Proceedings of "The 6^{th} SEAMS-UGM Conference 2011"
Topic, pp. xx–xx.

SURVEY OF METHODS FOR MONITORING ASSOCIATION RULE BEHAVIOR

Ani Dijah Rahajoe, Edi Winarko

Abstract. Much of the existing research in data mining focused on how to generate rules efficiently from static databases. However, it is often the database for generating the rules has been collected over a considerable period of time, making it subject to changes than not. As a result, it is possible that the underlying rules will also be subject to change as a function of time. As an example, consider the personalization of web sites, changes in topology of the web site may result in different user navigation behaviour. Observation on the changing of the rules could provide more useful information that the rules themselves. In this paper, we survey different methods for monitoring the behaviour of association rules over periods of time.
Keywords and Phrases: Data mining, Association rules, rule evolution, rule monitoring.

1. INTRODUCTION

Much of the existing research in data mining focused on how to generate rules efficiently from static datasets. However, it is often the dataset for mining has been collected over a considerable period of time making it subject to changes than not. Moreover, consumer behavior and preferences also change over time. As a result, it is possible that the underlying rules will also be subject to change as a function of time. As an example, consider the personalization of web sites, changes in topology of the web site may result in different user navigation behaviour.

Research associated with mining rule changes has been focused in several areas. In classification mining, there has been work on *concept drift*, see [1, 2]. Here concept drift refers to the phenomenon that some or all of the rules defining classes change over time. In association rule mining, a number of works on mining rule changes has been done with the objective to observe the behaviour of association

2010 Mathematics Subject Classification:

rules over several time periods [3, 4, 5, 6, 7]. In addition, other works focus on the changes that occurred over two periods of time [8, 9, 10]. Apart from these, a number of studies, called *incremental mining* has emerged. The objective is to update previously discovered rules incrementally when the underlying dataset is updated. By doing this, they can avoid scanning the whole dataset again. In incremental mining the statistical properties of known patterns are updated, instead of being recorded over time as in mining rule change. Incremental mining algorithms designed for maintaining discovered association rules can be found in [11, 12, 13, 14, 15].

In this paper, we survey different methods for monitoring the behaviour of association rules. We first give an overview of research in mining association rule changes. In this context, mining rule change generally consists of three steps. The most important step is the monitoring step, in which different methods to monitor the rules have been proposed. We will describe each step briefly, then present the classification of methods currently used in the monitoring. Finally, we describe how each method in each category is used to monitor the behaviour of association rules.

1.1. **Overview of Research in Mining Rule Change.** Based on their mining platform, research on mining association rule changes has been done in both temporal database and non-temporal database. However, only a few of them using temporal databases as a platform for data mining task, for example [3]. Saraee *et al.* [16] is the first to introduce a framework for mining association rules and sequential patterns from a temporal database, using the *ORES* temporal database management system. However, the framework does not consider the rule changes. This work focuses on two areas and their integration, i.e., data mining as a technique to increase the quality of data, and temporal databases as a technique to keep the history of data. A number of enhancements to the basic algorithm for mining association rules and sequential patterns is introduced. One of them is a new measure for mining association rules, called *time confidence*. Tansel *et al.* [3] study the problem of discovering association rules and their evolution from temporal databases. The proposed approach allows the user to observe the changes in association rules occur over periods of time. The observed changes include a decrease/increase in the support/confidence of an association rule and addition/removal of itemsets from a particular itemsets.

The problem of monitoring the support and confidence of association rules from non-temporal databases has been addressed in [4, 5, 6, 7]. Agrawal *et al.* [4] propose the method to monitor rules from 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 are represented and defined using shape operators. The user can then query the rule base by specifying some history specifications. In addition, the user can specify triggers over the rule base in which the triggering condition is a query on the shape of the history. In [5], Liu *et al.* propose a technique to use statistical methods to analyze the behavior of association rules over time. They focus on determining rules that are semi-stable, stable, showing trends over the several time periods. [6] proposes visualization techniques that allow the user to visually analyze association rules and their changing behaviours over a number of time periods. Baron *et al.* [7] introduce the GRM (*General Rule Model*) to model both the content and the statistics of a rule as a temporal object. Based on this two components of a rule, different types of pattern evolution are defined, such as changes of statistics or content, disappearance of a rule, the correlated changes of pairs of rules. In [17], Baron *et al.* study the evolution of web usage patterns using PAM (*PAttern Monitor*). The association rules that show which pages tend to be visited within the same user session are generated from a web server. They demonstrate how the mechanisms implemented by PAM can be used to identify interesting changes in the usage behaviour. In most of these works, the behaviour of rules is based on the behaviour of the rule's statistics, the changing in support and confidence values. They do not consider the changes in the rule contents.

