Finding Association Rules From Quantitative Data Using Data Booleanization

Susan P. Imberman, Ph.D.
College of Staten Island, City University of New York
imberman@postbox.csi.cuny.edu

Bernard Domanski, Ph.D.
College of Staten Island, City University of New York
domanski@postbox.csi.cuny.edu

Abstract

Finding association rules in data that is naturally binary has been well researched and documented. Finding association rules in numeric/categorical data has not been as easy. Many quantitative algorithms work directly on the numeric data limiting the complexity of the generated rules. In addition, as you create intervals from the numeric data the dimensionality of the problem increases significantly, causing execution time to blow up.

Quantitative data can be "booleanized" using simple thresholds as the basis for boolean classification. The "booleanized" data can be used in association rule algorithms to find interesting rules and patterns in this data. Once significant associations are found, we can increase the dimensionality on the selected interesting variables. We use an association rule algorithm, Boolean Analyzer, to look at rules. Finding significant rules is very dependent on how thresholds for booleanization are defined. We investigate the association rules generated by this algorithm when thresholds are defined by experts, and compare these rules to those calculated using mode, mean, median as threshold measures, as well as to rules derived by using thresholds found by k-means clustering.

Keywords: data mining, booleanization, association rules, dependency rules

1. Introduction

Finding association rules in two-valued categorical data has been well researched and documented (Agrawal, Imielinsk, and Swami, 1993). Problems occur when trying to find these types of rules in data with pure numeric (quantitative) or mixed numeric and categorical (qualitative) values. Several researchers have developed algorithms that work directly with strictly numeric data and mixed numeric and categorical data (Aumann and Lindell 1999, Fukada, T. et. al.(1996)(1996b)). Each has its own drawbacks and limitations as to the types of rules that result from this analysis.

Association rule algorithms find rules of the form $\mathbf{X} \Rightarrow \mathbf{Y}$ where **X** and **Y** are disjoint sets of items. Most notable of these algorithms are the Apriori algorithms described by Agrawal, Imielinsk, and Swami(1993). The data used in Agrawal, Imielinsk, and Swami(1993) was market basket data that is naturally binary (two-valued), that we refer to here as *boolean*. Either an item has been purchased by a customer and is in his/her market basket, indicated by a value of 1 (true), or it has not, indicated by a value of 0 (false). Rules generated by the Apriori algorithms showed direct associations between variables. Silverstein, C., Brin S., and Motwani, R. (1998), developed an algorithm that could find rules with negative dependencies. In doing so, Silverstein Brin, and Motwani broadened the definition of association rules calling these rules *dependency rules*. The data used by this algorithm was boolean as well. *Boolean Analyzer* (BA), Orchard, R. A. (1975) Domanski, B., (1996) Imberman, S. P. (1999), is an algorithm that also uses boolean data. The Boolean Analyzer not only calculates support and confidence estimates of a rule in the same vein as Agrawal, Imielinsk, and Swami(1993), but also calculates a probabilistic dependency (interestingness) measure (PIM) indicating the probability that a particular rule is significant with respect to the rule set. Rules generated by this algorithm are the same dependency rules generated by Silverstein Brin, and Motwani.

Finding rules in numeric data poses problems in efficiency. The number of rules defined by $\{x_1\ x_2\ x_3 \ldots x_n\} \in \mathbf{X} \Rightarrow \{y_1\ y_2\ y_3 \ldots y_m\} \in\ \mathbf{Y}$**,** where **X** and **Y** are each disjoint sets of boolean variables, is equal to $\mathbf{2}^{(\mathbf{m}+\mathbf{n})}$. An increase in the number of values that can be associated with any given variable increases the number of rules exponentially, thus causing execution time to increase significantly. Quantitative association rule algorithms attack this problem by placing the numeric values into discrete intervals, and then look at the associations formed. There are different methods in the literature for handling continuous variables (usually labeled as quantitative) and categorical values. Fukada, Yasuhiko, and Tokuyama (1996) (1996b) use geometric means to find numeric intervals for quantitative values. Their algorithm found association rules where the antecedent contained no more than two numeric variables and the consequent contained a single boolean variable. Aumann and Lindell (1999), used the distribution of a numerical value as the criteria for inclusion in the association rule. Their contention was that an association rule could be thought of as a population subset (the rule consequent) exhibiting some interesting behavior (the rule antecedent). They investigated two types of quantitative rules: categorical ⇒ quantitative rules and quantitative ⇒ quantitative rules.

