Discovery and Visualization of Interesting Patterns

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von Diplominformatiker Matthias Steinbrecher,
geboren am 01. Oktober 1980 in der Lutherstadt Eisleben

Gutachter:	Prof. Dr. Rudolf Kruse
	Prof. Dr. Michael R. Berthold
	PD Dr.-Ing. Detlef D. Nauck
Promotionskolloquium:	Magdeburg, den 15. Juli 2013

Contents

Abstract

Data storage space comes almost at no costs today. Accumulating data is therefore an ubiquitous task in basically every business organization. However, this collection process needs to be complemented with sophisticated data analysis techniques in order to detect patterns inside these data. Such patterns may indicate problems or opportunities. In both cases it is of paramount importance to detect the formation and development of such patterns early enough in order to take timely countermeasures. To reach a large range of users, such analysis methods have to be intuitively controllable, must provide instant feedback and offer suitable visualizations. In this thesis, I propose a framework to visualize and filter the temporal evolution of sets of association rules. I will show how linguistic terms (represented by fuzzy sets) can be used to quantify a rule's history (with respect to certain quantitative measures) and subsequently rank them to present only the most relevant ones to the user for further assessment. I will transfer the suggested filtering method to other model types, present the software platform on which the methods are implemented and provide empirical evaluations on real-world business data.

Zusammenfassung

Die Preise für Speicherplatz fallen stetig, da verwundert es nicht, dass Unternehmen riesige Datenmengen anhäufen und sammeln. Diese immensen Datenmengen müssen jedoch mit geeigneten Methoden analysiert werden, um für das Unternehmen überlebensnotwendige Muster zu identifizieren. Solche Muster können Probleme aber auch Chancen darstellen. In jedem Fall ist es von größter Bedeutung, rechtzeitig diese Muster zu entdecken, um zeitnah reagieren zu können. Um breite Nutzerschichten anzusprechen, müssen Analysemethoden ferner einfach zu bedienen sein, sofort Rückmeldungen liefern und intuitive Visualisierungen anbieten. Ich schlage in der vorliegenden Arbeit Methoden zur Visualisierung und Filterung von Assoziationsregeln basierend auf ihren zeitlichen Änderungen vor. Ich werde lingustische Terme (die durch Fuzzymengen modelliert werden) verwenden, um die Historien von Regelbewertungsmaßen zu charakterisieren und so eine Ordnung von relevanten Regeln zu generieren. Weiterhin werde ich die vorgeschlagenen Methoden auf weitere Modellarten übertragen, die Software-Plattform vorstellen, die die Analysemethoden dem Nutzer zugänglich macht und schließlich empirische Auswertungen auf Echtdaten aus Unternehmenskooperationen vorstellen, die die Wirksamkeit meiner Vorschläge belegen.

An ounce of prevention
is worth a pound of cure.

Henry de Bracton

Selected Publications

The main contributions of this thesis appeared in the following (chronologically ordered) publications.

[STEINBRECHER and KRUSE 2007a] M. Steinbrecher and R. Kruse. **Visualization of Possibilistic Potentials**. Vol. 4529 of Lecture Notes in Computer Science, pp. 295–303. 2007, Springer Berlin / Heidelberg.

[STEINBRECHER and KRUSE 2007b] M. Steinbrecher and R. Kruse. **Visualizing Interesting Rules through Belief Network Inspection**. In: V. Köppen and R. M. Müller, Eds., Business Intelligence: Methods and Applications, Vol. 23 of Studien zur Wirtschaftsinformatik, Essays in Honor of Prof. Dr. Hans-J. Lenz **Visualizing Interesting Rules through Belief Network Inspection**, pp. 95–101. 2007. Verlag Dr. Kovač, Hamburg.

[STEINBRECHER and KRUSE 2008a] M. Steinbrecher and R. Kruse. **Visualization of Local Dependencies of Possibilistic Network Structures**. Vol. 224 of Studies in Fuzziness and Soft Computing, pp. 93–104. 2008, Springer Berlin / Heidelberg.

[STEINBRECHER and KRUSE 2008b] M. Steinbrecher and R. Kruse. **Identifying Temporal Trajectories of Association Rules with Fuzzy Descriptions**. In: Proc. Conf. North American Fuzzy Information Processing Society (NAFIPS 2008), 2008, pp. 1–6.

[STEINBRECHER and KRUSE 2009a] M. Steinbrecher and R. Kruse. **Clustering Association Rules with Fuzzy Concepts**. In: A. Fink, B. Lausen, W. Seidel and A. Ultsch, Eds., Advances in Data Analysis, Data Handling and Business Intelligence, Proceedings of the 32nd Annual Conference of the Gesellschaft für Klassifikation e.V., Joint Conference with the British Classification Society (BCS) and the Dutch/Flemish Classification Society (VOC), Helmut-Schmidt-University, Hamburg, July 16-18, 2008, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 197–206. 2009, Springer Verlag.

[STEINBRECHER and KRUSE 2009b] M. Steinbrecher and R. Kruse. **Assessing the Strength of Structural Changes in Cooccurrence Graphs**. In: B. Mertsching, M. Hund and Z. Aziz, Eds., KI 2009: Advances in Artificial Intelligence, 32nd Annual German Conference on AI, Paderborn, Germany, Vol. 5803 of Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence, pp. 476–483. 2009, Springer Verlag.

[STEINBRECHER and KRUSE 2009c] M. Steinbrecher and R. Kruse. **Fuzzy Descriptions to Identify Temporal Substructure Changes of Cooccurrence Graphs**. In: Proceedings of 2009 IFSA/EUSFLAT, 2009, pp. 1177–1182.

[BORGELT et al. 2009] C. Borgelt, M. Steinbrecher and R. Kruse. **Graphical Models — Representations for Learning, Reasoning and Data Mining**. John Wiley & Sons, United Kingdom, 2nd Edn., 2009.

[STEINBRECHER and KRUSE 2010] M. Steinbrecher and R. Kruse. **Visualizing and fuzzy filtering for discovering temporal trajectories of association rules**. Journal of Computer and System Sciences, Vol. 76(1):77–87, 2010.

[KRUSE and STEINBRECHER 2010] R. Kruse and M. Steinbrecher. **Visual data analysis with computational intelligence methods**. Bulletin of the Polish Academy of Sciences, Vol. 58(3), 2010.

[KRUSE et al. 2010a] R. Kruse, M. Steinbrecher and M. Böttcher. WCCI 2010 Plenary and Invited Lectures, Chapter **Temporal Aspects in Data Mining**, pp. 1–22. 2010. Institute of Electrical and Electronics Engineering, Inc.

[KRUSE et al. 2010b] R. Kruse, M. Steinbrecher and C. Moewes. **Temporal pattern mining**. In: Proceedings of the International Conference on Signals and Electronic Systems (ICSES 2010), pp. 3–8. Institute of Electronics, Silesian University of Technology, 2010, IEEE Press, Piscataway, NJ, USA.

[KRUSE et al. 2011] R. Kruse, C. Borgelt, F. Klawonn, C. Moewes, G. Ruß and M. Steinbrecher. **Computational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze**. Computational Intelligence. Vieweg+Teubner, Wiesbaden, Germany, 2011.

1 Introduction and Motivation

In the year of birth of the author, one megabyte of hard disk memory came at a cost of around 300 $. By the time of filing this thesis, the price for one *gigabyte* has dropped way below the 10-cent mark.[1] The unanimous conclusion is that the costs for storage space are nowadays virtually negligible.[2]

The plummeting prices enabled companies to collect and maintain evermore growing volumes of data—be it customer details and customer orders, product life cycle information, quality assessment data, website activities and much more. However, having a tremendous amount of cheaply stored data at the one hand, does not imply that answers and insights can be extracted from this data at little expense, too. Actually, the situation is often quite the opposite.

Finding relationships, dependencies and regularities—all summarized as *patterns*—within data sets is a non-trivial, typically hard and costly task. But it is these patterns that may lead to observations which in consequence trigger actions to increase profit or to minimize costs.

Identifying different groups in the customer base can be used for individual advertising. Early detection of fraudulent behavior can to a great deal prevent financial losses. Tracking down latent product failures allows to circumvent inevitable full product recalls.

[1] Values are taken from [WWW: HDD]. Current prices were confirmed by local advertisements.

[2] The current strategy of the business software company SAP is to use non-volatile storage (such as hard disks or solid-state drives) solely as a backup medium and keep the entire database in memory [WWW: SAP].

All the sketched scenarios above embody the same objective, which is best known in its colloquial phrasing as *An ounce of prevention is worth a pound of cure*. That is: Would it have been possible to detect a pattern earlier? More precisely, subsequent questions that often arise are: How did these patterns evolve in the past? and How are they going to evolve in the near future? The first question implies the conjecture whether it would have been possible to detect a specific pattern earlier, that is, before a pricey countermeasure had to be taken or a certain customer group was inexcusably overlooked. The latter question expresses the desirable request for a predictive model. Both questions stress that the detection of change is a fundamental concept to be addressed when patterns are to be inferred from data.

1.1 Knowledge Discovery and Data Mining

The process of gaining new insights from large databases is widely known as *Knowledge Discovery in Databases (KDD)*.[3] A formal definition is, of course, hard to grasp as the notion is largely abstract in nature. It is consensus that a KDD process consists of several steps, one of which being Data Mining. Moreover, the process is interactive and sometimes also considered a cycle (that is, the results will affect the initial requirements and trigger an adjusted KDD process). The KDD notion was first stated by [FAYYAD et al. 1996] and almost every thesis related to Data Mining contains this early reference to it. I am not going to depart from this tradition as it helps to put the scope of this thesis into perspective:

> Knowledge Discovery in Databases is the process of identifying valid, novel, potentially useful, and ultimately understandable structure in data.

In Figure 1.1 below, I am using a simplified version of the KDD process since the contributions of this thesis are focused in the Data Mining step

[3] The notions *Knowledge Discovery in Databases* and *Data Mining* are often used interchangeably.

and the more elaborated version from [FAYYAD et al. 1996] differs in the other steps.

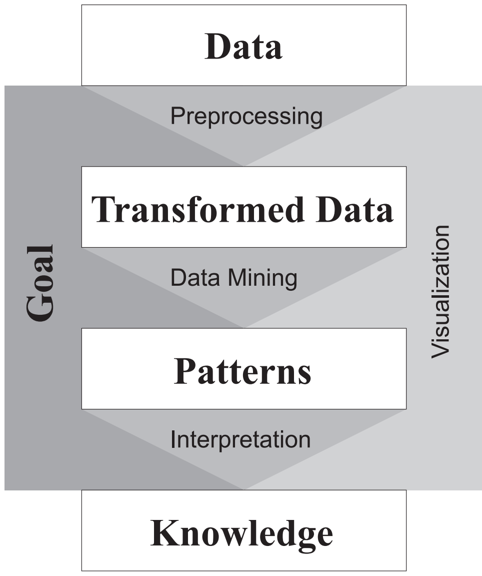


Figure 1.1: *Simplified KDD process. Stages are in bold letters, processes in sans-serif letters.*

Definition of the Goal. The initial step of every KDD process (that is quite often assumed to be obvious or casually stated) is the definition of the objective of the whole process. I do not mention this step here just as a preliminary one: it is actually affecting the entire KDD process! Without being aware of the exact questions (or at least the type of question) to be answered by the analysis results, it is hard—if not impossible—to select the appropriate data mining algorithm.

Preprocessing. This process encompasses the selection and transformation of the raw input data. It further comprises feature extraction and feature generation. The actual tasks to be carried out depend on the defined goal and thus on the selected data mining algorithm(s). The major part of the entire KDD workload is typically spent in this preprocessing step.

Data Mining. The data mining step induces the type of patterns that have been selected when defining the goal and for which the data has been preprocessed. There is a multitude of different tasks and algorithms which I do not intend to cover here. For an overview and thorough treatment of the KDD process refer to [BERTHOLD et al. 2010].

Interpretation and Evaluation. Finally, the patterns found in the data mining step need to be evaluated. Hypotheses have to tested and conjectures have to be confirmed. Often, the gained insights spark new goals and trigger a new and refined KDD process, thus creating an analysis cycle as in the widely-used CRISP-DM model[4] [CHAPMAN et al. 2000].

1.2 Cross-cutting Concerns

There are at least two aspects in the KDD process that were not yet addressed explicitly. One is represented in Figure 1.1 influencing the majority of the analysis process: visualization. It is sometimes included into the data mining step, however, as I will motivate in Section 1.2.1 below, it is a much more far-reaching aspect as to dismiss it as a fringe aspect. The second aspect addresses the temporal component of the domain under analysis. Section 1.2.2 will discuss this issue in more detail as it may be considered in different stages of the analysis.

1.2.1 Visualization

Visual representations of statistical data have been used for a long time and proved indispensable for understanding and presenting analysis results [CHEN et al. 2008]. The rapid progress in computer graphics hardware allowed for the development of interactive user interfaces that allow for an exploratory approach to the data to be visualized. The idea of using graphical representations not only for a static view on the data but

[4] CRoss Industry Standard Process for Data Mining. Established and promoted by SPSS, NCR and DaimlerChrysler.

rather as an intuitive and interactive tool for data analysis was summarized under the notion of *Visual Analytics*. The term was coined around 2004 and now represents a vivid field of application and research. As for KDD there is no fixed definition, however, a recent overview publication on the field ([KEIM et al. 2010]) contains the following definition:

> Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex datasets.

Moreover, the following goals are also named, and I repeat them here as they will later help to assess the quality of the proposed visualization techniques.

- Synthesize information and derive insight from massive, dynamic, ambiguous, and often conflicting data.
- Detect the expected and discover the unexpected.
- Provide timely, defensible, and understandable assessments.
- Communicate these assessments effectively for action.

Visualization is not only about displaying every piece of information that is available, it is also about carefully selecting those artifacts that shall be actually shown to the user. I do not have the hubris to claim a perfect visualization, but a proverb attributed to Antoine de Saint-Exupery nicely puts the above claim:

> Perfection is achieved, not when there is nothing more to add,
> but when there is nothing left to take away.

Often, the selection process of what to visualize and what to omit is harder to tackle than the actual visualization itself (which might be straightforward). The framework presented in this thesis is no exception: A considerable part is spent on discussing the filtering approach, the results of which are then visualized in an intuitive fashion.

1.2.2 Temporal Aspects

Taking temporal information into account is not explicitly specified in the KDD process but rather intended to be taken care of within each data mining algorithm. It can be considered a cross-cutting concern that—like visualization—even pertains to all KDD stages. Most of the collected data are already time-stamped. Due to its temporal nature, business data reflect external influences like management decisions, economic and market trends and thus captures the changes a business is interested in. But change is not necessarily related to problems. Change can also mean an opportunity (like an evolving group of customers) to a business. Therefore, (pattern) change detection is a vital task in order to survive and to compete.

Patterns are not likely to arise out of a sudden. They rather will evolve slowly over time. If the patterns represent evolving groups of customers or a changing subset of ordered products, it will take time until these patterns will become obvious. Conversely, after a problem has been addressed, it would be naïve to expect the patterns disappear immediately. Again, an effect of a counteraction will need time to become (in)visible. Thus, the temporal evolution of such patterns carries valuable—if not vital—information about the urgency of the underlying problem (or the effectiveness of the treatment).

However, a fully automatic approach has its limitations. In order to minimize response times to problems, data analysis results must be interpretable by technical staff that not necessarily has a statistical background. In addition, the analysis should be as transparent as possible to comprehend all inferences and conclusions that were drawn.

The approach I present in this thesis leverages the use of temporal information that is already present in every production database: the time stamps associated to the objects (tuples) under analysis. This immediately enables users to apply the presented framework with no change to the existing system.

1.3 Topic of this Thesis

This section introduces the main objective of this thesis as well as sketches of those aspects that I assign a high importance. Briefly put, the topics covered in Chapters 3 and 4 can be phrased as follows:

> This thesis deals with the ② *identification* and ③ *visualization* of ① *patterns* that exhibit a certain ④ *temporal behavior* that is considered ⑤ *interesting to the user*.

The emphasized and numbered phrases represent aspects that I consider important and thus are discussed in greater detail below.

① Pattern Types

Although the framework to be presented can be transferred to several pattern types,[5] I will introduce the arguments in favor of association rules[6] first. The main reason for this decision is the fact that the projects that sparked the ideas for the framework to be presented dealt with association rules in the first place. However, there are several other reasons as to why association rules were found to be a prominent choice for the framework.

Simplicity. The semantic of an association rule is understood to be a logic implication that is true in only a certain fraction of cases. Even though the logic implication is known to be prone to confusion by people that do not have skills in propositional logic [DURAND-GUERRIER 2003],[7] the experience with industrial partners and decision makers shows that if-then-like patterns are welcomed ways to represent dependences and

[5] Principles are suggested in Section 4.5 and examples are given during evaluation in Chapter 6.
[6] See Section 2.1 for an introduction.
[7] The confusion arises from mainly two causes: First, the fact that an implication with a false antecedent is generally true (instead of being colloquially "inapplicable") is often a source of confusion. Second, laymen often mistake "If not A, then not B" as a logic equivalence of "If A, then B".

relations of a certain domain. The intuitive and little number of parameters can also be seen as an appealing property in favor of association rules: only minimum support and minimum confidence[8] are necessary to fully specify the problem. Both parameters can easily be set and understood by users that not necessarily have a data mining background because they directly relate to relevant quantities. Compare it, for example, with the attribute evaluation measure needed for decision tree induction: here it is much harder for laymen to understand the difference between, say, information gain and χ^2 measure.

Stability. Projects with industrial partners have shown that there is another property (in addition to simple parameterization) that greatly enhances the acceptance of a data analysis technique: predictability. When studied in more detail, techniques are preferred where a small change in the input data or parameters (or both) leads to a result that has also changed just slightly. Counterexamples can, for example, be observed when inducing decision trees: even when the parameters are fixed, a small change in the input data (that is, an additional tuple, a changed tuple or a removed tuple) can lead to a completely different tree. With association rules induction, however, a change in the initial setting will typically lead to a few rules to be missing or to be newly created.[9] From the user point of view, rule sets can be refined by tuning parameters, rather than coming up with possibly dramatically different results. Not having such a type of recognition can be a deal-breaker when it comes to the decision whether a new technique shall be introduced in an existing analysis setting.

Derivability. There are, of course, other prominent data mining methods that are widely accepted in industrial applications. It would be rather restrictive for the proposed filtering approach of this thesis if it required the user to switch entirely to a different type of model (namely association

[8] See Section 2.1 for details.

[9] It is, of course, possible to craft a data set where a slight change in the parameters leads to drastic changes in the induced rule set. However, real-world applications have not shown such effects in practice, whereas the mentioned unstable behavior of decision trees can be observed quite frequently.

rules). Companies invest considerable amounts of money into building in-house data analysis solutions. It would be little appealing to them to evaluate a newly proposed method if this approach called for a revision or at least a refinement of their current development. I will argue in Section 4.5 how the presented framework can easily be used in a setting with, for example, decision trees or graphical models, thus greatly increasing the potential area of application.

Why not just frequent item sets? A look in the recent literature causes the impression that most effort is spent on improving the induction of so-called frequent item sets [BORGELT 2005, RÁCZ et al. 2005, BURDICK et al. 2003, SENO and KARYPIS 2001, HAN et al. 2000] which are a pre-stage from which association rules are generated in a subsequent step. In personal communications, some researchers even argued that "frequent item sets contain everything" and therefore "there is little need to bother with association rule 7 creation". I agree with the first phrase, since the item set $\{a, b, c\}$ potentially represents the rule set $\{b \to a,\ c \to a,\ bc \to a,\ a \to b,\ c \to b,\ ac \to b,\ a \to c,\ b \to c,\ ab \to c\}$ and thus can be considered a condensed representation. And indeed, if a classical market basket analysis reveals the frequent item set {*Lime*, *Cachaça*}, the supermarket owner has all he needs to exploit the pattern. But consider the following pattern: {*EHEC*, *sprouts*}. Since the number EHEC[10] cases was rather low (compared to the general population) a frequent item set induction algorithm would have had to run with an extremely small minimum support in order to generate the above pattern at all. A small minimum support, however, would lead to a huge number of patterns (out of which we could, of course, easily select those item sets that contain *EHEC*, but depending on the epidemic progression there might be multiple fomites). If we induced association rules, we would find the rule *EHEC* → *sprouts* with a confidence close to 1 much easier. To be clear here: both "approaches" are capable of finding the described pattern and to induce the above association rule we need to set the minimum support to the same low value. My argument is, that looking for high-confidence rules is much more in-

[10] enterohemorrhagic Escherichia coli

tuitive for a user with little or no background on frequent item set induction.

② Identification

Any algorithm intended for pattern induction can only serve as a recommendation agent suggesting patterns to the user. It is then the user's task to finally *identify* (and confirm) a pattern. To achieve this goal, (at least) two aspects need to be addressed. First, it is crucial to provide a responsive user interface. The notion *sub-second response time* is used to convey the requirement to get results faster than the average time span the human brain needs to reply to a stimulus [PLATTNER and ZEIER 2011]. The prototype used to implement the framework of this thesis meets that requirement.

Second, the user must be enabled to intuitively describe the type of patterns he is interested in (see Interestingness section below). This means of description must neither be to restrictive nor shall any unintuitive parameterization be necessary. Fine-tuning these parameters shall create an impression of revising the result set rather than just create a completely new one.

③ Visualization

As already mentioned in Section 1.2.1, I am going to apply appropriate graphical representations in order to visually verify the identified patterns. In line with the adage *a picture is worth a thousand words*, the human visual system is highly capable of assessing and recognizing patterns, provided we are able to restrict the set of candidate patterns to a size that is manageable by a single individual.

④ Temporal Behavior

Industrial and business data are usually collected incrementally over a certain period of time. Customer orders accumulate successively, the logistic processes behind the scenes—although more and more being sped up—take a few days and generate respective data traces. The quality of products may exhibit flaws after different periods of usage and thus leads to a constant stream of data. Since all these data are binary footprints of real-world processes and human behavior (including the behavior of human-designed devices and products), pretty much every data set that was (and currently is) being accumulated has time stamps in one or another way.[11] Humans are much better in comparing stimuli in relative terms than in assessing the absolute intensity of a single stimulus [FECHNER 1860]: telling the louder of two sound samples is much easier than determining the volume in decibel directly. The same holds true for effectively all other perceivable physical sizes. Comparing values in temporal succession can be considered an intrinsic human capability that should be leveraged when it comes to pattern evolution.

⑤ User Interestingness

I will suggest an approach using linguistic descriptions that allow to describe the shapes of pattern measure time series in a fuzzy manner. The user can use fuzzy partitions instead (manually specified or automatically generated from the data or a combination of both) to denote the vague class of shapes he is interested in.

1.4 Structure of this Thesis

Chapter 2 introduces and revisits the frameworks and mathematical underpinnings that are necessary for the remainder of the thesis. The visualization method is introduced in Chapter 3 after which the actual framework is presented in Chapter 4. The implementation of the theoretical concepts is presented in Chapter 5. I apply the proposed methods in Chapter 6 to several data sets before a conclusion is drawn and outlooks

[11] In most database schemas one can find dedicated attributes that represent the time when an entry was created or modified or both. If not, modern databases collect metadata internally and chances are that there is a creation time stamp for each entry.

are given in Chapter 7. In this thesis I use the following policy to decide which grammatical person to employ in the narration: I will use the first person singular whenever my own decisions, contributions or opinions are to be stressed. In all else situations I will use passive or first person plural.

2 Background

This chapter introduces the concepts needed to follow my arguments and proposals of this thesis. Section 2.1 introduces frequent item set mining before the connection to relational item sets in Section 2.2 is made. Cooccurrence graphs are sketched in Section 2.3 as another model type whose temporal analysis can benefit from the approaches in this thesis. Elements of fuzzy set theory are outlined in Section 2.4 before I close the chapter with a brief introduction into evolutionary algorithms in Section 2.5

2.1 Elements of Frequent Pattern Induction

The classic approach [AGRAWAL et al. 1993] as well as alternative techniques [ZAKI et al. 1997, BORGELT 2005, HAN et al. 2000] of association rule inference consists of first finding subsets of items (so-called item sets) that occur together in more than a predefined fraction (the minimum support) of transactions and then trying to identify a single item within each item set such that the probability of observing this item given the remaining items of the item set exceeds some other predefined threshold (the minimum confidence). By transaction we mean a tuple (like a row of a database table) with exclusively nominal attributes.

