Logical Identities Applied to Knowledge Discovery in Databases

James Buckley, Jennifer Seitzer, and Yongzhi Zhang

Computer Science Department
University of Dayton
300 College Park
Dayton, Ohio 45469-2160
{buckley, seitzer}@cps.udayton.edu

Yi Pan

Department of Computer Science
Georgia State University
Atlanta, GA 30303
pan@cs.gsu.edu

Abstract

Data mining is the process of extracting implicit, previously unknown, and potentially useful information from data in databases. It is widely recognized as a useful tool for decision making and knowledge discovery. Rule mining, however, is computationally expensive. Moreover, certain mathematical properties of mined rules have been given little attention. This paper applies logical identities to mined rules thereby producing additional rules that are much more efficiently acquired. We use simple properties of set theory to present a set of theorems applicable to association rules, and by using the support and confidence of mined association rules, we produce new association rules, each with its own support and confidence.

Keywords: Association Rule, Data Mining, Formal Logic, Knowledge Discovery, Negation

Introduction

Data mining is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data [Frawley 1991]. We are no longer looking for tabular answers or aggregations of the data; rather, we are looking for *patterns* within the data that reveal knowledge previously unknown. One of the most common applications of data mining is to generate all significant association rules between items in a data set. We can employ algorithms to mine a large collection of basket data type transactions for association rules between sets of items with some minimum specified confidence [Agrawal 1993] [Brin 1997] [Zaki 1997]. In this paper, we present some techniques to produce new rules more efficiently by manipulating mined association rules using logic and set theory.

The power of these rewrite rules is twofold. First, rules can be implicitly discovered by applying one or more of the identities to a set of mined rules. Some of these newly discovered rules may be difficult or impossible to determine using traditional data mining software. Secondly, the newly discovered rules can be done in O(1) time complexity. This is of particular importance for those rules that involve negation and whose time complexity would be much greater. The logical identities allow for the automatic and implicit discovery of new rules in an optimal amount of time.

Logical Identity Operations On Association Rules

In this work, we attempt to produce new rules from mined association rules by applying logical properties to them. It is interesting to examine the operations of confidence and support, which help quantify the strength of association rules. By using set theoretical operations on the involved sets of an association rule, along with confidence and support operations, we can increase the number of association rules relating to these sets.. For example, if we run a traditional data mining program, we will generate a set of association rules, T We can then apply logical identities to determine a new set of association rules, T'. It is important to note that T' can be achieved in time O(1).

Properties of association rules

In association rule mining, we use a database of sales transactions to discover relationships among items. We do this so that the presence of some items in a transaction will imply the presence of other items in the same transaction. To mine association rules, we extract a subset of items that demonstrate such a relationship. We then partition the subset into two, disjoint parts, the antecedent and consequent, generating a rule of the form $X \rightarrow Y$, where X and Y are the respective parts.

We use and augment a traditional mathematical model proposed in [Agrawal 1993] to formalize the problem of data mining association rules. We also assume the closed world assumption holds. Let $I=\{i_1, i_2,\ldots,i_n\}$be a set of all items. Let D be a set of transactions, where each transaction T is a set of items such that T is a subset of I. Let X, $Y \subset I$ be sets of items and $X \cap Y = \varnothing$. An *association rule* is an implication of the form $X \rightarrow Y$, meaning X implies Y. We say that the rule $X \rightarrow Y$ holds in the transaction set D with *confidence c* if *c*% of transactions in D that contain X also contain Y. The rule $X \rightarrow Y$ has *support s* in D if *s*% of transactions in D contain both X and Y. Thus, confidence denotes the strength of implication, and support indicates the frequencies of the occurring patterns in the rule [Ramakrishnan 1998] [Chen 1996].

In part of our work, we extract certain items from the antecedent (**X**) and consequent (**Y**) sets, and observe the relationships involving the remaining sets. We denote these extracted subsets as $\mathbf{X' \subset X}$ and $\mathbf{Y' \subset Y}$, respectively. All possible subsets of the extracted elements can be represented by the set $\mathbf{2^{X'}}$ ($\mathbf{2^{Y'}}$ respectively). Notice, all elements of $\mathbf{2^{X'}}$ ($2^{Y'}$) are also elements of $\mathbf{2^{I}}$, thus, $\mathbf{2^{X'} \subset 2^{I}}$. A transaction that is devoid of any extracted item is denoted $\mathbf{T' \subset I}$. To depict the collection of all such transactions (i.e., containing no extracted items), we introduce the new operator, 'tilde'.