Limitations on quantitative association rule algorithms, in general, center on the numbers of variables allowed in either the consequent or antecedent of the rules. In addition, Aumann, and Lindell (1999), Fukada, T. et. al.(1996b) do not allow for both boolean and multiple quantitative values to be present in both the consequent of the rule and/or in the antecedent. Furthermore, as the number of intervals per attribute increases linearly, since these algorithms are exponential, execution increases exponentially as well.

Srikant, R., and Agrawal, R.(1996) called the problem of finding association rules from quantitative data the "Quantitative Association Rules" problem. They pointed out that if too many intervals are defined for a variable, rules based on this variable might not hit minimum support thresholds. On the other hand, too large an interval results in confidence thresholds not being met.

Our approach is to use data booleanization to initially define two intervals on quantitative data. In most instances, quantitative data can be reduced into two values, i.e. a disk drive is busy or not busy, a patient has high blood pressure or blood pressure is low, customer spends over x amount for widgets or customer spends below that amount. By booleanizing a dataset, we can mine rules whose consequents and antecedents can be mixtures of both categorical and numeric data using existing algorithms. Once interesting rules are found, quantitative variables found in the most interesting rules can be further analyzed by using smaller intervals. By doing this, we find rules that have minimum support and then investigate their strength. We also avoid increasing dimensionality for variables that don't contribute to interesting associations. We have used this technique successfully to find dependency/association rules in computer performance data and oil futures data. Now the problem becomes, "How do we find good thresholds for booleanization?"

This paper is organized as follows: section 2 describes our booleanization techniques and a brief description of the Boolean Analyzer algorithm. Section 3 continues with an analysis of booleanization using computer performance data. Section 4 describes the statistical methods used. Section 5 gives our results using the booleanization methods on Oil Futures data. We finish in section 6 with a summary of our conclusions.

2. Data Booleanization

The generation of useful rules is highly dependent on the choice of "good" thresholds for data booleanization. It is the booleanization step, i.e. classification using a threshold, that most directly affects the statistical significance and strength of rules generated by the algorithms. In the past, we have used boolean thresholds that were set by an experts in the domain. By using an expert to create these thresholds, we implicitly input expert knowledge into the algorithm. While this is highly desirable, there are problems with using an expert. It is well documented that experts do not always agree. In addition, the domain knowledge we seek may already be implicit in the data. Experts may be able to set thresholds for the world at large, which may not accurately reflect the world of the data. Also, given a dataset with high dimensionality, determining thresholds by manual means can be time consuming. Most times, the system end user is not an expert, and often, there are domains for which there is no expert. Lastly, experts may be *fuzzy* on what the exact boolean boundary may be. In light of all these drawbacks, automated methods of booleanization can be a viable alternative, or an adjunct to booleanization by an expert.

There are many different ways of booleanizing data. The focus of this paper will be on determining thresholds using the mean of data values for each variable, the median of these values, the mode, and by partitioning the values into two groups using clustering techniques. Values above the threshold will take on a boolean value of 1 and below a value of 0. The Boolean Analyzer algorithm used in this paper, Orchard, R. A. (1975) Domanski, B., (1996), Imberman, S. P., (1999), not only generates rules but also identifies interesting rules by ranking them based on a probabilistic interestingness measure (PIM). The rules generated by automated methods will be compared to rules generated using expert defined thresholds. An analysis of the difference between rule PIM values will be done using standard statistical means.

