Monitoring the Evolution of Web Usage Patterns

Steffan Baron[1] and Myra Spiliopoulou[2]

[1] Institute of Information Systems, Humboldt-Universität zu Berlin
Spandauer Str. 1, Berlin 10178, Germany
sbaron@wiwi.hu-berlin.de
[2] Institute of Technical and Business Information Systems,
Otto-von-Guericke-Universität Magdeburg,
PO Box 4120, Magdeburg 39016, Germany
myra@iti.cs.uni-magdeburg.de

Abstract With the ongoing shift from off-line to on-line business processes, the Web has become an important business platform, and for most companies it is crucial to have an on-line presence which can be used to gather information about their products and/or services. However, in many cases there is a difference between the intended and the effective usage of a web site and, presently, many web site operators analyze the usage of their sites to improve their usability. But especially in the context of the Internet, content and structure change rather quickly, and the way a web site is used may change often, either due to changing information needs of its visitors, or due to an evolving user group. Therefore, the discovered usage patterns need to be updated continuously to always reflect the current state.
In this article, we introduce PAM, an automated *Pattern Monitor*, which can be used to observe changes to the behavior of a web sites visitors. It is based on a temporal representation of rules in which both the content of the rule and its statistical properties are modeled. It observes pattern change as evolution of the statistical measurements captured for a rule throughout its entire lifetime and notifies the user about interesting changes within the rule base. We present PAM in a case study on the evolution of web usage patterns. In particular, we discovered association rules from a web-servers log that show which pages tend to be visited within the same user session. These patterns have been imported into the monitor, and their evolution throughout a period of 8 months has been analyzed. Our results show that PAM is particularly suitable to gain insights into the changes a rule base is affected of over time.

1 Introduction and Related Work

Knowledge discovery is an iterative process that reflects the need of extracting knowledge from data that accumulate constantly [7]. The application expert periodically invokes a data mining tool to extract patterns from the data. Each invocation contributes new insights on the application domain, enriching the expert's domain knowledge and, occasionally,

motivating her to revise her beliefs. As the expert becomes gradually familiar with the patterns being extracted, she is increasingly interested in *changes* rather than in already known patterns. Especially in the context of the Internet, content and structure change rather quickly, and the way a web site is used may change often or even permanently, either due to the changing information needs of its visitors, or due to an evolving user group. Therefore, the discovered usage patterns need to be updated continuously to always reflect the current state. In the case of a heavily used web site with thousands of users per day, the question arises how this can be achieved with a reasonable effort.

One major problem is the large number of discovered rules, and the identification of *interesting patterns* has become a widely discussed topic [10,8]. Even for relatively small datasets this a problem and makes the inspection of all rules impractical. In some cases, it may be possible to assess the interestingness of rules manually or with respect to the application context. However, there are a number of approaches that offer potential solutions to this problem in an application independent manner [22,13,14].

When data, as in the case of web-server logs, is continuously collected over a potentially long period, the concepts reflected in the data will change over time. Due to internal and/or external factors, the distribution and/or the structure of the dataset may change. This requires the user to monitor the discovered patterns continuously, which is of particular importance for applications that timestamp data. One possible way to deal with the temporal dimension is to use an appropriate partitioning scheme. However, if the partitions are too big or too small it may be the case that important rules and/or changes are missed. Normally, partitioning is highly application depended. However, there is research into formal methods for application independent partitioning of data. Chen and Petrounias focus on the identification of valid time intervals for previously discovered association rules [10]. They propose a mechanism that finds (a) all contiguous time intervals during which a specific association holds, and (b) all interesting periodicities that a specific association has. Chakrabarti et al. propose the discovery of surprising, i.e. unexpected and therefore interesting, patterns in market basket analysis by observing the variation of the correlation of the purchases of items over time [9]. The underpinnings of that work come from time series analysis, so that the emphasis is on partitioning the time axis into such intervals that the rule statistics change dramatically between two consecutive intervals.