Definition 1
The **tilde**, **~**, operator depicts any transaction *not* containing any element of X' or Y'. That is,

$$\mathbf{\sim T' = 2^{I} - 2^{X' \cup Y'}}$$

where $\mathbf{2^{I}}$ represents the set of all possible transactions.

Identity Properties. Let A be a set of items, then the confidence and the support of the rule $A \rightarrow A$ are given as follows:

$$\mathbf{Con(A \rightarrow A) = 1}$$

Proof. By definition of confidence and our generalized concept for rules (transaction relative), the item set A implies A is always true. The confidence of the rule $A \rightarrow A$ is 100% which is 1.

$$\mathbf{Sup(A \rightarrow A) = N(A) / N}$$

Proof. By definition of support and our generalized concept for rules (transaction relative), the support of the rule $A \rightarrow A$ is the fraction of transactions in the database that contain all the items in the item set A. This is N(A)/N.

Converse Properties. Let A and B be sets of items. The confidence and support of the rule $A \rightarrow B$ are given as follows:

$$\mathbf{Sup(B \rightarrow A) = Sup(A \rightarrow B)}$$

Proof. This formula is true since all items in item sets A and B are exactly all those items in item sets B and A.

We now present four more properties. As these properties are more intuitive and are similar to those found in standard set theory, we do not provide proofs.

Commutative Properties. The commutative properties hold because the definition of support simply combines the antecedent and consequent sets and divides by the total number of transactions. Let A, B and C be item sets. Then, we can specify the following four formulas:

$$\mathbf{Sup(A \cap B \rightarrow C) = Sup(B \cap A \rightarrow C)}$$

$$\mathbf{Sup(A \cup B \rightarrow C) = Sup(B \cup A \rightarrow C)}$$

$$\mathbf{Con(A \cap B \rightarrow C) = Con(B \cap A \rightarrow C)}$$

$$\mathbf{Con(A \cup B \rightarrow C) = Con(B \cup A \rightarrow C)}$$

Associative Properties. Here, we appeal to the associativity of union and intersection. Let A, B, C and D be item sets. Then, we can specify the following four formulas:

$$\mathbf{Sup((A \cap B) \cap C \rightarrow D) = Sup(A \cap (B \cap C) \rightarrow D)}$$

$$\mathbf{Sup((A \cup B) \cup C \rightarrow D) = Sup(A \cup (B \cup C) \rightarrow D)}$$

$$\mathbf{Con((A \cap B) \cap C \rightarrow D) = Con(A \cap (B \cap C) \rightarrow D)}$$

$$\mathbf{Con((A \cup B) \cup C \rightarrow D) = Con(A \cup (B \cup C) \rightarrow D)}$$

Distributive Properties. Let A, B, C and D be item sets. Then, we can specify the following four formulas:

$$\mathbf{Sup(A \cup (B \cap C) \rightarrow D) = Sup((A \cup B) \cap (A \cup C) \rightarrow D)}$$

$$\mathbf{Sup(A \cap (B \cup C) \rightarrow D) = Sup((A \cap B) \cup (A \cap C) \rightarrow D)}$$

$$\mathbf{Con(A \cup (B \cap C) \rightarrow D) = Con((A \cup B) \cap (A \cup C) \rightarrow D)}$$

$$\mathbf{Con(A \cap (B \cup C) \rightarrow D) = Con((A \cap B) \cup (A \cap C) \rightarrow D)}$$

De Morgan's Laws. We use $\neg$ to denote complementation. Recall, the universe of transaction items is the set of all possible items, I. The complement $\neg$A is simply I-A. Let A, B, and C be item sets. Then, we can specify the following four formulas:

$$\mathbf{Sup(\neg (A \cup B) \rightarrow C) = Sup(\neg A \cap \neg B \rightarrow C)}$$

$$\mathbf{Sup(\neg (A \cap B) \rightarrow C) = Sup(\neg A \cup \neg B \rightarrow C).}$$

$$\mathbf{Con(\neg (A \cup B) \rightarrow C) = Con(\neg A \cap \neg B \rightarrow C)}$$

$$\mathbf{Con(\neg (A \cap B) \rightarrow C) = Con(\neg A \cup \neg B \rightarrow C).}$$