2.1 The Boolean Analyzer Algorithm

We summarize the Boolean Analyzer (BA) algorithm here and refer the reader to a more detailed outline in Domanski, B., (1996), and Imberman, S. P., (1999). Boolean Analyzer views each row of data as an observation in a boolean activity matrix. Each 0/1 vector XY represents a single entry in this matrix .

We partition the variables into two sets X and Y where $X \cap Y$ is $\emptyset$ and $X \cup Y$ is the set of all variables in the dataset. We next calculate a State Occurrence Matrix where each entry is the support of $X \Rightarrow Y$. The support of an implication is the percent occurrence of $X \cup Y$ in the dataset, Agrawal, Imielinsk, and Swami(1993)

Using the State Occurrence matrix, in addition to support and confidence measures, we calculate a *p*robabilistic *i*nterestingness *m*easure or *PIM*. Given a dependency rule $X \Rightarrow Y$, large positive PIM values indicate a strong direct dependency relationship and large negative PIM values show that X has a strong inverse dependency relationship to Y. Values close to or equal to zero indicate that the variables are independent and not related. PIM values are used to order the rules according to their "interestingness".

3. Testing On Computer Performance Data

We chose mean, median, and mode as possible methods for finding "good" thresholds. since these methods are easily calculated, and also serve as standard methods for partitioning data. These methods were examined to see how well they performed on a dataset where the rules within the dataset were known. A dataset composed of computer performance data was analyzed, consisting of 152 records and 6 variables. This dataset was based on an actual system where performance data was collected. The data was such that values for each variable were organized as follows:

- Variable1 - ascending date values
- Variable 2 - Number Of Transactions Run, was numeric and also increased over time.
- Variable 6 - Network Busy (utilization percent) values decreased over time.
- Variable 3 - CPU Busy, Variable 4 - Response Time, and Variable 5 - Disk Busy, were numeric and random (had no obvious relationship to the passage of time).

Dependency rules between Variable 1 and the rest of the variables were looked at. (How do the variables change over time?) The known dependency rules can be expressed as follows:

Table 1 - Known Rules

$X_1' : X_2'$	As time decreases, Number of Transactions decreases
$X_1 : X_2$	As time increases, Number of Transactions increases
$X_1' : X_6$	As time decreases, Disk Performance increases
$X_1 : X_6'$	As time increases, Disk Performance decreases
$X_1' : X_2'X_6$	As time decreases, Number of Transactions decreases and Disk Performance increases
$X_1 : X_2X_6'$	As time increases, Number of Transactions increases and Disk Performance decreases

Our goal was to verify whether the thresholds determined by the mean, median, and the mode of these values, could successfully be used to find the known rules in Table 1.

First, a random sample of 20 records was withheld. The remaining data was booleanized, and then fed into the Boolean Analyzer (BA) algorithm. Thresholds specified by an expert were used to examine the rules generated by BA on the remaining 132 observations. The 20 record sample was used to see if the generated rules would predict well over "new" data. This procedure was repeated 10 times keeping the expert defined thresholds constant. In 9 out of the 10 trials, the six known rules, as organized by their PIM values, appeared in the top 8 generated rules. In one trial, the six known rules appeared in the top 9 generated rules. Since the six known rules could be found using only Variable1 (Date), as the antecedent, it was only necessary to generate the 484 rules defined by all combinations of the other five variables in the consequent. (The number of rules is 484 since for each of the 5, 4, 3, 2, and 1 combinations of variables, each variable combination can have $2^{\#\text{ of variables}}$ different 0/1 vector values. Hence, 484 = [combination(5,5) * 2^5 + combination(5,4) * 2^4 + combination(5,3) * 2^3 + combination(5,2) * 2^2 + combination(5,1) * 2^1] * 2. We multiply by 2 since Variable1 can be 0 or 1.)