In the last years, a number of methods and techniques for maintaining and updating previously discovered knowledge have emerged which are able to deal with dynamic datasets. A widely used approach is that of *incremental mining* in which the knowledge about already extracted patterns is re-used in subsequent periods. Originally, the emphasis of incremental mining was on optimizing the miner's performance from one invocation of the miner to the next. Most of this research focuses on the updation of association rules [11,2,21,23], frequent sequences [24], and clusters [12]. They aim at efficiently updating the content of discovered rules, thus avoiding a complete re-run of the mining algorithm on the entire updated dataset.

The DELI Change Detector of Lee et al. uses a sampling technique to detect changes that may affect previously discovered association rules [18,17]. It invokes an incremental miner to modify the patterns if this turns out to be necessary.

Ganti et al. propose the DEMON framework for data evolution and monitoring across the temporal dimension [16]. DEMON focuses on detecting systematic vs. non-systematic changes in the data and on identifying the data blocks (along the time dimension), which have to be processed by the data miner in order to extract new patterns. The emphasis is on actualizing the knowledge base by detecting changes in the data, rather than detecting changes in the patterns.

Another avenue of research concentrates on the similarity of rules and on the statistical properties of rules by considering the *lifetime* of patterns, i.e., the time in which they are sufficiently supported by the data [15,9,10,16]. Ganti et al. propose the framework FOCUS for the comparison of two datasets and the computation of an interpretable, qualifiable deviation measure between them, whereby the difference is expressed in terms of the model the datasets induce [15].

However, all of these proposals consider only part of a pattern, either its content, i.e., the relationship in the data the pattern reflects, or the statistical properties of the pattern. In [3], we took a first step towards an *integrated* treatment of these two aspects of a rule. We proposed the *Generic Rule Model* (GRM) which models both the content and the statistics of a rule as a temporal object. Based on these two components of a rule, different types of pattern evolution were defined. Additionally, a simple monitor was implemented which used a user supplied deviation threshold to identify interesting changes to pattern statistics.

Pattern change is usually caused by concept drift. As Kelly et al. point out in [20], adaptive classification algorithms, such as adaptive Bayesian

networks are designed to overcome concept drift by considering the impact of each individual new record on the existing classifier and adapting it accordingly. However, the rapid accumulation of records, as in web-server logs, makes the consideration of the impact of each record ineffective. Moreover, individual records that come in large numbers are noisy and reflect trends only partially, especially for trends that manifest themselves slowly, such as a change in preferences or demographics of a user group.

The problem of interestingness arises also when evaluating the changes that have affected a rule. A commonly used approach to protect the domain expert from inspecting too many rule changes is the definition of limits for e.g. the steepness of change for the observed statistical measurements. When these limits are exceeded, the corresponding rule change is considered to be interesting.

The temporal aspects of patterns are taken into account in the rule monitors of [1,8] and [3,4,6]. In [8], Liu et al. count the significant rule changes across the temporal axis. They pay particular attention on rules that are "stable" over the whole time period, i.e. do not exhibit significant changes, and juxtapose them with rules that show trends of significant increase or decrease. Significance tests form the basis of the experiments. In [1], upward and downward trends in the statistics of rules are identified using an SQL-like query mechanism. Closer to our work is the research of Liu et al. on the discovery of "fundamental rule changes" [19]: 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 timepoints by applying a χ^2-test. In our previous work [3,4], we model rules as temporal objects, which may exhibit changes of statistics *or content* during the observation period, and we focus on surprising changes, such as the disappearance of a rule and the correlated changes of pairs of rules. In [6], we make the distinction between "permanent" rules that are always present (though they may undergo significant changes) and those that appear only temporarily and indicate periodic trends, and discuss methods for identifying them in a *progressive study.*

In this study, we present PAM, an automated pattern monitor, and its theoretical underpinnings. In a case study on the evolution of web usage patterns, we show how the mechanisms implemented by PAM can be used to identify interesting changes in the usage behavior. In particular, we discovered association rules from a web-servers log that show which pages tend to be visited within the same user session. These patterns have been imported into the monitor, and their evolution throughout a period of 8 months has been analyzed.

In the following section, we will introduce the theoretical framework PAM is based upon and its architecture. In Sec. 3 our experimental results are summarized. Sec. 4 concludes our study.