Other works in mining association rule changes focus on detecting the changes from two datasets, i.e., to find rule changes that occur from one dataset to another [8, 9, 10]. Ganti *et al.* [8] present a general framework for measuring changes or differences in two sets of association rules from two datasets. They compute a deviation measure which makes it possible to quantify the difference between two datasets in terms of the model they induce, called FOCUS. In [9], Dong and Li introduce a new kind pattern, called *emerging pattern*. The support differences of association rules mined from two datasets are used to detect the emerging patterns. Liu *et al.* [10] study the discovery of *fundamental rule changes*. They consider rules of the form $r_1, \ldots, r_{m-1} \rightarrow r_m$ and detect changes on support or confidence between two consecutive time periods by applying a chi-square test.

The rest of the paper is organized as follows. Section 2 describes three basic steps in mining association rule changes. Section 3 describes several monitoring methods based on statistical test, while Section 4 describes monitoring methods based on visualization. Section 5 presents methods to monitor rule behaviour in two datasets. The conclusion and future work are given in Section 6.

2. BASIC STEPS IN MINING RULE BEHAVIOR

Most of works in mining association rule changes divides the process into three steps. The first step is partitioning the dataset to extracts the portion of the data for each time period. The second step is generating the rules, and the final step is monitoring the rules. This section briefly describes each of these steps.

Step 1: Partitioning the dataset. In this step, the important parameters are the length of the interval during which data is accumulated, and the number of such intervals. Then, we walk over the dataset D to extract a subset of dataset for each time period. Let $t_i = [b_i, f_i)$ be a time period where b_i denotes its starting time point and f_i denotes its end. Time periods $t_1, t_2, \ldots, t_n$ are consecutive, non overlapping fixed length time periods. D_i denotes the portion of dataset that is valid during the time period t_i.

Step 2: Mining rules from sub-datasets. Two different approaches are generally used to mine the rules from the set of sub-datasets. The first approach is to mine each sub-dataset D_i in sequence. Let R_i be the set of temporal association rules from D_i, then after the mining we have the rule sets of $R_1, R_2, R_3, \ldots$, accordingly. If R is the set of rules that will be monitored in the next step, R is defined as $R = \{r | r \in (R_1 \cup R_2 \ldots \cup R_q)\}$. It is possible for a rule $r \in R$ to appear in R_i but not in R_j $(i \neq j)$ because r may not satisfy *minsup* and/or *minconf* in D_j. This approach is used in [3, 4, 5, 7, 17].

The second approach is to mine the rules from one sub-dataset, and apply the resulting rules on other sub-datasets to calculate the support and confidence values on them. It means that only initial mining session is launched (on D_1). At each later time period, an instance of each existing rule is created, computing the statistic values from the sub-dataset in the corresponding time period. If R is the set of rules that will be monitored in the next step, R is defined as $R = \{r | r \in R_1\}$. Thus, for each rule, we get a sequence of support and confidence values. This approach is used in [6].

The first approach results in larger number of rules than the second one. However, the users may find this is more useful as it gives more detailed view of the whole data. In the second approach, since the monitoring is focused only on the rules generated from the first time period, it cannot be used to detect new rules that appear in the next time periods. It can only detect rules that disappear in the next time periods.