These Monte Carlo methods were repeated using mean and median thresholds. Since the mean and median changes with each newly generated 132 observation sample, thresholds were recalculated for each trial. The data was booleanized and then fed into the Boolean Analyzer algorithm. This was repeated 10 times each for mean and median. In 10 out of 10 trials for the mean thresholds, the first six rules, as organized by their PIM values, found by the Boolean Analyzer algorithm were the six known rules. These rules were also found in the test sample data. For the median thresholds, again, in 10 out of 10 trials the first six rules found by the Boolean Analyzer algorithm were the six known rules and these rules were able to predict on the test sample data.

Results from these trials seemed to indicate that the automated thresholds might produce more accurate results than the expert defined thresholds. In fact, the thresholds for mean, median and the expert, calculated using the full 152-observation dataset in Table 2 shows that the major difference between the three statistics is for Variable 3, CPU busy. Based on this, one can also conclude that the choice of thresholds has a strong impact on the results obtained. Also, in the absence of an expert, mean and median look like excellent methods for choosing thresholds. One might wonder why the expert selected 80% as the threshold for *booleanization* of the CPU busy % variable - in point of fact, booleanization is not necessarily meant to classify the data into two equal categories. Rather, especially in this case, the threshold is meant to create two categories for *significant* CPU usage and *insignificant* usage. The performance expert worries about situations where excessive processing is occurring, and would naturally wonder what else (what other measurements) is correlated with excessive CPU activity.

Table 2 - Thresholds For Synthetic Data Set

	Date	Number of Transactions	CPU busy	Response Time	Disk Busy	Network Busy
mean	34319	4806	47	5	49	47
median	34319	4910	43	4	51	45
expert	34334	5000	80	4	40	50

Mode failed when applied to this data as a possible automated threshold method. Each record of Variable 1(dates) contained a unique date value. Therefore, there was no frequency to be found. The same was true for Variable 2 (number of transactions).

4. Statistical Methods

Boolean Analyzer *attaches* to each discovered rule a PIM value. The PIM imposes an order or ranking on the set of rules. Although we were able to show that the known rules were found easily by using mean, median, and expert-defined thresholds, the question remained, "*Was the order imposed on the rules found by all methods comparable*?"

Two different non-parametric statistical tests were used, the Spearman Rank-Order Correlation Coefficient and the Kendall Tau b Correlation Coefficient. The Spearman Rank-Order Correlation Coefficient uses the magnitude of the differences between one ranking to another to get a measure on the association between the two rankings. This statistical test was selected to see how well the rules' PIM values, generated by each threshold method, correlated with each other. Was each calculated PIM the same for each method? The Kendall Rank-Order Correlation Coefficient uses the number of agreements and disagreements in the ordering of the two variables' rankings to find a measure of association between the two. The way that the PIM orders the rules is significant. Looking at the differences in the numbers of agreements and disagreements between the relationship orderings tells if each method ordered these rules similarly. Queries like, "*Find the top 30 rules*" are affected by the order placed on the rules. In comparing the rules generated by BA, we only compared the subset of all rules from the partition of Variable1(Time) versus the other five variables. These rules represented an independent set of rules whose PIM values are not dependent on the measurement of other relationships in this set (Imberman, (1999)). There were 64 rules of this type. The Kendall Tau b and Spearman rank order correlation coefficients were calculated, using the ***SAS®*** CORR procedure.

Spearman correlation results for expert defined thresholds versus thresholds for mean and median showed correlations of 89% and 84% for mean and median respectively. The Kendall Tau b showed a 76% and a 70% correlation for mean and median respectively. Both correlation coefficients indicated a good correlation between the automated threshold determination and those thresholds defined by an expert. When we looked at how well mean and median correlated to each other we saw an even higher correlation. The correlation coefficient results for the Spearman were 93% correlated and for the Kendall Tau b 84% correlated.

Results from the Spearman Correlation Coefficient indicate that there was excellent correlation between the measures of the two rankings. The Kendall Tau b showed a reasonably high correlation between the way the two were ordered. Therefore, analysis of the computer performance dataset lead to the following conclusions:

- Expert, mean and median boolean thresholds can find known rules.
- Mean and median thresholds may yield more accurate results than thresholds from an expert.
- There are problems with mode as an automated method.
- Rules found using all three methods correlate very well with regard to their PIM values and hence their ordering.