2 Theoretical Framework

In [5], we introduce the theoretical foundations of PAM which are briefly described in the following. PAM builds on a temporal rule model which consists of a rules' content, i.e., the relationship in the data the pattern reflects, and the statistical measurements captured for the rule.

2.1 Temporal Rule Model

As basis for the temporal representation of patterns the *Generic Rule Model* (GRM) is used [3]. According to the GRM, a rule R is a temporal object with the following signature:

$$R = ((ID, query, body, head), \{(timestamp, statistics)\})$$

In this signature, ID is a system generated identifier, which ensures that all rules with the same body (antecedent) and head (consequent) have the same ID. It is used to identify a rule non-ambiguously throughout its entire lifetime. The *query* is the data mining query or similar specification of values for the mining parameters. Note that *query* and ID are invariant across the time axis. Contrary to it, the statistics may vary between two timestamps. The statistics depend on the rule type: We currently consider the support, confidence and certainty factor of association rules. A detailed discussion of the components of the rule signature can be found in [3].

2.2 Detecting Significant Pattern Changes

A mechanism that identifies changes to a rules statistic which exhibit a particular strength we denote as *change detector*. We use the notion of statistical significance to assess the strength of pattern changes. In particular, we use a two-sided binomial test to verify whether an observed change is statistically significant or not.

For a pattern ξ and a statistical measure s at a time point t_i it is tested whether $\xi.s(t_{i-1}) = \xi.s(t_i)$ 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 pattern ξ an alert is raised for each time point t_i at which the null hypothesis is rejected.

Example. Let $sup(t_{i-1}) = 0.1825$ be the support of pattern ξ at time point t_{i-1}, i.e. over dataset D_{i-1}. Let the number of sessions in D_i that support the pattern be 608 (*successes*) upon a total of 2914 sessions in D_i. We test the null hypothesis H_0 that the support of ξ has not changed significantly:

$$H_0: \quad \xi.sup_{i-1} = \xi.sup_i$$
$$H_1: \quad \xi.sup_{i-1} \neq \xi.sup_i$$

At $\alpha = 0.01$, the confidence interval ranges from 0.1896 to 0.2287. Since the true support value at time point t_{i-1} is smaller than the lower boundary of the confidence interval we reject H_0 and state that the support has changed significantly from time point t_{i-1} to time point t_i.

These tests are applied to the set of all patterns that appear in a given period. All significant pattern changes are additionally checked for their temporal dimension, i.e., whether they are only of temporary nature. We differentiate between two cases, (a) the value of the statistic returns immediately to its previous level, and (b) the value remains stable at the new level for at least one more period. For this purpose, we again use the binomial test and check if there is a significant change in period t_{i+1} annulling the change in period t_i. If so, the second change is not reported to the user because it only represents the return of the measure to its actual level. Instead, the significant pattern change is marked as a *core alert* which may be used as an indicator for a beginning concept drift.

2.3 Heuristics for Detecting Interesting Pattern Changes

As opposed to the change detector, the heuristics are used to track changes to the statistics of patterns starting at the time point at which they have emerged for the first time, even if they are not continuously in the rule base. Therefore, they can reveal potentially interesting changes also for those patterns that do not satisfy the thresholds given in the mining query in a particular period. Since the change detector is only aware of patterns that are present in the rule base, this property may be of particular interest to the analyst.

Occurrence-based Grouping. Patterns observed in a particular period reflect the properties of the underlying dataset within this specific

time interval. On the other hand, patterns that are present in each period reflect (part of) the invariant properties of the population. If such patterns change this may be of particular interest to the user.

We, therefore, group rules with respect to their *stability* over time and use the term *occurrence* to denote the proportion of periods in which the rule is present, i.e., the percentage of time points in which its statistics exceeded the threshold values specified in the mining query. In particular, let f be the frequency of appearance of a pattern, defined as the ratio of time points at which the observed statistic measure exceeds the threshold specified by the domain expert. The range of f is $[0, 1]$, which we partition into the intervals $I_L = [0, 0.5)$, $I_M := [0.5, 0.75)$, $I_H := [0.75, 0.9)$, $I_{H+} := [0.9, 1)$ and $I_{permanent} := [1, 1]$. Alternatively, $I_{permanent}$ can be set to $[0.9, 1]$ for large values of n, i.e. for a large number of discrete time points. Then, we label a pattern ξ as L, M, H, $H+$ or *permanent* according to the interval at which $f(\xi)$ belongs. Patterns labeled as H or $H+$ are characterized as *frequent* patterns, whereas L and M form the set of *temporary* patterns.

