Identifizierung kurz- und langfristiger Musteränderungen in Transaktionsdaten

Steffan Baron
Institut für Wirtschaftsinformatik,
Humboldt-Universität zu Berlin,
Spandauer Str. 1, D-10178 Berlin
sbaron@wiwi.hu-berlin.de

Abstract

Entscheidungsunterstützung ist ein kontinuierlicher Prozeß, der neben den Auswirkungen der eigenen Entscheidungen auch die Einflüsse externer Faktoren berücksichtigen muß. Da diese in den meisten Fällen nicht direkt, d.h. über ein separates Attribut, in den Daten enthalten sind, sind sie nur indirekt über die zeitliche Dimension ihres Einflusses zu identifizieren. Soll versucht werden, Musteränderungen ihren auslösenden Faktoren zuzuordnen, ist es deswegen wichtig, daß die temporale Komponente der Ergebnisse nicht vernachlässigt wird.

Diese Arbeit soll einen Überblick darüber geben, welche Probleme zu berücksichtigen sind, wenn die temporalen Eigenschaften von Mustern in die Analyse einbezogen werden sollen. Außerdem werden die im Rahmen des *Pattern Monitor* PAM implementierten Lösungen zur Identifizierung kurz- und langfristiger Musteränderungen diskutiert.

1 Einleitung

In den vergangenen Jahren ist die Anzahl und der Umfang elektronisch gesammelter Datensätze enorm angestiegen, und die Entwicklung effizienter Methoden zur Entdeckung nützlichen und verwertbaren Wissens in den Daten ist zu einer großen Herausforderung geworden. Während dabei traditionell Kriterien wie Performanz und Skalierbarkeit im Vordergrund standen, wurde in jüngerer Zeit auch den dynamischen Eigenschaften der untersuchten Daten Rechnung getragen, indem Techniken entwickelt wurden, die sich der Erhaltung und Aktualisierung einmal entdeckten Wissens widmen und häufig als *inkrementelle Mining-Techniken* bezeichnet werden [Ayan *et al.*, 1999; Cheung *et al.*, 1996; Thomas *et al.*, 1997; Wang, 1997]. Schwerpunkt dieser Arbeiten ist die effiziente Aktualisierung des Inhalts der entdeckten Muster im Fall von Modifikationen des ursprünglichen Datensatzes. Andere Arbeiten betrachten die Ähnlichkeit von Mustern und ihre statistischen Eigenschaften [Ganti *et al.*, 1999; Chen and Petrounias, 1999; Ganti *et al.*, 2000].[1]

[1] Eine detaillierte Diskussion dieser Arbeiten ist in [Baron and Spiliopoulou, 2002] zu finden.

Ist der Anwendungsexperte erstmal mit den Zusammenhängen, die die Daten reflektieren, vertraut, liegt jedoch sein primäres Interesse in der Identifizierung von Änderungen, denen diese Zusammenhänge unterliegen [Chakrabarti *et al.*, 1998]. In [Baron *et al.*, 2003] wurde der Versuch unternommen, interessante Musteränderungen mit möglichst geringem Aufwand zu erkennen. Basierend auf einem Regelmodell, das neben dem eigentlichen Muster auch die Entwicklung seiner statistischen Eigenschaften erfaßt [Baron and Spiliopoulou, 2001], wurde ein *Pattern Monitor* entwickelt, der grundlegende Änderungen in der Regelbasis anhand der Änderungen einer ausgewählten Menge von Mustern erkennen konnte. Die Auswahl der Muster kann dabei aufgrund verschiedener Kriterien wie z.B. subjektives oder objektives Interesse des Anwendungsexperten erfolgen [Tan *et al.*, 2002; Freitas, 1998; Gago and Bento, 1998], oder aufgrund ihrer Eigenschaft, grundlegende Zusammenhänge im untersuchten Datensatz zu repräsentieren [Baron *et al.*, 2003]. Für den Anwendungsexperten ist es jedoch häufig von besonderem Interesse, die zeitliche Dimension der entdeckten Regeländerungen zu bestimmen – sind sie nur kurzfristiger Natur, d.h. handelt es sich um kurzfristige Abweichungen von der *Norm*, oder spiegeln sie eher einen langfristigen Trend wider, der auf grundlegende Änderungen in der Population zurückzuführen ist.

