A framework for discovering interesting business changes from data

M Böttcher, D Nauck, C Borgelt and R Kruse

Since the world with its markets, innovations and customers is changing faster than ever before, the key to survival for businesses is the ability to detect, assess and respond to changing conditions — rapidly and intelligently. Discovering changes, and reacting to or acting upon them before others do, has therefore become a strategic issue for many companies.

Many businesses collect huge volumes of data. Commonly this data is continuously gathered over long periods and thus reflects changes in the parts of the business from which it has been derived. To control their business operations and to gain a competitive edge, it is crucial for businesses to detect these changes — early and precisely. However, existing data analysis techniques are insufficient for this task. The widely used method for defining key performance indicators is too weak to detect changes early enough, and requires time-consuming in-depth analysis before decisions can be made. State-of-the-art knowledge discovery techniques, on the other hand, provide the required level of detail, but assume that the domain under consideration is stable over time.

This paper presents a framework that detects changes within a data set at virtually any level of granularity. The underlying idea is to derive a rule-based description of the data set at different points in time and to subsequently analyse how these rules change. While rules are themselves a very comprehensible representation of knowledge, further techniques are required to assist the data analyst in interpreting and assessing changes. Therefore the framework also contains methods to discard rules that are non-drivers for change, and to assess the interestingness of the detected changes.

1. Introduction

Triggered by astonishing improvements in information and communications technology, the last decade has heralded the change from an industrial to an information age. Mass customisation replaced the mass production of the industrial age, and knowledge became the key driving economic input. Since the world with its markets and innovations is changing faster than ever before, the key to survival in the information age is the ability to detect, assess and respond to new trends and events — rapidly and intelligently. Discovering new information and using it before others do has become a strategic issue. As Don Keough, the former president of Coca-Cola, pointed out: ‘Who[ever] has information fastest and uses it wins’ [1]. Therefore it is crucial for companies to collect and stockpile data about processes, markets, customers and products in order to discover new information.

As a response to these trends, significant research into developing processes for automatic information gathering is currently being conducted, from which the term data mining has been coined. The main goal of data mining is to find information in data that is interesting, novel and potentially useful [2]. It utilises and combines methods from areas such as statistics and artificial intelligence as well as from soft computing and machine learning. Cluster and regression analysis, Bayesian networks, decision trees and association rules are just a few examples.

In parallel, business intelligence (BI) has become popular as an umbrella term for tools and technologies geared at transforming data into information and knowledge. Data mining is regarded as one of the fastest developing and most sophisticated BI technologies.

To market BI technologies, the term ‘proactive’ has recently become one of the major selling points. However, almost any BI technology, regardless of whether it is part of commercial software (like Siebel or SAP BI) or still a research subject, merely treats proactivity as a quality of predictive analytics. For example, state-of-the-art predictive analytic techniques are widely used to detect whether a customer is about to churn in the near future. It therefore answers the question: 'What are the main factors for churn at present?' Of course, such an approach is proactive since it yields information that helps to prevent a customer from churning. Nevertheless, it is proactive only on a short-term scale:

- it does not help prevent customers from becoming dissatisfied
- it does not help prevent customers from considering whether to churn
- it does not help to proactively eliminate factors which are likely to become triggers for churn on a medium- and long-term scale.

A human's competence to spot problems before they occur is significantly determined by their ability to register and analyse how certain aspects of the domain of interest change. In fact, it would be impossible for humans to draw any inferences about the future if they only had a snapshot of what a domain looks like at present. Data mining methods, however, assume implicitly that the domain under consideration is stable over time and thus provides a rather static view on the patterns and knowledge hidden in the gathered data. This is undesirable in time-stamped domains, since the data then captures and reflects external influences like management decisions, economic and market trends, and changes in customer behaviour. Data mining methods should account for this dynamic behaviour and hence consider the data as a sequence along the time axis. Such domains are very common in practice, since data is almost always collected over long periods — or, as Kimball [3] noted: 'The time dimension is the one dimension virtually guaranteed to be present in every data warehouse, because virtually every data warehouse is a time series.'

From this perspective the question: 'Which patterns exist?' as it is answered by state-of-the-art data mining technology, is replaced by the question: 'How do patterns change?' To continue our previous example of churn management, the analysis of pattern change would enable a business to answer questions like: 'Which factors are gaining more influence on customer churn and may be significant in the future?' Emerging causes for customer churn can thus be detected before they affect a large group of customers and business processes can be adapted in time. Generally, systematic pattern change is a pattern itself and obviously of high interest in the decision-making process — its predictive nature allows for proactive business decisions on a medium- and long-term scale. In fact, the detection of interesting and previously unknown changes in data, for which this paper proposes a general framework, not only allows the user to monitor the impact of past business decisions but also to prepare today's business for tomorrow's needs.