Then, for a pattern ξ an alert is raised for each time point t_i at which the pattern changes the group it belongs to. Certainly, in order to assess the reliability of patterns, a sufficiently long training phase has to be passed through before meaningful group changes can be observed; in the short run, this approach will be very sensitive to patterns that vanish or emerge.

As already mentioned above, one important peculiarity of this approach is that it can identify many significant rule changes which cannot be observed by significance tests. This is caused by the fact that this method will raise an alert when a rule appears/vanishes in/from the rule base. In such cases, changes to the statistics of a rule will usually be significant. However, if a pattern disappears from the rule base its time series is interrupted and a significance test cannot be applied.[1]

Corridor-based Heuristic. For this heuristic, we define a *corridor* around the time series of a pattern. A corridor is an interval of values which is dynamically adjusted at each time point to reflect the range of values encountered so far. In particular, for a pattern ξ and a statistic measure s, we compute the mean m_i and standard deviation $stddev_i$ of the values $\{\xi.s(t_j)|j = 0, \ldots, i\}$. The corridor at time point t_i is defined as the interval $I(t_i) := [m_i - stddev_i, m_i + stddev_i]$, having a width of one

[1] One possible way to bypass this problem is to use the pattern monitor proposed in [6] which computes the statistics of a rule directly from the underlying dataset.

standard deviation in each direction of the mean. Then, for pattern ξ an alert is raised for each time point t_i at which the value of the time series is outside the corridor $I(t_i)$.

The corridor-based heuristic takes account of the values already encountered for a given pattern. It is insensitive to oscillations of the time series around the threshold value used to discover rules. However, it is sensitive to changes that depart from past values but still remain in the interval, and it can only be defined reasonably for late time points: at time point t_1, a pattern change is most likely to be signaled because the mean and the standard deviation are not well-defined. Thus, the corridor-based heuristic is more appropriate for a retrospective study of the data; for a progressive study, a sufficient number of mining sessions must be performed first. As in the case of the occurrence-based grouping of patterns, this approach may also identify significant changes which cannot be covered by significance tests because corridor violations are tracked starting in the first period a pattern is visible. Therefore, an alert may also be raised when the pattern has disappeared from the rule base. However, such changes may be important for the user, e.g. if the reason for the disappearance is of interest.

Interval-Based Heuristic. For this heuristic, we partition the range of values of the time series into *intervals* of equal width. In particular, we consider the interval $[\tau, M]$, where M is the maximum permissible value per definition of the statistical measure under observation (e.g. support), while τ can be either a threshold provided by the application expert or the minimum permissible value per definition of the statistical measure. This range is partitioned into k equal subintervals. Then, for pattern ξ an alert is raised for each time point t_i at which the value of the time series is in a different interval than for t_{i-1}.

Example. Consider a time series on rule support and a pattern ξ, whose time series on support we denote as $\xi.sup(t_i), i = 0, \ldots, n$. Further, assume that $k = 4$, i.e. the range should be split into four intervals. Then, the range is $[\tau_{sup}, 1]$, where τ_{sup} is the support threshold specified in association rules' discovery.[2] For $\tau_{sup} = 0.2$, we would have four intervals, namely $I_1 = [0.2, 0.4)$, $I_2 = [0.4, 0.6)$, $I_3 = [0.6, 0.8)$ and $I_4 = [0.8, 1]$. A pattern change is signaled for ξ at each $t_i, i \geq 1$ such that

$$\xi.sup(t_i) \in I_j \wedge \xi.sup(t_{i-1}) \in I_{j'} \wedge I_j \neq I_{j'}.$$

[2] This threshold is part of the *query* in the signature of a rule.

