Mutual Exclusion Rule Mining in Transaction Datasets

Hardik Modi, Amit Awekar G.MODI@IITG.AC.IN, AWEKAR@IITG.AC.IN
Indian Institute of Technology Guwahati, Assam, India 781 039

Abstract

We address the problem of mining Mutual Exclusion Rules (MER), that is to bring out itemsets that do not co-occur. As an example: *a person either prefers AndroidOS or iOS, but not both.* MER has potential applications such as deciding optimum caching strategy, introducing diversity in *top-k* search results and identifying competitors. To the best of our knowledge, this is the first work to address MER problem. We formally define MER and give a support and confidence based framework. We then prove the monotonic property of support and leveraging on it present an algorithm *'M-Apriori'*. Our algorithm achieves upto 197 times speedup over naive approach to mine MER. We have mined two real life datasets to demonstrate application of MER.

1. Introduction

MER is a rule of the form $X \oplus Y$, where X and Y are disjoint itemsets. Specifically, we are interested in transactions that contain either item of X, but no item of Y and vice versa. MER by definition differs from Association, Negative Association and Dissociation Rules. No existing Association Rule (AR) mining algorithm can be used to mine MER. We could find only one work on mutually exclusive items (Tzanis & Berberidis, 2007). However, they mine mutually exclusive items, not rules.

1.1. Applications

MER can serve many data mining tasks, which in turn serve many applications. Figure 1 describes the importance of knowledge mined by MER. A concrete example is identifying competitors. MER captures an important aspect of business logic, that is; MER is a two way implication. It gives equal weights to both LHS and RHS of a rule. A rule saying $AndroidOS \oplus iOS$ provides information that Google and Apple are competitors.

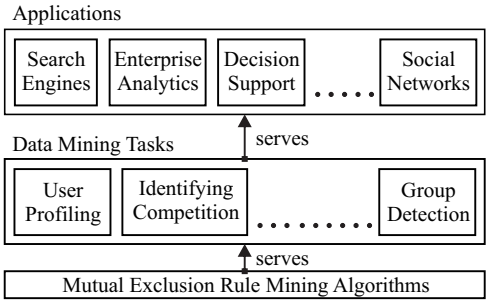

Figure 1. Applications of MER

Knowledge mined by MER can be applied to applications where there is contention and the goal is to minimize redundancy. MER can be used for solving variety of problems such as: introducing diversity in search results, web page caching, deciding balanced diet, identifying redundant entities.

2. Definitions

Definition 1: *Let $I = \{i_1, i_2, i_3..., i_n\}$ be n distinct items or binary attributes and D be a finite multiset of transactions. $D = \{T_1, T_2, ...T_m\}$, where $T_i \subseteq I$ and $1 \leq i \leq m$. We define Mutual Exclusion Rule to be of the form $X \oplus Y$; where $X, Y \subset I$ and $X \cap Y = \phi$.*

Support signifies popularity. We define it as the percentage of transactions that contain either item from the itemset. This is in contrast with support of an itemset for AR, where we count the number of transactions containing all the items of the itemset.

Definition 2: *For an item set $X = \{x_1, x_2, ...x_i\}$, where $X \subseteq I$; its support denoted as $support(X)$ is,*

$$support(X) = \frac{\sigma(X)}{\mid D \mid} \tag{1}$$

where $\sigma(X) = |\{t : (x_1 \in t \vee x_2 \in t \vee ... \vee x_i \in t)\}|$, that is count of transactions that contain either item of X

Appearing in Proceedings of the 1st Indian Workshop on Machine Learning, IIT Kanpur, India, 2013.

Given a support threshold, *minsup*, and two itemsets X, Y such that $Y \subsetneq X$.

$$support(X) < minsup \rightarrow support(Y) < minsup \quad (2)$$

We call this property as *monotonic property of support*.

Confidence depicts trust in the rule. Higher the value of confidence, means more disjoint are the itemsets with respect to transactions. All rules having confidence above specified threshold *minconf* are valid.