2. From KPIs to change mining

2.1 *Key performance indicators (KPIs)*

Currently, the technology provided by almost all business analytics software, like Siebel or SAP BI, to analyse change are key performance indicators (KPIs). These are quantifiable measurements that aim to assess the improvement of a business's critical success factors and to measure the progress towards its business goals. Updated at regular intervals, they give insight on how certain aspects of a business evolved in the past and how it might perform in the future. KPIs usually are long-term considerations. Their definition and the way they are measured may only change as the business goals change.

Technically, a KPI is often an aggregated value obtained by simple descriptive statistics. For example, if one of the business's goals is to improve customer retention, a reasonable KPI would be the relative frequency of churners within a certain time period.

While KPIs are a useful tool for strategic control and decision making on an upper management level, they have several shortcomings on the operational level:

- discovered changes are biased to what a user expects from its business described in terms of business goals — many other changes, in particular those which were not anticipated beforehand, remain unrevealed,
- the descriptive statistics used are rather coarse-grained — changes within more subtle aspects of a domain cannot be detected.

The range of observable change is therefore significantly limited by the user's expectations about its business — many other interesting changes may remain unrevealed. For example, for the churn-related KPI defined above, changes within smaller populations, like customers over the age of 50, cannot be detected. In general, the obstacle connected with this type of analysis is the user's lack of knowledge about many interesting co-occurring attribute values.

2.2 *Association rule discovery*

The latter issue can be solved by association rule discovery (see Agrawal et al [4]). Given a data set, its goal is to detect all those attribute values which frequently occur together and form rules which predict their co-occurrence. The advantage of association rule discovery is the completeness of its results — it finds the exhaustive set of all patterns which exceed specified thresholds on certain significant metrics. For this reason it provides a rather detailed description of a data set's structure. On the other hand, however, the set of discovered rules is typically vast.

Formally, association rule discovery is applied to a set D of transactions $T \in D$. Every transaction T is a subset of a set of literals L. These literals are commonly called items and a subset $X \subseteq L$ is called an item set. Given a database table with nominal attributes, every combination of an attribute and its value can be seen as an item and records can be seen as transactions.

An association rule r is an expression $X \Rightarrow y$, where X is an item set and y an item. Its meaning is quite intuitive — given a database D of transactions, the rule above expresses that whenever $X \subseteq T$ holds, $y \in T$ is likely to hold too. If for two rules $r_1 : X_1 \Rightarrow y$ and $r_2 : X_2 \Rightarrow y$ we have $X_1 \subset X_2$, this is denoted by $r_1 > r_2$ and it is said that r_1 is a generalisation of r_2 and accordingly that r_2 is a specialisation of r_1.

The predictive power of a rule $r : X \Rightarrow y$ is measured by its confidence conf(r) defined as the ratio of transactions that contain y as well as X w.r.t. the number of transactions that contain X:

$$\mathrm{conf}(r) := \frac{|\{T \in D | X \cup \{y\} \subseteq T\}|}{|\{T \in D | X \subseteq T\}|} \qquad \text{..... (1)}$$

The significance of a rule $r : X \Rightarrow y$ is measured by its support, which is defined as the fraction of transactions that contain $X \cup \{y\}$:

$$\mathrm{supp}(r) := \frac{|\{T \in D | X \cup \{y\} \subseteq T\}|}{|D|} \qquad \text{..... (2)}$$

Using the customer churn example, suppose that we are given a data set, which contains customer descriptions and whether these customers churned or not, and that the following association rule has been discovered from it:

AGE > 50 $\Rightarrow$ CHURN=YES

The support of this rule is the relative frequency of customers that are over 50 years old and churners, i.e. it describes the relative size of a group. The confidence of this rule, in contrast, can be interpreted as the relative frequency of churners within the group of over 50 year old customers, i.e. it describes the frequency a certain property has within a group.

Remarkably, if we focus only on rules with no antecedent, the confidence is identical to the common definition of KPIs as a relative frequency. For example, the confidence of the rule

$\Rightarrow$ CHURN=YES

is identical to the customer churn KPI defined in the previous section. Hence association rules together with the confidence metric can be seen as a generalisation of KPIs from a data mining perspective.

2.3 Rule change mining

The underlying idea of our framework is to detect interesting changes in a data set by analysing the support and confidence of association rules along the time axis (see Fig 1). In contrast to key performance indicators as discussed above, this allows us to detect interesting changes in a data set at nearly any level of granularity. Furthermore, no assumption about what actually might change needs to be specified.

