Journal of Computer Science 7 (3): 394-399, 2011
ISSN 1549-3636

Mining Rare Itemset with Automated Support Thresholds

Kanimozhi Selvi Chenniangirivalsu Sadhasivam and Tamilarasi Angamuthu
Department of Computer Applications, Kongu Engineering College,
Perundurai, Erode, 638 052, India

Abstract: Problem statement: Frequent itemset mining is an important task in data mining to discover the hidden, interesting associations between items in the database based on the user-specified support and confidence thresholds. **Approach**: In order to find important associations, an appropriate support threshold has to be specified. The support threshold plays a key role in deciding the interesting itemsets. The rare itemsets may not found if a high threshold is set. Some uninteresting itemsets may appear if a low threshold is set. **Results**: This study proposes an approach to obtain the frequent itemsets involving rare items by setting the support thresholds automatically. Experimental results show that this approach produces rare and frequent itemsets in sparse and dense datasets. According to T20I6D100K, 97.76% of the FIs are generators wherein Mushrooms 1.38% of the FIs are the generators. **Conclusion**: The proposed algorithm produces both frequent and rare itemsets in an effective way. In future, computational efforts can still be reduced by implementing the algorithm as parallel algorithm.

Key words: Frequent itemset, rare itemset, minimal rare itemsets, automated support threshold, minimal Rare Itemsets(mRI), Frequent Generator (FG), Most_interesting_Group(MiG)

INTRODUCTION

Data mining is widely used in a variety of application areas such as banking, marketing and retail industry. Frequent itemset mining is a technique used in data mining to discover hidden associations that arise between various data items (Agrawal *et al*., 1993). In retail industry, the market basket analysis is intended for discovering which items tend to be purchased together in order to detain the purchase behavior of customers and to improve business. In general, the mining experts look into frequent pattern of purchase. Recently, the importance is being given for the discovery of infrequent or exceptional patterns like fraudulent credit card transactions, rare symptoms which leads to disease. In market basket analysis, some sets of items, such as milk and bread, occur frequently and can be considered as regular cases. When compared to milk and bread, some items like a gold chain and a ring are infrequently associated itemsets, but considered to be an important association. We may also uncover some rare associations that one cannot expect. The problem of finding rare or infrequent patterns has recently captured the interest of the data mining community.

The association that occurs between items is also known as an association rule. An association rule (Agrawal *et al*., 1993) is an expression of the form $X \rightarrow Y$, where X, Y are itemsets. It reveals the relationship between the itemsets X and Y. The fraction of transactions containing X also containing Y, i.e., $P(Y|X) = P(X \cup Y)/P(X)$ is called the confidence (conf) of the rule. The support (sup) of the rule is the fraction of the transactions that contain all items both in X and Y, i.e., $sup(X \rightarrow Y) = P(X \cup Y)$. To generate an interesting association rule, the support and confidence of the rule should satisfy a user-specified minimum support (minsup) and minimum confidence (minconf) respectively. In general, frequent associations are generated by high support and high confidence thresholds. The association rules with low support and high confidence also need to be generated. These associations are sometimes called as rare. Rare association rules earn special attention because they may express information of high interest to experts.

Apriori algorithm is extensively used as an association rule mining method. Although Apriori algorithm generates rare association rules when a low support threshold is set, its computational cost is high. It is unlikely to uncover all rare associations with minimum computational efforts.

As a remedy, this study proposes an approach to deal with rare item problem. The goal is to provide an algorithm for rare itemset mining with automated support thresholds.

Corresponding Author: Kanimozhi Selvi Chenniangirivalasu Sadhasivam, Department of Computer Applications, Kongu Engineering College, Perundurai, Erode, 638 052, India

The rest of the study is organized as follows: Initially, the paper revisits the problem of association rule mining and explores the need for automated support thresholds for the generation of rare association rules also then it explains the proposed algorithm Automated_Arpiori_rare. Then the paper reports the experimental results on the three datasets. Finally, the conclusions are pointed out.

