Analyzing the Interestingness of Association Rules from the Temporal Dimension

Bing Liu, Yiming Ma
School of Computing
National University of Singapore
3 Science Drive 2, Singapore 117543
{liub, maym}@comp.nus.edu.sg

Ronnie Lee
Department of Statistics and Applied Probability
National University of Singapore
3 Science Drive 2
Singapore 117543

Abstract

Rule discovery is one of the central tasks of data mining. Existing research has produced many algorithms for the purpose. These algorithms, however, often generate too many rules. In the past few years, rule interestingness techniques were proposed to help the user find interesting rules. These techniques typically employ the dataset as a whole to mine rules, and then filter and/or rank the discovered rules in various ways. In this paper, we argue that this is insufficient. These techniques are unable to answer a question that is of critical importance to the application of rules, i.e., can the rules be trusted? In practice, the users are always concerned with the question. They want to know whether the rules indeed represent some true and stable (or reliable) underlying relationships in the domain. If a rule is not stable, does it show any systematic pattern such as a trend? Before any rule can be used, these questions must be answered. This paper proposes a technique to use statistical methods to analyze rules from the temporal dimension to answer these questions. Experimental results show that the proposed technique is very effective.

6. Introduction

The objective of data mining is to find interesting/useful knowledge for the user. Rules are an important form of knowledge. Existing research has produced many algorithms for rule mining. These techniques, however, often generate too many rules, and most of the rules are of no use to the user [22, 23, 15]. In the past few years, a number of rule interestingness techniques were proposed to deal with the problem [12, 13, 14, 15, 21, 22, 23, 4, 16, 24, 25, 28]. They typically use the whole dataset to mine rules and then filter and/or rank the discovered rules in various ways to help the user identify interesting/useful ones. These techniques, however, have a major shortcoming. They are unable to answer a question that is crucial to the application of rules, i.e., can the rules be trusted or will the rules hold in the future? A rule has no use if it does not hold in the future or is not predictive.

In this paper, we argue that an important aspect of the rules has not been studied, i.e., the temporal aspect. Analyzing rules from the temporal dimension is important for real-life applications. In the past few years, we used a number of interestingness techniques to help our users find interesting rules [13, 14, 15]. Although these techniques helped a great deal, our users often still felt uncomfortable. The reason is that the users are concerned with the reliability of the rules. They want to know whether the rules will hold in the future, i.e., whether the rules indeed represent some stable (or reliable) underlying relationships in the domain. If a rule is not stable, does it show any systematic trend? The inability of the existing approaches to answer these questions lies with the fact that they do not study the dynamic behavior of rules. In this research, we analyze the interestingness of rules from the temporal dimension to solve the problem. This approach makes sense as most datasets are collected over time and represent the behaviors of the application domains in different time periods.

In this research, we focus on analyzing association rules. Association rule mining is commonly stated as follows [3]: Let $I = \{i_1, \ldots, i_n\}$ be a set of *items*, and D be a set of data tuples (records). Each data tuple consists of a subset of items in I. An *association rule* is an implication of the form $X \rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ holds in D with confidence c if $c\%$ of data tuples in D that support X also support Y. The rule has support s in D if $s\%$ of data tuples in D contains $X \cup Y$. Given a set of data tuples D (the database), the problem of mining association rules is to discover all rules that have support and confidence greater than or equal to the user-specified minimum support (called *minsup*) and minimum confidence (called *minconf*).

This paper presents a set of statistical methods to analyze the behavior of rules over time. The basic idea is as follows: The dataset is first partitioned into a few blocks or sub-datasets corresponding to the time periods (e.g., years, months or weeks) in which they were collected. The granularity of the time period is application dependent. We then mine rules from each block. Since the amount of data available is usually large, the partitioning does not lose significance of the rules discovered. After all the rules are found, we analyze the supports and confidences of the rules in these time periods, which allow us to answer the above questions and to find various types of important rules, e.g.,

- Stable rules: These rules do not change a great deal over time. Stable rules are more reliable and can be trusted. That is, they can be safely used in real-world performance tasks.
- Trend rules: These rules indicate some underlying systematic trends. If the trend rules are known, the user can take necessary actions to exploit desirable trends, and to reverse/delay undesirable trends.

Our experiments and practical applications show that the proposed technique is very effective. Testing on unseen future data confirms that the stable rules and the trend rules identified by our method are indeed reliable. Our experiments also indicate that although the number of rules generated from the data can be large, the number of stable rules and those rules that show some trends is actually quite small. It is possible to manually inspect them to identify those truly useful/actionable ones.