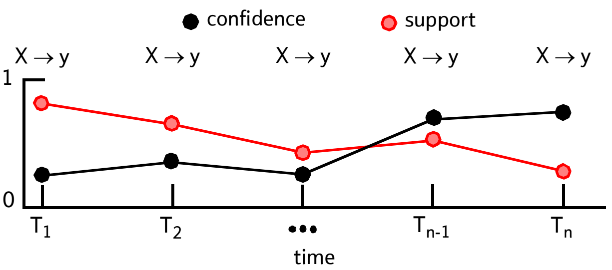

Fig 1 Deriving rule histories — the basis of rule change mining.

The starting point of such a rule change mining approach is as follows:

1. a time-stamped data set is partitioned into intervals along the time axis,
2. association rule discovery is then applied to each of these subsets,
3. this yields sequences — or histories — of support and confidence for each rule, which can be analysed further.

Of particular interest are regularities in the histories which we call change patterns. They allow us to make statements about the future development of a rule and thus provide a basis for proactive decision making.

Formally, the problem of rule change mining can be described as follows.

Let D be a time-stamped set of transactions and let $[t_0, t_n]$ be the smallest time span covering all its transactions. This interval $[t_0, t_n]$ is divided into $n > 1$ non-overlapping periods $T_i := [t_{i-1}, t_i]$, so that the corresponding subsets $D(T_i) \subset D$ each have a size $|D(T_i)| \gg 1$.

At the end of each period T_i a set of association rules $R(D(T_i))$ is generated from $D(T_i)$. Because confidence and support of each rule $r : X \Rightarrow y$ are now related to a specific transaction set $D(T_i)$ and thus to a certain time period T_i, their notations need to be extended. Starting from equations (1) and (2), this yields $\mathrm{supp}(r, T_i)$ and $\mathrm{conf}(r, T_i)$.

The temporal development of a rule r is thus described by n values for supp and conf. These values are inherently ordered by the time period they refer to and hence form sequences:

$$H_{supp(r)} := (\text{supp}(r, T_1),\ldots,\text{supp}(r, T_n)),$$
$$H_{conf(r)} := (\text{conf}(r, T_1),\ldots,\text{conf}(r, T_n)).$$

Accordingly they are called support history and confidence history of the rule r, respectively. The aim of rule change mining is to detect change patterns in rule histories. Trends, stabilities and cyclical variations are, for example, commonly encountered types. The detected change patterns and their detection methods should be chosen carefully and in such a way that any manual verification, post-analysis and search of detected change patterns is avoided.

To continue our example, suppose that rule change mining has discovered that the confidence of the rule:

$$\text{AGE} > 50 \Rightarrow \text{CHURN=YES}$$

has an upward trend. This, in turn, can be interpreted as that customer age is likely to become one of the key factors that influence churn in the future.

3. Framework for change detection

To summarise section 2.3, the objective of rule change mining is to reliably discover useful, interesting and interpretable change patterns hidden within a data set. Technically, this objective can be broken down into several tasks, which are interconnected and build upon each other.

As already said in the previous section our approach builds upon the idea of deriving association rules — as condensed representations of a data set's structure — at different points in time, which are then analysed for changes. As a consequence, histories of association rule measures, like support and confidence, are the basis for rule change mining. To derive a history, data sets collected during many consecutive periods have to be analysed for association rules. After each analysis session, the discovered rules have to be compared to those discovered in previous periods and their histories have to be extended. On the other hand, history values may be discarded if their age exceeds an application-dependent threshold. Consecutive update operations may not be temporally close to another because each association rule discovery session may take place directly after the collection period ended. Therefore rules and histories have to be stored on a long term basis. Taking all of the above into account the first task is:

1 association rules have to be discovered and their histories efficiently stored, managed and maintained.

If histories with a sufficient length are available, the next task is straightforward and constitutes the core component of rule change mining:

2 histories that exhibit specific change patterns have to be reliably detected.

Association rule discovery is generally connected with two problems:

- a vast number of rules will be detected, which is also referred to as the rule quantity problem,
- rules may be obvious, already known or not relevant, which is also referred to as the rule quality problem.

Since a history is derived for each rule, the rule quantity problem also affects rule change mining — it has to deal with a vast number of histories and thus it is likely that many change patterns will be detected. Moreover, as we will briefly discuss in section 4, methods that were developed to deal with this problem for association rules cannot be used in rule change mining.

Furthermore, there is also a quality problem — not all of the detected change patterns are equally interesting to a user and the most interesting are hidden among many irrelevant ones. This problem is partly inherited from association rule discovery, because the change patterns of obvious rules, for example, might be obvious too. On the other hand, however, a change pattern may increase the interestingness of a previously uninteresting rule significantly. Overall, the third task is:

3 histories with a change pattern have to be analysed for redundancies and evaluated according to their interestingness.