2 Problemstellungen und Lösungsansätze

Ein wichtiges Problem, das vorab gelöst werden muß, stellt die Bewertung der auftretenden Regeländerungen dar. Aufgrund der Vielzahl der zu erwartenden Musteränderungen muß eine geeignete Vorauswahl getroffen werden, um dem Anwendungsexperten nur die *interessanten* Musteränderungen zu präsentieren. Eine Möglichkeit besteht darin, Grenzen für die absolute und/oder relative Änderung festzulegen, deren Über- bzw. Unterschreitung auf interessante Regeländerungen hindeutet. Eine Variante dieses Verfahrens bezieht die Historie der statistischen Eigenschaften ein: Aufgrund der Erfahrungen aus vergangenen Perioden werden Korridore für die erfaßten statistischen Maßzahlen ermittelt, welche tolerierbare Änderungen repräsentieren [Baron and Spiliopoulou, 2003]. Gemäß dieser Vorgehensweise würden dem Anwendungsexperten nur jene Änderungen gemeldet, die die Grenzen des Korridors verletzen. Eine weitere Möglichkeit besteht darin, die Signifikanz einer Änderung durch statistische Testverfahren zu bestimmen, z.B. unter Verwendung des χ^2-Tests [Bing Liu and

Lee, 2001] oder des Binomialtests [Baron and Spiliopoulou, 2003]. Diese Verfahren sind jedoch nicht in der Lage, zwischen kurz- und langfristigen Änderungen zu unterscheiden. Langfristige Änderungen manifestieren sich häufig erst über einen langen Zeitraum – jede Änderung für sich genommen muß dabei nicht notwendigerweise auch signifikant im statistischen Sinne sein. Ein weiterer Vorschlag, die Anzahl der propagierten Regeländerungen zu vermindern, wurde in [Liu *et al.*, 2001] vorgestellt. Hier besteht die grundlegende Idee darin, Regeländerungen auf die Änderungen anderer Regeln zurückzuführen. Eine Regeländerung ist dabei nur dann von Interesse, wenn sie durch keine andere Regeländerung erklärt werden kann. Aber auch dieser Ansatz kann das eigentliche Problem der Beurteilung der zeitlichen Dimension der Regeländerung nicht lösen.

Ein weiteres Problem stellt die *Abwesenheit* von Regeln dar: Ein Muster, das über eine Anzahl von Perioden $t_1, \ldots t_n$ hinweg präsent war, könnte in Periode t_{n+1} zu wenig Support und/oder Konfidenz aufweisen, um die Anforderungen des Nutzers zu erfüllen. In diesem Fall wäre die Anwendung von Werkzeugen der Trend-Analyse nicht möglich, da die Zeitreihen der statistischen Eigenschaften unterbrochen wären. Eine mögliche Lösung für dieses Problem böte die Verwendung eines Pattern Monitor, wie er in [Baron *et al.*, 2003] vorgeschlagen wurde. Dieser Monitor verwendet die der Analyse zugrundeliegenden Daten direkt, um die Werte der statistischen Maßzahlen auch für jene Regeln zu bestimmen, die gemäß den in der Mining-Anfrage verwendeten Schwellen für z.B. Support und Konfidenz keine gültigen Muster darstellen.

3 Der Pattern-Monitor PAM

Im Gegensatz zu den in Abschnitt 1 erwähnten inkrementellen Mining-Techniken betrachten wir die gesammelten Daten nicht als homogene Einheit, sondern gehen davon aus, daß sich die Struktur und/oder die Zusammensetzung des Datensatzes im Zeitverlauf ändern können. Daten werden über einen potentiell langen Zeitraum $t_0, t_1, \ldots t_n$ gesammelt und zu bestimmten Zeitpunkten t_i analysiert, wobei sich die Analyse zum Zeitpunkt t_i auf die Daten D_i erstreckt, die im Zeitraum $t_i - t_{i-1}$ gesammelt wurden.[2]

3.1 Das temporale Regel-Modell

Jede Mining-Session liefert eine Menge von Mustern, die PAM unter Verwendung eines temporalen Regelmodells speichert. Das *Generic Rule Model* (GRM) umfaßt dabei sowohl den Inhalt der Regel, d.h. den Zusammenhang in den Daten, den das Muster beschreibt, als auch seine statistischen Eigenschaften [Baron and Spiliopoulou, 2001]. Gemäß dem GRM hat eine Regel R die folgende Signatur:

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

In diesem Modell ist ID eine automatisch generierte Kennung, die sicherstellt, daß alle Regeln, für die $body$ und $head$ gleich sind, auch die gleiche Kennung haben. Sie wird verwendet, um ein Muster über den gesamten Analysezeitraum hinweg eindeutig zu identifizieren. In $query$ ist die Mining-Spezifikation gespeichert, mit der das Muster entdeckt wurde, und in $body$ und $head$ der Zusammenhang, den das Muster beschreibt. Während alle bisher beschriebenen Bestandteile des Modells invariant sind, können sich die in $stats$ gespeicherten statistischen Eigenschaften des Musters im Zeitverlauf ändern. Außerdem wird in jeder Mining-Session ein Zeitstempel $timestamp$ erfaßt, der die erfaßten Statistiken bestimmten Zeitpunkten zuordnet.