2. Related Work

Existing research on rule interestingness focuses on a few directions, namely, template matching, unexpected rule identification, rule summarization and organization. We discuss them below in turn.

[12] proposes a template-based approach for finding interesting rules. This approach first asks the user to specify what rules he/she wants. The system then finds those matching rules.

[15, 23, 21] propose to used user expectations or beliefs to help him/her find unexpected rules. In this approach, the user first input his/her existing knowledge about the domain, and the system then finds those conforming and unexpected rules.

[14] presents a technique to summarize the discovered association rules using a small subset of the rules. [13] presents a technique to organize the discovered rules in such a way that they can be easily browsed by the user.

[4, 25, 18, 10] report a number of methods for ranking the discovered rules using some statistical interestingness measures.

All these techniques are different from our work as none of them analyzes rules over time. Thus, they are unable to answer the rule stability or reliability question.

[2] proposes to mine and to monitor rules in different time periods. The discovered rules from different time periods are collected into a rule base. Ups and downs in support or confidence over time (called history) are represented and defined using shape operators. The user can then query the rule base by specifying some history specifications. This is related to our work. However, there is a major difference. We analyze rules generated over different time periods using statistical tests rather than simply matching user specifications. By using statistical tests, we can formally analyze the significance of ups and downs of support and confidence. User specifications lack statistical foundation. One will not know whether the changes are significant and worth investigating.

[6] presents a framework for measuring changes in two models. The difference between the two models (e.g., two sets of itemsets, one generated from dataset D_1 and one generated from dataset D_2) is quantified as the amount of work (e.g., difference in supports) required to transform one model into the other. In a related work, [5] finds the support differences of association rules mined from two datasets and uses the differences to detect emerging patterns. These are different from our work as we analyze rules over a number of time periods, while [6, 5] only compare the supports/confidences of rules from two periods. Our method uses statistical methods to identify significant changes, trends and stable rules. Both [6, 5] do not perform these tasks.

3. Basic Steps of the Proposed Approach

This section gives an overview of the proposed approach, which consists of 3 steps:

1. **Partitioning the dataset**: The original dataset D is first partitioned vertically into a number of blocks (or sub-datasets) D_1, D_2, …, D_q according to the time periods, T_1, T_2, …, T_q, in which they were collected, e.g., years or months.
2. **Mining rules from sub-datasets**: We then mine rules from each sub-dataset D_i. Let the set of rules mined from D_i be R_i. The set R of rules that will be analyzed in the third step below is defined as follows:

   $$R = \{r \mid r \in (R_1 \cup R_2 \ldots \cup R_q)\}$$

   This means that if a rule appears in any rule set R_i it is considered a potentially interesting rule. The reason that we consider all such rules will be clear later.

   Clearly, a rule r in R may appear in R_i but not in R_j ($i \neq j$) because r may not satisfy minsup and/or minconf in D_j. For our analysis later, we need the supports and confidences of every rule in R in all time periods (or all D_i). Thus, the missing support and confidence information in certain time periods for each rule needs to be obtained. This can be done quite easily. We can first mine rules from each sub-dataset D_i to produce R. We then scan the data D once to obtain all the missing supports and confidences.

   Note that in rule mining, pruning is also performed to remove those insignificant rules. We use the technique given in [14].
3. **Analyzing rules over time:** After all the necessary information about supports and confidences of each rule r in R in all time periods is obtained, we analyze

the rules to give the user different types of interesting rules. A number of rankings of rules according to different statistical measures are performed to enable the user to view those most interesting rules first.

In the next section, we discuss step 3, which is the focus of this paper. Step 1 and 2 will not be discussed further as they are fairly straightforward.

4. Analyzing Rules Over Time

We now present a number of statistical tests to analyze the discovered rules in R from different perspectives. We assume that the confidence value (denoted by $conf_i$) and the support value (denoted by sup_i) of each rule in R in each time period T_i has been obtained by a mining algorithm. Our analysis aims to identify:

- semi-stable rules
- stable rules
- rules that exhibit trends.

Note that the statistical tests presented here are by no means the only tests that can be performed on rules. In fact, many other tests may also be used for various purposes. We also point out that the kinds of datasets, and the problems, dealt with here are different from time series data. In the latter, a given population is tracked over time. For our case, although the behavior of rules over time is of interest, each time period typically corresponds to a different population.

4.1 Semi-stable rules

Intuitively, a rule r in R is a *semi-stable rule* if none of its confidences (or supports) in the time periods, T_1, T_2, …, T_q, is statistically below minconf (or minsup). If some observed confidence $conf_i$ (or support sup_i) of the rule is less than minconf (or minsup), it may be due to chance. Semi-stable rules are in contrast with *stable rules* which have more stringent conditions (see the next sub-section).