Complement Properties. Let A, B, and C be item sets. Then, we can specify the following four formulas:

$$\mathbf{Sup(A \cap \neg A \rightarrow C) = 0}$$

$$\mathbf{Sup(A \cup \neg A \rightarrow C) = N(C) / N}$$

Proof. By the closed world assumption, it is obvious that the item set A$\cap\neg$A = $\varnothing$. Hence, the support of A$\cap\neg$A $\rightarrow$ C is zero. Again, by the closed world assumption, T = A$\cup\neg$A is every itemset, thus we have:

$$\begin{aligned} &Sup(A \cup \sim A \rightarrow C) \\ &= N(T \cap C) / N \\ &= N(C) / N. \end{aligned}$$

Contrapositive Properties. In traditional set theory, by the contrapositive property, we have

$$\mathbf{(\neg B \rightarrow \neg A) \leftrightarrow (A \rightarrow B)}$$

We will now illustrate how, in data mining, the contrapositive property holds via the mathematical relations of support and confidence.

Let A B be item sets. We can specify the following two theorems.

Theorem 1

$$\mathbf{Con(\neg B \rightarrow \neg A) = 1 - kc * (1 - Con(A \rightarrow B))}$$

where kc = N(A)/(N-N(B)), and N is the total number of transactions.

Proof. By definition, and by the fact that N($\neg$B) = N - N(B) is the total number of transactions that DO NOT contain B, we have

$$\begin{aligned} &Con(\neg B \rightarrow \neg A) \\ &= N(\neg B \cap \neg A) / N(\neg B) && \text{(by definition)} \\ &= N(\neg A \cap \neg B) / N(\neg B) && \text{(commute}\cap\text{)} \\ &= N(\neg A \cap \neg B) / (N - N(B)) && \text{(substitution)} \end{aligned}$$

Notice that, by the closed world assumption, the total number of transactions in the item set $\neg$A $\cap$ $\neg$B equals the total number of transactions in the entire dataset, minus N(A) + N(B) and less the overlapping part which is N(A $\cap$ B). So, we have:

$$\begin{aligned} &N(\neg A \cap \neg B) / (N - N(B)) \\ &= (N - (N(A) + N(B) - N(A \cap B))) / (N - N(B)) \\ &\quad \text{(substitution)} \\ &=((N - (N(B) + N(A) - N(A \cap B))) / (N - N(B)) \\ &\quad \text{(reorder)} \\ &=(N - N(B) - (N(A) - N(A \cap B))) / (N-N(B)) \\ &=(N - N(B)) / (N - N(B)) - (N(A) - N(A \cap B)) / (N - N(B)) \\ &=1 - (N(A) - N(A \cap B)) / (N - N(B)) \\ &\quad \text{(equivalence)} \\ &=1 - N(A) * (1 - N(A \cap B) / N(A)) / (N - N(B)) \\ &\quad \text{(factor N(A) out)} \\ &=1 - (N(A) / (N - N(B)) * (1 - N(A \cap B) / N(A)) \\ &=1 - kc(1 - Con(A \rightarrow B)) \\ &\quad \text{(by definition)} \end{aligned}$$

Theorem 2

$$\mathbf{Sup(\sim B \rightarrow \sim A) = 1 - ks + Sup(A \rightarrow B))}$$

where ks = (N(A)+N(B))/N.

Proof. Similarly, by definition, we have

$$\begin{aligned} &Sup(\sim B \rightarrow \sim A) \\ &= N(\sim B \cap \sim A) / N && \text{(by definition)} \\ &= N(\sim A \cap \sim B) / N && \text{(commute}\cap\text{)} \\ &= (N - (N(A) + N(B) - N(A \cap B))) / N \\ &\quad \text{(substitution)} \\ &=N / N - (N(A) + N(B)) / N + N(A \cap B) / N(A) \\ &\quad \text{(partition sum)} \\ &=1 - ks + Sup(A \rightarrow B). && \text{(by definition)} \end{aligned}$$

Principles of Inclusion and Exclusion. Let A, B, and C be item sets. We can specify the following four theorems:

Theorem 3
Con(A → (B ∪ C)) =Con(A → B) + Con(A → C) - Con(A → (B ∩ C))

Theorem 4 **Sup(A → (B ∪ C)) = Sup(A → B) + Sup(A →C) - Sup(A → (B ∩ C))**

Theorem 5 **Con((A ∪ B) →C) = k * (Con(C → A) + Con(C → B) - Con(C →(A ∩ B)))**
where k = N(C) / N(A ∪ B).

Theorem 6 **Sup((A ∪ B) →C) = Sup(C →A) + Sup(C →B) - Sup(C → (A ∩ B))**
.

An Example

Introduction

A simple transaction database is shown in Table 1. There are four item sets: A = Milk, B = Cereal, C = Spoon, Bowl, and D = Soap. The transaction number is labeled in the leftmost column and the items are labeled as column headings.