2.4 Identifying Atomic Changes

The change detector returns at each time point t_i the set of all patterns, whose observed statistic measure has changed with respect to the previous period. Normally, this set will be large, and some of the patterns may be correlated because they overlap in content. In such cases, it is likely that their changes are due to the same drift in the population. Therefore, we try to identify a minimal set of patterns that caused all change. For this purpose, we consider the components of each pattern, assuming that if a pattern change has occurred, it may be traced back to changes of the statistics of its components. We use the term *atomic change* for a change in a pattern which has no components that has itself experienced a change.

According to a rule's signature in Sec. 2.1, a rule has a body and a head. We observe them together as components of a pattern ξ that corresponds to the rule's itemset. If a pattern change on ξ is reported at time point t_i, it may be due to a change of one or more of its components. Therefore, at time point t_i we consider all combinations of the elements $e_1, \ldots, e_{length(\xi)}$ constituting ξ, i.e. $c(\xi) := \sum_{j=1}^{length(\xi)-1} \binom{length(\xi)}{j}$ components, excluding ξ itself. The following algorithm is used to attribute support changes of a rule to the support changes of its components:

> Let d be the detector used to raise alerts for changes in patterns.
> Let Ξ be the collection of patterns, for which an alert has been raised by d.
> We order the patterns in Ξ by length, keeping longer patterns first. Then, for each $\xi \in \Xi$
>
> 1. For each component ζ of ξ with length $length(\xi) - 1$:
> 2. If ζ is in Ξ
>   then we mark ξ as "removed" and break the loop
>   else we continue with the next component.

Undeleted patterns in Ξ are presented to the application expert. The algorithm assumes the base characteristic of association rules, namely that if a rule is frequent, then all its components are also frequent. Hence, for each pattern in the rule base produced at each time point t_i, all its sub patterns are also in the rule base. Since the detector considers all time series, any pattern that has experienced a change at t_i is placed in Ξ, independently of its components. Thus, if a component of a pattern ξ is found in Ξ, we attribute the change of ξ to it and ignore ξ thereafter. Since the algorithm moves progressively towards smaller patterns, a component

that was found to be responsible for a change in a pattern may itself be removed.

2.5 Architecture of PAM

Technically spoken, PAM encapsulates one or more data mining algorithms and a database back-end that stores data and mining results. The general structure of a PAM instance is depicted in Fig. 1. The core of

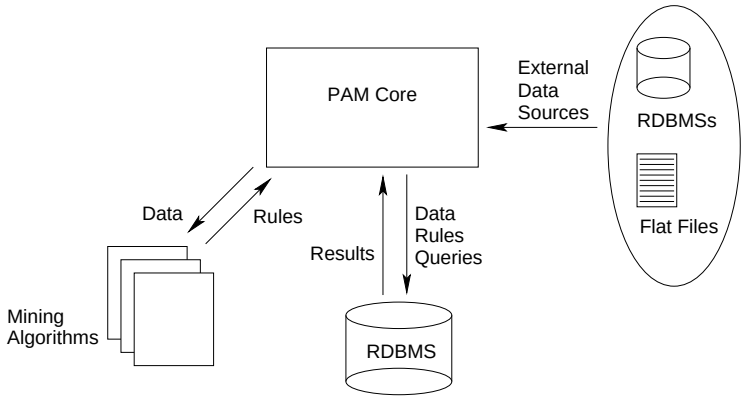

Figure 1. General structure of a PAM instance.

PAM implements the change detectors and heuristics described in the previous paragraphs. When incorporating new data, e.g. the transactions of a new period, these algorithms are used to identify interesting pattern changes. The core offers interfaces to external data sources like databases or flat files. In the database back-end not only the rules discovered by the different mining algorithms are stored, but also the (transaction) data. For example, in order to conduct significance tests it is sometimes necessary to access the base data. Mining results are stored according to the GRM which has been transformed into a relational schema. At present, any mining algorithm implemented in the Java programming language can be used within PAM.

3 Experiments

For the experiments, we used the log file of a web-server hosting a non-commercial web site, spanning a period of 8 months in total. All pages

on the server have been mapped to a concept hierarchy reflecting the purpose of the respective page. The sessionized log file has been splitted on a monthly basis, and association rules showing the different concepts accessed within the same user session have been discovered. In the mining step, we applied the association rule miner of the WEKA tools [25] using minimum support of 2.5% and minimum confidence of 80%.