Definition (semi-stable confidence rules): Let minsup and minconf be the minimum support and the minimum confidence, $conf_D$ and sup_D be the actual support and the actual confidence of a rule r obtained from the whole dataset D, $conf_i$ be the actual confidence of the rule in the time period T_i, and α be a specified significance level. The rule r is a *semi-stable confidence rule* over the time periods, T_1, T_2, …, T_q, if the following two conditions are met:

1. $sup_D \geq$ minsup and $conf_D \geq$ minconf.
2. For each time period (or sub-dataset), we fail to reject the following null hypothesis at the significance level α:

$$H_0: conf_i \geq \text{minconf}$$

Note that the first condition is to ensure that the rule r satisfies the user specified minsup and minconf thresholds in the whole dataset. To test the hypothesis in the second condition, we can use the statistical test on a single proportion [26, 17].

Test statistic

Consider testing the following null hypothesis (H_0) against the alternative hypothesis (H_1):

$$H_0: p \geq p_0$$
$$H_1: p < p_0$$

where p_0 is the given proportion. The *test statistic z* (with standard normal distribution) for the sample proportion, $\hat{p}$, is:

$$z = \frac{\hat{p} - p_0}{\sigma_{\hat{p}}} \qquad \sigma_{\hat{p}} = \sqrt{\frac{p_0(1-p_0)}{n}}$$

where n is the sample size. If z is less than its critical value at the significance level α, we reject the null hypothesis. In our case, p_0 is minconf and $\hat{p}$ is $conf_i$.

Example 1: Assume in a mining session we set minconf to 50%. There is a time period T_i in which the confidence $conf_i$ of a rule, $A \rightarrow B$, is 45%. There are 300 data tuples that satisfy the condition A, and out of these 300 tuples 135 tuples also satisfy the consequent B (i.e., $conf_i$ =135/300 = 45%). The question is whether $conf_i$ is statistically below 50%.

We solve the problem as follows: From the problem description, we obtain the population size n = 300, $\hat{p}$ = 45% and p_0 = 50%. We use α = 5%. The null and alternative hypotheses are:

$$H_0: p \geq 0.50$$
$$H_1: p < 0.50$$

The critical value for z is -1.65 at α = 5%, which can be obtained from statistical tables. Let us compute z,

$$\sigma_{\hat{p}} = \sqrt{\frac{p_0(1-p_0)}{n}} = \sqrt{\frac{0.5 * 0.5}{300}} = 0.0288675$$

$$z = \frac{\hat{p} - p_0}{\sigma_{\hat{p}}} = \frac{0.45 - 0.5}{0.0288675} = -1.73$$

The observed value of the test statistic z = -1.73, which is less than the critical value of z = -1.65. Therefore, we reject H_0, and accept the alternative hypothesis H_1. In other words, the difference between the sample proportion and the hypothesized value of the population proportion is large and this difference is unlikely to have occurred due to chance alone.

We now define *semi-stable support rules*.

Definition (semi-stable support rules): Let minsup and minconf be the minimum support and the minimum confidence, $conf_D$ and sup_D be the actual support and the actual confidence of a rule r obtained from the whole dataset D, sup_i be the actual support of the rule in the time period T_i, and α be a specified significance level. The rule r is a *semi-stable support rule* over the time periods, T_1, T_2, …, T_q, if the following two conditions are met.

1. $sup_D \geq$ minsup and $conf_D \geq$ minconf.
2. For each time period (or sub-dataset), we fail to reject the following null hypothesis at the significance level α:

$$H_0: \ sup_i \geq \text{minsup}$$

Here, again we can use the statistical test on a single proportion to test the null hypothesis.

Computation complexity: Let the number of rules in R be $|R|$, and the number of time periods be q. The complexity of semi-stability test is $O(q|R|)$. Since q is normally very small, and $|R|$ is very large (in thousands or more), the computation is thus linear in $|R|$.

4.2 Stable rules

A semi-stable rule only requires that its confidences (or supports) over time are not statistically below minconf (or minsup). However, the confidences (or supports) of the rule may vary a great deal. Hence, the behavior can be unpredictable. In practice, the user often wants rules that are stable over time, i.e., with predictable behaviors. We call such rules *stable rules*. Intuitively, a stable rule is a semi-stable rule and its confidences (or supports) over time do not vary a great deal, i.e., they are homogeneous.

