ISSN 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2012

Determining the Existence of Quantitative Association Rule Hiding in Privacy Preserving Data Mining

Dr.Sugumar Rajendran [1], Dr.Rengarajan Alwar[2], Dr.Saravanakumar Selvaraj [3]

[1] Department of CSE, Veltech Mutitech SRS Engineering College, Chennai, India
[2] Department of IT, Veltech Mutitech SRS Engineering College, Chennai, India
[3]Department of IT, Panimalar Institute of Technology, Chennai, India

Abstract: Determining the association rules is a core topic of privacy preserving data mining. This paper aims at giving an overview to some of the previous researches done in this topic, evaluating the current status of the field, and envisioning possible future trends in this area. The concept behind association rules are presented at the beginning. Comparison of different algorithms is provided as part of the evaluation.

Key words: Association rules, Apriori algorithm, Itemsets, data mining.

I. INTRODUCTION

Determining the association rules is at the heart of data mining. It detects hidden linkages of otherwise seemingly unrelated data. These linkages are rules. Those that exceed a certain threshold are deemed interesting. Interesting rules allow actions to be taken based upon data pattern. They can also help making and justifying decisions.

One of the most cited illustrations for mining association rules is the Market-Basket Problem. As [3] describes, the market-basket problem is about finding out what items people buy together without knowing the person so that marketers can position items accordingly in the store to generate higher volumes of sales and making other kinds of sales decisions. Some of the rules discovered maybe trivial, e.g. people who buy bread tend to buy butter. It is the extraordinary rules that are interesting, e.g. people who buy diapers also buy beers. It is the ability to discover the interesting rules that makes association rules discovery valuable and contributes to knowledge discovery.

The problem of discovering association rules can be generalized into two steps: (1) Fining large itemsets & (2) Generating rules from these itemsets. Previous researches are mostly along these lines and have grown into various dimensions. This paper aims at reviewing some of the past works in the field, evaluating the current status, and envisioning future trends. Most of the technical details of individual research are intentionally left out with the expectation that interested readers will read the original papers.

The next section first discusses the background theories behind discovering association rules. Then, it talks about researches in finding large itemsets and the comparison of different algorithms. Lastly, it talks about researches in generating rules. Before each subsection ends, a little discussion about the current status and future trends are presented. If available, applications of the ideas will also be mentioned. The last section is the conclusion.

II. ASSOCIATION RULES

Before discussing research in specific areas of mining association rules, it is worth reviewing the theories behind association rules, the different types of rules, and their generation. Association rules are defined as statements of the form $\{X_1,X_2,\ldots,X_n\} -> Y$ [3], which means that Y may present in the transaction if $X_1,X_2,\ldots,X_n$ are all in the transaction. Notice the use of may to imply that the rule is only probable, not identical. Note also, that there can be a set of items, not just a single item. The probability of finding Y in a transaction with all $X_1,X_2,\ldots,X_n$ is called confidence. The threshold (percentage) that a rule holds in all transactions is called support. The level of confidence that a rule must exceed is called interestingness [2].

There are different types of association rules. The simplest form is the type that only shows valid or invalid association. This Boolean nature of the rule dubs the name Boolean Association Rules. In our market-basket example, "People who buy skim milk also buy low fat oil" is a Boolean association rule. Rules that aggregate several association rules

together are called Multilevel or Generalized Association Rules [2]. These rules usually involve a hierarchy and mining is done at a higher concept level. For example, "People who buy milk also buy bread". In this example, milk and bread each contains a hierarchy of different types and brands, but mining at the lowest level may not produce very interesting rules.

A more complicated type of rules is the Quantitative Association Rules. This type of rules mines over quantitative (e.g. price) or categorical (e.g. gender) attributes, and is denoted in [1] by {<attribute:value>, <attribute:value>,…,<attribute:value>} -> <attribute:value>. For example, "People whose age is between 30 and 35 with income more than 75000 per year buy cars over 20000".

However, the above types do not address the fact that transactions are temporal in nature. For example, mining before a product is introduced to or after a product is discontinued from the market will both adversely affect the support threshold. In view of this, [5] introduced the concept of an attribute's lifetime into the mining algorithm of Temporal Association Rules.