2.1.1 Frequent Item Sets

Let U be a finite set of N *items*, that is, $|U| = N$. Any subset $I \subseteq U$ is called an *item set*. Further, we consider a so-called *transaction database*[1] $D = (d_1, \dots, d_n)$ with $d_i \subseteq U,\ i = 1, \dots, n$. A transaction $d \in D$ *covers* an item set I iff $I \subseteq d$. Consequently, we refer to

$$K_D(I) = \left[I\right]_D = \{k \in \{1, \dots, n\} \mid I \subseteq d_k\}$$

as the *cover* of I with respect to D. The $[\cdot]$-notation will later be useful when we discuss evaluation measures. Since the transactions in D need not necessarily be disjoint, we cannot collect all covered transactions as a set. Hence, we collect their unique indices. The following quantities define two important measures for item sets:

- absolute support: $\text{abs-supp}_D(I) \;=\; |K_D(I)| \;\geq 0$
- relative support: $\text{rel-supp}_D(I) \;=\; \dfrac{|K_D(I)|}{|D|} \in [0, 1]$

We will omit the index D if the underlying database is clear from the context. Obviously, the absolute support counts the number of transactions that cover I and therefore can be considered *supporting* it. In reality, only those item sets are of interest whose support exceeds a certain (user-specified) minimum support. The item sets are then called *frequent* item sets. Given a minimum absolute support of $s_{\min}$ we are interested in the set

$$F_D(s_{\min}) \;=\; \{I \subseteq U \mid \text{abs-supp}_D(I) \geq s_{\min}\}.$$

If the relative support $\sigma_{\min}$ is constrained, we get the following definition of the set of frequent item sets:

$$\Phi_D(\sigma_{\min}) \;=\; \{I \subseteq U \mid \text{rel-supp}_D(I) \geq \sigma_{\min}\}$$

Efficiently inducing the set F_D (or Φ_D) longtime occupied the scientific research field of frequent pattern mining.[2] An important property used

[1] For technical purposes we consider D to be a vector rather than a set.

[2] Even though there are still papers being submitted that deal with enhancing frequent item set induction, there are highly efficient algorithms available [HAN et al. 2000, BORGELT 2005] such that the research focus changed to post-processing sets of frequent item sets.

i	d_i
1	$\{a,b\}$
2	$\{a\}$
3	$\{b,c\}$
4	$\{b,c\}$
5	$\{a,b,d\}$
6	$\{b,d\}$
7	$\{c,d\}$
8	$\{c\}$
9	$\{a,b,c\}$
10	$\{a,b,e\}$

database D

$$\bigcup_{i=1}^{10} d_i = U = \{a,b,c,d,e\}$$

$$\begin{array}{llll}
K_D(\{a\}) & = \{1,2,5,9,10\}, & \text{abs-supp}_D(\{a\}) & = 5 \\
K_D(\{a,b\}) & = \{1,5,9,10\}, & \text{abs-supp}_D(\{a,b\}) & = 4 \\
K_D(\{a,b,c\}) & = \{9\}, & \text{abs-supp}_D(\{a,b,c\}) & = 1
\end{array}$$

Figure 2.1: *Example database with 10 transactions. Covers and absolute supports for three successively inclusive transactions are given to show the anti-monotonicity of the support.*

for restricting the search space of item sets is the so-called Apriori property of the support.[3] This property basically exploits the support's *anti-monotonicity*: The support of an item set cannot increase if further items are added. Each item of an item set constrains the cover as can be seen exemplarily in the cover relationship of the two-element item set $\{a,b\}$:

$$\forall a,b \in U: \; K_D(\{a,b\}) \;=\; K_D(\{a\}) \cap K_D(\{b\})$$

In other words: enlarging J to I will lower (or at most will not change) the support:

$$\forall J: \; I \supseteq J: \; \text{supp}_D(I) \leq \text{supp}_D(J)$$

Figure 2.1 illustrates this relationship. Starting with the item set $\{a\}$, further items are added and the covers and supports are shown. Finally, applied to the task of frequent item set induction, we get the insight that no superset of an infrequent item set (that is, an item set not reaching the minimum support) can be frequent:

$$\forall s_{\min}: \; \forall J: \; \forall I \supseteq J: \quad \text{abs-supp}_D(J) < s_{\min} \quad \Rightarrow \quad \text{abs-supp}_D(I) < s_{\min}$$

Algorithms based on extending frequent item sets to find potential larger frequent item sets can ignore all supersets of an infrequent item set since its support cannot increase.

[3] If neither *relative* nor *absolute* is specified, the discussed properties apply to both definitions.

2.1.2 Association Rules

An *association rule* $\rho = X \rightarrow Y$ satisfies $X, Y \subseteq U$, $X \cap Y = \emptyset$ and $Y \neq \emptyset$ and consists of the *antecedent* X and *consequent* Y. For the rest of this work we will assume that Y contains only one item, that is, $Y = \{y\}$ and often replace Y with $y \in U$ (or another lower-case letter clear from the context). For an association rule $\rho = X \rightarrow Y$ and an item $x \in U$ the statement $x \in \rho$ denotes that the item x is contained in either the rule's antecedent or consequent.

The semantic of a rule $X \rightarrow Y$ is that of an implication: If a randomly chosen transaction $d \in D$ covers the item set X, then it also covers the item set Y:

$$d \supseteq X \;\Rightarrow\; d \supseteq Y$$

The notion of an association rule and its semantics allows us to introduce more sophisticated evaluation measures. As a prerequisite for their definition, we need to define the semantics of an item set's probability. The probability $P(X)$ of an item set $X \subseteq U$ is chosen to coincide with X's relative support:

$$\forall X \subseteq U: \; P_D(X) \overset{\text{Def}}{=} \text{rel-supp}_D(X) = \frac{\big|[X]\big|}{|D|}$$

Note, although being a set, the item set X is not an event in the probabilistic sense! The item set X in $P(X)$ *represents* a probabilistic event rather than *being* that event itself. To actually arrive at a proper event, we agree on the following assumptions. To constitute a probability measure, P needs to be declared on a so-called *σ-algebra* and its argument has to be (or—as in our case—has to represent) an *event* [KOLMOGOROV 1933]. If we assume the index set of all transactions in database D to be the underlying *universe of discourse* (that is, each transaction $d \in D$ is considered an *elementary event*), then any item set $I \subseteq U$ fixes a subset of transactions by means of its cover $[I]_D \subseteq \{1, \ldots, n\}$. This subset is then a proper event and hence can be assigned a probability which we do so by assuming a uniform distribution over all transactions. To avoid unnecessary clutter, we use for an item set X the term $P(X)$ and $P\big([X]\big)$ interchangeably.

Given the ten transactions in Figure 2.1, the probability of the item set $\{b,d\}$ is then calculated as

$$P(\{b,d\}) = \frac{|\{5,6\}|}{10} = \frac{1}{5},$$

which is obviously the relative support of $\{b,d\}$.

We can easily embed the above-discussed agreements into the parlance used in binary classification. When association rules are used for market basket analysis, then there is usually no restriction on the items that antecedent and consequent may contain (apart from both being disjoint and the consequent being non-empty, of course). However, there are use cases where analysts are interested in special items in the consequent that are not allowed in the antecedent. Assume that a car manufacturer records the configuration of each vehicle that leaves the production plant. If failures are reported to service garages, a dedicated (class) variable in the record of that particular car is updated. Quality control personnel can now try to induce a classification model from that data source. If we focus on a certain error code (break failures, say) we get a binary class variable. For any binary classifier applied to tuples (also called *cases*) of a database D, there are four subsets of D that are of interest to assess the quality of that classifier:

- Predicted cases by classifier:
  - Case predicted to be positive
    (that is, belongs to a certain class)
  - Case predicted to be negative
    (that is, does not belong to a certain class)
- Actual state of the world:
  - Case is actually positive
  - Case is actually negative

The cardinalities of the intersections of these subsets are used to evaluate the classifier goodness. Since the actual state of the world is hard to impossible to assess, the classifier is usually applied to a database where the actual class assignments are known as kind of a ground truth. In such a setting, we can consider an association rule $X \rightarrow Y$ a special type of binary classifier: if an (previously unseen) input tuple meets the antecedent condition(s), the rule predicts the class encoded by the item in the consequent.[4] The below table summarizes the relations between an association rule and a binary classifier.

Notions in both concepts

Binary classifier	Association rule
Tuple is classified positive	Tuple matches antecedent
Tuple is actually positive	Tuple matches consequent

Given a rule $X \rightarrow Y$, the antecedent X and consequent Y divide the database D into four disjoint subsets. Figure 2.2 illustrates this. The ideal case, of course, would be reached when $[X] = [Y]$, that is, when the prediction by the rule's antecedent equals the actual state of the world (or the ground truth for that matter).

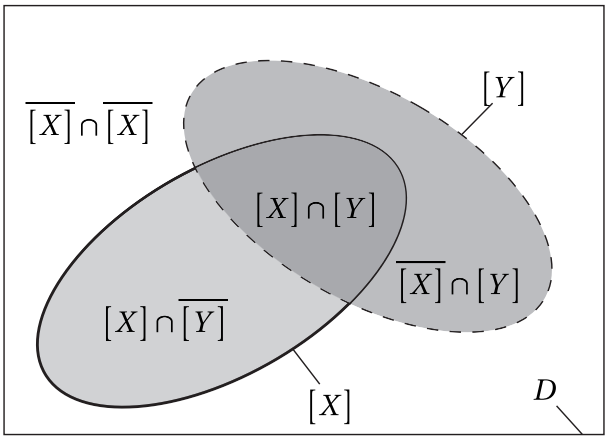


Figure 2.2: *The rule $X \rightarrow Y$ partitions the database D into four disjoint subsets that are represented by the white and three grayish shapes. The rule evaluation measures discussed below all make use of the cardinalities of these four subsets.*

All four subsets are well-known in the field of binary classification as depicted in Table 2.1.

[4] That is, we only would allow values of the class variable's domain to represent the consequent item.

		Actual world state	
		True	False
Predicted world state	True	True Positives (TP) $\left\|[X]\cap[Y]\right\|$	False Positives (FP) $\left\|[X]\cap\overline{[Y]}\right\|$
	False	False Negatives (FN) $\left\|\overline{[X]}\cap[Y]\right\|$	True Negatives (TN) $\left\|\overline{[X]}\cap\overline{[Y]}\right\|$

Table 2.1: *Disjoint subsets in binary classification. The cardinalities are used to derive association rule evaluation measures.*

Let us now discuss certain scenarios of Figure 2.2 where the overlap between X and Y differs. Figure 2.3 illustrates four constellations which carry certain meanings with respect to confidence and recall.

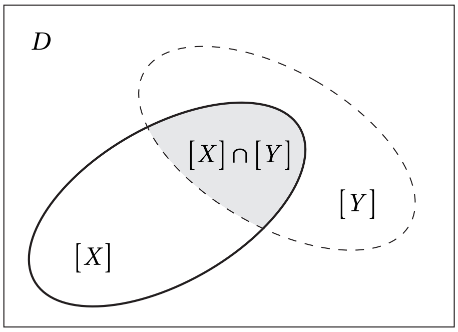
We now define and briefly discuss for a given rule $\rho = X \rightarrow Y$ some of the above-mentioned rule evaluation measures.[5] Be aware of the short-hand notation agreed upon above: the term $P(X \cup Y)$ actually translates into $P\Big([X \cup Y]\Big)$ which equals $P\Big([X] \cap [Y]\Big)$ for disjoint X and Y.

Support

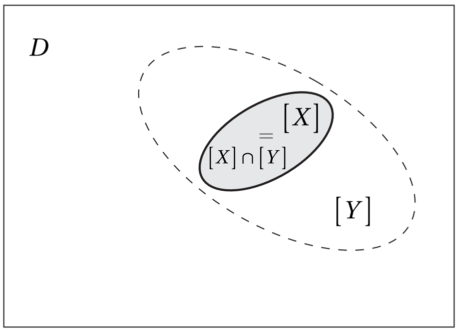
$$
\begin{aligned}
\text{rel-supp}_D(\rho) &= P(X \cup Y) && \in [0,1] \\
\text{abs-supp}_D(\rho) &= |K_D(X \cup Y)| && \in \mathbb{N}
\end{aligned}
$$

We define two support measures in analogy to the support measures for item sets. A transaction is counted to support a rule if it covers both antecedent and consequent. There are arguments to just consider the antecedent X in the above definitions [WWW: BORGELT 2] but I will stick to the widely used intuition given above.

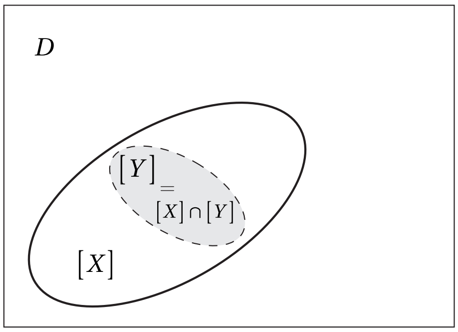
[5] Again, the index D is dropped if the respective database is unambiguous.



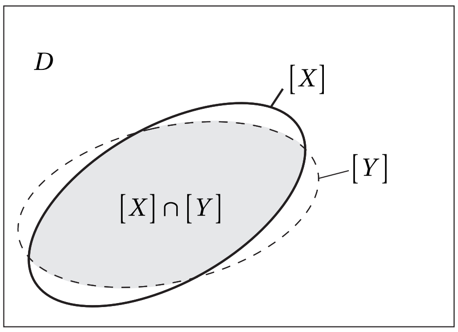
Low Confidence, low Recall
General case where X covers some relevant cases of Y but also a large quantity of irrelevant cases.



Maximal Confidence, low Recall
In this case all cases covered by X are relevant (that is, are also covered by Y) and therefore the confidence is 1. But the cardinality of X is small compared to the cardinality of Y. That is, only a small fraction of all relevant cases are covered and hence the recall is low.



Low Confidence, maximal Recall
Here, all relevant cases are covered by X and the recall is therefore 1. However, a lot of irrelevant cases are also covered leading to a low confidence.



High Confidence, high Recall
This is the kind of rules (at least in terms of confidence and recall) that the user shall be interested in: almost all relevant cases and only few irrelevant cases are covered by X.

Figure 2.3: *Different overlapping scenarios for a rule $X \rightarrow Y$.*

Confidence (also: Precision)

$$\mathrm{conf}_D(\rho) \quad = \quad P(Y \mid X) \quad = \frac{P(X \cup Y)}{P(X)} \quad = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \quad \in [0,1]$$

Simply put, the confidence represents the probability of the "applicability" of the rule. That is, given we observed the antecedent's items, it states the probability of also observing the consequent's items. Most algorithms for rule induction take a minimum confidence as a parameter (in addition to the minimum support) and return only rules that exceed that user-specified value.

Recall (also: Sensitivity, True Positive Rate (TPR))

$$\mathrm{recall}_D(\rho) \quad = \quad P(X \mid Y) \quad = \frac{P(X \cup Y)}{P(Y)} \quad = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \quad \in [0,1]$$

The recall denotes the consequent-specific proportions of cases matching the antecedent. A rule with high recall returns (or *recalls*) most of the cases that match the consequent. Since this can trivially be achieved by returning the entire transaction database, it must not be the only applied measure. However, it can help to balance between highly accurate rules (high confidence) and "enough" covered cases to be relevant (high recall).

Specificity (also: True Negative Rate (TNR))

$$\mathrm{spec}_D(\rho) \quad = \quad P(\overline{X} \mid \overline{Y}) \quad = \mathrm{conf}_D(\overline{Y} \to \overline{X}) = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} \in [0,1]$$

Using equivalence transformations from binary logic allows to infer from a rule $X \to Y$ its *modus tollens* $\overline{Y} \to \overline{X}$.[6] However since an association rule

[6] More specifically, the *modus tollens* is an inference rule that involves in addition to the rule $\overline{Y} \to \overline{X}$ the fact $\overline{Y}$ to infer the conclusion $\overline{X}$. The equivalence to $X \to Y$ can easily be verified as follows:

$$X \to Y \equiv \overline{X} \vee Y \equiv Y \vee \overline{X} \equiv \overline{Y} \to \overline{X}$$

If X and Y are sets, the same argument holds by exchanging $\vee$ for $\cup$ and the negation by the set complement.

only predicts correctly a certain fraction of cases (see confidence definition above), the specificity generally is different from the confidence and hence comprises an own proper rule evaluation measure.

Lift

$$\mathrm{lift}_D(\rho) \quad = \frac{P(X,Y)}{P(X)P(Y)} = \frac{\mathrm{conf}_D(\rho)}{P(Y)} > 0$$

The lift is an unbounded measure that compares the joint distribution of antecedent and consequent $P(X,Y)$ with a hypothetical independent distribution $P(X)P(Y)$. Note that the lift is therefore a symmetric measure, that is $\mathrm{lift}_D(X \to Y) = \mathrm{lift}_D(Y \to X)$. When substituting the confidence definition, the lift can also be interpreted as the ratio of the probability of the consequent given the antecedent (that is, the confidence) and the (marginal) probability of just the consequent (that is $P(Y)$).[7] In industrial cooperations, I found the lift definition using the confidence being more intuitive, however, some confusion was caused when telling that the lift is also symmetric. Nevertheless, it is a valuable and accepted measure to assess rule qualities and plays a major role in this thesis.

Given a sequence of temporally ordered time stamps $T = (1,\ldots,t_{\max})$, we can split a given transaction database D into a sequence of time frames $T(D) = (D_1,\ldots,D_{t_{\max}})$. For the sake of brevity we omit the D when we refer to evaluation measures with respect to a certain time frame D_t and just use the t as index. The time series of the absolute support of a rule ρ is defined as follows:

$$\tau_{\text{abs-supp}}(\rho) = \left(\text{abs-supp}_1(\rho),\ldots,\text{abs-supp}_{t_{\max}}(\rho)\right) \in \mathbb{R}^{|T|}$$

2.2 Relational Item Sets

Very often, the data under analysis comes from a relational data source such as a database system and therefore we need to define what transac-

[7] Of course, by same reasoning one can use the recall definition to find an alike statement for the lift.

G	P	S	C
f	y	n	n
f	n	y	y
m	n	y	n
m	n	n	y

$\Rightarrow$

	transaction
t_1	$\{G = f, P = y, S = n, C = n\}$
t_2	$\{G = f, P = n, S = y, C = y\}$
t_3	$\{G = m, P = n, S = y, C = n\}$
t_4	$\{G = m, P = n, S = n, C = y\}$

Table 2.2: *The table on the left is transformed into a set of transactions by combining the attribute and the respective value to an item.*

tions and items are in that parlance. Let us assume the data under analysis is contained in a single database table (which can be achieved always by joining together all relevant tables). Then this table's contents need to be preprocessed in order to be assessed as a transaction database.[8] Each row of the initial table will, obviously, become a transaction, thus resulting in all transactions having the same length (that is, the same number of items). Since different attributes of the original table my have identical values, we must ensure that they are mapped to unique items. Let us consider the relational database table in Table 2.2. The four columns G, P, S and C mean "Gender", "Pregnant", "Smoker" and "Cancer", respectively. The domains of the last three attributes are the same: $\{y, n\}$. If we just combined the values of each row into a transaction, we would end up with a collective set of items $U = \{f, m, y, n\}$. Technically, we cannot represent e. g. the first row of Table 2.2 into a proper item set since it would have to be $\{f, y, n, n\}$ which is not representable within the classic set notion as the item n would only be representable once. Even if this was possible, rules induced from such a transaction database would not be interpretable. For example, the rule $y \rightarrow n$ is ambiguous: Which y and which n are actually referred to?

[8] Note the clash of terminology here: The table of a relational database will be transformed into our transaction database.

As a solution, we create a new collective set of items U by prefixing each attribute value with its attribute (and the symbol "=" to increase readability). Now, the set U reads:[9]

$$U = \left\{\boxed{\text{G=f}}, \boxed{\text{G=m}}, \boxed{\text{P=y}}, \boxed{\text{P=n}}, \boxed{\text{S=y}}, \boxed{\text{S=n}}, \boxed{\text{C=y}}, \boxed{\text{C=n}}\right\}$$

Obviously, each row of Table 2.2 can now be expressed as a subset of U as can be seen on the right of the same table.

2.3 Cooccurrence Graphs

In this thesis we are going to deal exclusively with undirected graphs which we model as a tuple $G = (V, E)$ with vertices V and edge set E with

$$E \subseteq V \times V \setminus \{(v, v) \mid v \in V\},$$

and the constraint

$$(u, v) \in E \;\Rightarrow\; (v, u) \in E$$

to emphasize the undirected character. We will interpret the graphs as cooccurrence graphs where the edges determine the number of cooccurrences (of whatever kind). This is taken into account with an edge weight function for every edge $e = (u, v)$:

$$w : E \to \mathbb{N}_0 \;\text{ with }\; w(e) = w(u, v) = w(v, u).$$

In the figures, this weight is represented as the edge width, thus we use the notion *width* and *weight* interchangeably. Given a subset $W \subseteq V$, we can induce a subgraph $G_W = (W, E_W)$ with

$$E \supseteq E_W = \{(u, v) \mid u, v \in W \wedge (u, v) \in E\}.$$

In the remainder we will sometimes use such a subset W in the context of a graph; it is G_W that we then refer to. A threshold θ defines the subgraph $G_\theta = (V, E_\theta)$ with

$$E_\theta = \{(u, v) \mid (u, v) \in E \wedge w(u, v) \geq \theta\},$$

[9] The single items are boxed to stress that the tree characters make up one item. I will omit this box from now onwards.

that is, as the graph containing only edges with a weight greater or equal to θ. Both operations can of course be combined, that is, $G_{W,\theta}$ represents the subgraph of G induced by the node set W after having removed all edges with weight less than θ.

Since we will deal with sequences of graphs, we denote the temporal index as a superscript. All graphs share the same node set V and differ only in their edge sets or edge weights or both. Given a sequence $G^{(1)},\ldots,G^{(n)}$ of graphs, we define the sum of these graphs as follows: $G_\Sigma = (V, E_\Sigma)$ with

$$E_\Sigma = \bigcup_{i=1}^{n} E^{(i)} \quad \text{and} \quad w_\Sigma(u,v) = \sum_{i=1}^{n} w^{(i)}(u,v).$$

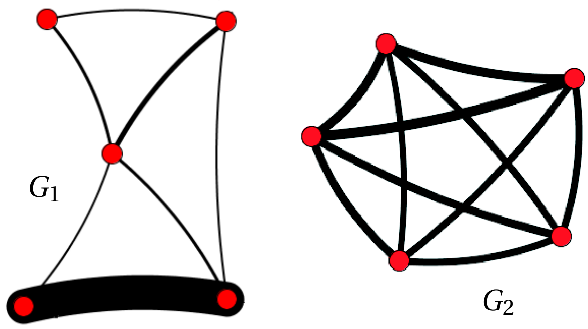
We use the following set of (sub)graph measures to quantify different aspects:

Size	$\mathrm{size}(G_W) = \lvert W\rvert$
Completeness	$\mathrm{comp}(G_W) = \dfrac{2\lvert E_W\rvert}{\lvert W\rvert^2 - \lvert W\rvert}$
Edge Weight	$\mathrm{wght}(G_W) = \sum_{e\in E_W} w(e)$

The size simply represents the number of nodes of the subgraph, whereas completeness refers to the relative number of edges compared to the maximal number. Zero represents an isolated graph (no edges) while a value of 1 designates a clique. Finally, the edge weight simply returns the sum of all edge weights without giving any clue about the distribution of these weights among the edges. Therefore, three additional measures are used: $\mathrm{avg}(G_W)$ calculates the arithmetic mean of all edge weights, $\mathrm{med}(G_W)$ returns the median of the weights and $\mathrm{dev}(G_W)$ represents the standard deviation of the weights. Fig. 2.4 illustrates these intentions with two graphs of the same size.

2.4 Elements of Fuzzy Set Theory

The introduction chapter emphasized (among others) one key requirement of the system under development, namely its simple and intuitive



Measure		G_1	G_2
size	Size	5.0	5.0
comp	Completeness	0.7	1.0
wght	Total Weight	5082.0	7906.0
avg	Average Weight	726.0	790.6
med	Median Weight	257.0	770.0
dev	Std. Dev.	1191.1	129.3

Figure 2.4: *Two graphs with the same number of nodes. G_1 lacks three edges to be complete, therefore the completeness is just 0.7, whereas the clique G_2 yields 1.0. The rather large difference between average weight and median weight for G_1 (in contrast to G_2) indicate an imbalanced edge widths distribution which is strengthened by the large standard deviation value. The two graphs obviously justify this finding. Note, that the layout does not have any influence on the measures and only acts as a visual cue.*

way of parameterization. The techniques offered by the field of Fuzzy Theory [ZADEH 1965] have proven over the decades to offer a robust and powerful tool set for building and applying vague and linguistic models. I am not going to elaborate a historic view of Fuzzy Theory, nor I am intending to present an exhaustive overview. I rather select only those artifacts that are necessary for this thesis. For a more thorough treatment of the topic, see e. g. [KRUSE et al. 2011], from which the below definitions are taken, unless otherwise stated.

One objective of this thesis is to allow queries like the following to run against a given rule base:

> *Show only rules that exhibit a slowly increasing lift and a fast increasing support.*

The crucial point is to handle *vague* terms like "slowly" and "fast" in an appropriate manner. In general, we have to determine how good the value of a certain physical quantity matches a given vague notion: Is a temperature of 40 °C hot? Or: Can a rule's confidence decline of -2 pp[10] per month still be considered stable?