A variation of the second approach is proposed in [18], by selecting a subset of rules generated in the first time period. If R is the set of rules that will be monitored, then R is a subset of R_1. It results in reducing the computational effort to a minimum while focusing only on interesting rules. If the user choose to monitor all rules in R_1, then it would be similar to the second approach described above. Choosing the rules to be monitored is generally user and application dependent. The rules that are interesting to one user may be of no interest to another user, and the interestingness of patterns varies from application to application. Regardless of which approach is used, the number of discovered rules could still be large. Several methods have been proposed to reduce the number of generated rules, for example by pruning [19, 20] or by using templates [21].

For the monitoring purpose, we need support and confidence values of every rule $r \in R$, in all time periods. Therefore, we need to obtain the missing support and confidence values in certain time periods for each $r \in R$. This can be done by rescanning the corresponding sub-dataset to calculate the support and confidence values. If a rule does not appear in a sub-dataset D_k, we set its support and confidence values in a time period t_k to zero.

Step 3: Monitoring rules over time. The direct and simple approach is to monitor the rule from one time period to other time periods by comparing the support and confidence of each rule from all time periods. This can be done using a graph, where x-axis represents the time line, and y-axis represent the support of a large itemset or support/confidence of a particular rule. It is useful when the user

Statistical based methods

Method Description	Statistical test	Authors
1. Detecting semi-stable rules	z test	Liu *et al.* 2001 [5]
2. Detecting stable rules	Chi-square test	Liu *et al.* 2001 [5]
3. Detecting rules that exhibit trends	Run test	Liu *et al.* 2001 [5]
4. Detecting rules with significant changes	Two-tail binomial test	Baron & Spiliopoulou 2003 [17]

Visualization based methods

Method Description	On display	Author
1. Visualize similar rules	A group of rules	Zhao & Liu 2001 [6]
2. Visualize neighbour rules	A group of rules	Zhao & Liu 2001 [6]
3. Visualize a permanent rule	One rule	Baron *et al.* 2003 [18]

Monitoring from two datasets

Method Description	Detection tool	Author
1. Detecting emerging patterns	Border-based algorithm	Dong & Li 1999 [9]
2. Detecting fundamental rule changes	Quantivative analysis Qualivative analysis	Liu *et al.* 2001 [10]

TABLE 1. Classification of methods to monitor rule behaviour

wants to see the fluctuation in a particular rule. This method is used in [3, 22]. However, it has two drawbacks. First, it often reports far too many changes and most of them are simply the snowball effect of some fundamental changes. Second, analysing the difference in supports/confidences may miss some interesting changes [10].

In this paper, we describe monitoring methods which are more advance than the above method. We classify these methods into three categories: statistical based methods and visualization based methods, and methods to monitor the rule from two datasets, as shown in Table 1. Statistical based methods use statistical test, while visualization based methods use visualization.

3. STATISTICAL BASED METHOD

The statistical test used in the following methods is applied on an individual rule. The test is performed on either the support or confidence of the rule. The focus of our discussion is on the test applied to the rule's confidence. The test on the rule support is analogous.

3.1. Detecting Semi-Stable Rules.

A rule $r \in R$ is a semi-stable rule if none of its confidences (supports) in the time periods $t_1, t_2, \ldots, t_n$ is statistically below *minconf* (*minsup*). Its formal definition is defined below.

Definition 3.1. Semi-stable confidence rules. *Let minsup and minconf be the minimum support and confidence, sup_D and $conf_D$ be the support and confidence of a rule r from the whole dataset D, $conf_i$ be the 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 $t_1, t_2, \ldots, t_n$, if the following two conditions are met:*

1. *$sup_D \geq minsup$ and $conf_D \geq minconf$*
2. *for each time period t_i, we fail to reject the following null hypothesis at significance level α: $H_o : conf_i \geq minconf$*

The first condition is used to ensure that the confidence of a rule r satisfies the minimum confidence threshold in the whole dataset. The second condition is tested using the *z test*.

3.2. **Detecting Stable Rules.** A semi-stable rule only requires its confidences (supports) over time are not statistically below *minconf* (*minsup*). However, the confidences (supports) of the rule may vary a great deal. Hence, the behaviour can be unpredictable. A stable rule is a semi-stable rule and its confidences (supports) are homogeneous.