MATERIALS AND METHODS

Frequent itemset mining is an interesting research area and studied widely (Agrawal *et al*., 1993; Agrawal and Srikanth, 1994; Zhou and Yau, 2007; Chandrakar *et al*., 2010; Poovammal and Ponnavaikko, 2009). Most of these studies address the issue of finding the most frequent itemsets that satisfy user-specified minimum support threshold. Algorithm like Apriori (Agrawal *et al*., 1993; Agrawal and Srikanth, 1994) uses the uniform minimum support threshold for all items. These algorithms assume that all items in the data are of the same kind and have similar occurrences in the database but this assumption is not relevant for real-life applications. In many applications, some items appear very frequently in the database, while others are not. One cannot claim that the frequent itemsets are alone interesting, but the rare items also. To identify the frequent and rare items, an appropriate minimum support has to be specified. Unless minimum support is set very low, rare association rules would never be generated.

Liu *et al*. (1999) note that some individual items can have such low support that they cannot be part of the associations generated by Apriori, even though they have very high confidence. This problem is solved by specifying each item in the database a Minimum Support (MIS) given by the user. It enables frequent items can have a higher minimum support and the rare items can have lower minimum support. Thus the MSApriori(Multiple Support Apriori) algorithm (Liu *et al*., 1999) finds some rare-itemset associations. Still the result depends on the user specified support threshold.

Significant effort has been made by (Lin and Tseng, 2006; Poovammal and Ponnavaikko, 2009) to alleviate these problems, such as adding the lift or conviction measure, which derives the minimum support dynamically from the item support, it also leaves out some interesting rules composed of more than two items because the specification is derived from frequent 2-itemsets.

Relative Support Apriori Algorithm (RSAA) (Yun *et al*., 2003) has been proposed to generate rare itemset rules without the need to specify the support threshold by the user. This algorithm assigns high support threshold for items with low frequency and low support threshold for items with high frequency. Thus RSAA generates exhaustive number of rules which includes less confidence rules.

Apriroi-Inverse algorithm (Koh and Rountree, 2005) has been proposed to find rules that may contain items over the maximum support threshold called as perfectly sporadic rules.

Another algorithm Apriori-Rare (Szathmary *et al*., 2007) has been proposed to find all minimal rare itemsets. This algorithm discovers two sets of items. One is Maximal Frequent Itemset (MFI) and the other one is minimal Rare Itemset (mRI). An itemset is a MFI if it is frequent but not all its supersets. An itemset is a mRI if it is rare but all its proper subsets are not. It also finds the generator of the Freqeunt itemsets (FIs). A Frequent Generator (FG) is an itemset which has no proper subset with the same support. These algorithms find (Exceptional) itemsets, but the problem of specifying an appropriate threshold still exists.

Though the use of automated support threshold for finding frequent itemsets involving rare itemsets have been studied in (Lin and Tseng, 2006; Selvi and Tamilarasi, 2007; 2009a; 2009b), still there are possibilities for improvement.

This study aims at making the user free from specifying any constraints including support constraint. It proposes an approach to calculate the minimum support threshold dynamically by scanning the records in the database so that all frequent, interesting and meaningful rules would be generated. It also proposes an approach to reveal the rare itemset rules with high confidence. Experiment results on three datasets show that the proposed technique is effective.

Frequent itemset mining: Consider a given transaction database $T = \{r_1, r_2, \ldots, rn\}$, where each record $ri, 1 \le i \le n$ is a set of items from a set I of items, i.e., $ri \subset I$. Let $I = \{i_1, i_2 \ldots, i_m\}$ be a set of items. Let R be a set of records in the database, where each record r is a set of items such that $r \subseteq I$. A subset of items is called an itemset.

Apriori algorithm: Apriori algorithm has been proposed in (Agrawal *et al*., 1993; Agrawal and Srikanth, 1994) for finding frequent itemsets. This algorithm is based on iterative level-wise search for frequent itemset generation. It uses a single minsup value at all levels to extract frequent itemsets. Prior to the generation of frequent itemsets, the algorithm generates all candidate itemsets in that level. A candidate k-itemset is an itemset having 'k' number of

items. A candidate k- itemset is said to be frequent if the support of the subset of candidate k-itemsets is greater than or equal to the user-specified minsup threshold. This algorithm is suitable for finding the frequent itmsets and not the rare itemsets. Unless the minsup value is fixed at a low value, the rare itemsets could not be found which leads to the explosion of frequent itemset generation.