Since both confidence and support are population proportions, we need a statistical test for population proportions. The Chi-square test [26, 17] is a popular choice for testing homogeneity of multiple proportions.

Definition (stable confidence rules): Let minsup and minconf be the minimum support and minimum confidence, $conf_i$ be the actual confidence of a rule in the time period T_i, and α be a specified significance level. The rule r in R is a *stable confidence rule* over the time periods, T_1, T_2, …, T_q, if the following two conditions are met:

1. r is a semi-stable confidence rule.
2. We fail to reject the following hypothesis at the significance level α:

$$H_0: conf_1 = conf_2 = \ldots = conf_q$$

Test of homogeneity: A test of homogeneity involves testing the null hypothesis (H_0) that the proportions, p_1, p_2, …, p_k, in two or more different populations are the same against the alternative hypothesis (H_1) that these proportions are not the same. That is,

H_0: $p_1 = p_2 = \ldots = p_k$

H_1: the population proportions are not all equal.

We assume that the data consists of independent random samples of size n_1, n_2, …, n_k from k populations. The data is arranged in a $2\times k$ contingency table (Figure 1). The numbers x_1, x_2, …, x_k, n_1-x_1, n_2-x_2, …, n_k-x_k listed inside

	Sample			
	1	2	…	k
Successes	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	**…**	$\boldsymbol{x_k}$
Failures	$\boldsymbol{n_1\text{-}x_1}$	$\boldsymbol{n_1\text{-}x_2}$	**…**	$\boldsymbol{n_1\text{-}x_k}$

Figure 1: A $2\times k$ contingency table

the $2k$ cells are called *observed frequencies* of the respective cells.

Let O be an observed frequency, and E be an expected or theoretical frequency for a cell in the above table. The statistic defined as

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

has a χ^2 distribution with

$$df = (Row - 1)(Column - 1)$$

degrees of freedom, where *Row* and *Column* are the number of rows and the number of columns, respectively, in the given contingency table.

Under the null hypothesis in a test of homogeneity, we would expect the frequency E (expected frequency) for a cell to be as follows [26, 17]:

$$E = \frac{\text{(Row total)(Column total)}}{\text{Sample size}}$$

Thus, χ^2 represents a normalized deviation from expectation. If all values were really homogeneous, the χ^2 value would be 0. If it is higher than a threshold value (5.99, at the 5% significant level and two degrees of freedom) we reject H_0. Note that the threshold value for any given significance level can be obtained from widely available tables for Chi-square distribution.

Example 2: In an application, we have three years of data with 30000 tuples. The data is partitioned into three sub-datasets according to years. We want to analyze a particular rule r, $A \rightarrow B$, found in the dataset. The confidences of r in different time periods are:

T_1 (year 1): confidence$_1$ = 70% (= 450/680)
T_2 (year 2): confidence$_2$ = 80% (= 400/500)
T_3 (year 3): confidence$_3$ = 58% (= 420/720)

Note that the confidence of r in D_i is computed using the support counts of A and $A \wedge B$ in D_i. These support counts are given in parentheses next to each confidence value. We want to test the null hypothesis that the confidences of r over the 3 time periods are the same using the significance level $\alpha = 5\%$.

The confidence information (using the support counts) can be presented in a 2x3 contingency table (Figure 2) containing 6 cells. The expected frequencies are included in parentheses next to the observed frequencies within the corresponding cells.

For the values in the table of Figure 2, the value for the

	T_1	T_2	T_3	Col. Total
satisfy $A \wedge B$	**450** (454.5)	**400** (334.2)	**420** (481.3)	1270
satisfy $A \wedge \neg B$	**230** (225.5)	**100** (165.8)	**300** (238.7)	630
Row total (satisfy A)	680	500	720	1900

Figure 2: The 2x3 contingency table of confidences

test statistic χ^2 is 62.7 [1]. If we use the significance level of 5%, the critical value for χ^2 is 5.99 with 2 degree of freedom (df = (2-1)(3-1) = 2). The observed χ^2 value is much larger than the critical value. Thus, we reject the null hypothesis, and conclude that the confidences of the rule over the three time periods are significantly different.

The above example tests the homogeneity of confidences of a rule over time. We can also use the same method to test the supports of a rule.

Definition (stable support rules): Let minsup and minconf be the minimum support and minimum confidence, sup_i be the actual support of a rule r in R in the time period T_i, and α be a specified significance level. The rule r is a *stable support rule* over the time periods, T_1, T_2, …, T_q, if the following two conditions are met:

1. r is a semi-stable support rule.
2. We fail to reject the following hypothesis at the significance level α:

   H_0: $sup_1 = sup_2 = \ldots = sup_q$

After a set of stable (confidence or support) rules are found, ranking can be performed according to the mean confidence or the mean support of each rule over time. This ranking allows the user to see rules that have higher average confidences or supports first.