3.2 Identifizierung interessanter Änderungen

Für die Identifikation interessanter Änderungen wurden eine Reihe von Heuristiken entwickelt, die jeweils unterschiedliche Aspekte der *Stabilität* von Mustern betrachten [Baron and Spiliopoulou, 2003].

Signifikanztests

Ein Weg, das potentielle Interesse des Nutzers bestimmten Musteränderungen gegenüber einzuschätzen, ist die Verwendung des Konzepts der statistischen Signifikanz. Demnach sind bspw. all jene Änderungen des Support interessant, die signifikant vom Wert der Vorperiode abweichen.[3] Wir benutzen einen zweiseitigen Binomialtest (Test auf Anteilswert) um festzustellen, ob für ein Muster ξ der Wert einer statistischen Eigenschaft s zum Zeitpunkt t_i für ein gegebenes Konfidenzniveau α signifikant vom Wert der Vorperiode abweicht:

$$\begin{aligned} H_0: &\quad \xi.s_{i-1} = \xi.s_i \\ H_1: &\quad \xi.s_{i-1} \neq \xi.s_i \end{aligned}$$

Demnach wird jede Musteränderung als interessant betrachtet, für die die Null-Hypothese verworfen werden muß. Für jede interessante Änderung wird zusätzlich geprüft, welche zeitliche Dimension die Änderung aufweist. Zur Vereinfachung unterscheiden wir zunächst nur zwei Fälle, entweder die Änderung ist nur temporär, und der Wert kehrt in der Folgeperiode zum ursprünglichen Niveau zurück, oder es handelt sich um eine dauerhafte Änderung, die für mindestens m Perioden anhält, wobei m ein Parameter ist, der an den jeweiligen Anwendungskontext angepaßt werden kann.

Dieser Test wird für alle in einer bestimmten Periode entdeckten Muster durchgeführt, sofern die Werte der statistischen Eigenschaften aus der Vorperiode tatsächlich vorliegen. Liegen nicht alle Werte vor, kann ein Mechanismus genutzt werden wie er in [Baron *et al.*, 2003] vorgeschlagen wurde. Der in dieser Arbeit vorgestellte Monitor arbeitete direkt auf den analysierten Daten, wobei für jedes beobachtete Muster eine Menge von SQL-Anfragen generiert wurde, die die Statistiken für das Muster direkt bestimmen. Wird ein solcher Ansatz durchgängig verwendet, können auch konventionelle Werkzeuge der Trend-Analyse eingesetzt werden.

Stabilitätsbasierte Gruppierung

Muster, die in einer bestimmten Periode erscheinen, widerspiegeln die Eigenschaften des Datensatzes in dieser Periode, während Muster, die in allen Perioden präsent sind, Teile der invarianten Eigenschaften der Daten repräsentieren.

[2] Aufgrund der Homogenitätsannahme analysieren inkrementelle Techniken im Zeitpunkt t_i die gesamten bis zum Zeitpunkt t_i gesammelten Daten.

[3] Obwohl es sich um ein objektives Maß handelt, verwenden wir den Begriff Heuristik, da bei weitem nicht alle interessanten Änderungen mit Hilfe von Signifikanztests bestimmt werden können.

Deswegen gruppieren wir Muster anhand der *Stabilität*, die sie im Zeitverlauf zeigen. Stabilität wird dabei als der Anteil der Perioden bestimmt wird, in denen ein gegebenes Muster sichtbar war, d.h. die in der Mining-Anfrage angegeben Schwellen für z.B. Support und Konfidenz überschritten hat, bezogen auf die Gesamtanzahl der Perioden. Muster, die eine vergleichbare Stabilität zeigen, werden zu Gruppen zusammengefaßt.

Sei f die Frequenz, mit der ein Muster präsent ist, und $[0, 1]$ der zulässige Bereich von f, den wir in die Intervalle $I_L = [0, 0.5)$, $I_M := [0.5, 0.75)$, $I_H := [0.75, 0.9)$, $I_{H+} := [0.9, 1)$ und $I_{permanent} := [1, 1]$ einteilen. Für eine große Anzahl an Perioden kann $I_{permanent}$ alternativ auch auf $[0.9, 1]$ gesetzt werden. Während wir Muster, die zur Gruppe H bzw. $H+$ gehören, als *häufige* Muster bezeichnen, bilden die Gruppen L und M die Menge der *temporären* Muster.

In jeder Periode werden die Muster einer Gruppe zugeordnet und eine interessante Änderung immer dann signalisiert, wenn ein Muster ξ in zwei aufeinanderfolgenden Perioden t_{i-1} und t_i zwei unterschiedlichen Gruppen angehört. Außerdem wird wieder die Frage betrachtet, welche zeitliche Dimension die Änderung aufweist. Dazu wird geprüft, ob das Muster in der darauffolgenden Periode in die ursprüngliche Gruppe zurückwechselt, oder ob es sich um eine dauerhafte Änderung handelt und das Muster für mindestens m Perioden in der neuen Gruppe verbleibt.