MSApriori algorithm: An extension of Apriori algorithm called MSApriori algorithm has been proposed in the literature (Liu *et al*., 1999) which attempts to discover frequent itemsets involving rare items. This algorithm assigns a minsup value known as MIS for each item and frequent itemsets are generated if an itemset satisfies the lowest MIS value among the respective items. This method derives the MIS values for items based on their support percentage. Here the frequent items are assigned with a higher MIS value whereas rare items are assigned with a lower MIS value. So, the MSApriori algorithm addressed the rare itemset problem and improves the performance over single minsup based algorithms.

Proposed method: This proposed method adopts both Apriori and MSApriori algorithms for frequent and rare item generation. It mines frequent items belonging to three different item groups namely Most_interesting_Group(MiG), Somewhat_interesting_Group(SiG),Rare_interesting_Group(Ri). MiG and RiG uses calculated levelwise automated support thresholds like Apriori whereas RiG uses itemwise support thresholds like MSApriori. The proposed method is known as Automated_Apriori_Rare.

Finding Most_interesting_Items: The itemset which has more support(sup) than average support of items in that level are known as Most_interesting_(MiG)n that level. Let 'l' represents the level and 'n' represents the number of items in that level. Average Support of itemsets (AvgSup) can be calculated as,

$$\text{AvgSup}_l = \frac{\sum_{i=1}^{n} \text{sup}_i}{n} \quad (1)$$

The first item group MiG consists of itemsets which has support greater than AvgSup.

Finding Somewhat_interesting_Items: The itemsets which have less support than AvgSup may turn to be somewhat interesting. The portion of itemsets which has low support than AvgSup has to be filtered as Somewhat_interesting_Items. The threshold used for filtering the SiG is known as MedianSup. The MedianSup can be calculated as:

$$\text{MedianSup}_l = \frac{\min \text{sup}_l + \max \text{sup}_l}{2} \quad (2)$$

where, SiG consists of itemsets with support between MedianSup and AvgSup.

Finding rare interesting items: The rare items are items that appear less frequently in the database. Rare interesting items are the items with less support and high confidence. The itemsets which have less support than AvgSup and MedianSup should be considered to find interesting rare itemsets. The transactions which consist of rare itemsets should be separated into a group RiT and the rare itemsets are found by scanning the itemsets in RiT. The remaining transactions iT are used to find the MiGs and SiGs.

FI are frequent itemsets that satisfies the assigned levelwise support thresholds. RIs are the rare itemsets that satisfies the item support thresholds.

Algorithm Automated_Apriori_Rare:
Input : Database D
Output: L, Frequent and rare itemsets in D

Method:

```
L1=find frequent 1 itemsets(D)
Group_Items(D);
Set RiT = { }
for each transaction r∈D do begin
        for each item i∈RiG do begin
                if i∈RiG and r∈RiT then
                RiT= RiT∪r
                endif
        end
end
 iT = { }
iT=RiT – {D}
/* Findig rare itemsets */
Set sup(a) as ItemSup(a) for all a∈I in RiT
RI = Rare_Item_Gen(RiT, I)
FI=Frequent_Item_gen(iT)
Merge(RI,FI)
```

Procedure Rare_Item_Gen(L, I)

L_1 = sort(I)
for (k=2; $L_{k-1} \neq \phi$; k++)
C_k = Candidate_Gen(L_{k-1});
for each transaction t∈Tdo
C_t=subset(C_k,t);
for each candidate $c \in C_t$ do
c.count ++;
end
$L_k = \{A \in C_K / sup(A) \geq ItemSup(A[1])\}$
end
return $L_k = \cup_k L_k$;
end

Procedure Frequent_Item_Gen(L, I)
L_1 = sort(I)
for (k=2; $L_{k-1} \neq \phi$; k++)
C_k = Candidate_Gen(L_{k-1});
for each transaction t∈Tdo
C_t=subset(C_k,t);
for each candidate c∈C_t do
c.count ++
end
AvgSup=Calculate_AvgSup (L_k)
MedianSup=Calculate_MedianSup(L_k)
Group_Items(L_k)
end
return $L_k = \cup_k L_k$;
end

Procedure Candidate-gen(L_{k-1})
for each itemset l_1 Є L_{k-1}
for each itemset l_2 Є L_{k-1}
perform join operation l1 l_2
if has_infrequent_subset(c, L_{k-1})
prune c;
else
add c to C_k;
end if
end
end
return C_k;

Procedure has_infrequent_subset(c, L_{k-1})
for each (k-1) subset s of c
if s is in L_{k-1}
return false;
else
return true;
end if
end

Procedure Group_Items(D)
MiG={ };SiG={ }; RiG={ };
for each item i∈Ddo
$sup = count(I,i) / |D|$
if sup≥AvgSup then
MiG= MiG∪i
Else
If MedianSup > AvgSup then
If sup≥MedianSup and sup< AvgSup then
SiG= SiD∪i
Else if MedianSup < AvgSup then
MiG= MiG∪i
Else
RiG=RiG∪i
Endif
Endif
end
End

This algorithm generates three groups of itemsets namely Most_Frequent_Itemset, Somewhat_Frequent_Itemset and Rare_Interesting_Itemset.