In spite of the various kinds of rules, the algorithm to discover association rules can generally be broken down into two steps:

(1) Find all large (frequent) itemsets - A large itemset is a set of items that exceeds the minimum support.
(2) Generate rules from the large itemsets

Since its introduction in [4], the Apriori algorithm has been the most mentioned algorithm for step 1. Many improvement [7, 13], e.g. speed up and scale up, of step 1 are about improving the Apriori algorithm by addressing its fallacy of generating too many candidate itemsets. There are also algorithms that are not based on Apriori [9,11,12] but aim at addressing the issues of speed of Apriori. Step 2 is mostly characterized by confidence and interestingness. There are researches about different way of generating rules [8] and alternative measure to interestingness [6, 10]. There are also researches about generating different types of rules [1, 5].

A. Finding Large Itemsets

Most of the earlier researches in mining association rules were actually done in this topic. That includes a milestone research of the Apriori algorithm. Various researches were done to improve the performance and scalability of Apriori included using parallel computing. There were also studies to improve the speed of finding large itemsets with hash table, map, and tree data structures.

Two of Apriori's predecessors are AIS [14] and SETM [15]. AIS and SETM generate candidate itemsets on-the-fly during the pass of database scan. Large itemsets from previous pass are checked if they are present in the current transaction. Hence, new itemsets are formed by extending existing itemsets. These algorithms turn out to be ineffective because they generate and count too many candidate itemsets that turn out to be small (infrequent) [4].

To remedy the problem, Apriori, AprioriTid, and AprioriHybrid were proposed in [4]. Apriori and AprioriTid generate itemsets by using only the large itemsets found in the previous pass, without considering the transactions. AprioriTid improves Apriori by only using the database at the first pass. Counting in subsequent passes is done using encodings created in the first pass, which is much smaller then the database. This leads to a dramatic performance improvement of three times faster than AIS and four times faster than SETM in one of their experiments in [4]. A further improvement, called AprioriHybrid, can be achieved when Apriori is used in the initial passes and switches to AprioriTid in the later passes if the candidate k-itemset is expected to fit into the main memory.

The problem with Apriori is that it generates too many 2-itemsets that are not frequent. [12] proposed a direct hashing and pruning (DHP) algorithm that reduced the size of candidate set by filtering any k-itemset out of the hash table if the hash entry does not have minimum support. This powerful filtering capability allows DHP to complete execution when Apriori is still at its second pass, according to a comparison experiment done by [12].

```
L1={large 1-itemsets};
FOR (k=2; Lk-1 != 0; i++ ) DO BEGIN
  Ck=apriori-gen(Lk-1);
  FORALL transactions t in D DO BEGIN
    Ct=subset(Ck,t);
    FORALL candidates c in Ct DO
      c.count++;
  END
  Lk={c in Ck | c.count >= minsup}
END
```

ISSN 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2012

Answer = Sum L_k;

FUNC apriori-gen(set L_{k-1}) BEGIN
  INSERT INTO C_k
  SELECT p.item$_1$, p.item$_2$,…,p.item$_{k-1}$,q.item$_{k-1}$
  FROM L_{k-1} p, L_{k-1} q
  WHERE p.item$_1$=q.item$_1$,…,p.item$_{k-2}$=q.item$_{k-2}$, p.item $_{k-1}$<q.item $_{k-1}$;

  FORALL itemset c in C_k DO
    FORALL (k-1)-subsets s of c DO
      IF (s not in L_{k-1}) THEN
        DELETE c from C_k;
END

Figure 1: Algorithm Apriori

L_1={large 1-itemsets};
C_1'=database D;
FOR (k=2; L_{k-1} != 0; i++) DO BEGIN
  C_k=apriori-gen(L_{k-1});
  C_k'=0;
  FORALL entries t in C_{k-1}' DO BEGIN
    C_t={c in C_k | (c-c[k]) in t.set-of-itemsets ^ (c-c[k-1]) in t.set-of-itemsets};
    FORALL candidates c in C_t DO
      c.count++;
    IF (C_t != 0) THEN C_k' +=<t.TID,C_t>;
  END
  L_k={c in C_k | c.count >= minsup}
END
Answer = Sum L_k;

Figure 2: Algorithm AprioriTid

Some further efforts to improve Apriori algorithm utilize parallel algorithm. [7] proposed 3 parallel algorithms based on Apriori to speed up mining of frequent itemsets. The Count Distribution (CD) algorithm minimizes communication at the expense of carrying out duplicate computations. The Data Distribution (DD) algorithm uses the main memory of the system to broadcast local data to all other nodes in the system. The Candidate Distribution algorithm is a load balancing algorithm that reduces synchronization between the processors and segments the database based upon different transaction patterns. These parallel algorithms were tested among each other and CD had the best performance against the Apriori algorithm. Its overhead is less than 7.5% when compared with Apriori by [7].