Because the aforementioned tasks build upon each other, they can be seen as layers of a processing framework. According to their task the layers are termed structural analyser, change analyser and interestingness evaluator, respectively. Figure 2 illustrates them and summarises the workflow.

4. Structural analyser

Given a time-stamped data set collected during a certain period, the task of the structural analyser is to discover and store the association rules hidden in it. Therefore the first component of this layer is an association rule mining system, while its second component is a database that stores and manages rules and their histories. Both components, together with the choice of the time periods, will be explained in the following. The design of the structual analyser is shown in Fig 3.

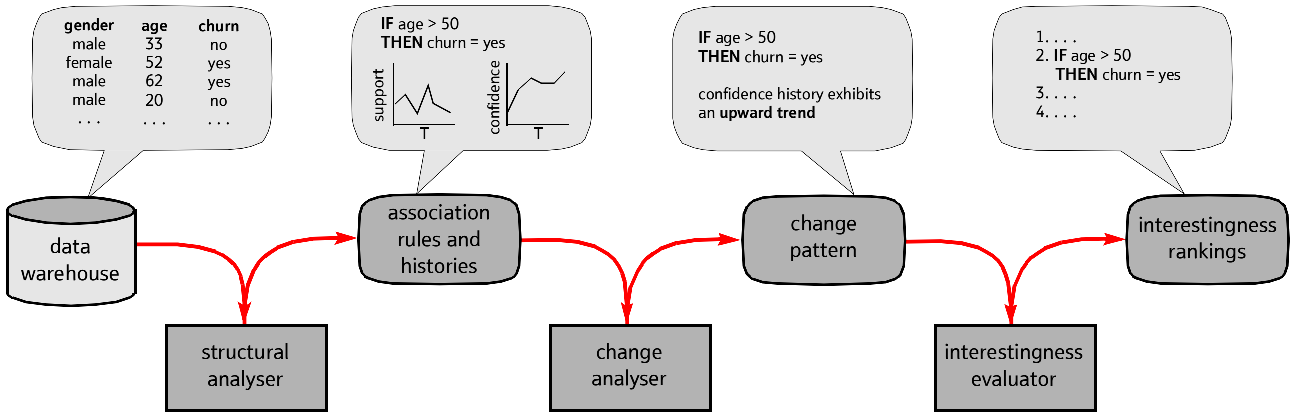

Fig 2 The framework's basic tasks and workflow.

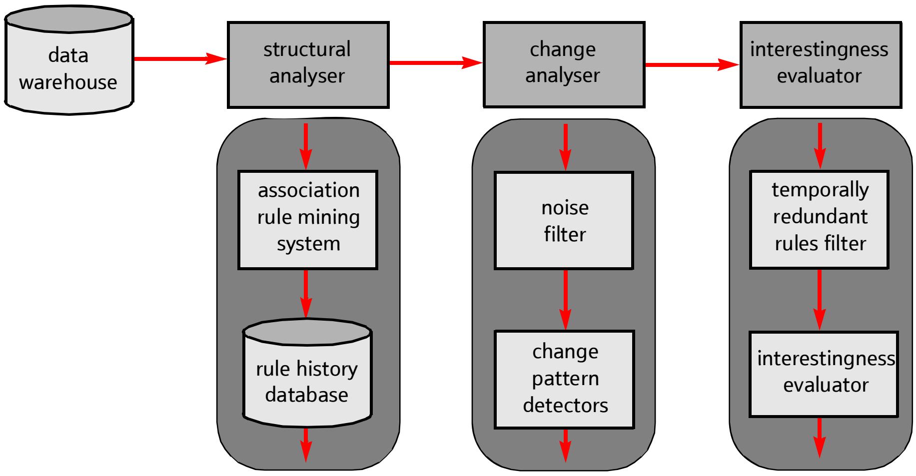

Fig 3 Detailed design of each layer.

In order to obtain the data set, the period length has to be chosen. This choice is crucial and difficult, because two aspects have to be considered — on the one hand, long periods lead to many transactions in the individual data sets for the different periods and thus can enhance the reliability of the support and confidence metrics. On the other hand, short periods allow us to measure a rule's statistics more frequently, which may lead to a more reliable detection of change pattern. Nevertheless, for practical convenience, periods in business applications are commonly measured in days, weeks, months, etc. In addition, the maximum number of periods is often restricted, for example, to fit a quarter or a year.

After the data set is available, association rule mining is applied to it. A typical system for association rule mining may consist not only of the rule miner itself, but also of methods for pruning, constrained mining and interestingness assessment. Such methods have been developed to cope with the rule quantity and rule quality problem, respectively, in each period.