Tab. 1 gives a general overview on the evolution of the number of page accesses, sessions, frequent single items and rules found in the respective periods. The last three columns give the number of rules that have emerged, remained in the rule base, or disappeared from one period to the next. The first period corresponds to the month October. It can

period	# accesses	# sessions	# freq. items	# rules	# new	# old	# disapp.
1	8335	2547	20	22	22	–	–
2	9012	2600	20	39	27	12	10
3	6008	1799	20	26	13	13	26
4	4188	1222	21	24	12	12	14
5	9488	2914	20	14	1	13	11
6	8927	2736	20	15	6	9	5
7	7401	2282	20	13	5	8	7
8	9210	3014	20	11	3	8	5

Table 1. General overview on the dataset.

be seen that in periods 3 and 4 (December and January) the site was visited less frequently than in other periods, although the number of discovered rules remains comparably high. The number of frequent single items is rather invariant, whereas the number of rules falls, especially in the second half of the analysis.

In order to assess the stability of the rules found, we grouped them according to their occurrence, i.e., the share of periods in which they were present (Tab. 2). In total, there were 58 distinct rules but only 11 rules are frequent according to the definition in Sec. 2.3, i.e. were present in at least 6 periods. Due to the large proportion of rules which appear only once or twice, there are strong changes to the mining results even for adjacent periods, especially in the first half of the analysis (cf. Tab. 1).

3.1 Applying Significance Tests and Heuristics

All experiments are solely based on the support of rules, i.e., in order to determine and assess rule changes only the support of a pattern was

# periods present	# rules	share
8	3	100.0
7	4	87.5
6	4	75.0
5	2	62.5
4	2	50.0
3	6	37.5
2	15	25.0
1	22	12.5

Table 2. Rules grouped by occurrence.

analyzed. However, when analyzing changes to the confidence of a rule the same methodology can be used. As pointed out in Sec. 2.2, we differentiate between short and long-term changes to the statistics of rules. For simplicity, we considered only two different scenarios in the experiments, either the value returns immediately to its previous level, or it remains the same for at least one more period (core alert).

The significance tests were applied to all cases throughout the entire case study.[3] In total, we observed 164 cases in 8 periods, from which 142 cases were checked for significance.[4] 17 support changes turned out to be also significant, from which 4 changes were core alerts. Tab. 3 shows the results of the significance tests and the results of the occurrence-based grouping and the corridor-based approach.[5] In the second column

approach	# changes	# significant	# core alerts
change detector	142	17	4
occurrence-based grouping	49	12	3
corridor-based heuristic	107	32	6

Table 3. Results of the different approaches.

the number of raised alerts is given. Obviously, the number of alerts is even greater than the number of cases to which the change detector was

[3] The term *case* refers to the appearance of one rule in one period.

[4] Cases from the first period cannot be checked for significance.

[5] For the interval-based heuristic we determined only one interesting change. It turned out that the support values of the rules were too small and that a large number of intervals would have been necessary to get useful results. We therefore decided to skip this approach.

applied. We noticed that the intersection of changes identified by one of the heuristics and the total number of cases throughout the analysis amounted only to 33. This complies with our assumption from Sec. 2.3 that the heuristics also reveal interesting changes for patterns that are invisible at the moment. Therefore, the change detector should be used in conjunction with at least one of the heuristics.

As described in Sec. 2.3, the heuristics may be used only after a sufficiently long training period. Due to the limited number of periods, we used the first 4 periods as a training phase and observed group changes only for the remaining periods. Fig. 2 shows examples of applying the occurrence-based grouping of patterns and a time series of the corridor-based heuristic. In the left figure, the horizontal lines at 0.5, 0.75 and

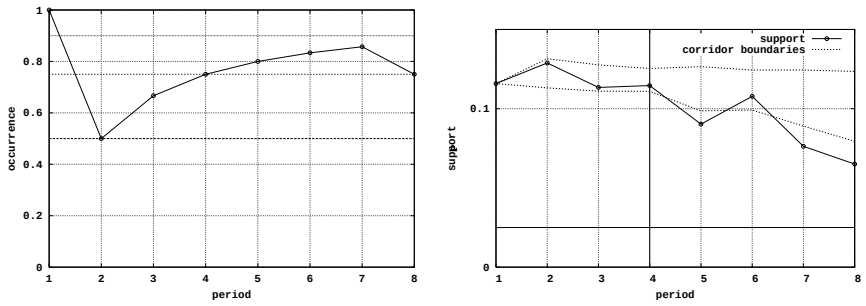

Figure 2. Examples of occurrences and corridors.