Definition 3.2. Stable confidence rules. *Let minsup and minconf be the minimum support and confidence, sup_D and $conf_D$ be the support and confidence of a rule r from the whole dataset D, $conf_i$ be the confidence of the rule in the time period t_i, and α be a specified significance level. The rule r is a stable confidence rule over the time periods $t_1, t_2, \ldots, t_n$, if the following two conditions are met:*

1. *r is a semi-stable confidence rule*
2. *we fail to reject the following null hypothesis at significance level α: $H_o : conf_1 = conf_2 = \ldots = conf_n$*

The second condition is tested using χ^2 test for testing homogeneity of multiple proportions.

3.3. **Detecting Rules that Exhibit Trends.** Sometimes the users are more interested in knowing whether changes in support or confidence of a rule are random or there is underlying trend. In this case, a statistical test called the run test can be used to detect if a rule's support or conficence values exhibit trend or not. The run test can find those rules that exhibit trends. But, it does not tell the types of trends.

3.4. **Detecting Significant Change.** In [17] a mechanism called *change detector* is used to identify significant changes. In this mechanism a two-tailed binomial test is utilized to verify whether an observed change is statistically significant or not.

For a a rule r and a statistical measure s at a time point t_i it is tested whether $r.s(t_{i-1}) = r.s(t_i)$ at a confidence level α. The test is applied upon the subset of data D_i accumulated between t_{i-1} and t_i, so that the null hypothesis means that D_{i-1} is drawn from the same population as D_i, where D_{i-1} and D_i have an empty intersection by definition. Then, for a rule r an alert is raised for each time point t_i at which the null hypothesis is rejected.

4. VISUALIZATION BASED METHOD

4.1. Visualizing Similar Rules. This method is used to visualize rules that have similar behaviour to a rule r. A commonly used method for searching similarity is to map objects into points in a high dimensional space, so that each point is represented as a series of values. Similarity of two objects is defined as the distance between their respective values.

In [6], euclidean distance is used as the distance function between two series. The distance $d(X, Y)$ between two series $X = \langle x_1, x_2, \ldots, x_n \rangle$ and $Y = \langle y_1, y_2, \ldots, y_n \rangle$ is

$$d = \sqrt{(\sum_{i=1}^{n}(x_i - y_i)^2}$$

where x_i and y_i are the values of X and Y in the ith time period, respectively. Given a similarity threshold value ϵ, if this distance is below user-defined threshold ϵ, we say that the two series are similar.

The parameter ϵ is a distance parameter that controls when two series should be considered similar. It could be either user-defined, or determined automatically. For a rule r with n time periods, ϵ is calculated as:

$$\epsilon = \frac{\sum_{i=1}^{n-1} |z_{i+1} - z_i|}{n - 1}$$

where z_i is a support (confidence) value of r at a period i. The bigger the value of ϵ, the more rules will be included as similar rules, with respect to r.

4.2. Visualizing Neighbour Rules. This method is used to visualize the neighbour rules of a rule r. A rule $r_1 : lhr_1 \to rhs_1$ is a neighbour of a rule $r_2 : lhr_2 \to rhs_2$ if the following two conditions are met:
(i). $rhs_1 = rhs_2$
(ii). $lhs_1 \supseteq lhs_2$ or $lhs_1 \subseteq lhs_2$
As an example, take a rule $r : A, B \to D$, then rules $r_1 : A \to D$, $r_2 : B \to D$, and $r_3 : A, B, C \to D$ are neighbour of r. But $r_4 : A, C \to D$ is not, because $\{A, B\} \not\supseteq \{A, C\}$ and $\{A, B\} \not\subseteq \{A, C\}$

4.3. Visualizing a Permanent Rule. A permanent rule is is a rule that is permanently supported by the data. All rules produced from the first partition D_1 are potential candidates for permanent rules and can be monitored. In real applications the number of rules may be huge. However, it shown in [18] that the number of permanent rules decreases relatively fast with the number of periods.