Computation complexity (without considering the final ranking): Since the number of cells in a contingency table is $2q$, the computation complexity of stability test is $O(q|R|)$. Since q is small and $|R|$ is very large, the computation is linear in $|R|$.

4.3 Rules that exhibit trends

In many applications, users are interested in knowing whether changes in support or confidence of a rule over time are random or there is an underlying trend. A statistical test called the *run test* [11] is often used to test whether a sequence of data is from a random process or exhibits trend. Below, we give a brief description of the run test in our application context.

Assume we have a rule that has the following confidences for T_1, …, T_6.

Time	confidence	change	
T_1	70%		
T_2	75%	+	
T_3	79%	+	Run 1
T_4	82%	+	
T_5	78%	-	Run 2
T_6	81%	+	Run 3

The direction of change in successive observations is shown by plus and minus signs. A *run* is a succession of plus or minus signs, surrounded by the opposite sign. If we read from top to bottom, we would say that we have a *run* of three +s, followed by a *run* of one -, and followed by a *run* of one +. In our example, there are 3 runs: one run of length three and two runs of length one.

The above sequence may suggest a trend toward increasing values of confidence. That is, these values of confidence may not reasonably be looked upon as being the items of a random sample.

The exact run test for trends is based on the probability density function (p.d.f) of W, the number of runs in a combined ordered sample consisting of two types of items, X and Y, say. Let m be the number of X items, and n be the number of Y items. Then as shown in [11], the p.d.f of W is given by:

when W is odd:

$$\Pr(W = 2k+1) = \left\{ \binom{m-1}{k}\binom{n-1}{k-1} + \binom{m-1}{k-1}\binom{n-1}{k} \right\} \Big/ \binom{m+n}{m}$$

when W is even:

$$\Pr(W = 2k) = \binom{m-1}{k-1}\binom{n-1}{k-1} \Big/ \binom{m+n}{m}$$

where $2k$ and $2k+1$ are elements that belong to the space of W, and k is a positive integer. When $k = 0$ (i.e., $W = 1$), we need special treatment. There are two cases:

1 If $m = 0$ or $n = 0$, $\Pr(W = 1) = 1$. That is, if $m = 0$ or $n = 0$, one run is the only possibility. This case clearly indicates a trend.

2 If $m \neq 0$ and $n \neq 0$, $\Pr(W = 1) = 0$. That is, if $m \neq 0$ and $n \neq 0$, it is not possible to be one run.

We are interested in testing:

H_0: sequence generated by a random process
H_1: sequence exhibits trend

The critical region of this test is of the form $W \leq v$. v is determined by computing $\alpha = \Pr(W \leq v; H_0)$, where α is the significance level (normally 5% or 10%). Let W_0 denote the observed number of runs (e.g., in our above

---

[1] One of the main assumptions of the χ^2 test is that the expected frequencies are not be too small [8]. The rule of thumb is that the χ^2 test is appropriate only when no expected frequency is less than 5. This seldom happens in our case as we normally deal with large datasets. When an expected frequency is below 5, extensions of the Fisher's exact test may be used, which, however, is quite cumbersome and not popular [8]. In our application, we find that for stability test the user is willing to give a range such that if the confidences or supports are within the range, the rule is considered stable.

example, $W_0 = 3$). If W_0 is greater than v, we fail to reject the null hypothesis. In the computation, we do not need to find the value of v. We can simply do the following: We compute $P = \Pr(W = 1) + \Pr(W = 2) + \ldots + \Pr(W = W_0)$. If $P \leq \alpha$, we reject H_0, and accept the alternative hypothesis H_1: *sequence exhibits trend*.

When m and n are small, it is easy to compute the probability. When they are large, the distribution of W can be approximated by a normal distribution [11] with mean

$$\mu = 2\frac{mn}{m+n} + 1$$

and variance $$\sigma^2 = \frac{(\mu-1)(\mu-2)}{m+n-1}$$

We can then use the standard normal approximation to test our hypothesis:

$$z = \frac{W - \mu}{\sigma}$$

The run test above finds those rules that exhibit trends. However, it does not tell the types of trends. We rank the rules according to the number of +s in each of them to show whether a rule is more likely to have an upward or a downward trend. Thus, the rules on the top are more likely to show upward trends, while the rules at the bottom are more likely to show downward trends.