However, the rule quality obviously does not affect rule change mining — whether or not a rule is interesting in a certain period does not directly influence the interestingness of its history. In fact, we assume that the interestingness of a change pattern primarily influences the interestingness of the underlying rule. Therefore interestingness measures for association rules should not be part of the rule mining system used within the scope of rule change mining.

On the other hand, the rule quantity problem affects rule change mining — a huge number of histories have to be processed and consequently far too many change patterns will be reported. In order to cope with this problem, pruning methods are broadly used to constrain the set of generated rules. Examples are non-redundant rule sets (see Zaki [5]) and informative rule sets (see Li et al [6]). From the perspective of rule change mining, such pruning methods treat rule sets independently from one another. However, in rule change mining we have many temporally ordered rule sets. Thus the rule property utilised for pruning — in general a measure based on rule statistics — may vary for some rules over time, but still match the pruning criterion in each rule set. Although these variations may render rules interesting, they are discarded by approaches for association rule pruning. Consequently conventional pruning approaches should not directly be used in conjunction with rule change mining.

The discovered rules and their statistics need to be stored and efficiently managed. Since the number of rules and thus the number of histories to analyse is vast, we use an Oracle database, which not only provides the required fast access and effective long-term storage, but also allows us to employ SQL for rule browsing and a simple descriptive statistical analysis of histories. The database scheme itself is geared towards the following frequent operations:

- determining whether or not a given rule is contained in the database,
- retrieving the histories for a given rule.

5. Change analyser

The task of the change analyser is to discover change patterns in rule histories. In this paper, however, we only discuss how histories are detected that are stable or exhibit a trend. The change analyser fulfills its task by a two step approach. In the first step, a filter is applied to the histories to reduce the noise contained in them. In a second step, statistical tests for trend and stability are conducted. Figure 3 shows the design of the change analyser; its two components are outlined below.

Any kind of data collected over time, like rule histories, inherently contains random variations referred to as noise. These random variations may influence subsequent analysis steps in such a way that wrong and misleading results are produced. An often-used technique for reducing this effect is smoothing. When properly applied, it reveals more clearly any underlying trend or stability. Within our framework we use double exponential smoothing (see Chatfield [7]).

Overall, double exponential smoothing is a simple and fast, yet effective, method, which can easily be automated. Nevertheless, the following implications have to be considered:

- after smoothing, association rule measures may be inconsistent with each other, e.g. given support and confidence histories, the equation conf(r, T_i) = supp(r, T_i) / asupp(r, T_i) will not hold generally after smoothing at least one of them,
- for some applications the focal point may be the detection of significant outliers — in this case smoothing should be applied very carefully, since it might disturb or discard them.

A trend is present if a sequence exhibits steady upward growth or a downward decline over its whole length. This definition is rather loose, and in fact no fully satisfactory definition exists for 'trend' [7]. For example, the perception of a trend depends partly on the length of a sequence, i.e. what seems to be a trend over a short time span, might be just part of a cyclical variation on a long-term scale.

From a data mining perspective, a trend describes the pattern that each value is likely to be larger or smaller than all its predecessors within a sequence, depending on whether the trend is upward or downward. Hence it is a qualitative statement about the current and likely future development of a sequence. However, taking aspects of interpretability and usefulness into account, such a statement is sufficient in the case of rule change mining — when faced with a vast number of rules and their histories, a user often has a basic expectation whether they should exhibit a trend and, if so, of what kind. By comparing their expectations with reality, the user will mostly be able to roughly assess the implications for their business. On the other hand, a user will rarely know in advance what trends should look like quantitatively, e.g. their shape or target values. Thus the user may be unable to exploit the advantages of more sophisticated trend descriptions, like regression models.

To choose a method for trend detection, it has to be taken into account that the number of sequences to examine is huge. Whenever a trend is reported, the user is basically forced to rely on the correctness of this statement, because it is unfeasible to verify each trend manually. In addition to the requirement of reliable detection, the method should incorporate no assumptions about any underlying model, because it is very unlikely that it will hold for all or at least most sequences. Therefore non-parametric statistical tests are the appropriate choice for trend detection, because they work without assuming any model. Furthermore, the probability that a trend is detected, although the sequence actually contains none, is bounded by the confidence level.

Within our framework we provide two statistical tests for trend, the Mann-Kendall test [8] and the Cox-Stuart test [9]. The Cox-Stuart test exploits fewer features of the sequence, leading to a computational effort that increases linearly with the sequence length. Although this may render the Cox-Stuart test susceptible to noise, because the influence of artefacts on the test result is stronger, it is considerably faster for long sequences. In contrast to this, the Mann-Kendall test is much more robust, but needs a computational effort that increases quadratically with the sequence length. Therefore it has to be determined which of the two issues — speed or robustness — is more important depending on the actual application scenario.