Es ist offensichtlich, daß diese Heuristik nur nach einer ausreichenden Anzahl von Trainingsperioden brauchbare Resultate liefern kann, da diese Methode in den ersten Perioden sehr empfindlich auf Änderungen der Regelbasis reagiert. Im Gegensatz zum Signifikanztest wird diese Heuristik für alle Muster ab der ersten Periode ihres Auftretens angewandt. Es ist wahrscheinlich, daß sie weitere interessante Änderungen identifizieren kann, da das Verschwinden bzw. Erscheinen eines Musters häufig mit starken Änderungen der statistischen Eigenschaften des Musters verbunden ist. Diese Änderungen können jedoch vom Signifikanztest nicht ohne weiteres erkannt werden, da das Muster in dieser bzw. der vorhergehenden Periode eventuell nicht in der Regelbasis vorhanden ist bzw. war.

Korridorbasierte Heuristik

Diese Heuristik definiert einen *Korridor* um die Zeitreihen der statistischen Eigenschaften eines Musters. Dieser Korridor entspricht einem Intervall von Werten und wird in jeder Periode angepaßt, so daß er alle bisherigen Werte der betrachteten Eigenschaft einbezieht.

Für ein Muster ξ und eine statistische Eigenschaft s berechnen wir den Durchschnitt m_i und die Standardabweichung $stddev_i$ der Werte $\{\xi.s(t_j)|j = 0, \ldots, i\}$. Der Korridor zum Zeitpunkt t_i ist definiert als das Intervall $I(t_i) := [m_i - stddev_i, m_i + stddev_i]$, mit einer Breite von $2 \times stddev_i$ und dem Mittelpunkt m_i. Eine interessante Änderung wird immer dann signalisiert, wenn für ein Muster ξ der Wert der Zeitreihe außerhalb des Korridors $I(t_i)$ liegt. Für den Zeitpunkt t_{i+1} wird wieder überprüft, ob der Wert der Zeitreihe nur temporär außerhalb des Korridors lag, oder ob er für m weitere Perioden außerhalb liegt. Für diese Heuristik wäre der letztgenannte Fall besonders interessant, weil er impliziert, daß die Werte der Zeitreihe schneller fallen bzw. ansteigen als sich der Korridor anpassen kann.

Während diese Methode unempfindlich gegenüber Schwankungen um den Schwellwert ist, der bei der Mustersuche verwendet wurde, reagiert sie auch auf Änderungen, die zwar von früheren Werten abweichen, jedoch immer noch im Intervall liegen. Natürlich können Mittelwert und Standardabweichung nur für eine hinreichend große Anzahl von Perioden sinnvoll bestimmt werden, d.h. sie ist entweder für eine retrospektive Analyse zu verwenden oder nur nach einer ausreichend langen Trainingsperiode anzuwenden. Aus den gleichen Gründen wie bei der stabilitätsbasierten Gruppierung von Mustern ist auch in diesem Fall mit der Signalisierung interessanter Änderungen zu rechnen, die nicht von Signifikanztests ermittelt werden können. Da die Überwachung des Korridors vom ersten Auftreten des Musters an durchgeführt wird, könnte der Korridor auch in einer Periode verletzt werden, in der das Muster eigentlich nicht existiert. Aber gerade solche Änderungen können für den Nutzer besonders interessant, kann er doch auf diese Weise den Grund für das Verschwinden eines Musters nachvollziehen.

Intervallbasierte Heuristik

Diese Heuristik partitioniert den Wertebereich einer statistischen Eigenschaft eines Musters $[\tau, M]$ in k Intervalle mit gleicher Breite, wobei M der maximal zulässige Wert für die jeweils beobachtete statistische Eigenschaft ist und τ entweder der in der Mining-Anfrage verwendete Schwellwert oder der minimal zulässige Wert für die jeweils beobachtete statistische Eigenschaft. Eine interessante Änderung wird immer dann signalisiert, wenn für ein Muster ξ der Wert der Zeitreihe zum Zeitpunkt t_i in einem anderen Intervall liegt als im Zeitpunkt t_{i-1}. Um Aussagen hinsichtlich der zeitlichen Dimension der Änderung machen zu können, wird überprüft, ob der Wert der Zeitreihe im Zeitpunkt t_{n+1} wieder im ursprünglichen Intervall liegt.

Es ist ersichtlich, daß diese Methode ebenso wie die Signifikanztests bereits ab der zweiten Periode angewandt werden kann. Die Anzahl der Intervalle k ist dabei ein Parameter, der an die spezifischen Bedürfnisse der Anwendung angepaßt werden kann.