0.9 represent the borders of the different groups of relative occurrence. In the first period the pattern is present (occurrence = 1), in the second period the pattern disappears and the occurrence drops to 0.5. However, since the training phase is not yet finished we do not observe a group change. In the remaining periods, the pattern is present and the occurrence grows, except for period 8 in which the pattern again disappears. In the other figure, it is shown how the corridor reacts on the changing support. In periods 5 and 7 we observe corridor violations. Although the support value for the last period is also outside the corridor no alert is raised. Instead, the alert in period 7 is marked as a core alert which may signal a beginning concept drift.

In summary, there were basically two different results: on the one hand, we had a small number of permanent patterns which changed only slightly throughout the analysis. On the other hand, there were many

temporary patterns, especially in the first half of the analysis, which changed almost permanently. For example, it turned out that rules which were present in only 2 periods were responsible for 31 corridor violations, whereas the permanent patterns caused only 8 violations.

3.2 Detecting Atomic Changes

For the results of the change detector and the heuristics we decomposed the rules and checked their itemsets for significant changes. Tab. 4 summarizes the results of this analysis. Again, the number of (significant)

approach	# changes	# significant
change detector	59	28
occurrence-based grouping	50	21
corridor-based heuristic	143	73
corridor & interval	156	77

Table 4. Number of significant itemset changes.

changes is much larger for the heuristics. Interestingly, in this case it seems sufficient to use only the corridor approach since it reveals almost all significant itemset changes. It can be seen that, no matter which approach is considered, only about 50% of the itemsets in a pattern that changed significantly also show significant changes.

4 Conclusions

In this article, we introduced PAM, an automated pattern monitor, which was used to observe changes in web usage patterns. PAM is based on a temporal rule model which consists of both the content of a pattern and the statistical measures captured for the pattern. Moreover, we described heuristics to identify not only significant but also interesting rule changes which take different aspects of pattern reliability into account. We argue that concept drift as the initiator of pattern change often manifests itself gradually over a long period of time where each of the changes may not be significant at all. Additionally, if a pattern disappears from the rule base it is important to know why it disappeared. We presented PAM in a case study on web usage mining, in which association rules were discovered that show which types of web pages tend to be viewed in the same user

session. Our results show that PAM reveals interesting insights into the evolution of the usage patterns

Challenging directions for future work include the application of the methodology to the specific needs of streaming data, and the identification of interdependencies between rules from different data partitions. In this scenario, each partition constitutes a transaction in each of which a number of items (rules) appear. These *transactions* can then be analyzed to discover periodicities of pattern occurrence and/or relationships between specific rules.

Acknowledgements: Many thanks to Gerrit Riessen for his helpful advice.