Computation complexity (without considering the final ranking): Since "m (or n) choose k" and finding the numbers m and n can be done in linear time (in q), the complexity of the trend analysis is $O(qW_0|R|)$. *S*ince both q and W_0 ($W_0 \leq q$) are small, the whole computation is linear in $|R|$. When q is large (more than 20 [11]), we can use the normal approximation. Then, the complexity becomes $O(q|R|)$ as z can be computed in $O(1)$.

5. Evaluation

We now report our experiment results. The objective is to show the effectiveness and efficiency of the proposed technique. We used 4 real life datasets, one from a vehicle insurance company, one from an education institution, and two from another education institution. We did not use public domain datasets such as those in UCI machine learning repository [19] because most of these datasets are not time or sequence related. For those datasets that have sequence, we do not understand them and thus do not know how to partition them meaningfully.

a. Finding different types of rules

Table 1 (next page) shows the datasets and the results of rule generation and statistical tests. Column 1 gives the name of each dataset. Column 2 gives the number of tuples in each dataset. Column 3 shows the number of partitions of each dataset. In our case, each partition contains one year of data. Column 4 gives the number of rules generated from all partitions or blocks, i.e., the size of R. Column 5 gives the number of rules generated from the original dataset D. Note that the minsup and minconf values were suggested by our users.

From column 4 and 5, we can see that the number of rules generated from each dataset is very large. Column 6 gives the number of rules left after pruning from all partitions or blocks. Column 7 gives the number of rules after pruning for the whole dataset D (we used the pruning method in [14]). From column 6 and 7, we observe that the number of rules after pruning is substantially smaller for each data. Column 8 gives the execution time (in seconds) for rule generation. Column 9 gives the execution time for statistical tests (running on Pentium II 350 with 128MB RAM). We see that statistical tests can be done very efficiently.

The following two tables (Table 2 and 3) show the statistical test results [2]. We used the significance level of 5% (a commonly used level [26, 17]) in all our statistical tests. In column 2 of both Table 2 and Table 3, we reproduce the number of rules to be analyzed for each dataset, which is shown in column 6 of Table 1. Table 2 gives the results of the analysis on rule confidence. We observe that although the number of discovered rules from each dataset is large, the number of stable confidence rules (column 3) and the number of rules that exhibit trends (column 5) are actually quite small. Column 4 gives the number of semi-stable rules (which does not include the stable rules). We observe that the education domains are quite stable as most of the rules are either stable or semi-stable. The insurance domain is, however, much more volatile. A large majority of rules are unstable. Rule support analysis given in Table 3 shows similar results. Since the number of stable rules and trend rules are not that large, manual analysis is possible.

5.2 Accuracy tests

The main aim of our technique is to find rules that can be trusted and used in the future. Thus, we need to see whether the stable rules identified will still be stable and trend rules will still maintain their trends in the future. For these experiments, we make use of unseen test sets. Since we partition each dataset into q blocks, we use the first q-1 blocks to find stable rules, semi-stable rules and trend rules. We then test them on the unseen qth block.

For comparison, we use mean squared error and mean absolute error to evaluate stable rules, semi-stable rules and the rest of the rules (called *unstable rules*). We will use the mean (expected) confidence (or support) of each

[2] In Chi-square test, it is assumed that the populations are independent. For the 3 education datasets, this is true as the data from different years represent different student populations. In the insurance case, the dataset records those insurers who had accidents and made claims. It is also reasonable to assume that the data from different years are independent.

Table 1: Application datasets and the generated rules

1	2	3	4	5	6	7	8	9
	tuples	blocks	rules in all blocks	rules in D	rules in all blocks (aft. prn)	rules in D (aft. prn.)	rule gen. (sec.)	analysis (sec.)
Insur	158151	7	81463	20177	5026	3797	80.52	9.34
Edu1	4780	5	46428	21516	2743	2743	7.28	7.36
Edu2-1	6540	5	4606	2776	1075	1057	2.74	0.61
Edu2-2	5320	5	11722	8531	1177	1174	5.76	1.43

Table 2: Analysis on rule confidences

1	2	3	4	5
CONF	all rules	stable	semi-stable	trend
Insur	5026	252	132	213
Edu1	2743	209	1919	35
Edu2-1	1075	215	603	52
Edu2-2	1177	433	626	10

Table 3: Analysis on rule supports

1	2	3	4	5
SUP	all rules	stable	semi-stable	trend
Insur	5026	20	20	223
Edu1	2743	494	782	166
Edu2-1	1075	314	239	86
Edu2-2	1177	257	241	62

rule over the q-1 time periods as the predicted confidence (or support) of the rule in the new (qth block) data. We want to see the error produced by each type of rules on the test data. Mean squared error (MSE) and mean absolute error (MAE) are commonly used in statistics to evaluate model accuracy. They are defined as follows:

$$MSE = \frac{\sum (Y_i - \hat{Y}_i)^2}{S} \qquad MAE = \frac{\sum | Y_i - \hat{Y}_i |}{S}$$

where Y_i is the confidence (or support) of a rule in the new data, and $\hat{Y}_i$ is the estimated confidence (or support) of the rule, which is the mean value of the confidences (or supports) of the rule in the q-1 blocks. S is the total number of rules tested.