A physical quantity (like temperature or confidence change rate above) is modeled as a so-called *linguistic variable*. *Linguistic terms* are used to describe the (typically real-valued) values of the linguistic variables. For the above examples, we could have the following setting:

Linguistic variable	Linguistic terms
Temperature	cold, warm, hot
Confidence change rate	decreasing, stable, increasing

In classical set theory, each linguistic term would be represented by a subset of the domain of the linguistic variable. Hence, each value then would either entirely match the linguistic term or not at all. Since it is not intuitive to select a crisp threshold for deciding whether a certain value of e. g. the confidence change rate is to be considered increasing, fuzzy sets are used to model degrees of membership to the linguistic terms. They can be considered a generalization of classical indicator functions to the unit interval: Each (classical) subset $X \subset \Omega$ can be defined via a so-called indicator function $\mathbb{1}_X$ on Ω that returns 1 for elements in X and 0 else:

$$\mathbb{1}_X : \Omega \to \{0,1\} \text{ with } \mathbb{1}_X(\omega) = \begin{cases} 1 & \text{if } \omega \in X \\ 0 & \text{if } \omega \notin X \end{cases}$$

We allow the $\omega \in \Omega$ to have gradual degrees of membership to X by generalizing $\{0,1\}$ to $[0,1]$. The new type of set is referred to as a *fuzzy set* with its corresponding indicator function being replaced by a membership function μ_X:

$$\mu_X : \Omega \to [0,1]$$

Definition 1 (Fuzzy Set, Set of Fuzzy Sets)
A fuzzy set μ of a reference set Ω is a function from Ω into the unit interval:

$$\mu : \Omega \to [0,1]$$

[10] *pp* stands for percentage point and represents the arithmetic difference of two percentages.

The set of all fuzzy sets of Ω *is denoted by* $\mathscr{F}(\Omega)$*:*

$$\mathscr{F}(\Omega) = \left\{\mu \mid \mu : \Omega \to [0,1]\right\}$$

Obviously, we will use fuzzy sets to model linguistic terms. From now on, we will use both notions interchangeably. Before we present an example, we need the following two definitions.

Definition 2 (Support of a Fuzzy Set)
Let $\mu \in \mathscr{F}(\Omega)$ *be a fuzzy set over* Ω*. The support of* μ *is defined as the subset of values* $\omega \in \Omega$ *that have positive membership degree with respect to* μ*:*

$$[\mu]_{>0} = \{\omega \in \Omega \mid \mu(\omega) > 0\}$$

Definition 3 (Fuzzy Partition)
A fuzzy partition Π *over a reference set* Ω *is a finite collection (subset) of fuzzy sets over* Ω *whose supports' union equals* Ω*:*

$$\mathscr{F}(\Omega) \supseteq \Pi(\Omega) = \{\mu_1, \ldots, \mu_{p_\Omega}\} \quad \textit{with} \bigcup_{\mu \in \Pi(\Omega)} [\mu]_{>0} = \Omega$$

We explicitly require the latter union condition to make sure that for every value of $\omega \in \Omega$ there is at least one fuzzy set which assigns to ω a positive membership degree. We will use the above-mentioned quotation to illustrate the fuzzy partitioning of the domain of the linguistic variable lift change rate which we will denote with Δ_{lift}. Change rates are numeric, that is in this case we get $\Omega = \mathbb{R}$. Let us further use a partition with five fuzzy sets, that is the linguistic terms are chosen to be as given the the following table:

English meaning	Linguistic term	Fuzzy set modeling the ling. term
fast decreasing	fast decr	$\mu_{\Delta_{\text{lift}}}^{(\text{fast decr})}$
slowly decreasing	slow decr	$\mu_{\Delta_{\text{lift}}}^{(\text{slow decr})}$
unchanged	unch	$\mu_{\Delta_{\text{lift}}}^{(\text{unch})}$
slowly increasing	slow incr	$\mu_{\Delta_{\text{lift}}}^{(\text{slow incr})}$
fast increasing	fast incr	$\mu_{\Delta_{\text{lift}}}^{(\text{fast incr})}$

Figure 2.5 shows one possible instance of the fuzzy partition $\Pi_{\Delta_{\text{lift}}}$. Given these prerequisites together with a current value of the lift change rate Δ_{lift} of x, we can easily compute its membership degree to the linguistic term fast incr as

$$\mu_{\Delta_{\text{lift}}}^{(\text{fast incr})}(x).$$

That is, a lift change rate of 1.8 would be considered 80 % fast increasing and only 20 % slowly increasing. From a formal viewpoint, we would need a function that maps the linguistic terms of a certain fuzzy partition (like fast incr from the partition of Δ_{lift}) to the actual fuzzy sets (like $\mu_{\Delta_{\text{lift}}}^{(\text{fast incr})}$) that represent it. I refrain from this mapping as it only would increase the technical complexity without adding to the understanding. Each fuzzy set will have a subscript and a superscript: the first denotes the respective measure whose domain is constrained whereas the latter denotes the linguistic term that the fuzzy set is describing. Therefore, we agree upon the following informal mapping:

> If term is a linguistic term of the fuzzy partition associated with the domain of a function (measure) m, then the fuzzy set $\mu_m^{(\text{term})}$ represents that linguistic term. If we refer to the change rate Δ_m of m, the fuzzy set $\mu_{\Delta_m}^{(\text{term})}$ shall represent this term.

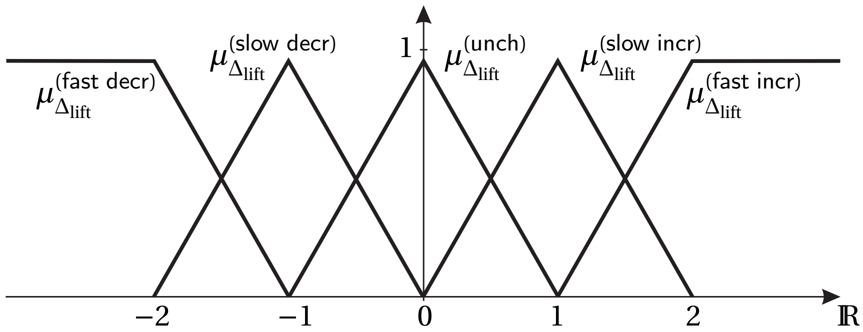


Figure 2.5: *Exemplary instance of a fuzzy partition* $\Pi_{\Delta_{\text{lift}}}$ *containing fuzzy sets modeling the linguistic terms of the linguistic variable* lift change rate.

Since we consider fuzzy sets a generalization of classical (crisp) sets, we can also transfer set operations like union, intersection and complement to the fuzzy framework.

Definition 4 (Fuzzy Union, Fuzzy Intersection, Fuzzy Complement) *Let* μ_A *and* μ_B *be two fuzzy sets on the same reference set* Ω*. The fuzzy union, fuzzy intersection and fuzzy complement are defined as follows:*

Fuzzy Union	$\mu_{A\cup B}(\omega) = \max\big(\mu_A(\omega), \mu_B(\omega)\big)$
Fuzzy Intersection	$\mu_{A\cap B}(\omega) = \min\big(\mu_A(\omega), \mu_B(\omega)\big)$
Fuzzy Complement	$\mu_{\overline{A}}(\omega) = 1 - \mu_A(\omega)$

The above definitions directly carry over to classical sets. For example, the indicator function of the union of two classical sets $A, B \subseteq \Omega$ can be written as

$$\mathbb{1}_{A\cup B}(\omega) = \max\big(\mathbb{1}_A(\omega), \mathbb{1}_B(\omega)\big).$$

The reason why I cover these operations is clearly its disjunctive, conjunctive and negating semantics. Since the generalized set notion allows for a broader range of modeling freedom, the question arises whether the three above-mentioned operations could also be generalized. That is: what are the classes of “and-like”, “or-like” and “not-like” operations (of which the operations in Definition 4 are instances). The idea is to find properties that characterize conjunctive, disjunctive and negating functions. The class of so-called triangular norms (short: t-norms) has been identified to represent the set of functions qualifying as fuzzy conjunctions. Likewise, triangular conorms (short: t-conorm) characterize the class of fuzzy disjunctions.

Definition 5 (Triangular Norm, t-Norm)
A binary function $\top : [0,1]^2 \to [0,1]$ *is called a* triangular norm*, or* t-norm *for short, if and only if the following properties hold:*

(T1)	*Commutativity*	$\top(x,y) = \top(y,x)$
(T2)	*Associativity*	$\top(x,\top(y,z)) = \top(\top(x,y),z)$
(T3)	*Monotonicity*	$y \le z \Rightarrow \top(x,y) \le \top(x,z)$
(T4)	*Boundary Condition*	$\top(x,1) = x$

A t-norm can be considered a conjunction that is generalized from binary definition space $\{0,1\} \times \{0,1\}$ to the continuous definition space $[0,1] \times [0,1]$ where the "corners" (that is, $(0,0)$, $(0,1)$, $(1,0)$ and $(1,1)$) yield the same results as the binary conjunction. Since Definition 5 is not unique, we can define a variety of different t-norms exhibiting different semantics. The function graphs are depicted in Figure 2.7.

Minimum t-Norm	$\top_{\text{min}} = \min(x,y)$
Algebraic Product	$\top_{\text{prod}} = x \cdot y$
Łukasiewicz t-Norm	$\top_{\text{Łuka}} = \max(x+y-1,0)$
Drastic Product	$\top_{-1} = \begin{cases} 0 & x,y \neq 1 \\ \min(x,y) & \text{else} \end{cases}$

The spectrum of t-norms is limited by $\top_{\text{drastic}}$ as the smallest t-norm and $\top_{\text{min}}$ being the largest t-norm. The above-mentioned t-conorms are defined in a similar way:

Definition 6 (Triangular Conorm, t-Conorm)
A binary function $\perp : [0,1]^2 \to [0,1]$ *is called a* triangular conorm*, or* t-conorm *for short, if and only if the following properties hold:*

(C1)	*Commutativity*	$\perp(x,y) = \perp(y,x)$
(C2)	*Associativity*	$\perp(x,\perp(y,z)) = \perp(\perp(x,y),z)$
(C3)	*Monotonicity*	$y \le z \Rightarrow \perp(x,y) \le \perp(x,z)$
(C4)	*Boundary Condition*	$\perp(x,0) = x$

Obviously, (T1)–(T3) directly match (C1)–(C3), whereas (T4) and (C4) differ in the identity element of the boundary condition. Again, some of the most prominent t-conorms are given below. Their function graphs are shown in Figure 2.8.

Maximum t-Conorm	$\perp_{\mathrm{max}}$	$= \max(x,y)$
Algebraic Sum	$\perp_{\mathrm{sum}}$	$= x+y-xy$
Łukasiewicz t-Conorm	$\perp_{\text{Łuka}}$	$= \min(x+y,1)$
Drastic Sum	$\perp_{-1}$	$= \begin{cases} 1 & x,y \neq 0 \\ \max(x,y) & \text{else} \end{cases}$

A last important operation that needs to be carried over to the fuzzy framework is the negation. Again, we claim that a fuzzy negation (or: fuzzy complement) shall coincide with the binary negation on the boundary, that is the complement of 0 shall be 1 and vice versa. In between we require a non-increasing behavior: If the argument decreases, the complement must not increase. We use the following definition:

Definition 7 (Fuzzy Negation)
A function $\sim: [0,1] \to [0,1]$ *satisfying the conditions*

$$\sim 0 = 1, \quad \sim 1 = 0$$

and

$$x,y \in [0,1]: \ x \le y \ \to \ \sim x \ge \sim y \quad \textit{(that is, } \sim \textit{ is non-increasing)}$$

is called a negation *or* complement.

Again, there is a variety of different negations. Four examples are given below. Their function graphs are shown in Figure 2.6.

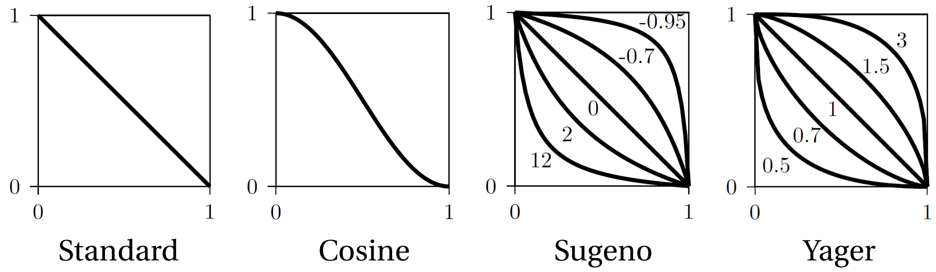


Figure 2.6: *Examples of fuzzy negations.*

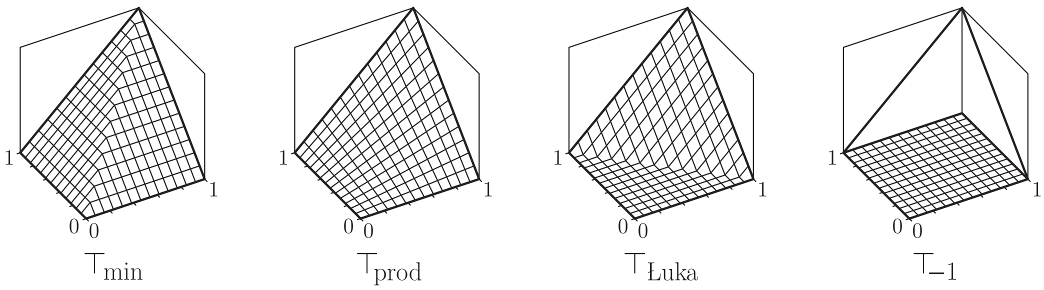


Figure 2.7: *Illustration of four prominent t-norms.*

$$\begin{array}{llcl}
\text{Standard Negation} & \sim x & = & 1 - x \\
\text{Cosine Negation} & \sim x & = & \frac{1}{2}\big(1 + \cos(\pi x)\big) \\
\text{Sugeno Negation} & \sim_\lambda x & = & \frac{1-x}{1+\lambda x}, \quad \lambda > -1 \\
\text{Yager Negation} & \sim_\lambda x & = & (1 - x^\lambda)^{\frac{1}{\lambda}}
\end{array}$$

More information on the topics discussed above can be found in [KRUSE et al. 2011]. If not stated otherwise, I will use the algebraic product, algebraic sum and standard negation as default t-norm, t-conorm and fuzzy negation, respectively. Using the definitions above allows us to compose

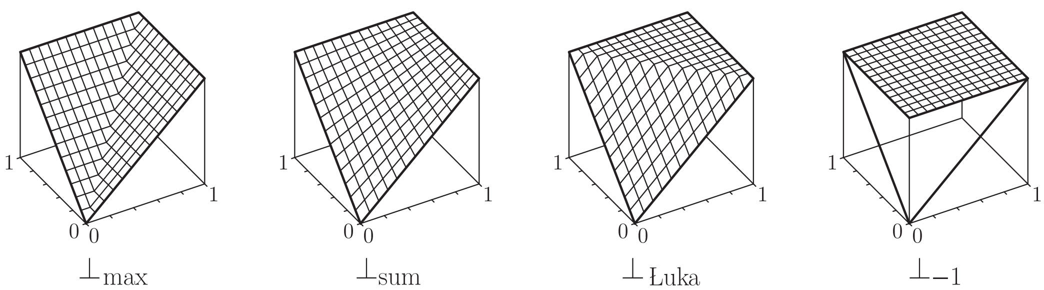


Figure 2.8: *Illustration of four prominent t-conorms.*

complex propositions and also to compute the membership degrees of them. We will refer to these propositions as linguistic or fuzzy concepts. The linguistic statement from the first paragraph of this section is now representable with the following fuzzy concept ξ:

$$\xi = \left\langle \Delta_{\text{lift}} \text{ is } \mathsf{slow\ incr} \text{ and } \Delta_{\text{supp}} \text{ is } \mathsf{fast\ incr} \right\rangle \tag{2.1}$$

Generally speaking, we define a fuzzy (or linguistic) concept with the following recursive definition:

Definition 8 (Fuzzy Concept, Linguistic Concept)
Let $m : \Omega \to \mathbb{R}$ be a function whose domain is equipped with a fuzzy partition Π_m. Then for all $\mu \in \Pi_m$ the proposition

$$\langle m \text{ is } \mu \rangle$$

is called a fuzzy concept *or* linguistic concept. *Further, for any fuzzy concepts ξ and ζ the following propositions are fuzzy concepts, too:*

$$\langle \xi \text{ and } \zeta \rangle, \quad \langle \xi \text{ or } \zeta \rangle \quad \textit{and} \quad \langle \text{not } \xi \rangle$$

It is obvious that we are going to model the junctors and , or and not with t-norms, t-conorms and fuzzy negations, respectively. Literal concepts, that is concepts of the form $\langle m \text{ is } \mu \rangle$ are then assigned a membership degree via the above-agreed implicit term-fuzzy set mapping.

Let us assume the current state of the model shows a lift change rate of x and a support change rate of y, we can compute the membership of that state with respect to the fuzzy concept ξ (equation 2.1) as:

$$\xi(x, y) = \top_{\text{prod}}\left(\mu_{\Delta_{\text{lift}}}^{(\mathsf{slow\ incr})}(x), \mu_{\Delta_{\text{supp}}}^{(\mathsf{fast\ incr})}(y)\right)$$

2.5 Elements of Evolutionary Algorithms

Evolutionary algorithms are a class of heuristic algorithms that can be applied to find (or approximate) solutions of optimization problems for which there are no efficient deterministic solvers. The principles underlying evolutionary algorithms are borrowed from the evolution theory [DARWIN 1859]. The basic idea is to encode solution candidates (so-called individuals) in a special—often string-based—representation (the chromosome) and randomly initialize a large amount of such individuals, forming a population. Different locations (character positions) of a chromosome correspond to different properties of the respective solution candidate and thus can be seen as the genes of the corresponding individuals. The actual values at these locations then define the specific characteristic of a solution and are the silicon equivalent of alleles. A fitness function assigns to each individual a value quantifying the goodness of the encoded solution. An evolutionary algorithm iterates through multiple epochs in each of which the following procedures are applied [GOLDBERG 1989]:

1. **Mutation and Crossover**
   Both operations are nature-inspired means to modify and create new individuals. Mutation randomly alters the alleles of genes of *single* individuals whereas during crossover gene segments of *two* individuals are exchanged thus mimicking sexual reproduction. The probability with which both operations occur and the number and regime in which segments of chromosomes are crossed over are parameters of the evolutionary algorithm.

2. **Selection**
   At the end of each epoch a new population is created out of the current one. There are, again, several strategies that can be applied here, all of which adhere to the "survival of the fittest"-principle: only the best individuals (according to the fitness function) are transferred to the new population. To ensure a constant population size, new individuals are added based on different strategies. These strategies also represent parameters of the evolutionary algorithm.

First experiments with evolutionary algorithms go back to John Holland [HOLLAND 1975] in the seventies. The general idea of evolutionary algorithms is to globally explore the solution space in the beginning and

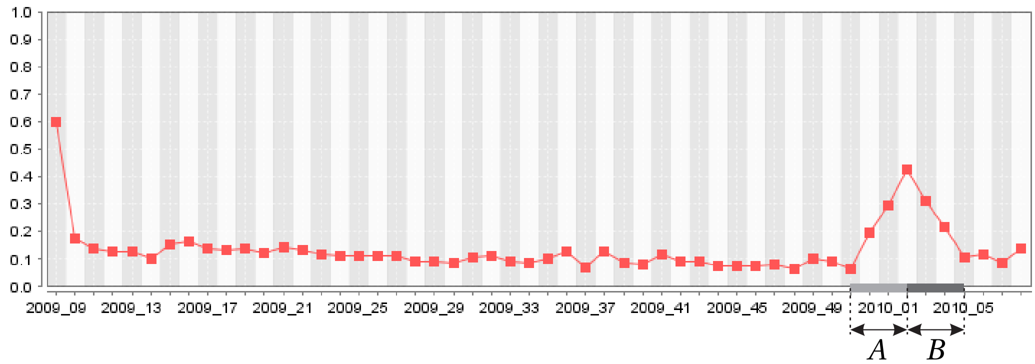


Figure 2.9: *The borders of the two time frame intervals A and B are set such that the membership to the concept* peak *is maximal.*

then enforce local optimization to find best individuals (encoding near-optimal solutions). The global exploration is achieved by first randomly initializing the first population and second in starting with relatively high mutation and crossover rates. The reduction of those rates in later epochs together with the selection process ensure that individuals with good fitness accumulate. Further reading is directed to [KRUSE et al. 2011].

I am using the above-mentioned evolutionary principle to find good compositions of (time frame) intervals that satisfy certain local constraints together with high membership degrees of linguistic concepts. This will allow to detect temporal composite patterns: A peak, for example, consists of a period of steep incline (of some measure), immediately followed by a steep decline (of the same measure). A plateau, however, allows for a variable space in between the two flanks. Figure 2.9 and Figure 2.10 show two examples. The peak in Figure 2.9 is clearly pronounced, whereas the shape of the time series in Figure 2.10 may be subject to discussion whether it can be characterized as a plateau or not. But this is exactly the key feature of the fuzzy matching approach of this thesis: allow the user to model generic concepts (such as “fast increasing”, “peak” or “plateau”) and then gradually fine-tune these concepts according to the data under analysis and the user’s reasoning.

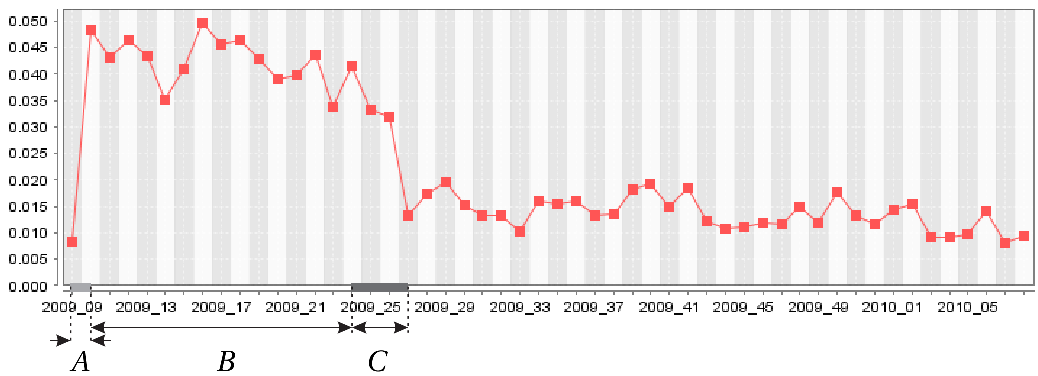


Figure 2.10: *The time frame intervals A and C mark the flanks of the concept* plateau *with B representing the middle part. In a "clean" instance of a* plateau, *B would be less wiggly and more straight. Obviously, this pattern would not get an utmost high membership degree to the concept, however, it might be pronounced enough to be worth looking at and therefore a means is needed (and provided in this thesis) to return such patterns.*

3 Visualization

This chapter is dedicated to a thorough treatment of how to intuitively represent patterns and their changes. After sketching the underlying data structures in Section 3.1, I will discuss the evolution of the glyphs (the visual entities used to represent the individual patterns) in Section 3.2 before assembling the ideas into a full model visualization in Section 3.3. Section 3.4 covers the way the temporal component is addressed. I conclude the chapter by comparing my suggestions to other existing techniques in Section 3.5.

3.1 Data Structures

The decision towards association rules as the model type under investigation[1] will pose the challenge of not only representing a large set of artifacts (that is, rules), each having a set of metric properties, but also to deal with the changes thereof. Figure 3.1 shows the entity relations that are involved between the model, the contained artifacts and their properties. Since most properties have a (time) series of values, there is also a 1-to-n relationship between properties and their values.

Let us illustrate this with a little example rule set R containing the following rules:

$$\rho_1 = a, b \rightarrow c, \quad \rho_2 = b \rightarrow d \quad \text{and} \quad \rho_3 = b, c \rightarrow e.$$

[1] See Section 1.3 on page 7.