Roughly speaking, a history is considered stable if its mean level and variance are constant over time and the variance is reasonably small. Similar to trends, a clear definition of stability is difficult. For example, a sequence may exhibit a cyclical variation, but may nevertheless be stable on a long-term scale. Depending on the problem domain, either the one or the other — stability or cyclical variation — may have to be emphasised.

From a data mining perspective stability describes the pattern that each value is likely to be close to a constant value, estimated by the mean of its predecessors. Thus it is, like a trend, a qualitative statement about the future development of a sequence. However, in contrast to a trend, it can easily be modelled in an interpretable and useful way, e.g. by the sequence's sample mean and variance. Generally, stable rules are more reliable and can be trusted — an eminently useful and desirable property for long-term business planning.

To test for stability we use a method based on the well-known χ^2 test. However, since the χ^2 test does not take the inherent order of a history's values into account, our method may infrequently also classify histories as stable, which actually exhibit a trend. Therefore we chose to perform the stability test as the last one in our sequence of tests for change patterns.

6. Interestingness evaluator

Since usually a vast number of change patterns will be detected, it is essential to provide methods which reduce their number and identify potentially interesting ones. This is the task of the interestingness evaluator. To reduce the number of change patterns the interestingness evaluator contains a novel redundancy detection approach, based on so-called derivative histories. Although this approach proves to be very effective, the number of rules may still be too large for manual examination. Therefore a component for interestingness evaluation is provided, which contains a set of interestingness measures. The structure of the interestingness evaluator is shown in Fig 3.

6.1 Temporally redundant rules filter

Generally, most changes captured in a history — and consequently also change patterns — are simply the snowball effect of the changes of other rules. Picking up our example from churn analytics, suppose our framework discovered that the support of the rule:

r_1: AGE > 50 $\Rightarrow$ CHURN=YES

shows an upward trend. That is, the fraction of customers over 50 who churn increased. However, if the fraction of males among all over 50 year old churners is stable over time, the history of:

r_2: AGE > 50, GENDER=MALE $\Rightarrow$ CHURN=YES

shows qualitatively the same trend. In fact, the history of rule r_2 can be derived from the one of r_1 by multiplying it with a gender-related constant factor. For this reason, rule r_2 is temporally redundant with respect to its history of support.

It is reasonable to assume that a user will generally be interested in rules with non-derivative and thus non-redundant histories, because they are likely key drivers for change. Moreover, derivative rules may lead to wrong business decisions. In the above example, a decision based on the change in rule r_2 would account for gender being one significant factor for the observed trend. In fact, gender is completely irrelevant.

In general, we call a history of a rule 'derivative' if it can be explained with histories of more general rules and can thus be discarded. Technically, this means a history is derivative if it can be constructed as a (non-trivial) mapping of the histories of more general rules. We identified three scenarios of derivativeness, two for support and one for confidence histories, which are frequently encountered when analysing data and which have intuitive meaning. For each scenario we developed a criterion and implemented statistical tests to check if a history meets it. Furthermore, it can be shown that, although existing redundancy detection techniques for association rules cannot directly be applied, our method is consistent with some of them, that is, it can be seen as a generalisation from static scenarios to sequences of rule sets (see Böttcher et al [10]).

6.2 Interestingness measures

To assess the interestingness of detected trends and stabilities it has to be considered that each history is linked to a rule, which, prior to rule change mining, has a certain relevance to a user. However, the detection of a specific change pattern may significantly influence this prior relevance. In this sense a rule can have different degrees of interestingness, each related to another history. However, there is no broadly accepted and reliable way of measuring a rule's interestingness up to now (see Tan et al [11]). Therefore we consider any statement about the interestingness of a history also as a statement about the interestingness of its related rule.

To assess stable histories two things should be considered:

- association rule discovery typically assumes that the domain under consideration is stable over time,

- measures like support and confidence are interestingness measures for rules themselves.

Taking all this into account, a stable history is in some way consistent with the above-mentioned assumption of association rule mining. It is summarised by the mean of its values, which in turn can then be treated as an objective interestingness measure. Here the variance of the history can be neglected, since it is constrained by the stability detection method.

To develop objective interestingness measures for trends is more complex due to their richness of features. For identifying salient features of a given trend, it is essential to provide reference points for comparison. As such we chose the assumptions a user naively makes in the absence of any knowledge about the changes in rules' histories. From a psychological perspective they can be seen as the anchors relative to which histories with a trend are assessed — a trend becomes more interesting with increasing inconsistency between its features and the user's naive assumptions. We identified three such assumptions and defined heuristic measures for the discrepancy between a history and an assumption.