Table 4 and 5 show the accuracy results on confidence and support respectively. We discuss Table 4 first. Column 2 of Table 4 gives the percentage of stable confidence rules that remain stable after considering the qth block (the final year) of data. That is, our statistical tests now consider one more year. In the case of insurance data, 96.4% of stable rules remain stable. 2.4% of the stable rules become semi-stable. For the other three datasets, the percentages of stable rules remaining stable are slightly smaller, but are still very large. We believe the reason is that these datasets are smaller and thus tend to have bigger variations over time. However, the remaining rules are almost all semi-stable. Column 4 shows that almost all semi-stable confidence rules are still semi-stable after the qth block is considered (note that the semi-stable rules here do not include stable rules). Column 5 shows that most trend rules still exhibit trends. The proportions are slightly less than those for stable rules and semi-stable rules because the number (q) of time periods in our experiments is small and thus the trend test is less stable. Column 6, 7 and 8 show the mean squared errors for each dataset when we use the mean confidences of stable rules, semi-stable rules and unstable rules to predict the confidences of these rules on the qth data block respectively. Column 9, 10, and 11 give the respective mean absolute errors for each dataset. We can see that stable rules commit least error. The error of unstable rules can be very large. For example, in the case of Edu1, for

Table 4: Accuracy results on confidence analysis

1	2	3	4	5	6	7	8	9	10	11
	stable		**semi stable**	**trend**	**MSE**			**MAE**		
CONF	**stable**	**semi-stable**	**semi-stable**	**trend**	**stable**	**semi-stable**	**unstable**	**stable**	**semi-stable**	**unstable**
Insur	96.4%	2.4%	91.15%	73.7%	0.0004	0.00060	0.00684	0.008	0.015	0.062
Edu1	67.0%	33.0%	98.68%	54.2%	0.0021	0.00641	0.04273	0.035	0.059	0.168
Edu2-1	72.6%	26.1%	91.69%	61.2%	0.0162	0.03490	0.06423	0.064	0.119	0.177
Edu2-2	69.5%	29.8%	96.60%	88.6%	0.0071	0.00770	0.02102	0.066	0.067	0.096

Table 5: Accuracy results on support analysis

1	2	3	4	5	6	7	8	9	10	11
	stable		**semi stable**	**trend**	**MSE**			**MAE**		
SUP	**stable**	**semi-stable**	**semi-stable**	**trend**	**stable**	**semi-stable**	**unstable**	**stable**	**semi-stable**	**unstable**
Insur	100.0%	0.0%	100.0%	84.1%	0.00000	0.00000	0.00002	0.0001	0.0002	0.002
Edu1	99.0%	0.2%	86.4%	50.8%	0.00011	0.00089	0.00236	0.0080	0.0230	0.039
Edu2-1	99.7%	0.3%	84.4%	62.3%	0.00006	0.00017	0.00011	0.0060	0.0100	0.011
Edu2-2	98.1%	0.8%	79.5%	51.8%	0.00011	0.00020	0.00028	0.0080	0.0110	0.012

unstable confidence rules, the error in confidence on the *q*th block is 16.8% on average, while the error in confidence for stable confidence rules is only 3.5%. The errors with the insurance data are smaller because this dataset is large and thus the variance is small. Note that the MSE and MAE values are not computed for trend rules, as it is not reasonable to use the mean confidence as the predicted confidence in the next time period due to the trend. Our trend test is mainly to give the user an indication that trends exist rather than predict the confidence of the rule in the future, for which we need a larger sample (large *q*) and further statistical analysis.

Table 5 gives the corresponding results on the analysis of supports. From column 2 and 3, we see that almost all stable support rules are still stable. A large majority of semi-stable support rules are also still semi-stable (column 4). Column 5 shows that most trend rules still exhibit trends. From column 6-11, we observe that the stable support rules commit least error. The results from both Table 4 and 5 confirm that stable rules and trend rules identified by our technique are indeed reliable, and can be trusted in the future.