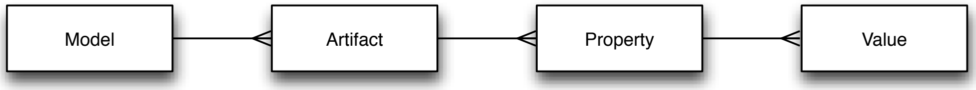


Figure 3.1: *Entity relationships amongst the model artifacts.*

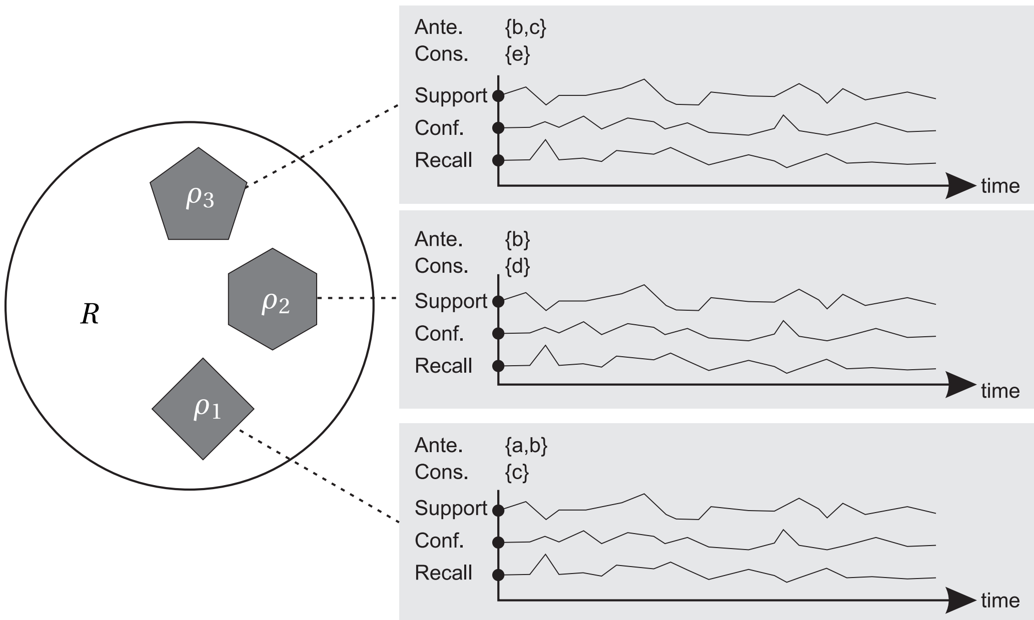


Figure 3.2: *Exemplary entity relationships amongst the model artifacts.*

Figure 3.2 illustrates the involved data: The only model instance is obviously R. The three rules are depicted as dark gray shapes. Each rule has assigned a map of properties. I use the term *map* here since it matches the implementation. The keys represent the property names. The values can be single-valued or list-valued. Single-valued properties comprise the antecedent and consequent, list-values properties obviously represent the time series of the individual rules evaluation measure values such as support, confidence, etc.

I will propose a representation for a single artifact (a so-called glyph) of a single point in time in the next section. The whole model is then represented naturally as the collection of those glyphs rendered into the same area. The temporal treatment is discussed in Section 3.4.

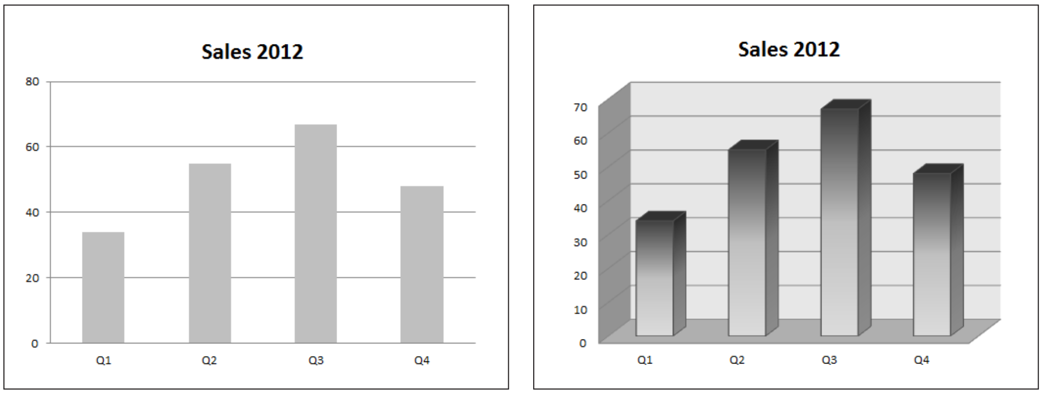


Figure 3.3: *Left: good data-to-ink ratio, right: bad data-to-ink ratio.*

3.2 Glyph Construction

As illustrated in Figure 3.2, each rule is assigned a collection of property-values pairs with most values being time series. I will consider here the visual representation of a single point in time. The data assigned to a single point in time of a single rule basically contains the rule itself (antecedent and consequent items) and a collection of evaluation measures values. One guideline I followed for the glyph creation was the claim for a good *data-to-ink ratio* as it was suggested by Edward Tufte in [TUFTE 2001]. The idea is to use as little ink (or pixels in terms of screen real estate) to encode all relevant information. Visual clutter shall be avoided if it carries no additional value (other than being questionably aesthetic). Figure 3.3 shows an example of bad and good data-to-ink ratio for a bar chart. It is even possible to encode additional information by reducing the actual pixel count (or leaving it at least constant) as shown in Figure 3.4. Even though I did not drive these ideas to the extreme, emphasis was put on having as little arbitrariness in the graphical representations as possible.

3.2.1 Encoding Antecedent and Consequent

The overall glyph shape shall be circular. The main reason for that decision was to make it rotation-invariant and thus avoid a potential source of

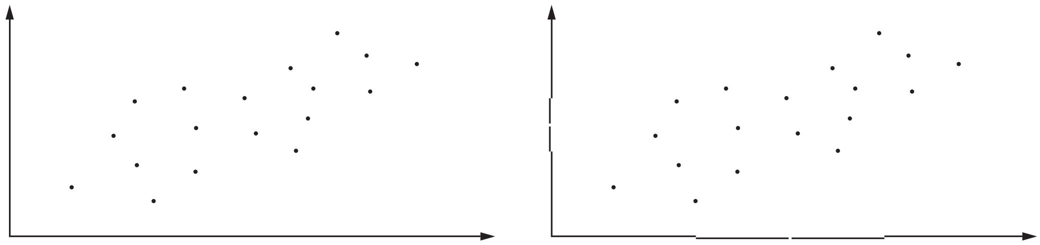


Figure 3.4: *While keeping data-to-ink ratio constant or even reducing it, one can still include information into the chart. The right chart encodes the arithmetic mean by the little gap on the axes while the standard deviation is represented by offsetting the respective segment of the axes. Examples adopted from [*TUFTE *2001].*

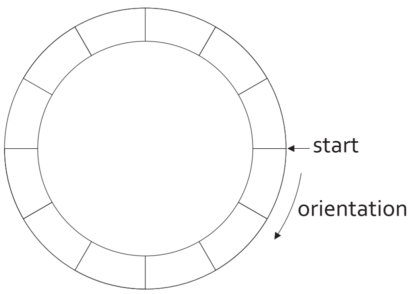


Figure 3.5: *Draft of the graphical representation of a single rule glyph. The outer rim segments encode the antecedent of the rule. The inner part will be used for the consequent (if it consists only of a single item).*

arbitrariness. A sketch of a rule glyph is depicted in Figure 3.5. The outer ring is reserved for encoding the rule's antecedent as follows: The outer ring is divided into as many equiangular segments as there are different items in the database. Each item gets assigned a unique segment (e. g. by sorting them lexicographically and assigning them counter-clockwise starting at the rightmost location). If an item is contained in the rule's antecedent, its respective segment is filled. Let's assume our item set contains twelve items (a to l), then the left glyph of Figure 3.6 shows the item-to-segment assignment. The middle glyph would then encode the antecedent $\{a, b, h\}$. The representation of the consequent could be naturally implemented by an inner ring using the same rationale as for the antecedent. The right glyph in Figure 3.6 would hence represent the rule

$$a, b, h \rightarrow f, l.$$

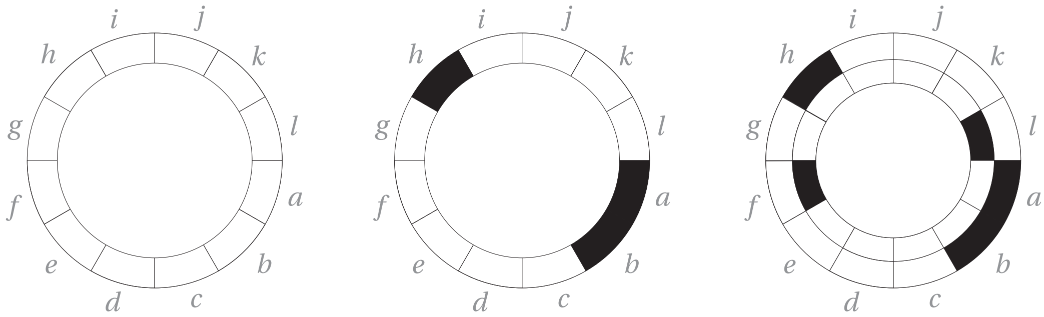


Figure 3.6: *Left: Item-to-segment assignment. Middle: Encoding the antecedent* $\{a, b, h\}$*. Right: Possible representation of consequent and encoding the rule* $a, b, h \rightarrow f, l$*.*

There are reasons why I do not use this latter approach to encode the consequent. First, when the total item set is rather large, it will be hard to distinguish the outer (antecedent) ring from the inner (consequent) ring. This becomes especially problematic as I will scale the glyphs according to certain rule evaluation measures in the next section. Second, in most real-world applications that I applied the visualization, the induced association rules had exactly one item in the consequent.[2] Therefore, I assign to each item a color and use the consequent item's respective color to fill the interior of the rule glyph. The same principle is applied to the outer ring for antecedent encoding when the origin of the rules is of the kind as discussed in Section 2.2. Each element of an attribute's domain $\mathrm{dom}(A_i)$ gets assigned a unique color (where colors may be reused across domains, that is, the color blue can encode a value of different attributes). Figure 3.7 shows an example of a glyph for such a rule. Let C be the attribute whose domain is the set of possible consequent items. The domains of attributes A, B and D represent the items that may occur in the antecedent. After assigning each value a unique color, the glyph of the rule

$$a_1, b_2, d_3 \rightarrow c_2$$

is shown in Figure 3.7.

[2] See Section 2.2 on page 22 for details.

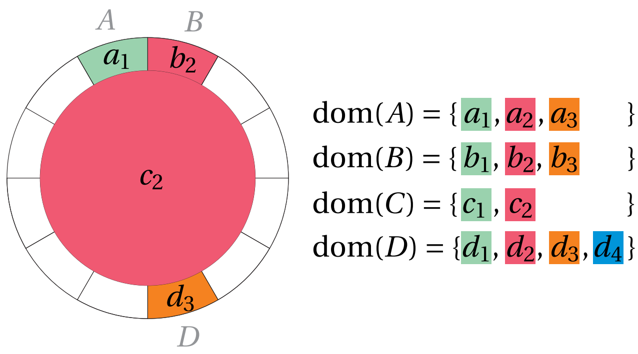


Figure 3.7: *Final version of rule glyph.*

3.2.2 Encoding Rule Evaluation Measures

After having a glyph that encodes a rule's antecedent and consequent, we need means for representing the quantitative details of a rule, that is, the values of its evaluation measures (of a single point in time—I will address the visualization of the temporal change of them later). I suggest to use the following glyph properties to encode the values of rule evaluation measures:

- x-location of glyph
- y-location of glyph
- size (area) of glyph
- style of the glyph center:
  - angle of pie chart in the glyph center
  - saturation of solid fill

The first three items should be clear from the discussions so far (and will be revisited in Section 3.3). The last item can be implemented in two different ways (depending on the user's intuition): In both cases the glyph center is used to encode a rule measure whose value is bounded (e. g. lies in the unit interval). One option is to let a pie chart occupy the center. The higher the rule measure value, the larger the pie segment. The top

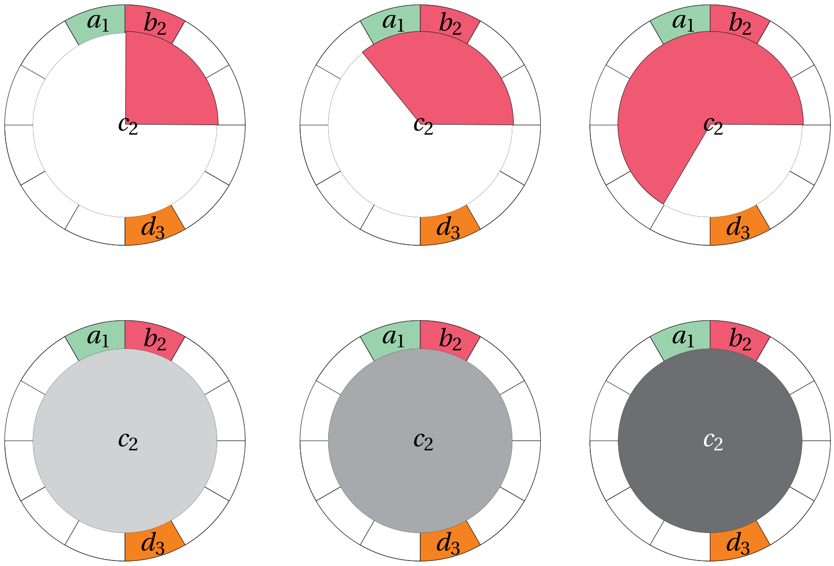


Figure 3.8: *Two options for the glyph centers: pie chart and saturated color.*

row of Figure 3.8 shows a series of glyphs with the pie chart encoding the values $^1/_4$, $^1/_3$ and $^2/_3$. Another option is to keep the fully filled inner circle and adjust the saturation of its color (which still denotes the consequent, remember). The bottom row of Figure 3.8 illustrates this.[3]

3.2.3 Encoding Overlapping Rules

Up to now, we assumed the rules to cover mutual disjoint sets of database objects, that is, every entry of the database was described by exactly one rule antecedent. This can be easily achieved by requesting a fixed set of attributes for every rule. If the user, however, is interested in general rules where database entries may be covered by multiple rules (e. g. because one rule is a specialization of another), we have to cater for this fact by depicting the mutual overlap.

Consider a population for which we assess the probability of having lung cancer. Let the cancer probability for a male person be 15%, that is, the rule

$$\text{Gender} = \text{male} \rightarrow \text{Cancer} = \text{y}$$

[3] I used gray shades here to make the effect apparent also in a monochrome print of this thesis.

	male		female	
	smoker	no smoker	smoker	no smoker
cancer	60	15	75	10
no cancer	140	285	225	190

Table 3.1: *Example database from which two rules (“male → cancer” and “male ∧ smoker → cancer”) were assessed and depicted in Figure 3.9.*

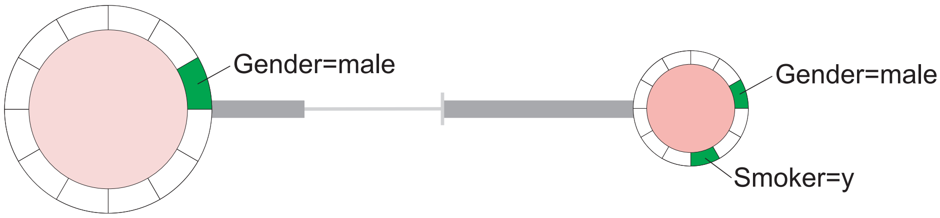


Figure 3.9: *Visualizing the overlap of two rules. Since “male∧smoker → cancer” is a specialization of “male → cancer”, the set of database cases covered by the first rule are fully contained in the set of covered cases by the second rule, hence the 100%-indication to the right. The common set of database cases comprises 40% of the cases covered by the more general rule, hence the smaller indicator to the left.*

has a confidence of 0.15. Let this confidence increase to 30% when the additional information that the person is a smoker is known. The respective rule is

$$\text{Gender} = \text{male} \wedge \text{Smoker} = \text{y} \rightarrow \text{Cancer} = \text{y}.$$

Clearly, all persons covered by the antecedent of the second rule are also covered by the antecedent of the first rule, hence they cover non-disjoint sets of cases. To depict this, we connect both rule glyphs by a line in a chart whenever their antecedents covers are not disjoint. Further, we compare the cardinality of the covers intersection to the support of both rules. The two ratios between intersection cardinality and the two rule supports are indicated as a bar chart on that connecting line. The 100%-mark is located in the middle, whereas the 0%-mark is on the rules' outer border. Figure 3.9 depicts the example situation of lung cancer above. The used numbers of cases are given in Table 3.1 for the sake of completeness.

3.3 Model Visualization

After having introduced the glyph for a single rule and the mutual overlap of two rules, it has to be decided how an entire rule model, that is a set of association rules, shall be represented. The main problem is to determine the location of each glyph. There are in principle two competing approaches: Use the values of two rule evaluation measures directly as the x- and y-location or use some dimension reduction technique to map a high-dimensional vector of rule evaluation measure values down into the plane. The latter approach is appealing as it would allow to use more than just two rule measure values for locating the glyphs. However, the resulting x- and y-coordinates would have no direct semantic; only the glyphs' proximity would carry any information. However, my objective was to encode as many quantitative information directly so that I am opting for the first method: using two rule evaluation measures to directly determine the location of the glyph.

With these ingredients at hand, I recommend the following mapping for assigning rule evaluation measures to the respective glyph properties:

Glyph property	Rule evaluation measure
x-location	Recall
y-location	Lift
size (area)	Support
glyph center	Confidence

This mapping has been empirically found to be most intuitive to users. Rules that cover more database cases are easier to spot due to their larger size. The confidence as the "correctness ratio" of the rule—either represented as a pie chart segment or the (de)saturation of the center—is directly perceivable as a more or less faint appearance. Rules with high lift will be located above rules having a low lift and finally, rules with a high recall stretch farther out to the right than rules with small recall.

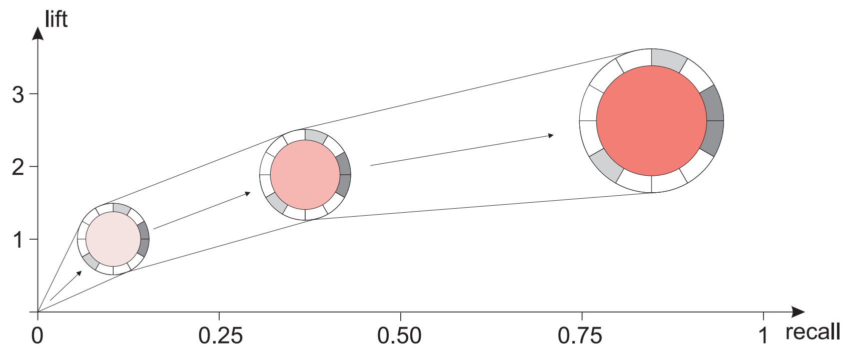


Figure 3.10: *A rule at three different times. Initially, the rule did not cover any database case, that is its initial icon is just a point at the origin. As time passes, the support increased (growing size of the glyph), the recall and lift did alike (glyph is moving to the upper right-hand corner). The user would be presented an animation with a smooth transition between the three states.*

3.4 Representation of Time

To present the temporal evolution of a rule set (with respect to the evaluation of selected measures), an animation is generated that displays the current state of the rule set at any given time (frame). Figure 3.10 depicts this idea with the same rule at three different times. If the consequent of the depicted rule is a failure class, this rule would be a candidate for a pattern that needs further investigation: the number of affected database cases (support) increased over time. The same can be stated for the lift and confidence. The latter means that the problem became more and more severe since its probability increased.

However, the more data there is under analysis, the more patterns and thus, rules, can be found. It is not unusual to have several hundreds induced from a database. Clearly, this would clutter the visualization beyond recognition. I therefore propose a method of thinning out the number of rules to be actually displayed in Chapter 4. Figure 3.11 is taken from a real-world application (and is discussed later in Chapter 6) and illustrates the clutter that occurs when too many rule glyphs are displayed.

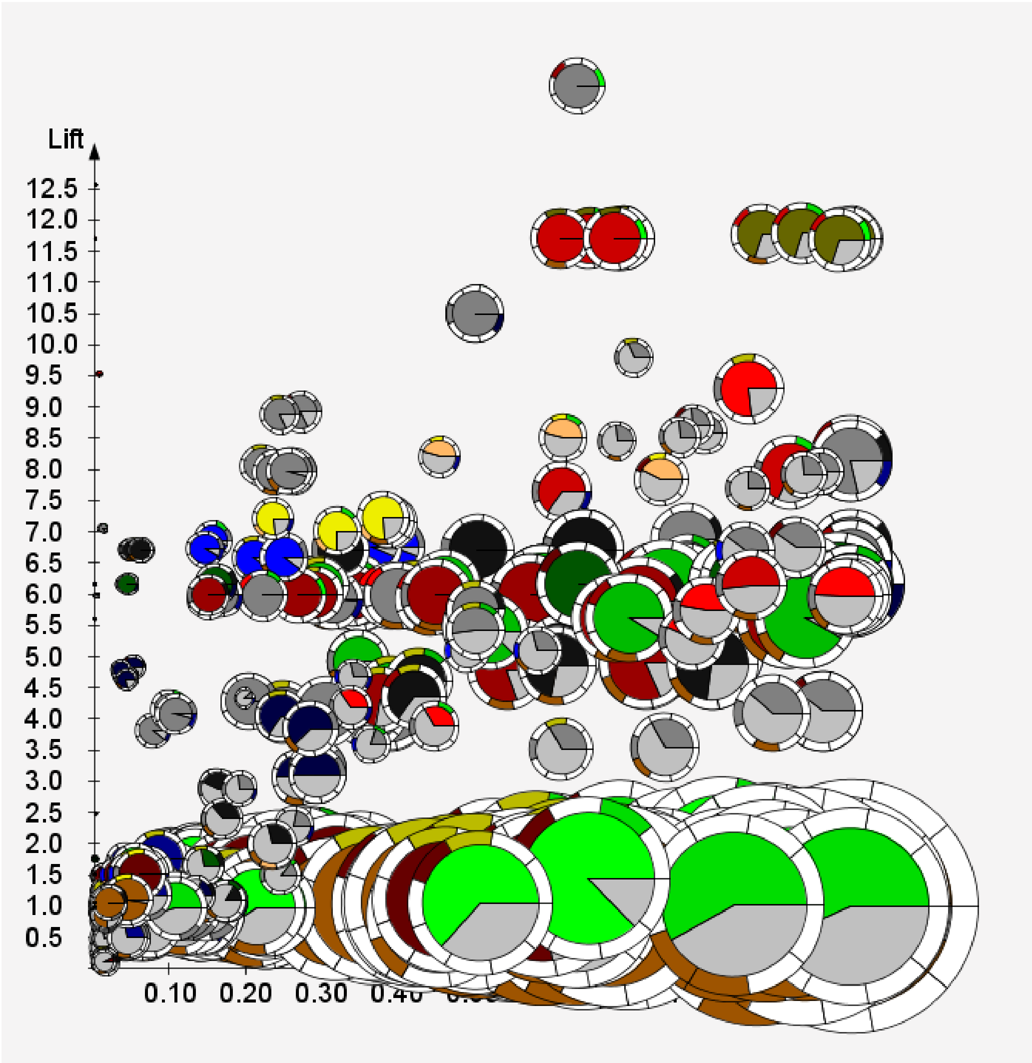


Figure 3.11: *Full rule set with 1585 rules from a real-world application (repeated and discussed in Section 6.4). It motivates the need for a filtering method in order to thin out the rule set.*

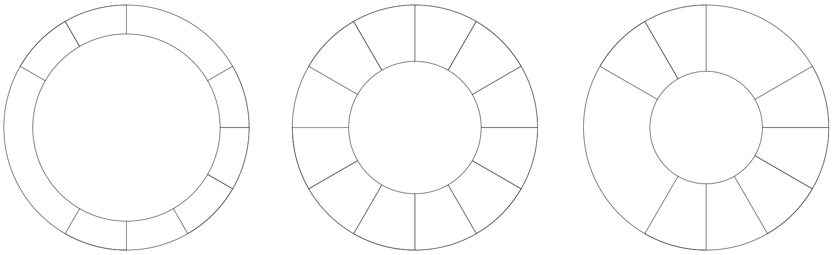


Figure 3.12: *The rule glyph could be loaded with even more information encoded into further features such as unequal segment angles, different inner-to-outer ratios or a combination of both. I refrained from such attempts as it would unnecessarily clutter the image and not aid the visual perception.*

3.5 Related Work

When it comes to multi-dimensional data visualization, several different well-known approaches come to mind. One of the most prominent being the Chernoff faces [CHERNOFF 1973] where numbers are mapped to certain face features (like eye size, face shape, nose angle and size, mouth position, etc.). Figure 3.13 illustrates an example data set where also the face location carries information (as it is the case for the rule glyphs introduced above). In some way my proposed glyphs share certain properties in common: Furthermost the fact that the glyph itself carries a multi-dimensional set of features (metric as for the evaluation measures and nominal as for the antecedent and consequent encoding). However, I tried to find a balance between the density of information pushed into the glyphs and the ease of readability. Therefore, I did not use different glyph shapes or other potential means of cramming additional information into the glyphs as Figure 3.12 might suggest.

WONG et al.[4] propose a three-dimensional association rule visualization that encodes evaluation measures (support) as well as the items of the rule itself. As visually appealing this approach may be, the data-to-ink ratio is weak and furthermost this approach does not comply well with representing temporal change of the rules' attributes.

[4] See Section A.1.1 on page 141 for more details.