3.3 Identifizierung minimaler Änderungen

Die Heuristiken liefern in jeder Periode t_i eine Menge von Mustern, deren statistische Eigenschaften eine *interessante* Änderung erfahren haben. Diese Menge wird im allgemeinen recht groß sein, und es ist möglich, daß verschiedene Muster inhaltlich überlappen. In einem solchen Fall ist es wahrscheinlich, daß die Änderungen der Muster auf die gleichen Änderungen in den Daten zurückzuführen sind. Deshalb versuchen wir, die minimale Menge der Muster zu bestimmen, die für die Änderungen verantwortlich sind. Unter der Annahme, daß wir die Änderung eines Musters auf seine Bestandteile zurückführen können, versuchen wir jene Bestandteile eines Musters zu bestimmen, die sich geändert haben, selbst jedoch keine Bestandteile haben, die von Änderungen betroffen waren.

Zu diesem Zweck betrachten wir $body$ und $head$ einer Regel, die gemeinsam das häufige *Itemset* des Musters bilden. Wird zum Zeitpunkt t_i eine Änderung für das Mu-

ster ξ signalisiert, könnte das auf eine Änderung eines Bestandteils des Musters zurückzuführen sein. Deshalb betrachten wir zum Zeitpunkt t_i alle möglichen Element-Kombinationen $e_1, \ldots, e_{length(\xi)}$ die das Muster ξ ausmachen, d.h. $c(\xi) := \sum_{j=1}^{length(\xi)-1} \binom{length(\xi)}{j}$, wobei wir das ursprüngliche Muster ξ ausschließen.

Wir verwenden den folgenden Algorithmus, um die Support-Änderung des Musters auf Support-Änderungen seiner Bestandteile zurückzuführen:

> Sei h die Heuristik, die zur Identifizierung interessanter Muster-Änderung verwendet wurde.
> Sei Ξ die Menge aller Muster, die gemäß der Heuristik h als interessant identifiziert wurden.
> Erst ordnen wir die Muster in Ξ nach abnehmender Länge und führen dann folgende Schleife aus:
> Für alle $\xi \in \Xi$
>
> 1. Für alle Bestandteile ζ von ξ mit der Länge $length(\xi) - 1$:
> 2. Wenn ζ in Ξ
>   entfernen wir ξ und brechen die Schleife ab, anderenfalls fahren wir mit dem nächsten Bestandteil fort.

Alle nicht gelöschten Bestandteile werden dem Nutzer präsentiert.

Der Algorithmus fußt auf der Grundeigenschaft von Assoziationsregeln, nach der alle Itemsets einer Regel ebenfalls häufig sind. Demzufolge befinden sich die Itemsets eines jeden Musters ebenfalls in der Regelbasis. Da die Heuristiken die Zeitreihen aller statistischen Eigenschaften betrachten, wird jedes Muster ξ, das zum Zeitpunkt t_i interessante Änderungen aufweist, in Ξ aufgenommen, unbeachtlich seiner Bestandteile. Ist in Ξ bereits ein Bestandteil ζ enthalten, der für die Änderung von ξ verantwortlich ist, wird die Änderung von ξ mit der Änderung von ζ begründet und ξ fortan ignoriert. Da der Algorithmus immer kürzere Muster betrachtet, könnte es sein, daß eine Komponente, die für die Änderung eines Musters verantwortlich gemacht wurde, ebenfalls aus Ξ entfernt wird.

3.4 Der PAM-Prototyp

Sämtliche Heuristiken sind im Kern von PAM implementiert (vgl. Abbildung 1). Sind die Transaktionen einer neuen Periode einzubeziehen, werden die Daten zunächst analysiert.

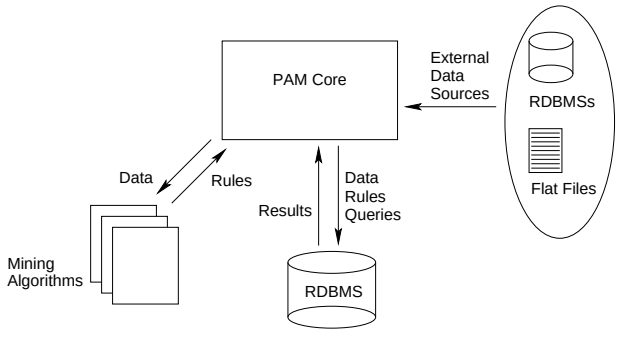

Abbildung 1: Aufbau einer PAM-Instanz.