References

1. R. Agrawal, G. Psaila, E. L. Wimmers, and M. Zaït. Querying shapes of histories. In *VLDB*, Zürich, Switzerland, 1995.
2. N. F. Ayan, A. U. Tansel, and E. Arkun. An Efficient Algorithm To Update Large Itemsets With Early Pruning. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 287–291, San Diego, CA, USA, August 1999. ACM.
3. S. Baron and M. Spiliopoulou. Monitoring change in mining results. In *3rd Int. Conf. on Data Warehousing and Knowledge Discovery - DaWaK 2001*, Munich, Germany, Sept. 2001.
4. S. Baron and M. Spiliopoulou. Monitoring the Results of the KDD Process: An Overview of Pattern Evolution. In J. M. Meij, editor, *Dealing with the Data Flood: Mining data, text and+multimedia*, chapter 6, pages 845–863. STT Netherlands Study Center for Technology Trends, The Hague, Netherlands, Apr. 2002.
5. S. Baron and M. Spiliopoulou. Detecting Long-Term Changes of Patterns over Accumulating Datasets. Submitted to *IEEE International Conference on Data Mining (ICDM'03)*, 2003.
6. S. Baron, M. Spiliopoulou, and O. Günther. Efficient Monitoring of Patterns in Data Mining Environments. In *Seventh East-European Conference on Advance in Databases and Information Systems (ADBIS'03)*, Dresden, Germany, Sep. 2003. Springer. To appear.
7. M. J. Berry and G. Linoff. *Data Mining Techniques: For Marketing, Sales and Customer Support*. John Wiley & Sons, Inc., 1997.
8. Y. M. Bing Liu and R. Lee. Analyzing the interestingness of association rules from the temporal dimension. In *IEEE International Conference on Data Mining (ICDM-2001)*, pages 377–384, Silicon Valley, USA, November 2001.
9. S. Chakrabarti, S. Sarawagi, and B. Dom. Mining surprising patterns using temporal description length. In A. Gupta, O. Shmueli, and J. Widom, editors, *VLDB'98*, pages 606–617, New York City, NY, Aug. 1998. Morgan Kaufmann.
10. X. Chen and I. Petrounias. Mining Temporal Features in Association Rules. In *Proceedings of the 3rd European Conference on Principles of Data Mining and Knowledge Discovery*, Lecture Notes in Computer Science, pages 295–300, Prague, Czech Republic, September 1999. Springer.

11. D. W. Cheung, S. Lee, and B. Kao. A general incremental technique for maintaining discovered association rules. In *DASFAA'97*, Melbourne, Australia, Apr. 1997.
12. M. Ester, H.-P. Kriegel, J. Sander, M. Wimmer, and X. Xu. Incremental Clustering for Mining in a Data Warehousing Environment. In *Proceedings of the 24th International Conference on Very Large Data Bases*, pages 323–333, New York City, New York, USA, August 1998. Morgan Kaufmann.
13. A. Freitas. On objective measures of rule surprisingness. In *PKDD'98*, number 1510 in LNAI, pages 1–9, Nantes, France, Sep. 1998. Springer-Verlag.
14. P. Gago and C. Bento. A Metric for Selection of the Most Promising Rules. In J. M. Zytkow and M. Quafafou, editors, *Principles of Data Mining and Knowledge Discovery, Proceedings of the Second European Symposium, PKDD'98*, Nantes, France, 1998. Springer.
15. V. Ganti, J. Gehrke, and R. Ramakrishnan. A Framework for Measuring Changes in Data Characteristics. In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 126–137, Philadelphia, Pennsylvania, May 1999. ACM Press.
16. V. Ganti, J. Gehrke, and R. Ramakrishnan. DEMON: Mining and Monitoring Evolving Data. In *Proceedings of the 15th International Conference on Data Engineering*, pages 439–448, San Diego, California, USA, February 2000. IEEE Computer Society.
17. S. Lee, D. Cheung, and B. Kao. Is Sampling Useful in Data Mining? A Case in the Maintenance of Discovered Association Rules. *Data Mining and Knowledge Discovery*, 2(3):233–262, September 1998.
18. S. D. Lee and D. W.-L. Cheung. Maintenance of Discovered Association Rules: When to update? In *ACM-SIGMOD Workshop on Data Mining and Knowledge Discovery (DMKD-97)*, Tucson, Arizona, May 1997.
19. B. Liu, W. Hsu, and Y. Ma. Discovering the set of fundamental rule changes. In *7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001)*, pages 335–340, San Francisco, USA, August 2001.
20. N. M. A. Mark G. Kelly, David J. Hand. The Impact of Changing Populations on Classifier Performance. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 367–371, San Diego, Aug. 1999. ACM.
21. E. Omiecinski and A. Savasere. Efficient Mining of Association Rules in Large Databases. In *Proceedings of the British National Conference on Databases*, pages 49–63, 1998.
22. P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Interestingness Measure for Association Patterns. In *Proc. of the Eighth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 2002.
23. S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka. An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD-97)*, pages 263–266, Newport Beach, California, USA, Aug. 1997.
24. K. Wang. Discovering patterns from large and dynamic sequential data. *Intelligent Information Systems*, 9:8–33, 1997.
25. I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 1999.