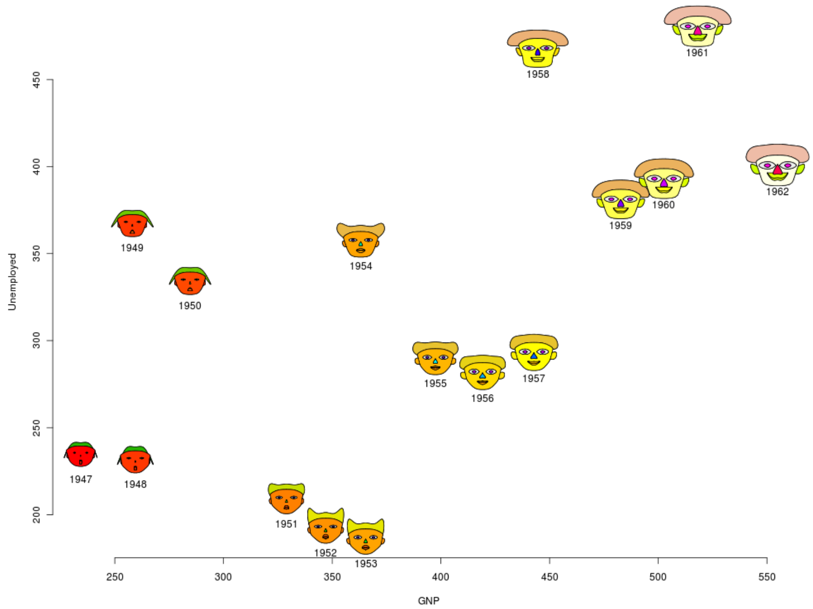


Figure 3.13: *Example data set illustrated using Chernoff faces [*OGASAWARA*].*

YANG[5] utilizes parallel coordinates [INSELBERG 1985] to represent the rule antecedent and consequent. The main reason against a visualization with parallel coordinates is the difficulty to implement an intuitive temporal change metaphor. Another reason is the inherent arbitrariness that is connected to parallel coordinate plots. The order of the axes, for example, totally governs the visual appearance of the plot. In the above-mentioned reference, the axes contain items for which there is generally no natural order. However, the order of them dominates the actual appearance: Different orders may lead to completely different (but semantically equivalent) plots.

HOFMANN et al.[6] use mosaic plots to encode rule evaluation measures. They restrict the rules to those having a common (single-item) consequent; a choice I also use later in the evaluation chapter to restrict the rule set. In addition, an exhaustive set of antecedents is represented which clearly is only useful for a small set of rules (for example, after having ap-

[5] See Section A.1.2 on page 143 for more details.
[6] See Section A.1.3 on page 144 for more details.

plied some kind of filtering technique as described in the next chapter). Indeed, I used similar visualizations in [STEINBRECHER 2006] and [STOBER et al. 2009] to present rule evaluation measures of small rule sets.

BRUZZESE and DAVINO[7] construct a binary item-to-rule membership matrix augmented with support and confidence values which is then "flattened" by some dimension reduction algorithm. The result of this mapping is a two-dimensional map where proximity can be seen as a measure of item-to-rule containment and also to assess similarities among the rules themselves. The animation of the contents of such a map (representing temporal similarity change) is quite appealing. Of course it requires a dimension reduction technique that delivers similar results when the input similarity matrix changes slightly (if the generated map would look completely different given a slightly changed input, the technique would be inapplicable for animating changes).

BLANCHARD et al.[8] devised a three-dimensional glyph encoding several rule evaluation measures. Even though I opted against three-dimensional visualizations here, the underlying ideas for designing the glyphs were similar: simple geometric shapes, easily perceivable quantities (size, opening angles, etc.) and balance between information content and comparability. If there should ever be a need to translate my suggested visualizations into the third dimension, the glyphs by BLANCHARD et al. should be considered a starting point.

3.6 Summary and Discussion

In this chapter I introduced a visualization method for association rules by suggesting a glyph that is capable of encoding a rule's antecedent and consequent together with the values of four rule evaluation measures. The temporal change of these measures is visualized by an animation that linearly interpolates between two consecutive time frames. Depending on the choices of the rule evaluation measures the proposed visualiza-

[7] See Section A.1.4 on page 145 for more details.
[8] See Section A.1.5 on page 147 for more details.

tion technique is capable of delivering a four-dimensional representation in two-dimensional space when a still image of one time frame is considered. When including the time, five dimensions can be displayed in a two-dimensional animation. The glyphs also encode the antecedent and consequent of the rule which, in theory, adds to the dimensionality but I do not count them as additional dimensions. They rather add a multinomial component to the glyph. Choosing the rule evaluation measures wisely allows to provide quite catchy rules of thumb to users that do not necessarily have a statistical background. Using the rule assignment suggestions given above, a user can identify potential interesting rules by looking for "large glyphs in the upper right-hand corner of the chart" [STEINBRECHER 2006, STEINBRECHER and KRUSE 2007a, 2008b, KRUSE and STEINBRECHER 2010].

Whenever glyphs are designed the question is how much bias is introduced that may mislead the user [WARD 2002, CHEN et al. 2008]. Ward distinguishes different types of biases with perception-based bias being the one that matters most for the presented glyph design. Perception-based bias addresses the fact that certain graphical features (such as lengths of bars starting at the same base) are easier to distinguish than for example angles (of pie charts). Figure 3.14 illustrates this phenomenon. The problem with pie charts mainly exists when used with multiple segments. In the rule glyphs I suggested there will only be two segments at most, thus reducing the perception bias. Since the general glyph shape is circular, pie charts fit better and use the space more efficiently. Speaking of circular shapes always triggers the psychological objection that areas are harder to compare than, say, lengths [SPENCE and LEWANDOWSKY 1991]. While it is indeed hard to assess the factor by which a given circle is larger (area-wise) than a reference circle, it is quite easy to determine the relative sizes, that is, which circle is larger than the other. This already gives valuable insight which rule might be more or less important. However, there is room for improvement when the user shall be required to assess absolute glyph properties.

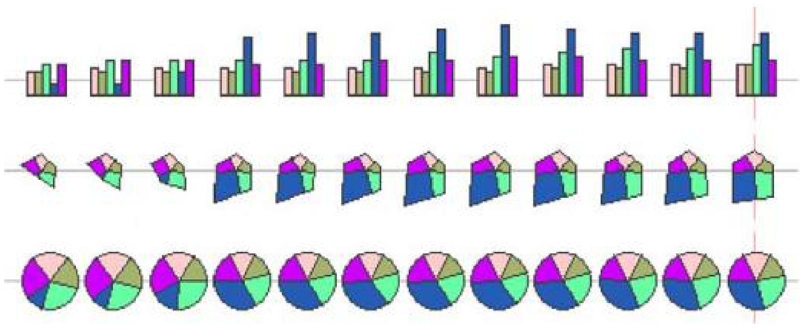


Figure 3.14: *Example glyphs with different perception bias: Attributes within and between glyphs are easier to compare with so-called profile glyphs (top row) [*WARD *2002].*

4
Linguistic Filtering

This chapter introduces a framework that is used to assess the temporal characteristics of time series induced by the evaluation measures of association rules. Section 4.1 describes the features that the framework will possess before Section 4.2 introduces the principal ideas after which they are translated into specific algorithms in Section 4.3 and Section 4.4. Section 4.5 transfers the ideas to other models than association rules after which the chapter finishes with related works in Section 4.6 and a summary in Section 4.7.

4.1 Requirements

The last chapter has shown that any glyph-based model visualization will eventually suffer from a cluttered view, especially when showing an animation of the temporal change of the respective glyph features. One could use this argument to rule out any glyph approach, however, I found this method appealing as it, on the one hand, allows to encode a multitude of features in a non-reductive[1] way and, on the other hand, was found intuitive by users (see Section 6.4.6 for details). The objective in this chapter is to develop a framework that allows to filter sets of association rules (and other models as we will see later) based on the temporal evolution of their respective properties (that is, rule evaluation measure

[1] By "non-reductive" I refer to methods that do not employ any means of dimension reduction such as multi-dimensional scaling (MDS) or principal component analysis (PCA) which would render the new axes rather meaningless compared to true evaluation measure values.

values). I will discuss three aspects in greater detail, which all are a requirement to a successful application of the filtering approach.

Intuitiveness

Whatever method will be used to filter sets of model artifacts (here: sets of association rules), it will much likely offer parameters that affect the results. These parameters shall be intuitive and have a specific meaning that can easily be understood and judged by users that not necessarily have a statistics or data mining background. Further, suggestions for these parameters should be offered as default settings extracted from the model under analysis.

This requirement is particularly important as association rules are rather user-friendly in terms of parameterization (see Section 1.3). Putting on top of this an over-engineered approach with artificial parameters would entirely shrink the acceptance and usefulness of the framework.

Instant Feedback

Even with the above-mentioned intuitive parameters, deep insight into the model structure may only be gained when there is an immediate feedback on any parameter change. This will allow ad-hoc changes to acquire a natural feeling on how the parameters affect the resulting model subset. Further, it can be foreseen that the learnt parameter-response interaction of one model allows for a quicker assessment of future models as the user may anticipate the impact of his changes more easily.

Visual Interaction

The last ingredient to user acceptance next to intuitive parameters and the instant response to changes is an intuitive interface to actually carry out these changes and assess the results. A simple user interaction (in terms of exploration of results and parameter setting) is therefore of paramount importance.

4.2 Filtering Approach

Filtering the rule set for predefined temporal patterns first and foremost calls for a language or framework in which to define the desired behavior. I decided to use an approach based on fuzzy concepts[2] to describe the temporal properties of the evolution of rule measures. The goal is to enable the user to specify linguistically what kind of change of the rule evaluation measures he is interested in. For example, the user may be interested in rules that exhibit a fast increase of the lift as well as a moderate increase of confidence.

When using the fuzzy approach described below, the user can specify individually what "moderate" or "fast" means by defining a fuzzy partition over the change rate domains of the rule evaluation measures of interest. Since we will be able to compute for every rule of the rule set a membership degree to which extent the respective rule evolution belongs to the user-specified concept, we can use a threshold to limit the set of resulting rules that are shown to the user or order all rules by descending membership degree.

The basic idea is to allow the user to define a fuzzy concept that contains linguistic variable assignments over the change rate domains of rule evaluation measures. Multiple such assignments are combined with well-known fuzzy connectives (t-norms or t-conorms). That is, the membership degree of any rule $X \to Y$ to the example fuzzy description stated above in the text:

$$\left\langle \Delta_{\text{lift}} \text{ is fast and } \Delta_{\text{conf}} \text{ is moderate} \right\rangle$$

will be evaluated as

$$\top\left(\mu_{\Delta_{\text{lift}}}^{(\text{fast})}(X \to Y), \mu_{\Delta_{\text{conf}}}^{(\text{moderate})}(X \to Y)\right)$$

[2] See Section 8 on page 34 for details.

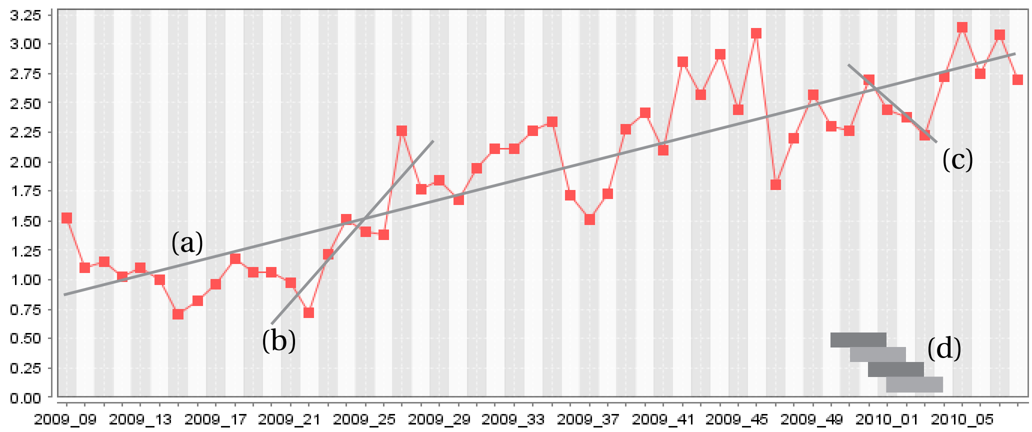


Figure 4.1: *Time series of the lift of a single rule (by week over one year). (a) shows the global linear trend, (b) and (c) illustrate local linear trends. (d) denotes sliding time frames for local trend estimation.*

where $\top$ is a t-norm that represents a fuzzy conjunction. Since we intend to assign a membership degree of the *change* of any rule evaluation measure (represented by the Δ in the linguistic variable name), we need to quantify this change rate from the data set.

4.2.1 Local and Global Changes

However, things are not as easy as just sketched. We have to make clear which time scale we refer to when we mean "fast increasing lift". Figure 4.1 shows the time series of the lift of a single rule over a period of one year on a weekly granularity. There is obviously a global trend denoted with (a). To what degree this increase qualifies as "fast increasing" depends on the underlying fuzzy partition. If we focus on the calendar weeks (CW) 22–27, we can see a much steeper slope belonging to a lift increase from about 0.75 to 2.25 (b). Thirdly, towards the end of the series, (c) marks a period with a lift decline from about 2.6 down to 2.25. The filtering approach must be able to detect especially these local subtrends in order to be of any value for a real-world application.

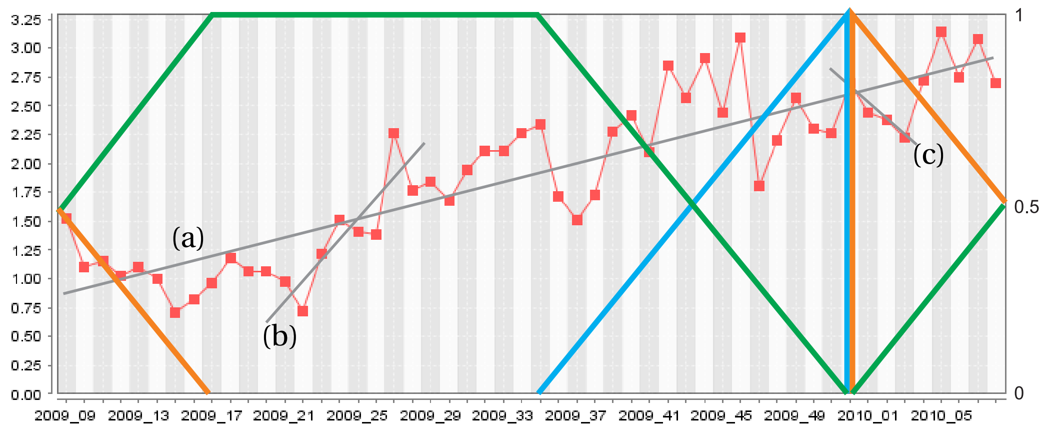


Figure 4.2: *Time series from Figure 4.1 with superimposed (shifted) fuzzy partition from Figure 4.3.*

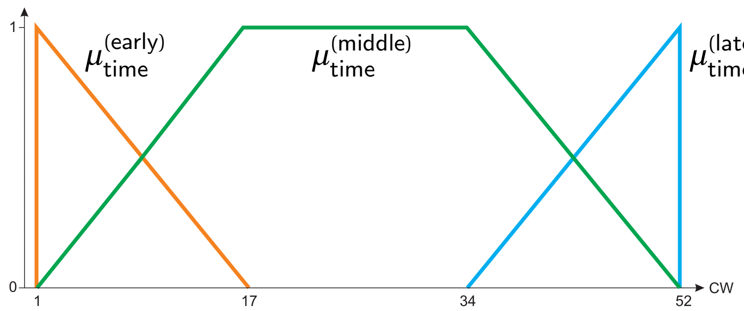


Figure 4.3: *The unshifted fuzzy partition of Figure 4.2.*

Up to now, the actual location of the recognized pattern did not matter. However, we might be interested in patterns arising only in certain parts of the time series. Hence, another notion of locality should be also applicable directly within the fuzzy concept: For example by querying for rules having "fast decreasing lift *early in the year*".

Here, a fuzzy partition on the time axis itself is used to represent linguistic terms such as "in summer", "early the year" or "last quarter". Figure 4.2 shows such a partition overlaid on top of the lift time series from Figure 4.1. Since the data the rule was extracted from started in CW 9/2009, the fuzzy partition is shifted to the left. Figure 4.3 show the unshifted version for better reference. The fuzzy triangular fuzzy sets at both ends represent the linguistic concepts "early in the year" and "end of the year".

The center fuzzy set is chosen to have a constant membership degree sum of one[3] and could be assigned the term "middle of the year". The additional term that constraints the temporal location is taken as a normal fuzzy concept and is connected to the remaining concept via the product t-norm $\top_{\text{prod}}$. Put together, the linguistic concept

"lift is fast decreasing early in the year"

will be evaluated as

$$\xi = \Big\langle \big(\Delta_{\text{lift}} \text{ is fast decr}\big) \text{ and time is early} \Big\rangle,$$

which is evaluated as

$$\xi(l,t) = \top_{\text{prod}}\Big(\mu^{(\text{fast decr})}_{\Delta_{\text{lift}}}(l), \mu^{(\text{early})}_{\text{time}}(t)\Big)$$

with l being the (local) lift change value and t being the point in time of that (local) lift change.[4]

Semantic Discussion

The question may arise why I opt for the product t-norm to combine the linguistic concept with the temporal constraint. The idea of constraining parts of the input stems from band-pass filters [SHENOI 2005] which are electrical devices that block certain portions of a frequency spectrum and let pass other regions. The transfer function of such a filter basically weighs an input signal. Applied to our scenario, for example, the fuzzy sets $\mu^{(\text{early})}_{\text{time}}$ could be considered a low-pass filter for the membership degrees of concepts evaluated at the respective location of the filter. Applying the transfer function to the signal is represented by a multiplication and I therefore opt for the product t-norm.

[3] I will later (Chapter 6, page 118) argue that a constant sum of one is not a hard requirement, however, it is advisable to start out with proper fuzzy partitions.

[4] I will suggest shortly, how the point in time is computed and how the local subseries for e. g. the lift decrease can be identified.

Global Trends

The global trend of a rule evaluation measure time series is estimated by a simple and straightforward linear regression: For any rule, the values of the desired rule evaluation measures are calculated for every time frame that the data set contains. A simple but quite robust way is the mentioned regression line that is fitted into the point set. The slope of this line serves as an indicator of the overall linear trend of the rule measure. Figure 4.1 (a) denotes such a global trend.

Local Trends

For identifying local trends, I employ a sliding-window approach: For each point in time a time windows of a pre-specified width is used to fit a linear regression line into the subset of the time series belonging to the current window. Figure 4.1 (d) shows four of such windows, one of which was used to compute the local slope of the pattern (c). Semantically, we are approximating the first-order derivative of the actual time series with this approach which nicely fits into the underlying rationale: Finding local changes. When temporal constraints are used (see Figure 4.3), the center of the sliding window is used to compute the membership degree. The sliding-window approach yields another time series from which we currently use only the maximum and minimum value. These two extrema are used when computing membership degrees to linguistic terms and concepts.

4.2.2 Composite Patterns

As motivated in Section 2.5, I also intend to detect so-called composite patterns, that is, patterns that consist of a temporal succession of local changes. The most interesting pattern is probably the peak which I use here to illustrate the detection paradigm. Figure 4.4 is repeated here from the background chapter. Obviously, the time frames A and B must appear in a special temporal relation in order to mimic a peak: A has to meet

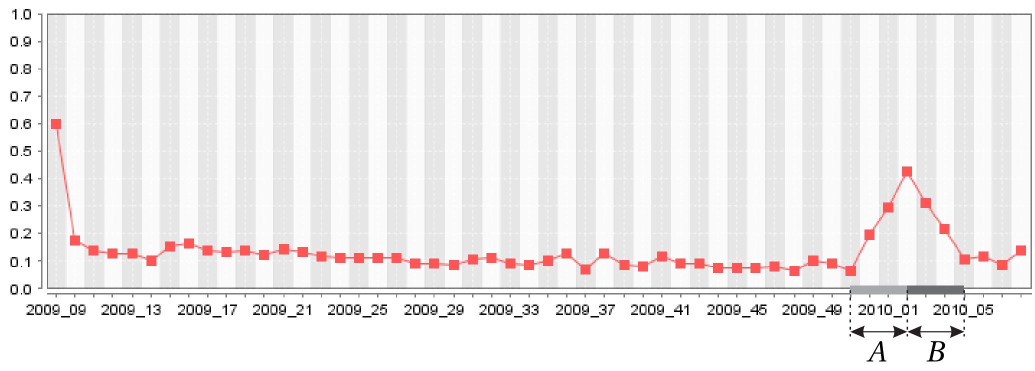


Figure 4.4: *The borders of the two time frame intervals A and B are set such that the membership to the concept* peak *is maximal.*

B. When referring to temporal relations, Allen's temporal algebra is most widely known [ALLEN 1983]. Figure 4.5 shows all 13 different relations between two intervals.[5] Of course, for our purpose, only the relations "before" and "meets" (and their counterparts) are relevant as there is no way to define overlapping intervals inside one time series. In order to find composite patterns, it is necessary to find intervals (like A and B in the peak example) that maximize the membership degrees of their respective linguistic terms and also satisfy the temporal relation.

For a peak we need to find:

- An interval A whose time subseries has large membership degree to the concept "(confidence) is fast increasing".[6]
- An interval B whose time subseries has large membership degree to the concept "(confidence) is fast decreasing".
- Intervals A and B that meet each other.

I propose to use an evolutionary algorithm to find good candidates for the intervals A and B. First, we are satisfied with near-optimal candidates

[5] Note, that all but the equality relation can be read in two different ways (stressing X or Y), thus resulting in 13 different relations.
[6] The example uses a confidence time series.

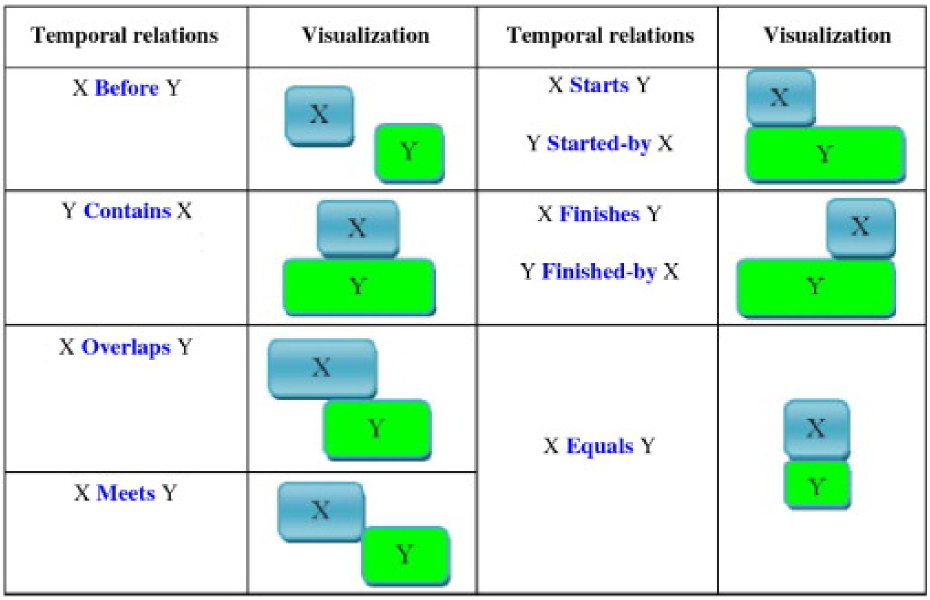


Figure 4.5: *Allen's 13 temporal relations. Image taken from [*COOK *et al. 2009].*

and second, we can easily implement the three above-mentioned criteria into a fitness function. Also, the structure of the individuals is straightforward: They consist of the borders of the intervals A and B. The fitness function is e. g. the sum of the membership degrees to both linguistic concepts, penalized when violating the temporal order. I will discuss a specific implementation of an evolutionary algorithm with all parameters and operations later in Section 4.3.

4.2.3 Summary

Before I will rephrase all discussed matter in a clean formal way, let us recapitulate which elements the linguistic filtering framework contains.

1. Specify a set of linguistic concepts that refer to the (local or global) temporal changes of rule evaluation measures. These concepts may also include temporal constraints as well as composite patterns.

2. Provide fuzzy partitions of the domains of the change rate of the measures from step 1 and the time line.

3. Evaluate for every rule the membership degrees for the linguistic concepts from step 1.

4. For every linguistic concept, order the rules according to their membership degrees such that for a given threshold the set of rules can easily be determined whose membership degrees exceed this threshold.

The rationale is as follows: Whenever the user changes a fuzzy set of the fuzzy partition, the rule set visualization that matches the concept being edited is updated instantly. Therefore, there is an immediate feedback and the user is able to determine visually, whether the currently edited fuzzy set (e. g. for the linguistic value “unchanged”) really meets his intentions (e. g. by conceiving that the resulting rules do not change their vertical position).

4.3 Formal Treatment

This section precisely rephrases the ideas given in the section above and complements central parts with examples.

We start with the initialization phase of the analysis, that is the rule induction and time series computation in Section 4.3.1. After that, the actual membership degree computation is discussed in Section 4.3.2.

4.3.1 Rule Induction and Time Series Computation

We discretize the time span of the underlying database D into $T \in \mathbb{N}$ frames. From these frames we derive the corresponding databases D_1 to D_T. From each D_t, $t \in \{1, \ldots, T\}$ we induce a set of association rules R_t by an algorithm $\mathscr{A}$ of the user's choice.[7] The collective set of rules R is then just the union of all the rules of all frames:

$$R = \bigcup_{1 \le t \le T} R_t = \bigcup_{1 \le t \le T} \mathscr{A}(D_t; \sigma_{\min}, c_{\min})$$

[7] I will use the well-known Apriori algorithm [AGRAWAL et al. 1993] for that purpose.