Scalability is another important research area for data mining because databases are getting bigger everyday. Hence, algorithms must be able to "scale up" to handle large number of data. With the work of [7] as the foundation, [13] tried to make DD and CD scalable by the Intelligent Data Distribution (IDD) algorithm and Hybrid Distribution (HD) algorithm respectively. IDD addresses the issues of communication overhead and redundant computation in [7] by using aggregate memory to partition candidates and move data efficiently. HD improves over IDD by dynamically partitions the candidate set to maintain good load balance. Experiment results show that the response time of IDD is 4.4 times less than DD on a 32-processors system and HD is 9.5% better than CD on 128 processors [13].

Another scalability study of data mining was done in [11] by introducing a light-weight data structure called Segment Support Map (SSM) that reduces the number of candidate itemsets needed for counting. SSM contains the support count for the 1-itemset. The individual support counts are added together as the upper bound for k-itemsets. Applying this to Apriori, the effort to generate 1-itemset is saved by simply inspecting those SSM support counts that exceed the support threshold. Furthermore, those 1-itemsets that do not meet the threshold will be discarded to reduce the number of higher level itemsets to be counted.

Another study to improve Apriori by using a novel data structure is the frequent pattern tree, or FP-tree. It stores information about the frequent patterns. A FP-tree-based method, called FP-growth, is also proposed to mine frequent patterns and does not involve candidate generation. Not only did [9] show that FP-growth performs an order of magnitude better than Apriori, it also showed that it is more scalable.

From iterative methods like Apriori and DHP to innovative use of data structures like SSM and FP-tree, research in mining large itemsets has made it more scalable and efficient. The need for faster and more scalable algorithms will continue because databases are getting bigger and bigger everyday. Research in distributed algorithms for finding large itemsets will gain more and more attention as more databases are integrated together. This presents a new level of challenge that demands more flexible algorithms in view of different representation of similar data, e.g. zip code maybe represented by string or integer, so that data do not need to be normalized before mining. More researches in parallel algorithms are also expected as grid computing is gaining interest.

Compare to the numerous works done to search for better algorithms to mine large itemsets, the qualifying criterion – support threshold – and the mechanism behind it – counting – have received much less attention. The problem with support threshold is that it requires expert knowledge, which is subjective at best, to set this parameter in the system. Setting it arbitrarily low will qualify itemsets that should be left out, vice versa. Moreover, as database size increases, the support threshold may need to be adjusted. The next section will highlight a few researches that address some of these issues.

B. Generating Association Rules

Research in rule generations mainly focus on newer algorithms, deriving more types of association rules, and interestingness of the rules. Newer algorithms mostly employ new strategy like parallel computing [7] and Evolutionary Algorithm (EA) [8]. Newer rules types add dimension and quality to the traditional Boolean rule type. Examples are quantitative association rules [1] and temporal association rules [5]. Newer criteria [6, 10] on interestingness tend to be more objective than support and confidence.

Most of the association rules are generated by counting the number of occurrence of the rule in the database – the confidence. Therefore, it is intuitive to partition the set of all frequent itemsets and count in parallel. Together with the 3 parallel algorithms to mine frequent itemsets, [7] presented a parallel implementation for rule generation using the previously-stated approach.

In recent years, EA has been widely adopted in many scientific areas. EA borrows mechanisms of biological evolution and applies them in problem-solving. It is especially suitable for searching and optimization problems. Hence, the problem of mining association rules is a natural fit. [8] used EA to generate association rules. It takes a population frequent itemsets as the initial population. Using EA that includes crossover and mutation of these itemsets, the population will evolve into one that contains itemsets with better and better fitness function. When the desired number of frequent itemsets is left in the population, the algorithm will stop. This novel way of generating/searching association rules allows for overlapping intervals in different itemsets. For example, one frequent itemset can have interval from [10, 20] and another one can be [15, 25]. This is a sharp contrast to other techniques that divided the attributes into non-overlapping intervals. Rules that fall across two intervals may not be possible for discovery, hence a loss of information. This algorithm allows new rules to be discovered.