- Stability

  Unless other information is provided, a user assumes that histories are stable over time. This assumption does not mean that he expects no trends at all, but expresses his naive expectations in the absence of precise knowledge about a trend. It should be noted that this is consistent with conventional association rule mining, which implicitly assumes that the associations hidden in the data are stable over time. The confidence histories of the rule $XY \Rightarrow z$ in Fig 4 would violate the stability assumption because its trend is very clear.

- Non-rapid change

  Since users shape its business, they will be aware that the domain under consideration changes over time. However, they will assume that any change is continuous in its direction and moderate in its value. For example, if a business starts a new campaign, it will probably assume that the desired effect evolves moderately, because, for instance, a campaign may not have an impact on all customers immediately. On the other hand, a rapid change in this context attracts more attention, because it may hint at an overwhelming success or an undesired side effect. For example, the history of the rule $Y \Rightarrow z$ in Fig 4 would be very interesting according to the non-rapid change assumption because the depicted trend is very pronounced and steep.

- Homogeneous change

  If the support of a rule (item set) changes over time, it is assumed that the rate and direction of changes in the support of all its specialisations are the same. This basically means that the observed change in the rule (item set) does not depend on further items. For example, a user may know that the fraction of satisfied customers increases. The homogeneous change assumptions states that the observed change in satisfaction affects all customers and not only selected sub-populations, e.g. females over 40. If, on the other hand, the confidence of a rule changes over time, it is assumed that the confidence of all more specialised rules changes at the same rate. For example, the fraction of satisfied males among all male customers may increase. According to the homogeneous change assumption a user would conclude that among all married male customers the fraction of satisfied ones increases at the same rate. For example, the history of the rule $XY \Rightarrow z$ in Fig 4 would be very interesting because its shape is completely different from those of its more general rules.

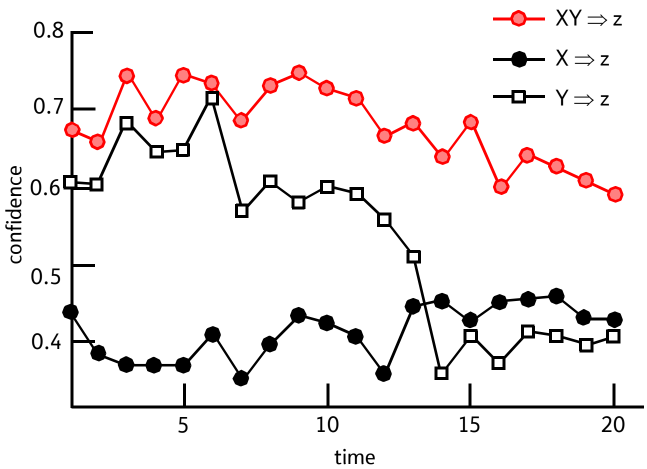

Fig 4 Examples of interesting histories which exhibit a trend.

7. Conclusions

Many businesses collect huge volumes of time-stamped data. This data reflects changes in the domain from which it has been derived. It is crucial for the success of most businesses to detect these changes, correctly interpret their causes and finally to adapt or react to them. Hence there is a significant need for data mining approaches that are capable of finding the most relevant and interesting changes in a data set.

In this paper we proposed a framework for discovering interesting trends and stabilities in the support and confidence histories of association rules. We have shown that it provides detailed knowledge about how a business evolves over time. Our framework's capabilities are far beyond what is possible with today's state-of-the-art data analysis systems, but still in line with traditional change analysis methods, such as KPIs.

We successfully applied our framework to two problem domains which are very significant for a telecommunications company — customer analytics, to investigate what is likely to drive customer satisfaction in the future, and fault analytics, to analyse the drivers of change in the number of faults.

Figure 5 shows the implementation of our framework. The prototype not only features the methods described in this paper but also a variety of filters to support tailored reporting that matches the needs of individual job roles and users.

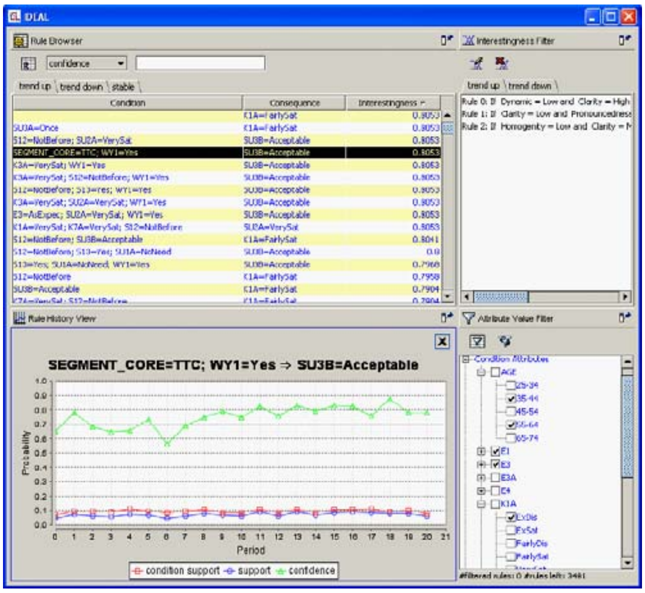

Fig 5 Prototypical implementation of the framework.