For every rule $\rho \in R$ we derive a time series $\tau_m(\rho)$ for each evaluation measure $m \in M$:

$$\forall \rho \in R : \forall m \in M : \ \tau_m(\rho) = \left(m_{D_1}(\rho), \ldots, m_{D_T}(\rho) \right)$$

The pseudocode for these steps are given in Algorithm 1. We first initialize the rules result set R to the empty set in line 1 and iterate then over all T time frames in line 2. Let A_T be the attribute in the database that tells for each case to which time frame it belongs, that is we assume the domain of A_T being $\{1, \ldots, T\}$. Any temporal attribute can obviously be transformed into such a form by means of binning. Line 3 then selects only those cases that belong to the current time frame t. We then induce rules in line 4 from the respective temporal database D_t by means of a previously chosen rule induction algorithm. The resulting rules are added (unified) to the result set R in line 5. The rule evaluation measures discussed in Section 2.1.2 can all be derived from the antecedent support, consequent support and (joint) support of a rule. That is, for each rule $\rho = X \rightarrow Y$ we need the values $P(X)$, $P(Y)$ and $P(X,Y)$ for every time frame. This happens from line 7–line 14.

4.3.2 Membership Degree Computation

For each fuzzy concept $\xi \in \Xi$, we can compute the membership degrees for all rules $\rho \in R$ according to the elaborations in Section 2.4 starting on page 34.

Given a user-specified threshold θ, we can easily restrict a subset of rules $\Gamma \subseteq R$ that meet a given concept ξ to the extent θ:

$$\forall \xi \in \Xi : \ \Gamma(\xi, R, \theta) = \left\{ \rho \in R \mid \xi(\rho) \geq \theta \right\} \subseteq R$$

Depending on the type of linguistic concepts in ξ, we need to distinguish between global membership computation (Algorithm 2), local membership computation (Algorithm 3) and the membership with respect to composite patterns (Algorithm 4).

Input : $D, T, \sigma_{\min}, c_{\min}$
Output: R
$R \leftarrow \emptyset$;
for $t \leftarrow 1, \ldots, T$ **do**
    $D_t \leftarrow \boldsymbol{\sigma}[A_T = t](D)$;
    $R_t \leftarrow$`Apriori`$(D_t, \sigma_{\min}, c_{\min})$;
    $R \leftarrow R \cup R_t$;
end
for $\rho = X \rightarrow Y \in R$ **do**
    for $t \leftarrow 1, \ldots, T$ **do**
        Determine $P(X)$, $P(Y)$ and $P(X, Y)$ from D_t;
        ρ.`lhssupp`$[t] \leftarrow P(X)$;
        ρ.`rhssupp`$[t] \leftarrow P(Y)$;
        ρ.`jointsupp`$[t] \leftarrow P(X, Y)$;
    end
end
return R;

Algorithm 1: Preparation

Global Membership Computation

Algorithm 2 starts with initializing the result map $\mathscr{M}$ to the empty set in line 1. If e. g. the lift change Δ_{lift} is referenced in ξ with the corresponding linguistic term fast, and if the membership degree of ρ to that linguistic term is 0.7, then the following mapping would be an element of $\mathscr{M}$:

$$\Big(\big(\Delta_{\text{lift}}, (\mathsf{fast})\big) \;\mapsto\; 0.7\Big) \in \mathscr{M}$$

Line 2 iterates over all rule evaluation measures in ξ for which we need to calculate a mapping. The fuzzy sets representing the linguistic terms of the measure changes are defined on the change rate domain of these measures which is $\mathbb{R}$. The actual trend is estimated by a linear regression `LinReg` in line 4. This algorithm takes a $2 \times T$ matrix (line 3) as input (the time frame indices $1, \ldots, T$ with its corresponding rule measure values). The slope α of the linear regression is used as the argument for which a membership degree to the linguistic term is calculated in lines 5 and 6 and put into the map $\mathscr{M}$. With all membership degrees at hand, the overall membership degree to the given concept ξ can then be computed as given in definition 8 and explanations on page 34.

Input : rule ρ, fuzzy concept ξ that is composed of the linguistic terms $\{\mu_1, \dots, \mu_n\}$
Output : map $\mathscr{M}$ of membership degrees of ρ to all linguistic terms referenced in ξ

$\mathscr{M} \leftarrow \emptyset$;
foreach *rule evaluation measure m referenced in* ξ **do**
  $\boldsymbol{M} \leftarrow \Big((1, \dots, T), \tau_m(\rho)\Big)^\top \in \mathbb{R}^{2\times T}$;
  $(\alpha, \beta) \leftarrow \mathtt{LinReg}(\boldsymbol{M})$;
  $\mu_{\Delta_m}^{(\mathsf{term})} \leftarrow$ the term referenced for m in ξ;
  $\mathscr{M} \leftarrow \mathscr{M} \cup \Big(\big(\Delta_m, (\mathsf{term})\big) \mapsto \mu_{\Delta_m}^{(\mathsf{term})}(\alpha)\Big)$;
end
return $\mathscr{M}$;

Algorithm 2: Global membership computation

Local Membership Computation

Algorithm 3 sketches the procedure for computing local membership degrees. The sliding window approach for that[8] requires a predefined window width which we denote by $w \in \mathbb{N}$. That is, we need to compute $T - w + 1$ different local slopes (line 3). For all $T - w + 1$ different windows the slopes α_i are estimated in lines 4 and 5.[9] In lines 7 and 8 we keep the minimal and maximal α_i since we do not yet know whether we match against a linguistic term denoting a positive change (such as fast increasing) or a negative change (such as fast decreasing). Depending on that nature of the linguistic term, we branch into line 11 or line 13 to add the new term-membership mapping. As for globals membership degree computation, the overall membership degree to the given concept ξ can then be computed as given in definition 8 and explanations on page 34.

[8] See Section 4.2.1 on page 61.

[9] The proposition $\rho.m[i]$ be the value of the rule evaluation measure m in time frame i in the style of lines 10–12 of Algorithm 1. The full sequence over all time frames was also noted as $\tau_m(\rho)$.

Input : rule ρ, fuzzy concept ξ that is composed of the linguistic terms $\{\mu_1, \ldots, \mu_n\}$, window width w

Output : map $\mathscr{M}$ of membership degrees of ρ to all linguistic terms referenced in ξ

$\mathscr{M} \leftarrow \emptyset$;

foreach *rule evaluation measure* m *referenced in* ξ *with local context* **do**

  for $i \leftarrow 1, \ldots, T - w + 1$ **do**

    $\boldsymbol{M}_i \leftarrow \begin{pmatrix} i & \cdots & i + w - 1 \\ \rho.m[i] & \cdots & \rho.m[i + w - 1] \end{pmatrix} \in \mathbb{R}^{2 \times w}$;

    $(\alpha_i, \beta_i) \leftarrow \texttt{LinReg}(\boldsymbol{M}_i)$;

  end

  $\alpha_{\min} = \min(\alpha_1, \ldots, \alpha_{T-w+1})$;

  $\alpha_{\max} = \max(\alpha_1, \ldots, \alpha_{T-w+1})$;

  $\mu^{(\mathsf{term})}_{\Delta_\mathrm{m}} \leftarrow$ the term referenced for m in ξ;

  if term *denotes positive change* **then**

    $\mathscr{M} \leftarrow \mathscr{M} \cup \Big(\big(\Delta_m, (\mathsf{term}) \big) \mapsto \mu^{(\mathsf{term})}_{\Delta_\mathrm{m}}(\alpha_{\max}) \Big)$;

  else

    $\mathscr{M} \leftarrow \mathscr{M} \cup \Big(\big(\Delta_m, (\mathsf{term}) \big) \mapsto \mu^{(\mathsf{term})}_{\Delta_\mathrm{m}}(\alpha_{\min}) \Big)$;

  end

end

return $\mathscr{M}$;

Algorithm 3: Local membership computation

Composite Pattern Membership Computation

I will use a real-world example to illustrate the pseudocode for composite pattern matching. The used pattern will be a peak concept. It will become clear from the example how other shapes would be implemented. I chose not to present the pattern matching idea on yet another level of generality (catering for general patterns) as this would require the introduction of some sort of abstract grammar to represent the composite patterns and then their translation into a specific evaluation function.

Figure 4.7 shows the confidence of an association rule on a weekly basis over a period of one year, that is, $T = 52$. A peak pattern exhibits an increasing flank which is met or followed by a decreasing flank. More precisely, we seek for two intervals $[l_A, r_A]$ and $[l_B, r_B]$ of $[1, 52]$ where the linguistic terms confidence is increasing and confidence is decreasing, respectively have a high membership degree (conjuncted via a t-norm).

I will use a genetic algorithm to find good candidates for these intervals. The borders of the intervals form the chromosome and the nature of the actual composite pattern will be incorporated into the fitness function. The fitness function for a peak pattern is shown in Algorithm 4. Line 1 sets the artificial fitness value to be returned if the chromosome is invalid (e. g. when the intervals overlap). Just in case the interval borders are in wrong order we can repair this by swapping them which is done in lines 3–5 for interval A and in lines 8–10 for interval B. We require each interval to be of minimal size 3, that is all shorter intervals are rejected. Lines 12 and 13 take care of this. If the two intervals overlap or have a too wide gap in between (here: wider than 2) the chromosome is also rejected with worst fitness in line 14. After all validity tests are passed, we compute the slopes of the linear trends on the intervals. Lines 15 and 16 set the data on which in lines 17 and 18 the linear regression is executed. The return value of the fitness function in line 19 is the (fuzzy) conjunction of the two membership degrees corresponding to the linguistic terms that describe the time series behavior at the peak's two flanks.

Let us illustrate such a matching with the time series of Figure 4.7. The genetic algorithm will start with a randomly initialized population of 15 chromosomes. The algorithm will run at most 300 epochs. The fuzzy partition that is needed to get the membership degrees of the flank slopes is shown in Figure 4.6. Table 4.1 shows the intervals encoded by the five fittest chromosomes. Clearly, the most pronounced peak ⑤ has by far the highest fitness. The other peaks can be visually confirmed, too. The algorithm has returned other (less fit) chromosomes. These, however, consisted of rather wide intervals and my be avoidable by constraining the interval width in the fitness function.

Input : time series $\tau_m(\rho)$, chromosome $\chi = [l_A, r_A, l_B, r_B]$
Output : fitness of χ with respect to $\tau_m(\rho)$

WORST_FITNESS $\leftarrow -\infty$;
if $l_A > r_A$ **then**
    $t \leftarrow l_A$;
    $l_A \leftarrow r_A$;
    $r_A \leftarrow t$;
end
if $l_B > r_B$ **then**
    $t \leftarrow l_B$;
    $l_B \leftarrow r_B$;
    $r_B \leftarrow t$;
end
if $r_A - l_A < 2$ **then return** WORST_FITNESS;
if $r_B - l_B < 2$ **then return** WORST_FITNESS;
if $(r_A > l_B) \vee (l_B - r_A > 2)$ **then return** WORST_FITNESS;
$\boldsymbol{M}_A \leftarrow \begin{pmatrix} l_A & \cdots & r_A \\ \tau_m(\rho)[l_A] & \cdots & \tau_m(\rho)[l_A] \end{pmatrix}$;
$\boldsymbol{M}_B \leftarrow \begin{pmatrix} l_B & \cdots & r_B \\ \tau_m(\rho)[l_B] & \cdots & \tau_m(\rho)[l_B] \end{pmatrix}$;
$(\alpha_A, \beta_A) \leftarrow \texttt{LinReg}(\boldsymbol{M}_A)$;
$(\alpha_B, \beta_B) \leftarrow \texttt{LinReg}(\boldsymbol{M}_B)$;
return $\top\left(\mu^{(\text{incr})}_{\Delta_\text{conf}}(\alpha_A), \mu^{(\text{decr})}_{\Delta_\text{conf}}(\alpha_B)\right)$;

Algorithm 4: Fitness function for composite pattern *peak*.

Fitness	Chromosome	Ref. in Fig. 4.7
0.000873	[17,19], [19,21]	①
0.001371	[13,15], [15,17]	②
0.002891	[39,41], [41,43]	③
0.004617	[26,28], [28,30]	④
0.069894	[44,46], [47,49]	⑤

Table 4.1: *Identified peaks with increasing membership degrees. The fitness function is shown in Algorithm 4. The fuzzy partition on the confidence change rate domain is depicted in Figure 4.6.*

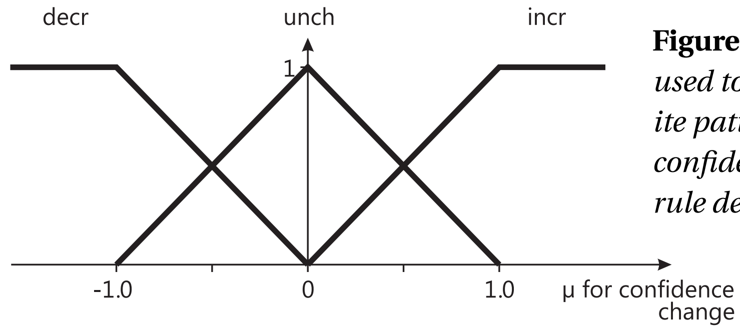


Figure 4.6: *Fuzzy partition used to identify the composite pattern* peak *inside the confidence time series of a rule depicted in Figure 4.7.*

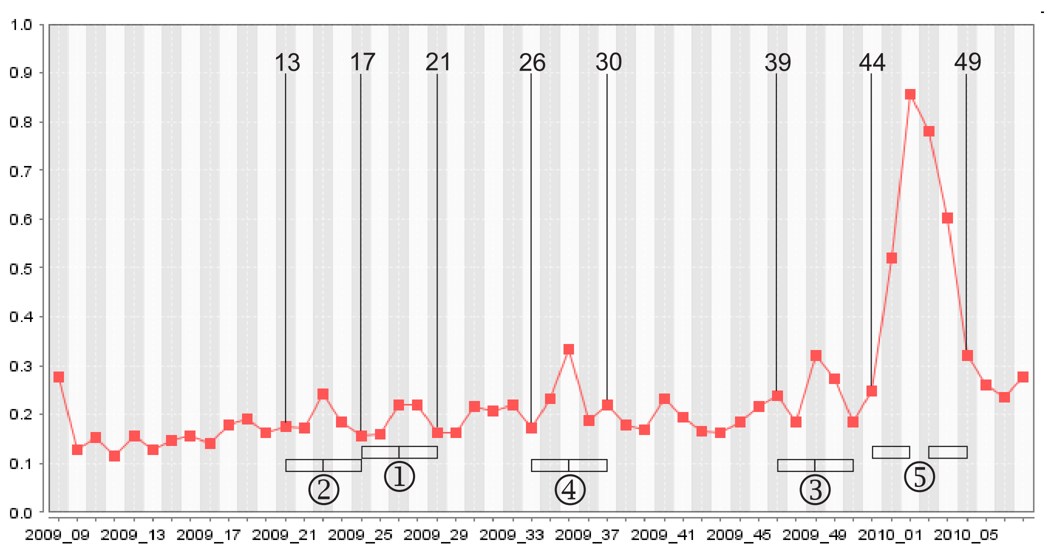


Figure 4.7: *Confidence time series of an association rule with several identifiable peaks. The series is used to illustrate the fitness computation of Algorithm 4. The identified peaks are numbered and correspond to the rows of Table 4.1.*

4.4 Fuzzy Partition Induction

Up to now, we did assume the fuzzy partitions on the rule evaluation measures to be given beforehand. Certainly, one can develop an intuition for appropriate partitions and set these manually according with some rules of thumb. However, since these fuzzy partitions are data-dependent, it suggests itself to propose some straightforward heuristic to initialize them in a reasonable manner.

For estimating a three-term fuzzy partition from the underlying data, I suggest the following heuristics.

Heuristic for Global Trends

If only linguistic terms with respect to global change trends are used in the filtering concept, Algorithm 5 can be used to induce in initial fuzzy partition. It basically looks for the minimum and maximum change rate and inserts a left-open, a right open and a triangular fuzzy set. Line 1 resets the set of all slopes, then we iterate over all rules in line 2. Line 3 and line 4 calculate the global trend whereas line 5 adds the new value to set A. Line 7 declares the left-open fuzzy set that models the linguistic term decreasing by increasing membership degree from zero leftwards and reaching membership degree 1 at the minimal change rate. Likewise line 9 declares the right-open fuzzy set modeling the linguistic term increasing by increasing membership degree from zero rightwards, reaching 1 at the maximum change rate. Line 8 declares the center fuzzy set representing the linguistic term unchanged by using a triangular fuzzy set with center at zero and the ends being the minimal and maximal change rates. Line 10 finally returns the entire partition.

Heuristics for Local Trends

In case the local trends are referenced inside the linguistic concept, Algorithm 6 proposes a fuzzy partition induction heuristic. Again, we iterate over all rules in line 2 but do another nested loop over all time

Input : measure m, rule set R
Output : Fuzzy partition Π_{Δ_m}

$A \leftarrow \emptyset$;
for $\rho \in R$ **do**
  $\boldsymbol{M} \leftarrow \Big((1, \ldots, T), \tau_m(\rho)\Big)^\top \in \mathbb{R}^{2 \times T}$;
  $(\alpha, \beta) \leftarrow \texttt{LinReg}(\boldsymbol{M})$;
  $A \leftarrow A \cup \{\alpha\}$;
end

$$\mu^{(\mathsf{decr})}_{\Delta_\mathrm{m}}(x) \leftarrow \begin{cases} 0 & x > 0 \\ \frac{x}{\min(A)} & \min(A) < x \leq 0 \\ 1 & x \leq \min(A) \end{cases};$$

$$\mu^{(\mathsf{unch})}_{\Delta_\mathrm{m}}(x) \leftarrow \begin{cases} \frac{x - \min(A)}{-\min(A)} & \min(A) \leq x \leq 0 \\ \frac{\max(A) - x}{\max(A)} & 0 < x \leq \max(A) \\ 0 & \text{else} \end{cases};$$

$$\mu^{(\mathsf{incr})}_{\Delta_\mathrm{m}}(x) \leftarrow \begin{cases} 0 & x < 0 \\ \frac{x}{\max(A)} & 0 \leq x < \max(A) \\ 1 & x \geq \max(A) \end{cases};$$

return $\left\{\mu^{(\mathsf{decr})}_{\Delta_\mathrm{m}}, \mu^{(\mathsf{unch})}_{\Delta_\mathrm{m}} \mu^{(\mathsf{incr})}_{\Delta_\mathrm{m}}\right\}$;

Algorithm 5: Heuristic for fuzzy partition induction with respect to global change rates of measure m.

frame windows of length w in line 4. This inner loop computes the local change rates (see Algorithm 3 for details) and keeps track of them via A_{local} in line 7. Depending on whether the fuzzy concept references a positive or negative local change, we add the minimum or maximum of A_{local} to A in lines 9–13. The remaining four lines 15–18 correspond to the lines 7–10 of Algorithm 5, that is they set up the three fuzzy sets.

4.5 Application to Other Models

4.5.1 Decision Trees

Decision trees [BREIMAN et al. 1984, QUINLAN 1986] are a widely spread technique that allows for an intuitive interpretation of the results. The underlying principle is a recursive partition of the data set based on a greedy attribute selection. Since every path from the root node to a leaf node can be read as an association rule, it is possible to use the framework proposed in this thesis without deviating from the concept of decision trees. The highlighted branch of the decision tree in Figure 4.8, for example, represents the association rule *If* $A = a_2$ *and* $B = b_1$, *then* $C = c_i$ for all values c_i of C. Thus, results from applications using decision trees can easily be transformed to be compatible with the framework presented in this thesis.

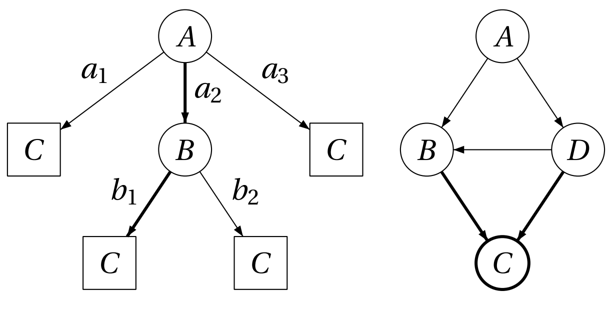


Figure 4.8: *Left: The bold path of the decision tree corresponds to the association rules* $A = a_2 \wedge B = b_1 \rightarrow C = c$, $c \in \text{dom}(C)$ *tree. Right: Bayes network. The highlighted subgraph represents the association rules* $B = b \wedge D = d \rightarrow C = c$ *for all* $b \in \text{dom}(B)$, $d \in \text{dom}(D)$ *and* $c \in \text{dom}(C)$.

Input : measure m, rule set R, window width w
Output : Fuzzy partition Π_{Δ_m}

$A \leftarrow \emptyset$;
for $\rho \in R$ **do**
  $A_{\text{local}} \leftarrow \emptyset$;
  for $i \leftarrow 1, \ldots, T - w + 1$ **do**
    $\boldsymbol{M}_i \leftarrow \begin{pmatrix} i & \cdots & i + w - 1 \\ \rho.m[i] & \cdots & \rho.m[i + w - 1] \end{pmatrix} \in \mathbb{R}^{2 \times w}$;
    $(\alpha_i, \beta_i) \leftarrow \texttt{LinReg}(\boldsymbol{M}_i)$;
    $A_{\text{local}} \leftarrow A_{\text{local}} \cup \{\alpha_i\}$;
  end
  if *m is referenced with respect to positive change* **then**
    $A \leftarrow A \cup \left\{\max(A_{\text{local}})\right\}$;
  else
    $A \leftarrow A \cup \left\{\min(A_{\text{local}})\right\}$;
  end
end

$\mu^{(\text{decr})}_{\Delta_\text{m}}(x) \leftarrow \begin{cases} 0 & x > 0 \\ \frac{x}{\min(A)} & \min(A) < x \le 0 \\ 1 & x \le \min(A) \end{cases}$;

$\mu^{(\text{unch})}_{\Delta_\text{m}}(x) \leftarrow \begin{cases} \frac{x - \min(A)}{-\min(A)} & \min(A) \le x \le 0 \\ \frac{\max(A) - x}{\max(A)} & 0 < x \le \max(A) \\ 0 & \text{else} \end{cases}$;

$\mu^{(\text{incr})}_{\Delta_\text{m}}(x) \leftarrow \begin{cases} 0 & x < 0 \\ \frac{x}{\max(A)} & 0 \le x < \max(A) \\ 1 & x \ge \max(A) \end{cases}$;

return $\left\{\mu^{(\text{decr})}_{\Delta_\text{m}}, \mu^{(\text{unch})}_{\Delta_\text{m}} \mu^{(\text{incr})}_{\Delta_\text{m}}\right\}$;

Algorithm 6: Heuristic for fuzzy partition induction with respect to local change rates of measure m.

4.5.2 Graphical Models

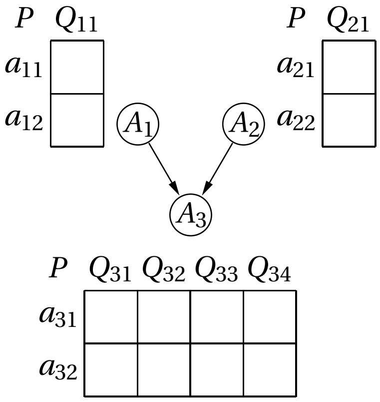
Graphical models [LAURITZEN and SPIEGELHALTER 1988, PEARL 1988, BORGELT et al. 2009] comprise another data mining method that has attracted a lot of research effort and led to many successful industrial applications [GEBHARDT et al. 2003, STEINBRECHER et al. 2008, KRUSE et al. 2010b, STEINBRECHER and KRUSE 2007b]. One type of graphical models are Bayes networks. A Bayes network uses an directed, acyclic graph $\vec{G} = (U, \vec{E})$ to encode the decomposition of a multidimensional probability distribution. The decomposition induced by the graph consists of a set of conditional distributions assigned to each node given its direct predecessors (parents). Their product equals the joint probability distribution of the underlying graphical model. For a directed graphical model with attribute set $U = \{A_1, \ldots, A_n\}$ and network structure $\vec{G} = (U, \vec{E})$ this quite general definition can be specialized with the chain rule to

$$P\Big(A_1 = a_1, \ldots, A_n = a_n\Big) = \prod_{i=1}^{n} P\Big(A_i = a_i \;\Big|\; \bigwedge_{A_j \in \mathrm{parents}_{\vec{G}}(A_i)} A_j = a_j\Big).$$

We refer to the functions $P(A \mid \mathrm{parents}(A))$ as *potentials*. More precisely, for every attribute A_i we are dealing with

$$q_i = \prod_{A_j \in \mathrm{parents}_{\vec{G}}(A_i)} \big|\mathrm{dom}(A_j)\big|$$

different probability distributions over $dom(A_i)$; one for every combination of the parent attribute values of A_i. Every such combination is denoted as $Q_{i1}, \ldots, Q_{iq_i}$. As an example consider a Bayes network with three attributes: A_1 represents air conditioning type, A_2 engine type and the class value is determined by A_3. For the sake of simplicity we assume their domains to be binary. The left part of Figure 4.9 depicts the graphical structure as well as the layout of the three potentials. Since root nodes cannot have parents, note that Q_{11} and Q_{21} represent "empty combinations", that is marginal distributions $P(A_1)$ and $P(A_2)$. We summarize the potentials of an attribute as a *potential table* as it is sketched in the right of Figure 4.9. The different combinations of parent attribute values are gathered column-wise, that is every column represents a probability distribution (as indicated by the shaded column) and therefore sums up to unity.