Zu diesen Zweck bedient sich der Kern externer Mining-Bibliotheken. Zur Zeit werden die WEKA-Tools verwendet [Witten and Frank, 1999], aber prinzipiell kann jeder in Java implementierte Algorithmus genutzt werden. Neben den Mining-Ergebnissen werden auch die Transaktionen in das angeschlossene Datenbanksystem importiert, wo sie dann für weitere Analysen durch die verschiedenen Heuristiken zur Verfügung stehen. Der Kern verfügt außerdem über weitere Schnittstellen für den Datenimport aus externen Quellen und über eine eine Visualisierungskomponente, welche die Ergebnisse der Analysen ausgibt.

4 Fallstudie

Die Eignung der vorgeschlagenen Lösungen wurde unter anderem im Rahmen einer Analyse der Entwicklung von *Web Usage Patterns* untersucht [Baron and Spiliopoulou, 2003].

Für die Analyse wurde das Logfile einer nichtkommerziellen *Website* verwendet, das die Zugriffe auf den *Server* über einen Zeitraum von acht Monaten enthält. Alle Seiten des Servers wurden gemäß ihres Zweckes auf eine Konzepthierarchie abgebildet. Aus den Einträgen im Logfile wurden *Sessions* erstellt und das Logfile auf monatlicher Basis geteilt. Anschließend wurde eine Assoziationsanalyse durchgeführt, bei der ermittelt wurde, welche Seiten (Konzepte) der Website innerhalb der gleichen Session besucht wurden. Hierfür wurde ein Wert von 2,5% als untere Grenze für den Support und ein Wert von 80% als untere Grenze für die Konfidenz verwendet.

Tabelle 1 gibt einen allgemeinen Überblick über die Anzahl der Seitenzugriffe, der Sessions, der häufigen *Items* (Konzepte) und die Anzahl der Regeln in den jeweiligen Perioden. Der erste Monat der Analyse entspricht dem Monat

Periode	Zugriffe	Sessions	Items	Muster
1	8335	2547	20	22
2	9012	2600	20	39
3	6008	1799	20	26
4	4188	1222	21	24
5	9488	2914	20	14
6	8927	2736	20	15
7	7401	2282	20	13
8	9210	3014	20	11

Tabelle 1: Allgemeiner Überblick über den Datensatz.

Oktober. Obwohl die Anzahl der Zugriffe und der Sessions in den Perioden 3 und 4 (Dezember und Januar) stark abnimmt, bleibt die Anzahl der häufigen Konzepte fast konstant, wohingegen die Anzahl der Regeln in der zweiten Hälfte der Analyse merklich abnimmt.

Tabelle 2 zeigt die Entwicklung der Regelbasis in den verschiedenen Perioden. Es ist ersichtlich, daß die Regelbasis in der zweiten Hälfte der Analyse weniger Änderungen ausgesetzt ist.

Um die Stabilität der gefundenen Muster einschätzen zu können, gruppierten wir die Muster nach dem relativen Anteil der Perioden, in denen sie präsent waren. Es zeigte sich, daß nur 11 von insgesamt 58 Regeln häufig waren,

Periode	Muster	neu	alt	vers.
1	22	22	–	–
2	39	27	12	10
3	26	13	13	26
4	24	12	12	14
5	14	1	13	11
6	15	6	9	5
7	13	5	8	7
8	11	3	8	5

Tabelle 2: Entwicklung der Regelbasis.

d.h. in mindestens sechs Perioden vertreten waren (Tabelle 3). Wegen der großen Anzahl der Muster, die nur in ein

Perioden	Muster	Anteil
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

Tabelle 3: Stabilität der Muster.

oder zwei Perioden erscheinen, gab es selbst in aufeinanderfolgenden Perioden starke Änderungen in der Regelbasis, insbesondere in der ersten Hälfte des Analysezeitraums (Tabelle 2).

Tabelle 4 stellt die Ergebnisse der verschiedenen Heuristiken gegenüber.[4] Die Signifikanztests wurden für *alle beobachteten* Musteränderungen durchgeführt. Für die anderen Heuristiken enthält die Spalte „Änderungen“ die Anzahl der als interessant eingestuften Musteränderungen. Die beiden letzten Spalten enthalten die Anzahl der Änderungen, die signifikant waren, und die Anzahl der signifikanten *langfristigen* Änderungen, wobei zur Vereinfachung von einer langfristigen Änderung ausgegangen wurde, wenn der Wert der Zeitreihe nicht unmittelbar zum ursprünglichen Niveau zurückkehrte.[5]

Heuristik	Änderungen	sign.	langfr.
Signifikanztest	142	17	4
stabilitätsbasiert	49	12	3
korridorbasiert	107	32	6

Tabelle 4: Ergebnisse der Heuristiken.

[4] Aufgrund des niedrigen Support-Niveaus konnte unter Verwendung der intervallbasierten Heuristik nur eine einzige interessante Musteränderung identifiziert werden, weswegen nur die anderen Heuristiken betrachtet wurden.

