilistic string similarity joins*," in Proc. ACM SIGMOD, Indianapolis, IN, USA, 2010.

[25] Y. Tong, L. Chen, and B. Ding, "*Discovering threshold-based frequent closed itemsets over probabilistic data*," in Proc. IEEE 28th ICDE, Washington, DC, USA, 2012.

[26] L. Wang, R. Cheng, D. Lee, and D. Cheung, "Accelerating *probabilistic frequent itemset mining: A model-based approach*," in Proc. 19th ACM CIKM, Toronto, ON, Canada, 2010.

[27] Z. Zou, J. Li, and H. Gao, "*Discovering frequent sub graphs over uncertain graph databases under probabilistic semantics*," inProc. 16th ACM SIGKDD, Washington, DC, USA, 2010.

[28] L. Wang, D. Cheung, R. Cheng, S. Lee, and X. Yang, "*Efficient mining of frequent itemsets on large uncertain databases*," IEEE Trans. Knowl. Data Eng., vol. 24, no. 12, pp. 2170–2183, Dec. 2012.

© 2014, IJARCSSE All Rights Reserved

[29] Y. Tong, L. Chen, Y. Cheng, and P. S. Yu, "*Mining frequent itemsets over uncertain databases*," in Proc. VLDB, Istanbul, Turkey, 2012.

[30] L. Le Cam, "*An approximation theorem for the Poisson binomial distribution*," Pacific J. Math., vol. 10, no. 4, pp. 1107–1478, 1960.

[31] A. Volkova, "*A refinement of the central limit theorem for sums of independent random indicators*," Theory Probab. Applicat.,vol. 40, no. 4, pp. 791–794, 1995.

[32] Y. Hong, "*On computing the distribution function for the sum of independent and non-identical random indicators*," Dep. Statit., Virginia Tech, Blacksburg, VA, USA, Tech. Rep. 11_2, Apr. 5, 2011.

[33] The Lahar Project [Online]. Available: http://lahar.cs.washington.edu/displayPage.php?path=./content/Download/RFIDData/rfidData.html

© 2014, IJARCSSE All Rights Reserved