A_i	Q_{i1}	$\cdots$	Q_{ij}	$\cdots$	Q_{iq_i}
a_{i1}	θ_{i11}	$\cdots$	θ_{ij1}	$\cdots$	θ_{iq_i1}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_{ik}	θ_{i1k}	$\cdots$	θ_{ijk}	$\cdots$	θ_{iq_ik}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_{ir_i}	θ_{i1r_i}	$\cdots$	θ_{ijr_i}	$\cdots$	$\theta_{iq_ir_i}$

Figure 4.9: *Left: An example network which induces the layout of the potential tables. Right: A general potential table. Each column represents a (conditional) probability distribution.*

For the sake of brevity, we will denote the probability of the kth value of A_i given its parent attributes assume the jth value combination Q_{ij} by θ_{ijk}, that is

$$P(A_i = a_{ik} \mid \text{parents}_{\vec{G}}(A_i) = Q_{ij}) \;=\; \theta_{ijk}.$$

The values of a potential table are the parameters we referred to above and we now seek for an appropriate visualization. Note that every database entry can be mapped to exactly one table entry θ_{ijk}. That is, an attribute and its parents induce a partition of the underlying database: two database entries are in the same equivalence class if they share the same attribute values. The main idea is to interpret each equivalence class as an association rule and use rule evaluation measures to quantify the properties of the corresponding visual cues.

When one attribute of such a network models a designated class variable, then together with its parent attributes they can be used to immediately write the underlying probabilistic parameters as association rules. The example network in Figure 4.8 in that sense would give rise to the rules of type *If* $B = b_i$ *and* $D = d_j$*, then* $C = c_k$ for all values that B, D and C can assume.

4.5.3 Cooccurrence Graphs

This section sketches how the linguistic filtering can also be applied to cooccurrence graphs.[10] The objective is to answer questions of the following type (given a sequence of cooccurrence graphs):

> "First, what are interesting candidates for subgraphs that it would be worth looking at over time?"

and

> "Second, given a (still intractable large) set of subgraphs, which graphs become more sparse and less balanced over time?"

Before I turn to the algorithmic part of my approach, we need to negotiate which types of substructures within the graphs are most interesting to users. I will exploit the edge weights for this purpose. Several measures are needed to quantify for every subgraph aspects such as size, completeness, edge balance, etc.

If the cooccurrence graphs represent visits of different web pages within the same online shop portal, then it might be desirable to know whether customers are able to use the web portal as intended by the owners. Are there dead ends where users are stuck? What are the "hot spot" sites, that is, the pages that attract the most users and are visitors able to find the recently introduced shortcut to related pages? How do the accesses to the support area of the site change after renewing the navigational aids, etc.

I explicitly stress that subgraphs that are heavily interconnected with large edge weights only provide us with a *hint* that there may be an interesting visiting pattern. However, we can never conclude transitivity just from the cooccurrence graph! This is due to the fact that it only represents *binary* cooccurrences. Even a fully connected graph does not tell us anything about individual events. The sets of cooccurring events whose car-

[10] See Section 2.3 for details.

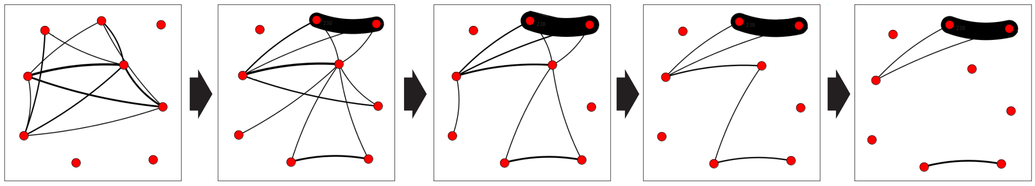


Figure 4.10: *The temporal evolution of the graph induced by a set of nine nodes. The number of edges is decreasing with time resulting in an almost isolated graph. Simultaneously, the edges that are remaining grow more and more unbalanced, that is, the deviation of the edge weights is increasing. Both time series of the corresponding measures* comp *and* dev *are shown in Figure 4.11 and Figure 4.12, respectively.*

dinality is represented by the edge weights even might be mutually disjoint. However, these subgraphs are found to be valuable hints that are worth being investigated.

Focusing on the before-mentioned type of aspects one can identify highly connected subgraphs with large edge weights to be one type of substructures that are most interesting to users. Another type may be single edges just connecting two nodes or substructures that are highly interconnected but with a large imbalance in the edge weights. The latter might represent two active sets of websites (large edge weights) between which users are able to navigate back and forth (numerous edges in between but with small weights since not every user is likely to use the offered navigational freedom).

The last arguments call for measures that on the one hand capture the mentioned properties of subgraphs and on the other hand allow to build a fuzzy partition on their domains since we are not going to ask for subgraphs *with 9 nodes and edges with weights greater than 100* but for *large* subgraphs with *moderately sized* edges.

Candidate Graph Generation

As we are now equipped with measures to assess certain aspects of subgraphs that we would like to track over time, the remaining question is how to determine such candidate graphs? It is clear that a brute-force approach (testing all subsets of nodes as potential subgraph node sets) fails immediately due to runtime problems, even for small node sets. I therefore promote the following heuristic: The graphs of all time frames are added as shown in Section 2.3 to arrive at the sum graph G_Σ (or simply the cooccurrence graph if we ignore the time frames). Next, a threshold θ is chosen and the components $\mathscr{C}_{G_\Sigma} = \{C_1, \ldots, C_j, \ldots, C_m\}$ of $G_{\Sigma,\theta}$ are taken as the candidate subgraphs. The choice of θ can be entirely left to the user (e. g. by offering a graphical preview tool that shows the components instantly whenever the user selects a new threshold via a slider) or θ may be determined in such a way to limit either the number of components or the (average) size of the components.

Matching Against Linguistic Concepts

Whatever way of determining the granularity of components is chosen, we are left with a set of mutual disjoint node sets $\mathscr{C}_{G_\Sigma}$ that are used to create a sequence of subgraphs $\left\langle G^{(i)}_{C_j} \right\rangle$, $i = 1, \ldots, n$, $j = 1, \ldots, m$ (one sequence for every subgraph induced by the node set) that are evaluated against the user-specified temporal behavior description. A time series is generated for every measure referenced in this user description. The temporal change within this time series is computed and the degree of membership to the user description is calculated. I will employ a simple regression approach, that is, we fit a regression line into the time series and interpret its slope as an indicator of decrease, stability and increase.

The example concept from the motivation of this section is repeated here:

"Completeness is decreasing and std. deviation is increasing"

Translated into a linguistic concept, the user may specify

$$\left\langle \Delta_{\text{comp}} \text{ is } \mathsf{decr} \text{ and } \Delta_{\text{dev}} \text{ is } \mathsf{incr} \right\rangle,$$

which is evaluated to

$$\top\left(\mu^{(\mathsf{decr})}_{\Delta_{\text{comp}}}(C_j), \mu^{(\mathsf{incr})}_{\Delta_{\text{dev}}}(C_j)\right),$$

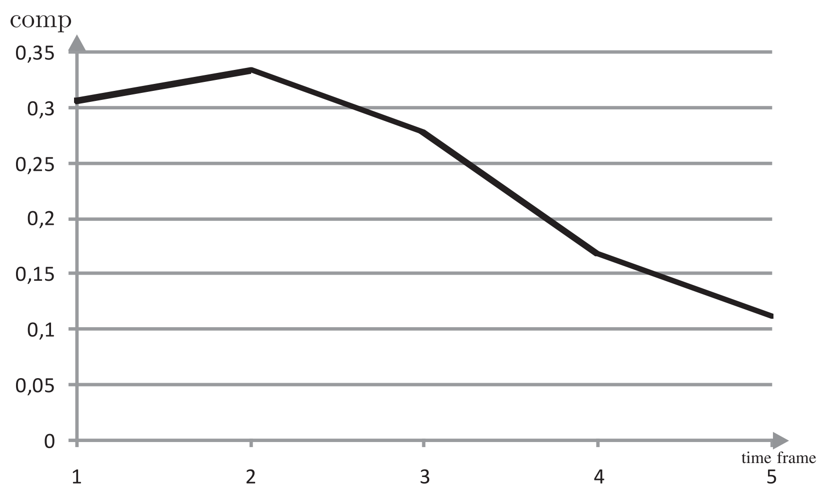


Figure 4.11: *Time series with decreasing trend for the completeness of the edge weights for the five graphs of the time frames depicted in Fig. 4.10.*

where $\top$ represents a t-norm modeling the fuzzy conjunction. Figure 4.10 depicts an example subgraph consisting of 9 nodes. Five time frames are shown with the respective edge weights. The graph is obviously becoming less dense with time, that is, the completeness is decreasing. The chart for this measure is depicted in Figure 4.11. In analogy to this, Figure 4.12 shows the increasing deviation of the edge weights which is attributed to the emergence of the strong cooccurrence (the sudden appearance in this case can be explained with time frames that were too large to appropriately cover the short period during which this strong cooccurrence emerged). If we equipped the change rate domains (that is, domains of the slopes of the regressions lines of the two time series) with appropriate fuzzy partitions (as I will do it in the experiments section) we could calculate the membership degree of this node set to the above-mentioned linguistic concept.

Summarizing, I state the following procedure:

1. Given a sequence $G^{(0)},\ldots,G^{(n)}$ of cooccurrence graphs with their sum graph being G_Σ.

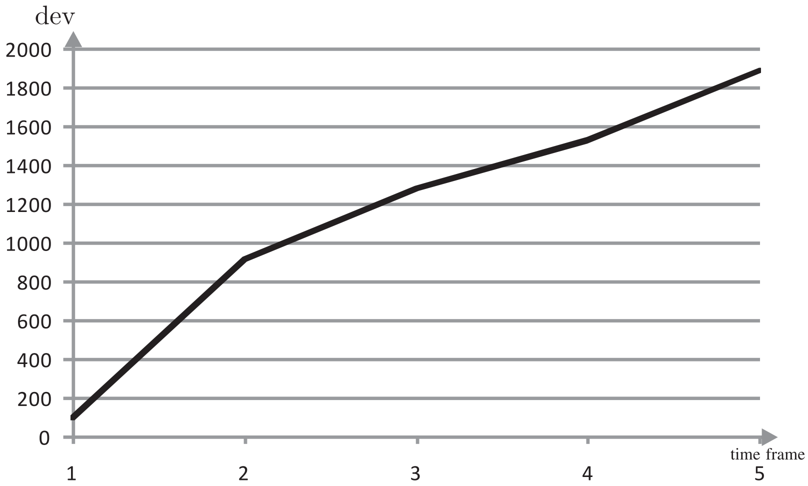


Figure 4.12: *Time series with increasing trend for the standard deviation of the edge weights for the five graphs of the time frames depicted in Fig. 4.10.*

2. Based on an appropriate value θ, we calculate the candidate graph node sets $\{C_1, \ldots, C_m\}$ which are the vertices of the components of the graph $G_{\Sigma,\theta}$.

3. The user provides a set of linguistic concepts that refer to the temporal change of the graph measures.

4. Provide fuzzy partitions for every domain of the change rate of the measures used in the descriptions of step 3.

5. Evaluate for every graph G_{C_j} the degree of membership to the linguistic concepts of step 3.

6. For every linguistic concept sort the graphs in descending order with respect to their membership degrees.

4.6 Related Work

My proposed approach leaves the definition of an association rule untouched which allows to carry over the linguistic filtering approach to other model types (such as cooccurrence graphs as shown). However, there are also investigations that augment the rule definition itself by temporal validity constraints. ALE and ROSSI[11] define time windows within which the rule is actually valid. This approach is not unlike the work of LEE et al.[12] who modify the rule definition and actually address problems in stream mining: The support of newly arrived items is necessarily lower than the support of older item (since they had more time to "gather up"). The authors use a special support definition to alleviate this effect. However, in both cases the notion of an association rule is deeply modified. On the one hand, this allows to address some problems in greater depth. On the other hand, it renders the straightforward transfer of the proposed approaches to other model types almost impossible.

AU and CHAN[13] actually use a very similar approach to asses the time series of rule evaluation measures. In their approach, they consider linguistic terms (such as "lift is increasing") new items and infer yet another rule set from it. Those meta rules can then be used to detect certain collective behavior such as "in 20% of those cases where the lift increased, the confidence did alike".

Other works deal with in principle the opposite challenge: Given a time series, which linguistic concepts match against it best? Results from such approaches are known as temporal linguistic summaries [KACPRZYK and WILBIK 2010] and can considerably help better understand the underlying process. Subsets of time series are also known as *motifs* for which linguistic assignments can be induced as well [MOEWES and KRUSE 2009].

4.7 Summary and Discussion

In this chapter I thoroughly covered my contributions to the task of filtering time series against linguistic concepts [STEINBRECHER and KRUSE 2008a, 2009b, 2010, KRUSE et al. 2010a,c]. In this thesis, I am applying these techniques to time series of evaluation measures of sets of associa-

[11] See Section A.2.1 on page 149 for more details.
[12] See Section A.2.2 on page 149 for more details.
[13] See Section A.2.3 on page 150 for more details.

tion rules. I will sketch later in the evaluation chapter how the technique can also be applied to other models' measure series [STEINBRECHER and KRUSE 2009c].

In Section 4.1 of this chapter, I claimed three requirements to be met in order to achieve a positive user acceptance: intuitiveness, instant feedback and visual interaction. Intuitiveness has been covered in this chapter: The application of fuzzy concepts to describe linguistically the (local and global) temporal behavior of time series enables a straightforward usage. Further, most of the necessary parameters (such as the fuzzy partitions representing the linguistic terms) can be suggested automatically from the underlying data. The remaining parameters (like window widths, population sizes, etc.) are still meaningful and can still be chosen with common sense. I purposely refrained from introducing methods to detect higher-order trends inside the time series. That is, no quadratic or cubic trends can be directly described and detected. The main reason for this decision is the difficulty to model the respective parameters linguistically. In a linear (global) trend model $y_t = \alpha t + \beta + \epsilon$, the parameter α has an intuitive and commonly known name: slope. This makes it easy describe and recognize certain values of α as e. g. *increasing* or *stable*. In a quadratic model $y_t = \alpha t^2 + \beta t + \gamma + \epsilon$ the dominating shape parameter α now is much harder to describe in linguistic terms. It certainly governs the *"opening"* or *"aperture"* of the trend function, but it is harder to come up with a reasonable set of descriptive linguistic terms describing its shape change rate.[14] In addition to that, a quadratic trend—if it is not too weakly pronounced— can be approximated by two linear trends just as described with the *peak* composite pattern above.

The remaining two requirements (instant feedback and visual interaction) I still owe and I will cover them in the next chapter as they are crucial parts of the software package that was used to implement the major contributions of this thesis.

[14] The linguistic terms of the change rate (of the shape of a quadratic trend) would, of course, still be decr, unch and incr. But the notion to whose change rate they refer is more intangible.

5 Implementation

In this chapter, I will introduce the user interface and software stack that was used to evaluate the previously presented ideas. The evaluation results provided in the next chapter were all created with this software. Most of the figures in this chapter are screen shots and as such consume considerable amount of space. In order to enhance readability, I placed most of these figures into Appendix B.

5.1 The Information Miner 2.0 Platform

The Computational Intelligence Group[1] led by Prof. Dr. Rudolf Kruse at the Otto-von-Guericke University Magdeburg is not only known for its constant output of high-quality research results in the area of data mining. Another major strength is the aim to forge the scientific results into solutions that are then applied successfully by industrial partners. Over the past years, a considerable number of data analysis tools and visualization methods have been developed. The Information Miner platform was introduced as a logical consequence in order to leverage their concentrated strengths: data-intensive algorithms have mostly been implemented as C++ command line tools[2] whereas the visualization tools are merely developed in Java.

[1] See `http://fuzzy.cs.uni-magdeburg.de/wiki/pmwiki.php`.
[2] See e. g. `http://borgelt.net/software.html`.

The first iteration of the Information Miner [RÜGHEIMER and KRUSE 2005] has been built with an academic focus. I rebuilt the entire application to form it into an orchestration layer between high-performance data analysis applications and the visualization components.

Figure 5.1 shows the main window with a simple example analysis flow loaded. In line with the initial iteration of the Information Miner (and basically all other data mining suites[3]) the analysis flow is modeled as a directed graph. Nodes represent so-called tasks that turn input data into output data. Edges indicate the data flow and feed the input data into tasks and output data towards the next task(s). Tasks that are sources (that is, having no inputs from previous tasks) represent data sources like CSV files, database connections or data generators. Tasks that are leaves (that is, having no successor tasks) typically indicate visualization tools or data serialization mechanisms. The tree view on the left of the main window contains all available tasks to build powerful analysis flows. I will sketch those that are relevant for this thesis in the next section.

The Information Miner is available at the workgroup website[4] and under constant development. Currently, it consists of 50k+ lines of code and is implemented in Java 1.6 (and invoking native executables that ship with the Java application).

5.2 Real-world Analysis Workflow

Let us now turn to the typical workflow that was used to come up with most[5] of the results of the next chapter. We assume that no preprocessing has been done so far, that is, we start from scratch with a single database table that contains all relevant data.[6] One dedicated attribute $A_T^{(\mathrm{orig})}$ rep-

[3] Like KNIME, RapidMiner, WEKA, etc.

[4] See workgroup website `http://fuzzy.cs.uni-magdeburg.de/wiki/pmwiki.php?n=Forschung.InformationMiner2`.

[5] "Most" means that in one case the rule set was small enough to assess it solely with the visualizations from Chapter 3 and no filtering was applied.

[6] Obviously, this table was created by joining together the original tables in the database system.

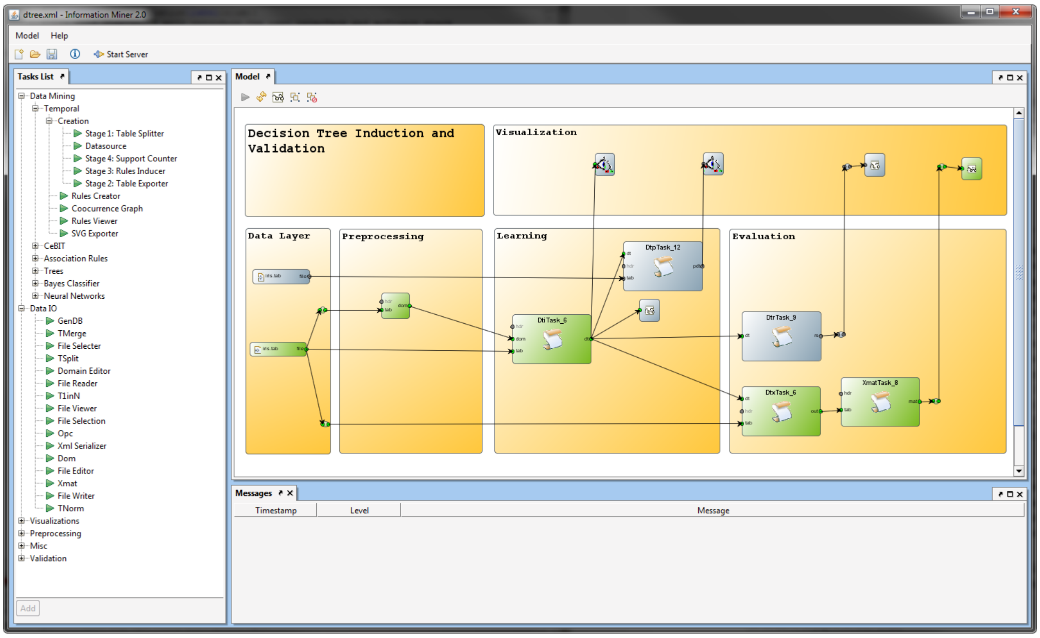


Figure 5.1: *A demo analysis workflow loaded into Information Miner 2.0.*

resents the original time stamp attribute. It will be discretized into suitable attribute A_T according to which we will subdivide the dataset. The preprocessing consists of the following steps:

1. Splitting the table according to (the values of) A_T
2. Exporting the split tables
3. Induce rules from the exported tables
4. Determine supports of rules for each time frame
5. Filter and visualize the rules

Each of the steps 1–4 will be implemented by a dedicated task which we are briefly discussing below. Step 5 will be, of course, treated in greater detail as it represents the applied contributions of this thesis.

5.2.1 Table Splitting

The task `TableSplitterTask` will carry out the step in line 3 of Algorithm 1 on page 66, that is, create as many database tables D_t as the attribute A_T has values. The task has the following input slots:

Input slot	Description	Format
`schema`	Table schema description in XML format of the tables D_t to be created. Normally, this schema will be identical to that of D but by omitting attributes it is simple to create projections.	XML
`db`	JDBC connection to the actual database.	POJO
`recipe`	A *recipe* can contain instructions how to translate values of attributes read from D before they are written into D_t. It is optional and can be used to induce missing values, discretize metric attributes or create items out of attribute values (see Table 2.2 on page 23).	XML
`props`	Property file that contains information on D, $A_T^{(\text{orig})}$, A_T as well as the respective mappings.	property file

After this task is executed, the database to which `db` points to will contain tables corresponding to D_t for each time frame that we are interested in. The output of the task is an ordered list of these time frame values. This list is serialized and can be loaded later without having to run the `TableSplitterTask` again.

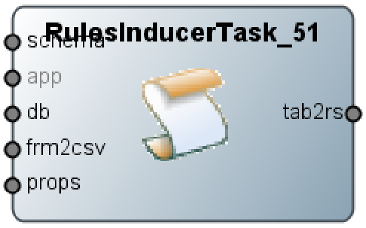
5.2.2 Table Export

The `TableExporterTask` exports the previously created database tables D_t to into CSV files for further treatment (here: running the Apriori algorithm on each of them). The task has the following input slots:

Input slot	Description	Format
`frm`	The list of time frame values $\{1, \ldots, T\}$ (leading to the respective database tables D_t).	XML
`schema`	Same as for `TableSplitterTask`. Used here only to know which attributes to export.	XML
`db`	JDBC connection to the actual database where to export from.	POJO
`props`	Property file that contains information on where on disk to store teh CSV files and how to name them.	property file

After this task has executed, the contents of all D_t are stored as CSV files on disk.

5.2.3 Rules Induction



The `RulesInducerTask` carries out the step in line 4 of Algorithm 1 on page 66, that is, it will execute the Apriori algorithm on each of the temporally sliced tables D_t. I use Christian Borgelt's freely available Apriori implementation [WWW: BORGELT 1] which takes CSV files as inputs: the ones we just exported.

Input slot	Description	Format
`schema`	Deprecated	
`app`	Appearances file which can be used to restrict the appearances of items in the rules. If there is a dedicated class attribute in D, the app file can be used to induce only rules with items relating to that attribute in the consequent.	ASCII
`db`	Deprecated	
`frm2csv`	Contains information which files to run Apriori on.	XML
`props`	Property file that contains parameters for Apriori such as minimum support and minimum confidence.	property file

After this task has executed, there will be an ASCII file for each $t \in \{1, \ldots, T\}$ containing the association rules that were induced from D_t.

5.2.4 Support Counting

The SupportCounterTask carries out the remaining steps of lines 4–15 of Algorithm 1 on page 66. That is, it will load the rules created in the previous steps and determine the antecedent, consequent and joint support for each rule. This is necessary as some rules might not have been found in all time frames but we need their supports in order to get a complete time series of any needed evaluation measure. The following input slots are required:

Input slot	Description	Format
db	JDBC connection to the actual database where to query the supports from (via SQL select count(*) queries).	POJO
tab2rs	Contains information which files to load the rules from.	XML
props	Property file that contains information on how to compose the SQL queries for support counting (date/time format, etc.)	property file

After the execution of this task, all information is collected to visualize and filter the rules. The output slot rules stores this information such that from now onwards, no database access is needed. The complete chain of tasks needed to preprocess the data in table D is depicted in Figure 5.2.