	Milk	Cereal	Spoon, Bowl	Soap
1	x	x	x	
2	x	x	x	x
3			x	
4	x		x	x
5		x		
6	x		x	x
7	x	x		x

Table 1. A simple transaction database

From Table 1, we can determine the following:
N = 7
N(Milk) = 5
N(Cereal) = 4
Con(Milk → Cereal) = .6
Sup(Milk → Cereal) = .43

Illustration of logical identities. We first examine the contrapositive properties:

Con(~Cereal → ~Milk)
= 1 – kc * (1 - Con(Milk → Cereal))
= 1 - (5 / (7 - 4)) * (1 – .6) = .3

Sup(~Cereal → ~Milk)
= 1 – ks + Sup(Milk → Cereal))
= 1 - (5 + 4) / 7 + .43 = .14

Next, we examine the principles of inclusion and exclusion. We additionally find that Con(Milk → Spoon, Bowl) = .8 and Con(Milk → (Cereal ∩ Spoon, Bowl)) = .4. Hence:

Con(Milk → (Cereal ∪ Spoon, Bowl))
= Con(Milk → Cereal) + Con(Milk → Spoon, Bowl) - Con(Milk → (Cereal ∩ Spoon, Bowl))
= .6 + .8 – .4 = 1

Sup(Milk → (Cereal ∪ Spoon, Bowl))
= Sup(Milk → Cereal) + Sup(Milk → Spoon, Bowl) - Sup(Milk → (Cereal ∩ Spoon, Bowl))
= .43 + .57 – .29 = .71

Again, in Table 1, we find that Con(Spoon, Bowl → Milk) = .8, Con(Spoon, Bowl → Cereal) = .4, Con(Spoon, Bowl → (Milk ∩ Cereal)) = .4 and k = .83. Hence:

Con((Milk ∪ Cereal) → Spoon, Bowl)
= k (Con(Spoon, Bowl → Milk) + Con(Spoon, Bowl → Cereal) - Con(Spoon, Bowl → (Milk ∩ Cereal)))
= .83 (.8 + .4 – .4) = .66

Similarly,

Sup((Milk ∪ Cereal) → Spoon, Bowl)
= .57 + .29 – .29 = .57

Lastly, we examine De Morgan's laws:

Sup(~(Milk ∪ Cereal) → Spoon, Bowl)
= Sup(~Milk ∩ ~Cereal → Spoon, Bowl) = .14

Sup(~(Milk ∩ Cereal) → Spoon, Bowl)
= Sup(~Milk ∪ ~Cereal → Spoon, Bowl) = .43

Conclusions And Future Work

Conclusion

In this work, we produced new rules from mined association rules by applying logical identities. Various properties of rules were defined and proved, thus providing the framework for a system of logical identities. A new operator, tilde, was defined that differentiates the concepts of set complementation and negation. If we run a traditional data mining program, we will generate a set of association rules, T We can then apply logical identities to determine a new set of association rules, T'. It is important to note that T' can be achieved in time O(1).

Future Work

The authors would like to expand upon the previously described work by examining the potential for generating predictive data mining rules that would be based upon the probabilistic nature of the database combined with transitivity of the identity rules. The concept of negation in association rules is a novel aspect of our work and we would like to continue to investigate the generation and usefulness of such rules. Lastly, the authors would like to examine the incorporation of fuzzy logic concepts into the logical identities to produce fuzzy association rules.

References

Agrawal, R, Imielinski, T., and Swami, A. "Mining association rules between sets of items in large databases," *ACM SIGMOD Bulletin*, May 1993, pp. 207-216.

Brin, S., Motwani, R., Ullman, J., and Tsur, S. "Dynamic Itemset Counting and Implication Rules for Market Basket Data," *ACM SIGMOD*, May 1997.

Chen, M., Han, J., and Yu, P. "Data Mining: An Overview from a Database Perspective", IEEE Transactions on Knowledge and Data Engineering, Vol 8, No. 6, December 1996.

Frawley and Piatetsky-Shapiro, editors, *Knowledge Discovery in Databases*, chapter Knowledge Discovery in Databases: An Overview, AAAI Press/The MIT Press. 1991.

Ramakrishnan, R., *Database Management Systems*, McGraw-Hill, 1998.

Zaki, M., Parthasarthy, S., Ogihara, M., and Li,W. New Algorithms for Fast Discovery of Association Rules", *3rd International Conference on Knowledge Discovery and Data Mining*, August 1997.