References

1 Watterson K: 'A Data Miner's Tools', BYTE Magazine (October 1995).

2 Fayyad U M, Piatetsky-Shapiro G, Smyth P and Uthurusamy R: 'Advances in Knowledge Discovery and Data Mining', AAAI Press and MIT Press, Menlo Park and Cambridge, MA, USA (1996).

3 Kimball R: 'Data Warehouse Toolkit: Practical Techniques for Building High Dimensional Data Warehouses', John Wiley & Sons (1996).

4 Agrawal R, Imielinski T and Swami A: 'Mining Association Rules Between Sets of Items in Large Databases', in Proceedings of the ACM SIGMOD International Conference on Management of Data, pp 207—216, Washington DC (1993).

5 Zaki M J: 'Mining Non-Redundant Association Rules', Data Mining and Knowledge Discovery, 9, No 3, pp 223—248 (2004).

6 Li J, Shen H and Topor R: Mining Informative Rule Set for Prediction', Journal of Intelligent Information Systems, 22, No 2, pp 155 —174 (2004).

7 Chatfield C: 'Time-Series Forecasting', Chapman and Hall/CRC (2001).

8 Mann H: 'Nonparametric Tests Against Trend', Econometrica, 13, pp 245—259 (1945).

9 Cox D and Stuart A: 'Some Quick Sign Tests for Trend in Location and Dispersion', Biometrika, 42, pp 80—95 (1955).

10 Böttcher M, Spott M and Nauck D: 'Detecting Temporally Redundant Association Rules', in Proceedings of 4th International Conference on Machine Learning and Applications, pp 397—403, Los Angeles, USA, IEEE Computer Society (2005).

11 Tan P-N, Kumar V and Srivastava J: 'Selecting the Right Objective Measure for Association Analysis', Information Systems, 29, No 4, pp 293—313 (2004).

Mirko Böttcher studied computer science and psychology at the University of Magdeburg, Germany. Before joining BT's Intelligent Systems Research Centre in 2005, he graduated with distinction with an MSc in computer science. Prior to and during his studies he worked as a software developer for the German Aerospace Centre and as a consultant for middleware technologies. His research interests are in the areas of data mining and machine learning.

He is currently working on the development of novel methods for change detection within temporal data and data streams.

Detlef Nauck is a Chief Research Scientist in BT's Intelligent Systems Research Centre in Adastral Park where he is currently leading an international team of researchers in Intelligent Data Analysis. He holds a Masters degree in Computer Science (1990) and a PhD in Computer Science (1994), both from the University of Braunschweig. He also holds a Venia Legendi (Habilitation) in Computer Science from the Otto-von-Guericke University of Magdeburg (2000), where he is a Visiting Senior Lecturer. He has published seven books and more than 80 papers and is a regular member of programme committees for conferences on computational intelligence. He has won two BCS gold medals and is an Associate Editor of IEEE Transactions on Systems, Men and Cybernetics — Part B.

His current research interests include the application of Computational Intelligence, Intelligent Data Analysis and Machine Learning in different areas of Business Intelligence. His current projects include research in proactive intelligent data exploration and platforms for real-time business intelligence and predictive analytics in customer relationship management.

Christian Borgelt studied computer science and physics at the University of Braunschweig, Germany. Having obtained a diploma in computer science in 1995, he worked first in the Daimler-Benz research centre, Ulm, Germany. Since 1996 he has been a research assistant at the University of Magdeburg, Germany. In 2000 he obtained his PhD in computer science from the University of Magdeburg with a thesis on ‘Data mining with graphical models’. In 2006 he obtained the venia legendi for computer science, also from the University of Magdeburg. His research interests include decision and regression trees, association rules, frequent graph mining, clustering algorithms, neural networks and computational geometry.

Rudolf Kruse studied mathematics and computer science at the University of Braunschweig, Germany. In 1980 he obtained his PhD from the University of Braunschweig, while working as a research assistant in the department of stochastics. In 1984 he obtained the venia legendi for mathematics, also from the University of Braunschweig. In 1986 he became Professor for computer science at the University of Braunschweig. He has held the chair for practical computer science (neuro-fuzzy systems) at the University of Magdeburg, Germany since 1996. In 2005 he was appointed a Fellow of the IEEE. Currently his main research interests are fuzzy and neuro-fuzzy systems and their application for data analysis and data mining.