Research is also done in the types of association rules. At the beginning of data mining research, Boolean association rules dominated the field. Later on, more focus was put on Quantitative Association Rules. Quantitative association rules are rules over quantitative and categorical attributes like age and marital status. Mining these rules involve partitioning the values of the attribute, but may loss information as a result of the division. [1] introduced an algorithm based on Apriori to mine quantitative rules. It also introduced a partial completeness to measure information loss due to partitioning, and "greater than expected" interest as interestingness measure. Partial completeness is directly proportional to information loss. Given the minimum support and partial completeness by the user, the system can figure out the number of partitions needed. A quantitative rule is interesting only if it has "greater than expected" support and/or confidence specified by the user. The algorithm scales up linearly with experimental data [1].

The time dimension is one that exists in all transaction. Therefore, it should be included in finding large itemsets, especially when not all items exist throughout the entire data gathering period. [5] introduced the temporal concept by limiting the search for frequent itemsets to the lifetime of the itemset members. It also introduced the concept of temporal support, in addition to the normal support and confidence. The lifetime of an item is defined by the first and last time an item appears in the database. The temporal support is the minimum interval width. Thus, a rule is considered as long as there is enough support or temporal support. A byproduct of this approach is that old, obsolete itemsets can be deleted.

Any associations discovered by the above algorithms are eligible to be rules. The quality of those rules is measured by confidence. However, only those rules with confidence above a certain level are interesting and deserve attention. Most algorithms define interestingness in terms of user-supply thresholds for support and confidence. The problem is that these algorithms rely on the users to give suitable values. A new algorithm, called APACS2, is proposed in [6] that does not require such guesswork, but makes use of an objective interestingness

measure called adjusted difference. Moreover, APACS2 can discover both positive and negative association rules. [10] presented a new concept of relatedness as an alternative approach to determine interestingness.

APACS2 uses adjusted difference as an objective interestingness measure. Adjusted difference is defined in terms of standardized difference and maximum likelihood estimate. More details can be found in [16] regarding the statistical theories. If the magnitude of the adjusted difference is greater than 1.96, i.e. 95 percentiles of the normal distribution, the association is regarded as significantly different and hence, interesting. If the adjusted difference is positive, it means the rule is likely, vice versa. The directional nature of adjusted difference gives association rules discovery a new dimension.

Interestingness can be subjective – using support and confidence – or objective – using adjusted difference like [6]. An opposite concept – relatedness – was introduced by [10] to examine relationship between two items based on their co-occurrence frequency and context. Relatedness is meant to be used in lieu with interestingness to quantify association rules. Relatedness has three components: (1) average predictive ability of the presence of one item given the presence of the other; (2) the intensity of the occurrence of an item-pair with respect to other items; (3) the substitutability of another item for the items in the item-pair. These three measures give the strength of the relationship in terms of the frequency of the rule in relation to other items.

The study of rule generation started from a single Boolean type using subjective measures of support and confidence. It has grown to include various rule types with objective measures like adjusted difference and relatedness. The new researches have added both quality and quantity to rule generation. The quality is improved by using more objective measures to qualify rules. The quantity is increased by novel methodologies that enable mining rules in overlapping intervals and negative associations.

It is surprising to see that this topic is not researched more. The quality of the rules a system determined interesting is equally, if not more, important than the speed and scale to find these rules because the real goal of data mining is to mine interesting rules. Only interesting rules are useful to help decision making. Uninteresting or trivial rules, albeit from larger database quicker, do not. Therefore, more effort should continue to pour in to investigate more objective measures so that data mining can be parameter-free and operational, thus less subjective with higher quality. This way, domain experts can focus on interpreting the rules as oppose to worrying about how to tune the mining parameters to produce meaningful rules.

As more and more new data types are created, multimedia data for instance, more researches can also be done to define more association rule types. This may allow newer behavioral pattern to be analyzed and outcome predicted. The ability to profile multimedia data in its raw data format for data mining is particularly useful in medicine [17] or even homeland security.

III. CONCLUSION

The topic of discovering association rules has been studied over a decade. Most of the foundation researches have been done. A lot of attention was focus on the performance and scalability of the algorithms, but not enough attention was given to the quality (interestingness) of the rules generated. In the coming decades, the trend will be to turn the attention to the application of these researches in various areas of our lives, e.g. genetic research, medicine, homeland security, etc. As databases are integrated and the data themselves are getting bigger and bigger, algorithmic research about how to scan faster and more will receive less attention. Rather, distributed algorithms that allow sharing of workload in a grid computing environment should gain more awareness.