5. Oil Futures Data

The results with the computer performance data are encouraging. The question that remained was whether or not the Boolean Analyzer was general enough to yield similar results in a different domain, using a different expert. In addition to using mean and median as automated methods, clustering was thought to be a method worthy of consideration. If the data does not have a continuous distribution and is possibly multimodal, then clustering algorithms might better capture this distribution. The data used was actual measurements obtained from a Wall Street oil futures firm. The data described crude oil prices, trading volume, and crude oil stockpiles. Data selection applied to this dataset resulted in a subset of the original data. Data selection was done according to criteria defined by an expert.

The dataset had 5,782 entries and 6 variables. The six variables were:

- Open Interest,
- Settlement Price (dollars paid per barrel of oil),
- Volume (Number of barrels of oil)
- PADD 1 Total Crude Inventory (represents the amount of oil stored at the New York Harbor delivery point)
- PADD 2 (the delivery point for the mid-continent)
- PADD 3 delivery point located on the gulf coast).

The oil futures dataset had a natural partition between the reported oil stock data and the crude oil price/volume data. It was not necessary to look at all the rules since only the ones generated by this one partition were considered significant by the expert.

5.1 Correlation Results For Oil Futures

Boolean thresholds elicited from the expert were applied to the target data resulting in a booleanized data set. The same was done for mean, median and thresholds determined by mode. When the mode for open interest variable was found, its mode value was zero. Since open interest shouldn't have a zero value, errors in the data were suspected. This suspicion was confirmed with the domain expert. Hence, mode is sensitive to errors in the data, making it less attractive for threshold automation than methods using mean and median. There were four records having this erroneous data. These were deleted from the dataset with the resulting dataset having 5,778 records, and all further analyses were done on this subset.

To cluster the data, each value was transformed into its *Z*-score. The transformed data was clustered using the FASTCLUS procedure in the ***SAS®*** statistical package. FASTCLUS uses a K-means algorithm to do clustering. Clusters corresponding to high values in the original target data were assigned a value of one, clusters with low values a zero. The Kendall Tau b and Spearman rank order correlation coefficients were calculated, using the ***SAS®*** CORR procedure as before. The results are in table 3.

Table 3 - Statistical Results From Oil Futures Data

	Correlation Results Between An Expert and Automated Methods		Correlation Results Between An Expert and Automated Methods With Removal of Fuzzy Variable	
Threshold Method	Spearman	Kendal Tau b	Spearman	Kendal Tau b
Mean	72 %	52 %	96 %	85 %
Median	70 %	50 %	98 %	90 %
Mode	66 %	48 %	84 %	69 %
Cluster	21 %	15 %	37 %	26 %

Looking at the Spearman correlation coefficient in column 1 of Table 3, we see a 72% and 70% correlation for mean and median respectively. Although not as strong as the results obtained from the computer performance data, the correlation to the expert's rules was still evident. The values for the Kendall Tau b were not high enough to show even good correlation. One possible explanation for these results was attributed to a poor boolean threshold set by the expert for the variable associated with PADD3 crude oil stockpiles. For the other five variables the expert was able to give concrete boolean thresholds. For PADD3, the expert was not sure where an exact boundary should be. The next step was to test to see if this fuzzy boundary affected the correlation coefficients between the expert rules and the automated ones. The variable for PADD3 was removed from the dataset and the same analysis was repeated on the remaining five variables. The results of the five variable analysis are shown in column 2 of Table 3.