RESULTS

The performance of the proposed algorithm is compared with Apriori__Rare. Since the Apriori_Rare algorithm generates frequent itemsets and rare itemsets like the proposed algorithm, this algorithm is taken for comparison. The proposed algorithm is implemented in Java. Apriori_rare is run on CORON platform (Szathmary and Napoli, 2005). The experiments are carried out on an Intel Pentium IV 2.33 GHz machine running under Fedora 10 Operating system with 2 GB RAM. Testing of the algorithms was carried out on three different datasets. T20I6D100K, C20D10K and MUSHROOMS datasets are taken from CORON platform. The characteristics of these datasets are illustrated in the Table 1 given below.

The Apriori_Rare algorithm runs each time with different user specified minsup thresholds. For each dataset some optimum support thresholds have been picked up and run. The proposed algorithm is an automated algorithm which does not require any minsup values rather it calculates itself on their own and run. The results are illustrated in Table 2.

Table 1: Dataset characteristics

Dataset	#Transactions	#Attributes	#Non empty attributes	Average # of attributes	Density
T20I6D100K	99,922	1000	893	19.9	1.99%
C20D10K	10,000	385	192	20.0	5.19%
MUSHROOMS	8416	128	119	23.0	17.97%

Table 2: Comparison of #FIs, #FGS, $\frac{FGs}{FIs}$

	Apriori_Rare				Automated_Apriori_Rare (Proposed)		
Dataset	MinSup (%)	#FIs	#FGs	# $\frac{FGs}{FIs}$	#FIs (MiG+SiG)	#FGs	$\frac{FGs}{FIs}$
T20I6D100K	10.00	7	7	100.00%	143,544	140,322	97.76%
	0.75	4,710	4,710	100.00%			
	0.50	26,836	26,305	98.02%			
	0.25	155,163	149,447	96.32%			
C20D10K	30.00	5,319	967	18.18%	86,848	28851	33.22%
	20.00	20,239	2,671	13.20%			
	10.00	89,883	9,331	10.38%			
	5.00	352,611	23,051	6.54%			
	2.00	1,741,883	57,659	3.31%			
MUSHROOMS	40.00	505	153	30.30%	643,358	8898	1.38%
	30.00	2,587	544	21.03%			
	15.00	99,079	3,084	3.11%			
	10.00	600,817	7,585	1.26%			

Table 3: Comparison of mRIs vs RiG

Dataset	Apriori_Rare		Automated_ Apriori_Rare
	MinSup (%)	#mRIs	#RiGs
T20I6D100K	10.00	907	445,634
	0.75	211,578	
	0.50	268,915	
	0.25	537,765	
C20D10K	30.00	230	8,455
	20.00	400	
	10.00	901	
	5.00	2002	
	2.00	7,735	
MUSHROOMS	40.00	254	5922
	30.00	409	
	15.00	1846	
	10.00	3077	

DISCUSSION

According to T20I6D100K most of the FIs are generators wherein Mushrooms Dataset few of the FIs are only generators. The FIs generated by the Automated_Apriori_Rare is obtained by Apriori_Rare when the support threshold is slightly less than 0.25% in the case of T20I6D100K. For C20D10K dataset the number of FIs generated by the proposed algorithm is achieved by Apriori_Rare only when the minsup value is less than 10%. In case of Mushrooms dataset the proposed algorithm matches with Apriori_Rare when the minsup value is set greaten than 10% level.

The Table 3 shows the number of mRIs generated by Apriori_Rare and also the number of items in the RiG of the proposed algorithm. Since mRIs are RiGs represent the same kind of itemsets, these can be compared. The RiGs generated by the Automated_Apriori_Rare is obtained by Apriori_Rare algorithm when the support threshold is slightly less than 0.25% in the case of T20I6D100K. For C20D10K dataset, the proposed algorithm attains its level when the minsup value is set less than 2% for Apriori_Rare. In case of Mushrooms dataset, the proposed algorithm matches with Apriori_Rare when the minsup value is set greater than 10% level. This shows that the proposed algorithm produces the frequent itemsets involving rare items in an effective way.