With more and more data is created outside of traditional database, data mining and rule discovery will grow out of scanning database tables into accessing data in its raw format, e.g. video clips. Performance and scalability issues will become more prominent. Newer rule types maybe necessary to facilitate new data analysis. More objective measures of interestingness may also be required to avoid manipulation of rule discovery criteria by domain experts to produce desired results. Discovery association rules will continue to thrive as a research topic. Base on the research and development in the past decade, its form and focus areas are expected to be dramatically different in the next decade.

REFERENCES

[1] Ramakrishnan Srikant, Rakesh Agrawal: Mining quantitative association rules in large relational tables.ACM SIGMOD Record, Proceedings of the ACM SIGMOD international conference on Management of data SIGMOD, Volume 25 Issue 2, 2006, pp.512-527.

[2] Ming-Syan Chen, Jiawei Han, Philip S. Yu: Data Mining: An Overview from a Database Perspective. IEEE Trans. On Knowledge And Data Engineering. 1996

[3] Data Mining Lecture Notes http://www-db.stanford.edu/~ullman/mining/mining.html

[4] Rakesh Agrawal, Ramakrishnan Srikant: Fast Algorithms for Mining Association Rules (1994)

[5] Juan M. Ale, Gustavo H. Rossi: An approach to discovering temporal association rules. Proceedings of the 2000 ACM symposium on Applied computing – Volume 1. March 2000

[6] Keith C. C. Chan, Wai-Ho Au: An effective algorithm for mining interesting quantitative association rules. Proceedings of the 2007 ACM symposium on Applied computing. April 2007

[7] M. J. Zaki, M. Ogihara, S. Parthasarathy, W. Li: Parallel data mining for association rules on shared-memory multi-processors Proceedings of the 1996 ACM/IEEE conference on Supercomputing (CDROM) Supercomputing '96. November 1996

[8] J. Mata, J. L. Alvarez, J. C. Riquelme: Evolutionary computing and optimization: An volutionary algorithm to discover numeric association rules Proceedings of the 2002 ACM symposium on Applied computing. March 2002

[9] Jiawei Han, Jian Pei, Yiwen Yin: Mining frequent patterns without candidate generation. ACM SIGMOD Record , Proceedings of the 2000 ACM SIGMOD international conference on Management of data SIGMOD '00, Volume 29 Issue 2. May 2000

[10] Rajesh Natarajan, B. Shekar: Data mining (DM): poster papers: A relatedness-based data-driven approach to determination of interestingness of association rules. Proceedings of the 2005 ACM symposium on applied computing SAC '05. March 2005

[11] Laks V. S. Lakshmanan, Carson Kai-Sang Leung, Raymond T. Ng: The segment support map: scalable mining of frequent itemsets. ACM SIGKDD Explorations Newsletter,Volume 2 Issue 2. December 2000

[12] Jong Soo Park; Ming-Syan Chen; Yu, P.S.: Using a hash-based method with transaction trimming for mining association rules. Knowledge and Data Engineering, IEEE Transactions on Volume 9, Issue 5, Page(s):813 – 825, 1997

[13] Eui-Hong (Sam) Han, George Karypis, V. Kumar: Scalable Parallel Data Mining for Association Rules. Transaction on Knowledge and Data Engineering, 12(3): P. 728-737. 2000

[14] Rakesh Agrawal, Thomasz Imielinski, and Arun Swami: Mining association rules between sets of items in large database. In Proc. Of the ACM SIGMOD Conference on Management of Data, P. 207-216, May 1993.

[15] Maurice Houtsma and Arun Swami: Set-oriented mining of association rules. Research Report RJ 9567, IBM Almaden Research Center, October 1993.

[16] K. C. C. Chan and A. K. C. Wong: APACS: A System for the Automatic Analysis and Classification of Conceptual Patterns. In Comput. Intell. 6, P. 119-131. 2010.

[17] Simeon J. Simoff, Chabane Djeraba, Osmar R. Zaïane: MDM/KDD2002: multimedia data mining between promises and problems. ACM SIGKDD Explorations Newsletter, Volume 4 Issue 2. December 2002.