Results improved tremendously with the removal of this variable. Both correlation coefficients showed high correlation between the expert and the automated methods of mean and median, a lower correlation between the expert and mode, and a poor correlation between the expert and the clustering technique. Results were encouraging, but the question remained, did things get better because the problem was reduced to five variables or was there really a significant correlation between the expert and these automated methods? The variable for PADD3 was reintroduced into the dataset. Since mean seemed to be good method of automation, the expert boolean threshold for that variable was replaced with the mean boolean threshold. The analysis was repeated. The same was done with replacement of the expert boolean threshold with the median boolean threshold. These results are shown in Table 4. The correlation coefficients indicate that the previous high coefficients were not due to the reduction in the number of variables. In fact, the substitution of a mean or median boolean boundary for a fuzzy boundary given by an expert can yield good results. Mean and Median look like good methods of automated boolean threshold determination, with mode not as strong a method, and clustering a poor one.

Table 4 - Statistical Results With Substituted Expert Threshold Value

	PADD3 booleanized with Mean calculated thresholds instead of expert defined threshold		PADD3 booleanized with Median calculated thresholds instead of expert defined threshold	
Threshold Method	Spearman	Kendal Tau b	Spearman	Kendal Tau b
Mean	98 %	91%	97 %	87 %
Median	97 %	85 %	96 %	83 %
Mode	86 %	69 %	86 %	68 %
Cluster	27 %	20 %	30 %	23 %

6. Conclusions and Futures

We found that the rules generated using boolean thresholds elicited from an expert correlate well with the rules generated using automated methods of mean and median. Rules generated from expert defined thresholds correlate less significantly with rules found by boolean thresholds determined by using mode, and poorly with those found using clustering.

Mean and median thresholds can be used in combination with expert-defined thresholds when an expert is unsure about a specific variable's threshold. Mean and median thresholds can also yield good results in the absence of an expert. Mode is not as good an automated method since it is more sensitive to errors, and it cannot capture thresholds for unique data values such as date values, and it does not correlate as well to an expert's knowledge as does mean and median.

We believe the Boolean Analyzer algorithm shows great promise as a new exploratory data analysis tool. The sensitivity of analyses to different thresholds and threshold determination procedures needs to be further examined so we can better understand how to apply BA to new measurement domains.

7. References

Agrawal, R., T. Imielinsk, and A. Swami. "Mining Association Rules between Sets of Items in Large Databases." *Proceedings of the ACM SIGMOD International Conference on the Management of Data,* 207-216, 1993.

Aumann, Y. Lindell, Y. "A Statistical Theory for Quantitative Association Rules." *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 261 -270*, 1999.

Domanski, B. "Discovering the Relationships Between Metrics." *The Proceedings of the 1996 Computer Measurement Group.* December 1996, San Diego California, 309 – 313, 1996

Fukada, T., Yasuhiko, M., Sinichi,, M., Tokuyama, T. "Data Mining Using Two-Dimensional Optimized Association rules: Scheme, Algorithms, and Visualization." *Proceedings of the ACM SIGMOD International Conference On Management of Data,* June 1996, Montreal Canada, 13-23, 1996.

Fukada, T. Yasuhiko, M. Sinichi, M. Tokuyama, T. "Mining Optimized Association Rules for Numeric Attributes,." *Proceedings of the ACM SIGMOD International Conference On Management of Data,* June 1996, Montreal Canada, 13-23, 1996b.

Imberman, S. P. "Comparative Statistical Analyses of Automated Booleanization Methods For Data Mining Programs." Ph. D. Dissertation, Doctoral Program in Computer Science, City University of New York Graduate Center, 1999.

Orchard, R. A. "On the Determination of Relationships Between Computer System State Variables." *Bell Laboratories Technical Memorandum,* January 15, 1975

Silverstein, C., Sergey Brin, and Rajeev Motwani. "Beyond Market Baskets: Generalizing Association Rules to Dependence Rules." *Data Mining and Knowledge Discovery,* 2(1):39-68, 1998

Srikant, R., Agrawal, R. "Mining Quantitative Association Rules in Large Relational Tables." *Proceedings of the ACM SIGMOD International Conference On Management of Data,* June 1996, Montreal Canada,, 1-12, 1996.