CONCLUSION

This study presented an approach for finding both frequent and rare itemset mining based on the Apriori framework. It uses both levelwise and itemwise support thresholds for mining. These thresholds are automatically calculated and used by the algorithm. Experimental results show that the algorithm produces the most frequent items and the rare interesting items with the use of automated support thresholds. Since the algorithm operates by splitting the database into two, it can be implemented as parallel algorithm in future.

REFERENCES

Agrawal, R. and R. Srikanth, 1994. Fast algorithms for mining association rules in large database. Proceedings of the 20th International Conference on Very Large Data Bases, (VLDB'94), Morgan Kaufmann Publishers Inc. San Francisco, CA, USA., pp: 487-499.

Agrawal, R., T. Imielinski and A. Swami, 1993. Mining association rules between sets of items in large databases. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, (SIGMOD'93), ACM New York, NY, USA., pp: 207-216. DOI: 10.1145/170035.170072

Chandrakar, I., Y. UshaRani, M. Manasa and K. Renuka, 2010. Hybrid algorithm for privacy preserving association rule mining. J. Comput. Sci., 6: 1494-1498. DOI: 10.3844/jcssp.2010.1494.1498

Koh, Y. and N. Rountree, 2005. Finding sporadic rules using apriori-inverse. Adv. Know. Discovery Data Min., 3518: 153-168. 153-168, DOI: 10.1007/11430919_13

Lin, W.Y. and M.C. Tseng, 2006. Automated support specification for efficient mining of interesting association rules. J. Inform. Sci., 32: 238-250. DOI: 10.1177/0165551506064364

Liu, B., W. Hsu and Y. Ma, 1999. Mining association rules with multiple minimum supports. Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (KDD'99), ACM New York, NY, USA., pp: 337-341. DOI: 10.1145/312129.312274

Poovammal, E. and M. Ponnavaikko, 2009. Utility independent privacy preserving data mining on vertically partitioned data. J. Comput. Sci., 5: 666-673. DOI: 10.3844/jcssp.2009.666.673

Selvi, C.S.K. and A. Tamilarasi, 2007. Association rule mining with dynamic adaptive support thresholds for associative classification. Proceeding of the International Conference on Computational Intelligence and Multimedia Applications, (ICCIMA'07), IEEE Computer Society Washington, DC, USA., pp: 76-80. DOI: 10.1109/ICCIMA.2007.90

Selvi, C.S.K. and A. Tamilarasi, 2009a. An automated association rule mining technique with cumulative support thresholds. Int. J. Open Problems Comput. Sci. Math., 2: 427-438. http://www.kurims.kyoto-u.ac.jp/EMIS/journals/IJOPCM/Vol/09/IJOPCM%28vol.2.3.7.S.9%29.pdf

Selvi, C.S.K. and A. Tamilarasi, 2009b. Mining association rules with dynamic and collective support thresholds. Int. J. Eng. Technol., 1: 236-240. http://www.ijetch.org/papers/044.pdf

Szathmary, L. and A. Napoli, 2005. CORON: A Framework for levelwise itemset mining algorithms. Proceedings of the 3rd International Conference on Formal Concept Analysis (ICFCA '05), Lens, France, pp: 110-113. http://hal.inria.fr/inria-00001201/en/

Szathmary, L., A. Napoli and P. Valtcev, 2007. Towards rare itemset mining. Proceeding of the 19th IEEE International Conference on Tools with Artificial Intelligence, (ICTAI'07), IEEE Computer Society, Washington, DC, USA., pp: 305-312. DOI: 10.1109/ICTAI.2007.180

Yun, H., D. Ha, B. Hwang and K.H. Ryu, 2003. Mining association rules on significant rare data using relative support. J. Syst. Software, 67: 181-191. DOI: 10.1016/S0164-1212(02)00128-0

Zhou, L. and S. Yau, 2007. Association rule and quantitative association rule mining among infrequent items. Proceedings of the 8th international workshop on Multimedia data mining, (MDM'07), ACM New York, NY, USA. DOI: 10.1145/1341920.1341929