It may also be possible that there are no more permanent rules at one point in time. In order to avoid this problem, the approach used in [18] is as follows. Depending on the number of initially discovered rules, rules that disappear from the rule base in subsequent periods could be kept in memory until their support values violate a given threshold, or they are absent for more than a given number of periods. For example, a rule would be removed from memory if it appears in less than 50% of the periods, or if it is absent for more than three periods. Depending on the number of occurrences different rule groups could be defined (e.g., 100%, 90%, 75%, and 50% rules). For example, using these groups, a rule change could be considered interesting if it leads to a group change.

5. MONITORING RULE BEHAVIOR OVER TWO DATASETS

5.1. **Detecting Emerging Patterns.** Emerging patterns (EPs) are defined as itemsets whose support increase significantly from one dataset, D_1, to another, D_2. EPs can also be defined as itemsets whose ratios of the two supports are larger than a given threshold ρ. EPs can be large in size, and may have very small support. Since apriori property is no longer holds for EPs and because there are too many candidates, then naive algorithms such as Apriori algorithm [23] are not efficient for discovering EPs. Therefore, an efficient algorithm for discovering EPS was proposed in [9].

5.2. **Detecting Fundamental Rule Changes.** Fundamental rule changes are defined as changes that cannot be explained by other changes (its formal definition will be described below). To detect fundamental rule changes, two techniques are used: quantitative analysis and qualitative analysis. Quantitative analysis measures the magnitude of change, while qualitative analysis finds the direction of change. We describe each analysis by focusing our discussion on the change of the rule's support. The change of the rule confidence can be done in the same way. This method considers the rule r of the form r: $a_1, a_2, \ldots, a_n \rightarrow y$.

5.2.1. *Quantitative Analysis.* In order to perform this analysis, we need to calculate the expected support of a rule r in t_2, which is defined as follows:

1. If r is a 1-condition rule, its expected support in t_2 is its support in t_1
2. If r is a k-condition rule ($k > 1$) of the form $r : a_1, a_2, \ldots, a_n \rightarrow y$, then r can be considered as a combination of two rules, a 1-condition rule r_{one} and a k-condition rule r_{rest}, where

$$r_{one} : a_i \rightarrow y \qquad r_{rest} : a_1, a_2, \ldots, a_j \rightarrow y$$

and $\{a_1, a_2, \ldots, a_j\} = \{a_1, a_2, \ldots, a_n\} - \{a_i\}$. Let $sup_t(x)$ be the support of a rule x at a time period t. The expected support of r in t_2 with respect to r_{one} and r_{rest} are

$$E_{r_{one}}(sup_{t_2}(r)) = min\left(\frac{sup_{t_1}(r)}{sup_{t_1}(r_{one})} \times sup_{t_2}(r_{one}), 1\right) \quad (1)$$

$$E_{r_{rest}}(sup_{t_2}(r)) = min\left(\frac{sup_{t_1}(r)}{sup_{t_1}(r_{rest})} \times sup_{t_2}(r_{rest}), 1\right) \quad (2)$$

After we know the expected support of a rule r in t_2, we can check if the change in support of a rule r from t_1 to t_2 is fundamental or not. The change is fundamental if:

Rule	Support at t_1	Support at t_2
r_1: $a \to y$	5.2%	4.4%
r_2: $b \to y$	6.0%	4.3%
r_3: $a, b \to y$	2.1%	4.2%

TABLE 2. Rule support at two time periods

	$E(sup_{t_2}(r_1))$	$sup_{t_2}(r_1)$
satisfy r_1	52	44
do not satisfy r_1	948	956

TABLE 3. The 2x2 contingency table of r_1 support

1. r is a 1-condition rule and its support is significantly different from its expected support, or
2. r is a k-condition rule ($k > 1$), and $E_{r_{one}}(sup_{t_2}(r))$, $E_{r_{rest}}(sup_{t_2}(r))$ and $sup_{t_2}(r)$ are significantly different, for all r_{one} and r_{rest} combinations.