[5] Obwohl sich alle Auswertungen hier nur auf den Support beziehen, kann die gleiche Analyse auch auf Basis der Konfidenz durchgeführt werden.

Es ist auffällig, wie hoch die Anzahl der interessanten Musteränderungen ist, die auch signifikant sind. Die Anzahl der interessanten Änderungen ist sogar größer als die Gesamtzahl der Musteränderungen. Insgesamt waren nur 33 interessante Änderungen, die durch eine der Heuristiken identifiziert wurden, in den 142 Regeländerungen vertreten. Das stützt unsere Annahme, daß die Heuristiken auch für Muster, die aktuell gar nicht in den Ergebnissen vertreten sind, interessante Änderungen erkennen können.

Wie bereits erwähnt kann ein Teil der Heuristiken erst nach einer ausreichenden Anzahl von Trainingsperioden sinnvolle Ergebnisse liefern. Aufgrund der geringen Anzahl an Perioden wurden jedoch Gruppenänderungen bzw. Verletzungen des Korridors bereits ab Periode 5 erfaßt. Abbildung 2 zeigt eine Visualisierung der Anwendung der stabilitätsbasierten Heuristik. Die horizontalen Linien bei

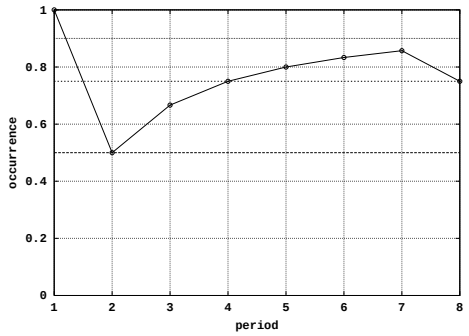

Abbildung 2: Stabilitätsbasierte Heuristik.

0.9, 0.75 und 0.5 repräsentieren die jeweiligen Gruppen-Grenzen. Während das Muster in der ersten Periode erscheint, verschwindet es in der zweiten Periode, und der relative Anteil fällt auf 50%. In den Folgeperioden ist das Muster dann wieder vertreten, während es in der letzten Periode erneut abwesend ist. Nach Ablauf der Trainingsperiode sind keinerlei Gruppenwechsel zu verzeichnen.

Abbildung 3 zeigt ein Beispiel für die Anwendung der korridorbasierten Heuristik. In den Perioden 5 und 7 kommt

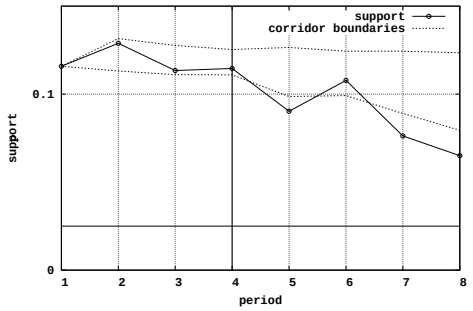

Abbildung 3: Korridorbasierte Heuristik.

es zu Verletzungen des Korridors. Obwohl der Wert der

Zeitreihe auch in Periode 8 außerhalb des Korridors liegt, wird keine interessante Änderung signalisiert. Da der Wert bereits zum zweiten Mal außerhalb des Korridors liegt, wird die Änderung in Periode 7 als langfristig gekennzeichnet, was auf eine grundlegende Änderung im Datensatz hindeuten könnte.

Zusammenfassend wurden zwei wichtige Erkenntnisse gewonnen. Zum einen gab es nur wenige häufige Muster, die im Zeitverlauf nur sehr moderaten Änderungen ausgesetzt waren. Zum anderen gab es insbesondere in der ersten Hälfte der Analyse eine Vielzahl von temporären Mustern, die sich fast ununterbrochen änderten. Es zeigte sich, daß die Regeln, die nur in zwei Perioden vorhanden waren, für 31 Verletzungen des Korridors verantwortlich waren. Im Gegensatz dazu verursachten die *permanenten* Muster gerade acht Verletzungen.

Im letzten Schritt wurde versucht, die interessanten Regeländerungen den verursachenden Bestandteilen zuzuordnen. Tabelle 5 faßt die Ergebnisse dieser Analyse zusammen. Wieder können die Heuristiken, insbesondere die korridorbasierte, sehr viele interessante Änderungen erkennen, die bei einer traditionellen Analyse verborgen geblieben wären. Auffällig an diesem Ergebnis ist einerseits, daß unbeachtlich welche Methode verwendet wurde nur rund die Hälfte der Itemset-Änderungen signifikant sind, und andererseits, daß die korridorbasierte Heuristik allein fast alle (signifikanten) Itemset-Änderungen erkennen kann, wie aus der Kombination der Heuristiken in der letzten Zeile hervorgeht.