Definition 3: *For itemsets* $X = \{x_1, x_2, ...x_i\}$ *and* $Y = \{y_1, y_2, ...y_j\}$ *where* $X \subseteq I, Y \subseteq I$ *and* $X \cap Y = \phi$

$$conf(X \oplus Y) = 2 - \frac{\sigma(X) + \sigma(Y)}{\sigma(X \cup Y)} \quad (3)$$

3. *'M-Apriori'* Algorithm

We present a two phase algorithm *'M-Apriori'* to mine MER. Figure 2 depicts these phases.

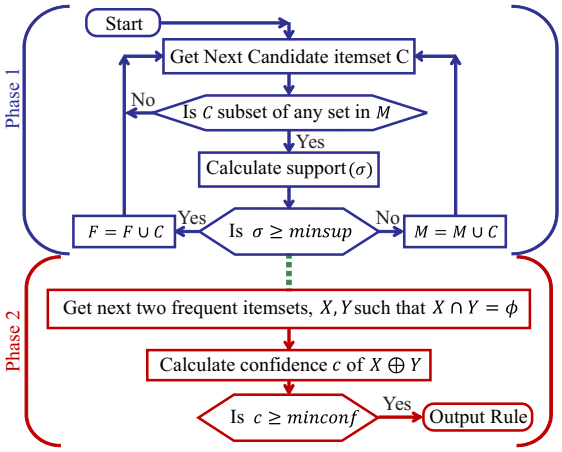

Figure 2. Flowchart for 'M-Apriori'

Phase 1 mines all itemsets that are above the specified *minsup*. It traverses the itemset lattice in BFS manner starting from largest itemset. In each level, it generates candidate itemsets from frequent itemsets of previous level. For each candidate itemset, *'M-Apriori'* first checks weather it is a subset of any known infrequent itemset. In that case, the candidate itemset is infrequent and thus it avoids a dataset scan. During Phase 2, it tries to form candidate MER from frequent itemsets, and checks them against confidence threshold $minconf$. All rules that qualify both thresholds are written to the output file. Specifically, *'M-Apriori'* achieves speed up by generating less candidate itemsets and exploiting the monotonic property to reduce dataset scans. Identifying an itemset of size n as infrequent, reduces $2^n - 2$ dataset scans. We have implemented *'M-Apriori'* using C++. Access to the code repository and datasets is made public on `https://hardikmodi@bitbucket.org/hardikmodi/`

4. Experiments and Result Discussion

4.1. Relevance of MER

To showcase relevance of MER we have mined two real life datasets: 1)Research interests of computer science faculty members in IITs and IISc and 2) Nutritional content in food items. For dataset (1) top rule with confidence 1 is as below:

{Operating Systems, Embedded Systems, Formal Verification, VLSI, Artificial Intelligence}
is mutually exclusive of
{Machine Learning, Theory of Computation, Information Security}

The above rule can be interpreted as *faculty members working in Operating Systems OR Embedded Systems OR Formal Verification OR VLSI OR Artificial Intelligence are usually not interested in Machine Learning OR Theory of Computation OR Information Security and vice versa*. It is important to note MER does not comment on the relation between items within an itemset.

4.2. SpeedUp of *'M-Apriori'*

Naive approach to mine MER is to generate all possible rules and check them against *minsup* and *minconf* thresholds. We compare the performance of *'M-Apriori'* with naive approach to mine MER in synthetically generated datasets with size ranging from thousand to ten million transactions and 10 to 14 items. *'M-Apriori'* achieves upto 197 times speedup over naive approach. Speedup increases with increase in number of transactions and number of items.

5. Conclusion and Future Work

'M-Apriori' efficiently mines MERs from real world datasets. Knowledge of mutually exclusive items is novel, not trivial and useful. Mining *top-k* MER, developing parallel algorithms, incremental mining of MER are some immediate opportunities to explore.

References

Tzanis, G. and Berberidis, C. Mining for mutually exclusive items in transaction databases. *International Journal of Data Warehousing and Mining*, 3:45–59, 2007.