Then, χ^2 test is used to check if the support is significantly different from the expected support. It means that if r is a 1-condition rule, the different is significant if we fail to reject null hypothesis $H_o : E(sup_{t_2}(r)) = sup_{t_2}(r)$ at significance level α. If r is a k-condition rule, the different is significant if we fail to reject null hypothesis $H_o : E_{r_{one}}(sup_{t_2}(r)) = E_{r_{rest}}(sup_{t_2}(r)) = sup_{t_2}(r)$ at significance level α, for all r_{one} and r_{rest} combinations.

As an example, consider the example shown in table 2 (adopted from [10]). Since r_1 is a 1-condition rule, to test if the change in support of r_1 is fundamental, we have null hypothesis $H_o : E(sup_{t_2}(r_1)) = sup_{t_2}(r_1)$, where the $E(sup_{t_2}(r_1)) = sup_{t_1}(r_1) = 0.052$. Assume that the size of dataset at each period is 1000 tuples, we have 2×2 contingency table in which each cell contains observed frequencies of tuples that satisfy r_1 and do not satisfy r_1, for each time period, as shown in table 3. From this table, we can compute the value of chi-square, which is equal to 0.7. Using significance level of 5% and degrees of freedom 1, the critical value is equal to 3.84. Since the chi-squre value is smaller than the critical value, we do not reject the H_o, and conclude that the support of r_1 is not significantly different[1]. It means that r_1 does not show fundamental change in support.

5.2.2. *Qualitative Analysis.* In qualitative analysis the magnitude of change is ignored and is only focus on the direction of change, i.e., *increase, drop* or *noChange* from t_1 to t_2. Given a rule r, Definition 5.1 is used to determine if the changes of its support or confidence is fundamental or not.

Definition 5.1. Fundamental rule change in support/confidence via qualitative analysis. *The support (or confidence) change in a rule r from t_1 to t_2,*

[1]The computation is performed using chi-square calculator, which is available on the web at http://schnoodles.com/cgi-bin/web_chi.cgi

	Case I	Case II	Case III	Case IV	Case V	Case VI	Case VII
r_{one}:	increase	drop	noChange	increase	noChange	drop	noChange
r_{rest}:	increase	drop	noChange	noChange	increase	noChange	drop
r:	increase	drop	noChange	increase	increase	drop	drop

Figure 1. Explainable change combinations used in qualitative analysis

where r is a k-condition rule (k¿1), is said to be a fundamental support (or confidence) change if for all r_one and r_rest combinations, the directions of support (or confidence) changes of r_one, r_rest and r in time period 2 do not belong to any of the 7 cases in Figure 1.

6. CONCLUSION

All of the proposed methods, either statistical or visualization based methods, consider only the statistical properties of the rule, i.e., its support or confidence. In [7], the first step toward an integrated treatment of two aspects of a rule, its content and statistics, has been made, by proposing the GRM which models both the content and the statistics of a rule as a temporal object. In our next research, we will combine statistical and visualization methods for observing evolution of temporal association rules generated from interval sequence data.

References

[1] Hembold, H.M., Long, P.M.: Tracking drifting concepts by minimizing disagreements. Machine Learning **114** (1996) 27–45

[2] Hickey, R.J., Black, M.M.: Refined time stamps for concept drift detection during mining for classification rules. In Roddick, J.F., Hornsby, K., eds.: Proceedings of the 1st International Workshop, TSDM 2000. Volume 2007 of LNAI., lyon, France, Springer (2000) 20–30

[3] Tansel, A.U., Ayan, N.F.: Discovery of association rules in temporal databases. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, Distributed Data Mining Workshop, New York City, New York, USA (1998)

[4] Agrawal, R., Psaila, G.: Active data mining. In Fayyad, U.M., Uthurusamy, R., eds.: Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD'95), Montréal, Québec, Canada (1995) 3–8

[5] Liu, B., Ma, Y., Lee, R.: Analyzing the interestingness of association rules from the temporal dimension. In: Proceedings of IEEE International Conference on Data Mining (ICDM'01), Silicon Valley, USA (2001) 377–384

[6] Zhao, K., Liu, B.: Visual analysis of the behavior of discovered rules. In: Workshop Notes in ACM-SIGMOD 2001 Workshop on Visual Data Mining, San Fransisco, CA, USA (2001)