Heuristik	Änderungen	signifikant
Signifikanztest	59	28
stabilitätsbasiert	50	21
korridorbasiert	143	73
kombiniert	156	77

Tabelle 5: Signifikante Itemset-Änderungen.

5 Zusammenfassung

In dieser Arbeit wurden einige Probleme vorgestellt, die bei der Einbeziehung der temporalen Eigenschaften von Mustern zu berücksichtigen sind. Für viele der genannten Probleme wurden erste eigene Lösungen vorgestellt, die in PAM, einem Pattern Monitor, implementiert sind. PAM basiert auf einem temporalen Regelmodell, das sowohl den Inhalt eines Musters als auch deren statistische Eigenschaften integriert betrachtet, und das geeignet ist, die Änderungen von Mustern im Zeitverlauf zu erkennen und zu bewerten. Erste Experimente haben gezeigt, daß die Heuristiken zur Erkennung kurz- und langfristiger Musteränderungen geeignet sind, zusätzliche und wertvolle Details über die untersuchten Datensätze und die darin entdeckten Muster zu erkennen.

Derzeit wird an einer Erweiterung von PAM gearbeitet, die die Verarbeitung der Ergebnisse von Cluster-Analysen ermöglicht. Herausforderungen für künftige Arbeiten bietet die Adaption der vorgestellten Techniken auf die spezifischen Anforderungen der Verarbeitung von Datenströmen und die Untersuchung intertemporaler Abhängigkeiten zwischen Mustern.

Literatur

[Ayan *et al.*, 1999] Necip Fazil Ayan, Abdullah Uz Tansel, and Erol 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.

[Baron and Spiliopoulou, 2001] Steffan Baron and Myra Spiliopoulou. Monitoring Change in Mining Results. In *Proceedings of the 3rd International Conference on Data Warehousing and Knowledge Discovery*, Munich, Germany, Sep. 2001. Springer.

[Baron and Spiliopoulou, 2002] Steffan Baron and Myra Spiliopoulou. Monitoring the Results of the KDD Process: An Overview of Pattern Evolution. In Jeroen Meij, editor, *Dealing with the data flood: mining data, text and multimedia*, chapter 6. STT Netherlands Study Center for Technology Trends, The Hague, Netherlands, Apr. 2002.

[Baron and Spiliopoulou, 2003] Steffan Baron and Myra Spiliopoulou. Monitoring the Evolution of Web Usage Patterns. In *1st European Web Mining Forum at ECML/PKDD 2003*, Cavtat-Dubrovnik, Croatia, Sep. 2003.

[Baron *et al.*, 2003] Steffan Baron, Myra Spiliopoulou, and Oliver 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.

[Bing Liu and Lee, 2001] Yiming Ma Bing Liu and Ronnie 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.

[Chakrabarti *et al.*, 1998] Soumen Chakrabarti, Sunita Sarawagi, and Byron Dom. Mining Surprising Patterns Using Temporal Description Length. In Ashish Gupta, Oded Shmueli, and Jennifer Widom, editors, *VLDB'98*, pages 606–617, New York City, NY, Aug. 1998. Morgan Kaufmann.

[Chen and Petrounias, 1999] Xiaodong Chen and Ilias 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.

[Cheung *et al.*, 1996] David W. Cheung, Vincent T. Ng, and Benjamin W. Tam. Maintenance of Discovered Knowledge: A Case in Multi-Level Association Rules. In *KDD'96*, 1996.

[Freitas, 1998] Alex Alves Freitas. On Objective Measures of Rule Surprisingness. In Jan M. Zytkow and Mohamed Quafafou, editors, *Principles of Data Mining and Knowledge Discovery, Proceedings of the Second European Symposium, PKDD'98*, Nantes, France, 1998. Springer.

[Gago and Bento, 1998] Pedro Gago and Carlos Bento. A Metric for Selection of the Most Promising Rules. In Jan M. Zytkow and Mohamed Quafafou, editors, *Principles of Data Mining and Knowledge Discovery, Proceedings of the Second European Symposium, PKDD'98*, Nantes, France, 1998. Springer.

[Ganti *et al.*, 1999] Venkatesh Ganti, Johannes Gehrke, and Raghu 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.

[Ganti *et al.*, 2000] Venkatesh Ganti, Johannes Gehrke, and Raghu 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.

[Liu *et al.*, 2001] Bing Liu, Wynne Hsu, and Yiming 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.

[Tan *et al.*, 2002] Pang-Ning Tan, Vipin Kumar, and Jaideep 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.

[Thomas *et al.*, 1997] Shiby Thomas, Sreenath Bodagala, Khaled Alsabti, and Sanjay 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.

[Wang, 1997] Ke Wang. Discovering Patterns from Large and Dynamic Sequential Data. *Intelligent Information Systems*, 9:8–33, 1997.

[Witten and Frank, 1999] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 1999.