6. Conclusions

Association mining is an important data mining task. Its application is, however, hampered by the fact that it often generates a large number of rules. In this paper, we argue that the large number of rules is only part of the problem. Lack of systematic analysis procedures to help the user understand the behavior of rules is also a major issue. Without understanding the behavior of a rule over time, the rule cannot be used in practice. This paper presented a number of statistical tests to analyze the behavior of rules. Using these tests, we are able to identify stable rules and trend rules, and at the same time remove those unreliable (unstable) rules. Experiment results showed that the proposed technique is very effective and efficient.

References

[1]. Aggarwal, C., and Yu, P. "Online generation of association rules." *ICDE-98*, 402-411.

[2]. Agrawal, R. and Psaila, G. "Active data mining." *KDD-95*, 1995.

[3]. Agrawal, R. and Srikant, R. “Fast algorithms for mining association rules.” *VLDB-94*, 1994.

[4]. Bayardo, R. and Agrawal, R. "Mining the most interesting rules" *KDD-99*, 1999.

[5]. Dong, G. and J. Li. “Efficient mining of emerging patterns: discovering trends and differences.” *KDD-99*, 1999.

[6]. Ganti, V., Gehrke, J., and Ramakrishnan, R. "A framework for measuring changes in data characteristics" *POPS-99*.

[7]. Ganti, V. Gehrke, J. and Ramakrishnan, R. “DEMON: Mining and Monitoring Evolving Data.” *ICDE-2000*.

[8]. Hamilton, L. C. *Modern data analysis: a first course in applied statistics*, Brooks/Cole, 1990.

[9]. Han, J. and Fu, Y. “Discovery of multiple-level association rules from large databases.” *VLDB-95*.

[10]. Hilderman, R. and Hamilton, H. "Principles for Mining Summaries Using Objective Measures of Interestingness." *ICTAI-00*, 2000.

[11]. Hogg, R. & Craig, A. *Introduction to mathematical statistics*. Macmillan Publishing, 1970.

[12]. Klemetinen, M., Mannila, H., Ronkainen, P., Toivonen, H., and Verkamo, A.I. “Finding interesting rules from large sets of discovered association rules.” *CIKM-94*, 1994.

[13]. Liu, B., Hu, M. and Hsu, W. “Multi-level organization and summarization of the discovered rules.” *KDD-2000*.

[14]. Liu, B. Hsu, W. and Ma, Y. “Pruning and summarizing the discovered associations.” *KDD-99*.

[15]. Liu, B., Hsu, W. Mun, L. and Lee, H. “Finding interesting patterns using user expectations.” *IEEE Trans. on Knowl. & Data Eng*, vol: 11(6), 1999.

[16]. Liu, H., Lu, H., Feng, F and Hussain, F. “Efficient search of reliable exceptions.” *PAKDD-99*, 1999.

[17]. Mann, P. S. *Introductory statistics*. John Wiley & Sons, 1998.

[18]. Megiddo, M. and Srikant, R. “Discovering predictive association rules.” *KDD-98*, 1998.

[19]. Merz, C. J, and Murphy, P. UCI repository of machine learning databases, 1996. [http://www.cs.uci.edu/~mlearn/MLRepository.html].

[20]. Ng, R., Lakshmanan, L. Han, J. “Exploratory mining and pruning optimizations of constrained association rules.” *SIGMOD-98*, 1998.

[21]. Padmanabhan, B., and Tuzhilin, A. “A belief-driven method for discovering unexpected patterns.” *KDD-98*, 1998.

[22]. Piatesky-Shapiro, G., and Matheus, C. “The interestingness of deviations.” *KDD-94*, 1994.

[23]. Silberschatz, A., and Tuzhilin, A. “What makes patterns interesting in knowledge discovery systems.” *IEEE Trans. on Know. and Data Eng.* 8(6), 1996, pp. 970-974.

[24]. Suzuki, E. “Autonomous discovery of reliable exception rules.” *KDD-97*, 1997.

[25]. Tan, P-N. and Kumar, V. “Interestingness measures for association patterns: a perspective.” *KDD-2000 Workshop on Post-processing in Machine Learning and Data Mining*, 2000.

[26]. Walpole, R & Myers, R. *Probability and statistics for engineers and scientists*. Prentice Hall, 1993.

[27]. Zaki, M. “Generating non-redundant association rules.” *KDD-2000*, 2000.

[28]. Zhong, N., Ohshima, M., & Ohsuga, S. “Peculiarity Oriented Mining and Its Application for Knowledge Discovery in Amino-acid Data.” *PAKDD-2001*.