[7] Baron, S., Spiliopoulou, M.: Monitoring change in mining results. In Kambayashi, Y., Winiwarter, W., Arikawa, M., eds.: Proceedings of the 3rd International Conference on Data Warehousing and Knowledge Discovery (DaWak'01). Volume 2114 of LNCS., Munich, Germany, Springer (2001) 51–60

[8] Ganti, V., Gehrke, J., Ramakrishnan, R.: A framework for measuring changes in data characteristics. In: Proceedings of the 18th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Philadelphia, Pennsylvania (1999) 126–137

[9] Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. In: Knowledge Discovery and Data Mining. (1999) 43–52

[10] Liu, B., Hsu, W., Ma, Y.: Discovering the set of fundamental rule changes. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA (2001) 335–340
[11] Cheung, D.W.L., Ng, V., Tam, B.W.: Maintenance of discovered knowledge: A case in multi-level association rules. In: Knowledge Discovery and Data Mining. (1996) 307–310
[12] Cheung, D.W.L., Lee, S.D., Kao, B.: A general incremental technique for maintaining discovered association rules. In: Database Systems for Advanced Applications. (1997) 185–194
[13] Ayan, N.F., Tansel, A.U., Arkun, M.E.: An efficient algorithm to update large itemsets with early pruning. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA (1999) 287–291
[14] Omiecinski, E., Savasere, A.: Efficient mining of association rules in large dynamic databases. In Embury, S.M., Fiddian, N.J., Gray, W.A., Jones, A.C., eds.: Proceedings of the 16th British National Conference on Databases (BNCOD'98. Volume 1405 of LNCS., Cardiff, Wales, U.K., Springer (1998) 49–63
[15] Ganti, V., Gehrke, J., Ramakrishnan, R.: DEMON: Mining and monitoring evolving data. In: Proceedings of the 16th International Conference on Data Engineering (ICDE'00), San Diego, CA, USA, IEEE Computer Society Press (2000) 439–448
[16] Saraee, M.H., Theodoulidis, B.: Knowledge discovery in temporal databases. In: IEE Colloquium on 'Knowledge Discovery in Databases', IEE, London (1995) 1–4
[17] Baron, S., Spiliopoulou, M.: Monitoring the evolution of web usage patterns. In: Proceedings of the First European Web Mining Forum (EMWF 2003), Cavtat-Dubrovnik, Croatia (2003) 181–200
[18] Baron, S., Spiliopoulou, M., Günther, O.: Efficient monitoring of patterns in data mining environments. In Kalinichenko, L.A., Manthey, R., Thalheim, B., Wloka, U., eds.: Proceedings of the 7th East European Conference on Advances in Databases and Information Systems (ADBIS'03). Volume 2798 of LNCS., Dresden, Germany, Springer (2003) 253–265
[19] Toivonen, H., Klemettinen, M., Ronkainen, P., Hatonen, K., Mannila, H.: Pruning and grouping of discovered association rules. In: ECML-95 Workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases, Heraklion, Greece (1995) 47–52
[20] Liu, B., Hsu, W., Ma, Y.: Pruning and summarizing the discovered associations. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA (1999) 125–134
[21] Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., Verkamo, A.: Finding interesting rules from large sets of discovered association rules. In Adam, N., Bhargava, B., Yesha, Y., eds.: Proceedings of the 3rd International Conference on Information and Knowledge Management, Gaithersburg, Maryland, ACM Press (1994) 401–407
[22] Koundourakis, G., Theodoulidis, B.: Association rules and evolution in time. In: Proceedings of Methods and Applications of Artificial Intelligence, Second Hellenic Conference on AI, SETN 2002, Thessaloniki, Greece (2002) 261–272
[23] Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases. (1994) 487–499

Ani Dijah Rahajoe: Faculty of Informatic Engineering,
Universitas Bhayangkara, Surabaya.
E-mails: anidrahayu@gmail.com

Edi Winarko: Faculty of Mathematics and Natural Sciences,
Universitas Gadjah Mada, Jogjakarta.
E-mails: ewinarko@ugm.ac.id