5.2.5 Filter and Visualize the Rules

After all necessary information has been collected, the last step is to load the user frontend containing the actual visualization and filtering tools. This is done via the RulesViewerTask which is depicted in Figure 5.3. Its input slots are as follows:

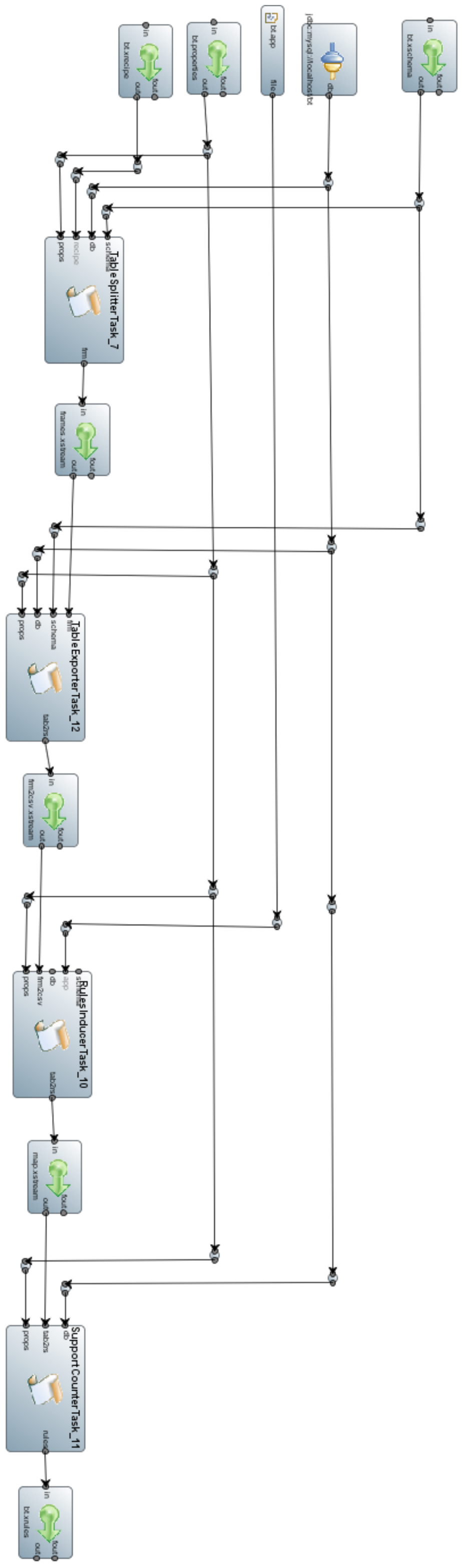


Figure 5.2: *The complete preprocessing phase as an Information Miner 2.0 model.*

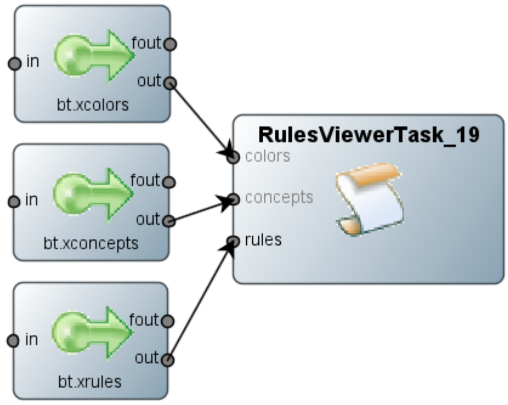


Figure 5.3: *The `RulesViewerTask` finally displays the frontend in which the filtering and visualization takes place. The `rules` input slot contains all necessary information about the rules under analysis. Slot `concepts` loads a preset of fuzzy partitions while `colors` is used to map certain items to colors in the rule visualization.*

Input slot	Description	Format
`colors`	Can be used to specifiy which color shall be used to represent the items in the rules' antecedents and consequents (see Section 3.2).	XML
`concepts`	Contains presets of fuzzy partitions that can be used to specify the linguistic terms.	XML
`rules`	The rules and their supports which have been collected during the preprocessing phase.	XML

Executing this task brings up a collection of windows that contain all relevant user interface elements.

Rules Glyph View

The main component is the rules view which is depicted in Figure B.1. It depicts the entire set of rules (①)denoting each rule by its glyph as introduced in Chapter 3. To rescale the horizontal axis, vertical axis and glyph size, the sliders at location ② can be used. To change the actual evaluation measures that are used to locate the glyphs (and determine their sizes), a drop-down menu at location ③ can be opened. It is depicted in Figure B.2 and shows the varieties of evaluation measures.

Finally, the slider at the bottom (④) allows to flick through the different time frames. The rule glyphs are smoothly animated in order to present a seamless transition[7] between the time frames.

Rules List View

The rules list view as depicted in Figure B.3 enumerates all rules in their textual form as a scrollable list. Each list item contains a checkbox by which the rule can be hidden in the rules glyph view. This allows to hide irrelevant rules from a filtering result which may still contain additional rules. The button bar at the bottom helps to hide entire selections of rules. The right part of each list item displays the absolute support, the confidence and the lift of each rule in the current time frame (selected by slider ④ in Figure B.1).

The rule set depicted in Figure B.1 (and respectively inFigure B.3)[8] makes clear that the number of rules (here: 907) is far too large for “manual” inspection. In addition to that, the number of time frames (here: 42) would make it hard too impossible to assess the trajectories of even a small set of rules. We will discuss the editors for filter composition below, after we addressed two other important detail views that are updated anytime the user clicks a rule.

Rule Details View

Whenever a rule is selected by clicking it, the textual representation of it is displayed at location ① of the rule details view as shown in Figure B.4. The time series of the absolute support[9], confidence, lift and recall are given in tabular form (②) below. This allows to precisely lookup values of a rule that was e. g. found to be peculiar by visual inspection.

[7] Note that the actual trajectory of a rule between two time frames has not necessarily to be linear. However, the number of time frames should be chosen such that consecutive frames are close enough such that no heavy change in between should be overlooked (here: 1 week time frame width).

[8] We will return to that rule set later in the evaluation chapter in Section 6.4.

[9] Support is named “Units” in Figure B.4.

Rule Time Series View

While the rule details view presents precise numbers for a set of evaluation measures, the rule time series view of Figure B.5 allows to display any evaluation measure graphically. The top chart (①) shows the time series of the antecedent support, consequent support and joint support as a line plot. We can clearly see a pronounced peak around the end of the year 2011 which we will cover later in the evaluation chapter. The bottom chart (②) shows a user-selectable measure time series. Again, all measures from the list in Figure B.2 can be chosen from in the rule measure drop-down list in location ③. The green line shows the average value of the series while the two yellow lines mark one standard deviation off the average value. The gray line sketches the global linear trend fitted into the time series. The controls in location ③ allow also to show quadratic and cubic trends, however, I stated in Section 4.7 not to use them for linguistic filtering.

Filter Concept Editor

The graphical interface discussed up to now only allowed to deal with the entire rule set. The filter concept editor finally enables the user to compose a linguistic concept[10] and match it against the rule set. Figure B.6 shows the filter concept editor where a single concept can be created and edited at a time. The recursive definition of linguistic concepts in Definition 8 on page 34 allows for the representation as an expression tree, that is, a tree structure where the nodes represent the operators and edges denote the argument relations. The tree view of Figure B.6 (①) is redrawn in Figure 5.4 to more clearly show the underlying filter expression.

The actual linguistic concept as I introduced in in Chapter 2 is represented by the left subtree with root node $\top_{\text{prod}}$ and reads as follows:

$$\left\langle \Delta_{\text{lift}}\ \mathsf{is\ incr}\ \ \text{and}\ \ \Delta_{\text{conf}}\ \mathsf{is\ incr} \right\rangle$$

[10] See Section 2.4 and Section 4.2 for a thorough treatment.

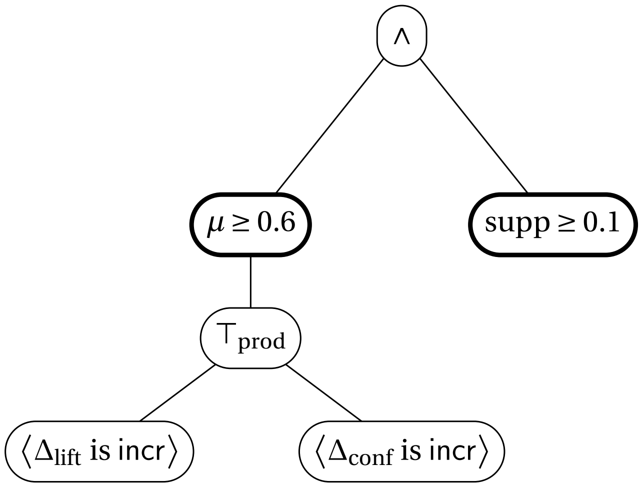


Figure 5.4: *Expression tree of the linguistic concept shown in Figure B.6. The bold nodes yield a crisp (boolean) result to be anded to the final decision whether a rule matches the filter or not.*

As we need a crisp, that is, boolean result in order to decide whether the glyph of the respective rule shall be drawn, the node $\mu \geq 0.6$ serves kind of a defuzzifier. Having boolean values at hand, we can further implement boolean expressions to create a richer set of filter operations. The concept in Figure 5.4 and Figure B.6 additionally restricts the support of the rules to a fixed interval. The properties of each node are shown (②) when the node is selected. Most adjustments take effect immediately, that is, the concept is constantly evaluated and the rule set is redrawn accordingly.

Fuzzy Partition Editor

As discussed in Chapter 4, the fuzzy partitions that encode the linguistic terms of the evaluation change rate domains play a central role in the applicability of the entire approach. Heuristics for initial suggestions were provided in Section 4.4. The fine-tuning of these partitions (or the complete redesign) can be done via the fuzzy partition editor shown in Figure B.7. The list ① contains all partition names (supplied by the `concepts` input slot into the `RulesViewerTaks`, see Figure 5.3). Selecting a partition will display the corresponding fuzzy sets in the area ②. The current view shows the fuzzy partition of the confidence change rate domain as it was induced by the heuristics presented in Section 4.4. The square handles at the apexes of each fuzzy set can be dragged with the mouse to adjust the shape and thus the linguistic meaning of the respective term. Again, the rule glyphs are updated instantly. Clicking and dragging the blue outlines of a fuzzy set can be used to move the fuzzy set horizontally.

In order to get an intuition on how a reasonable fuzzy partition should look like one has to have information about the change rate distributions of every measure of interest. The histogram in location ③ shows such a discretized distribution. When the "Histogram" button is clicked, the (local or global slopes) of all rules are assigned to twenty equidistant bins.[11] The resulting histogram is shown for the global confidence change rates in ③. To finally induce a fuzzy partition as suggested in Section 4.4 one uses the control in location ④. Setting e. g. a new 3-set partition will induce a left-open, a right-open and a triangular fuzzy set based on the (local or global—depending on the checkbox "Local") confidence change rate distribution. The new fuzzy partition will be in effect instantly.

Now that we have briefly sketched over the individual components of the graphical user interface, I will show how quickly a set of more than 900 rules can be easily filtered. I will focus on the handling and the actual user experience here. Detailed results with interpretations from real-world data sets follow in the next chapter.

5.3 A Sample Workflow

Let us now start a sample analysis session. The original database (table) D (which we will revisit again in the next chapter) contains 11 attributes and over 4.2 million rows. Each row represents a support request at a big telecommunications company. The time attribute was discretized into slices of 42 weeks. Running the workflow from Figure 5.2 will deliver all needed rules and their respective supports.[12]

The upper part of Figure 5.5 shows the full rule set of 904 rules in the first time frame. Clearly, this is no basis for a visual inspection. When looking

[11] Actually, 18 equidistant bins as the first and the last one extend to infinity.

[12] The rules were induced with Apriori invoked at minimum support 5% and minimum confidence 60%.

at the animation of the rule's trajectories across the screen of the 42 time frames, some rules literally stick out as they develop high lift values and thus move upwards out of the screen real estate. However, no reliable assessment is possible. If we would plot the trajectories of each rule as a straight line of 1 pixel width: the screen would be almost solid black because there would be 904 lines with 41 segments each.

Let us be interested in rules that exhibit increasing local confidences as well as globally unchanged lift. As the lift definition is closely related to the definition of the confidence[13] the lift quite often shows a similar behavior to the confidence given a (approximately) stable consequent probability:

$$\mathrm{conf}(X \to Y) = P(Y \mid X) \propto \frac{P(Y \mid X)}{P(X)} = \mathrm{lift}(X \to Y) \quad \text{with } P(X) \text{ constant.}$$

A stable lift together with an increasing confidence then would also make a statement on the increase of the overall consequent probability. We first induce the fuzzy partitions for the confidence and lift change rate domains. Figure B.8 and Figure B.9 show the resulting partitions. The reason for the asymmetric shape of the lift partitions can be seen when looking at the histograms of the lift change rates in Figure B.10: Most lift change rates are approximately zero, almost no negative change rate but a lot of different but larger change rates. The triangular fuzzy set representing "unchanged" and the left-open fuzzy set representing "decreasing" of Figure B.9 were adjusted manually in order to have flatter flanks: the fuzzy partition induction heuristic would yield almost vertical flanks for these as there are almost no negative change rates (see, again, Figure B.10) and I require the fuzzy set for "unchanged" to have its maximum at slope 0.

The linguistic concept is easily created node by node; the result is shown in Figure B.11. The bottom part of the figure shows how to set the local confidence increase that we are interested in: The drop-down list "Partition" contains all loaded and manually constructed fuzzy partitions. The drop-down list "Value" then will be updated with the respective fuzzy sets corresponding to the linguistic terms. If a local trend is intended to be

[13] See page 21.

referred to (as in this example for the confidence), the checkbox “Local trend” enables the radio buttons “Minimum” and “Maximum”. We chose “Maximum” here as we are interested in rules with strong local confidence increases.

After all tree nodes have been appropriately configured, we can easily filter the rule set (actually, when changing settings as described in the paragraph above, the rule set was already updated at every step, which can also be considered some kind of filtering): This is accomplished by adjusting the threshold of the concept that specifies the minimum membership degree (to the linguistic concept) that a rule has to have in order to be in the result set and thus be visible. Figure B.12 shows this slider as the only property of the “Threshold” node of the concept tree.

Let us compare four different thresholds to illustrate the increasingly restrictive filtering. The following table summarizes these steps:

Threshold	Figure
0%	Figure 5.5 (top)
10%	Figure 5.5 (bottom)
50%	Figure 5.6 (top)
90%	Figure 5.6 (bottom)

The bottom chart of Figure 5.5 shows already a considerable reduction in the number of rules. Obviously, most of the rules that disappeared had quite a large support which gives rise to the idea that the most relevant rules (with respect to our linguistic concept) will have a small support—a conjecture that will prove valid when raising the membership threshold further. At a value of 50% there are only 11 rules left. This could already be a base for an exhaustive evaluation: Assessing all 11 rules manually is a task that can easily be accomplished without risking to miss important clues due to emerging inattention. Let us push the threshold further to the boundary of 90%. We arrive at as few as four remaining rules in the bottom chart of Figure 5.6. To visually validate the match of the measure time series to the specified linguistic concept, the time series for confidence and lift of the rule ① are both shown in Figure 5.7: The trend of the lift is slightly increasing (more precisely: “slightly” with respect to the fuzzy partition that we specified—there might be scenarios where such a global increase from 7.6 to 8.2 could be considered “huge”); the global trend reference in the linguistic concept mitigates the dip around the turn of the year. The confidence, however, drops dramatically around the same time and climbs back up around CW 08/2012. It is this rapid increase that matched against the local confidence change term.

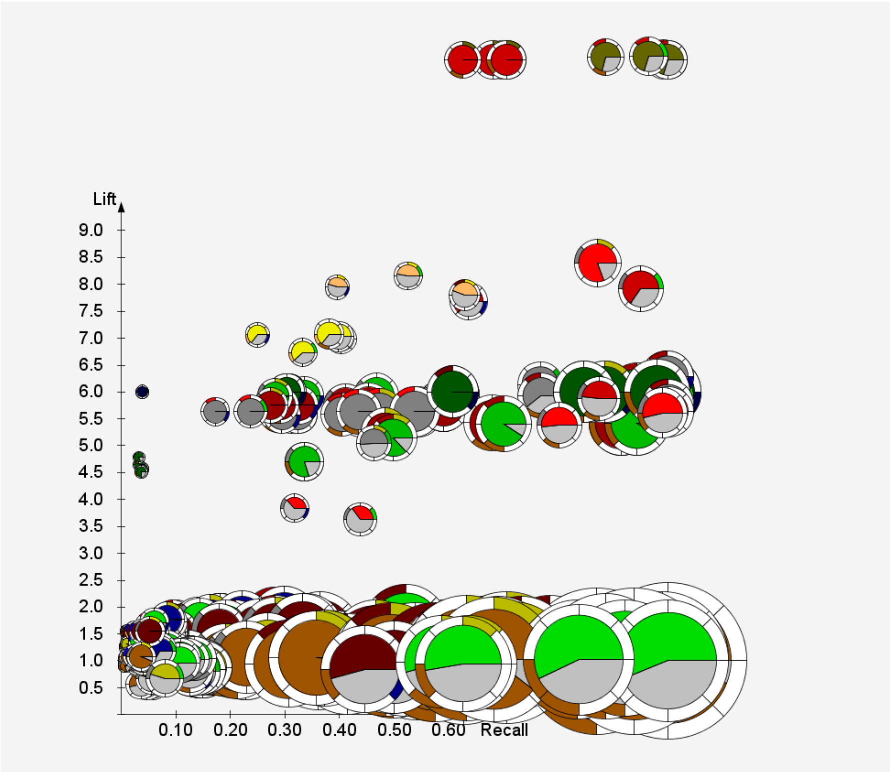
The entire analysis of this rule set with respect to the single concept can be accomplished in about three minutes. This allows to check for multiple concepts in little time which can greatly reduce the response time in case the analysis is done due to quality issues in the underlying domain.

5.4 Summary and Discussion

The (re)implementation of the Information Miner mainly pursued two objectives. First and foremost to provide empirical evidence that the scientific ideas withstand industrial requirements and use cases. The next chapter will show results from such projects. The second goal was to establish a framework for research and rapid prototyping. The framework mainly provides an extensible set of interconnectable tasks that depend on each other and that can interchange data. This minimalistic approaches allows for a steep learning curve and maximal freedom in implementing the tasks. Academic ideas need to be prototyped very quickly in order to decide whether the postulated algorithm or principle generates the planned results. These prototypes can be quickly plugged together and test-driven in the Information Miner.

The Information Miner was not considered and targeted a competitive solution to products such as KNIME, RapidMiner or WEKA. The requirements in terms of user assistance, training, support and quality control would clearly exceed the committable staff for development. Therefore, particular features can be made accessible with the above-mentioned solutions. Essential parts of the rule visualization presented in Chapter 3 have been implemented as a supplementary package to the KNIME analysis suite.

The chapter shall be closed with some software metrics and platform specifications. The Information Miner frontend is written entirely in Java 1.6. The total number of lines of code is about 53,000 of which approximately 22,000 are related to the visualization and filtering approaches presented in this thesis. The software is available under `http://bit.ly/R79aCB`.



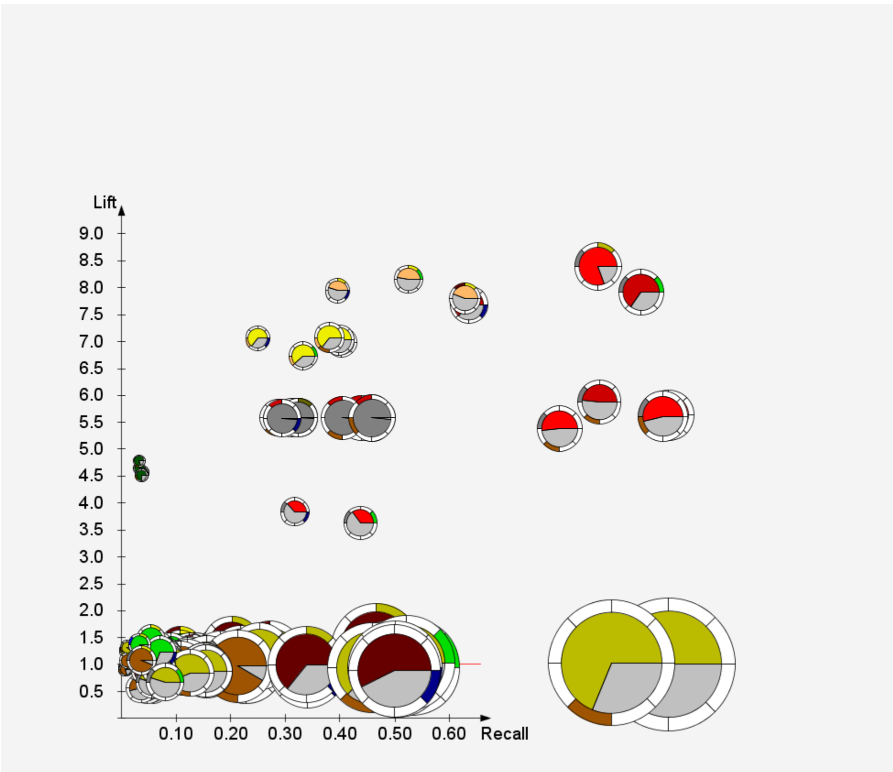


Figure 5.5: *Top: Full set of 904 rules that need to be filtered as they cannot be assessed manually in an exhaustive manner. Bottom: Using the linguistic concept in Figure B.11 and Figure B.12 with a threshold of 10% yields already a reduced set of rules.*

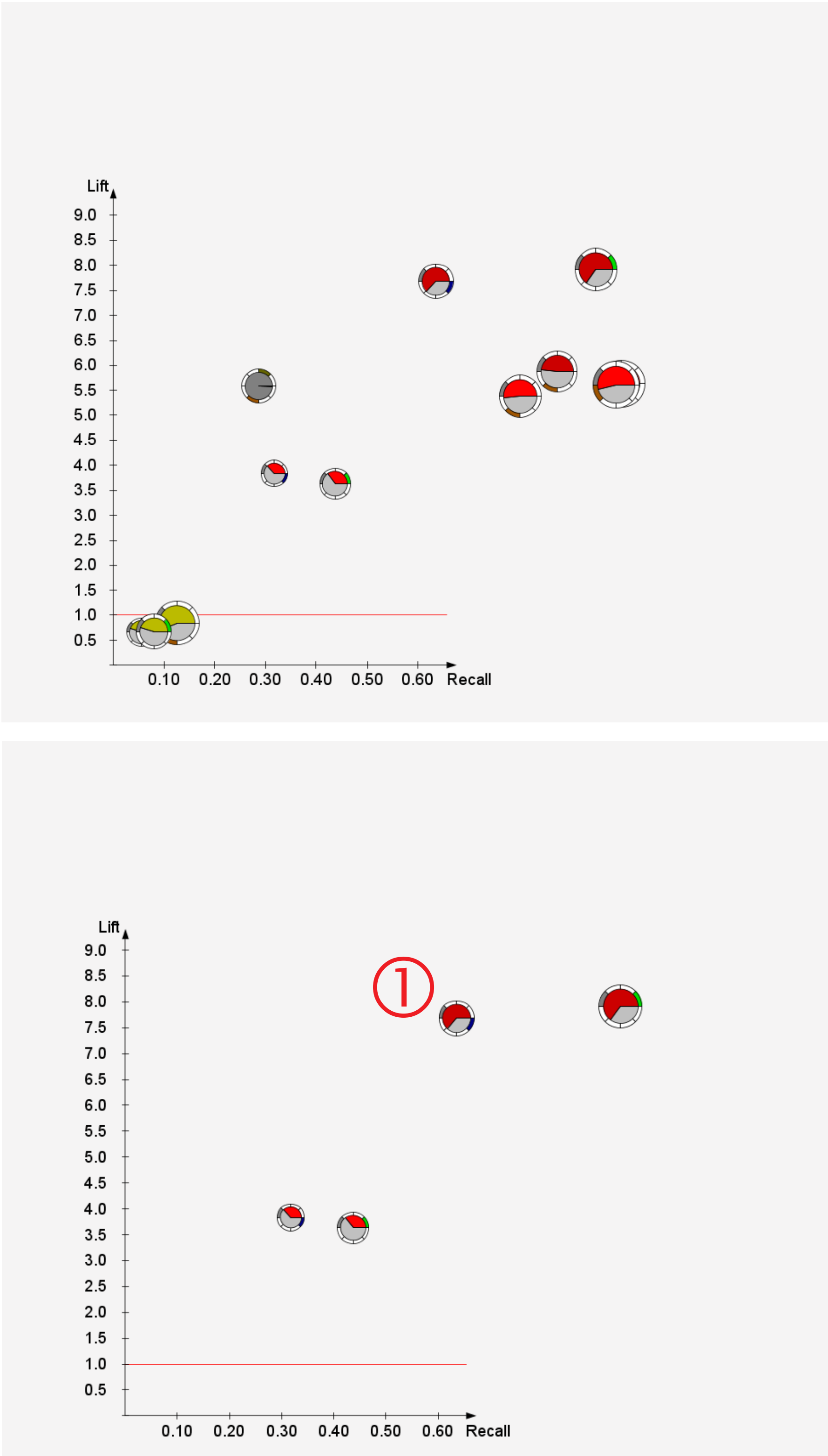


Figure 5.6: *Top: Using the linguistic concept in Figure B.11 and Figure B.12 with a threshold of 50% results 11 highly relevant rules. Bottom: Using a threshold of 90%, we are left with only four rules. The confidence and lift time series of rule ① are shown in Figure 5.7.*

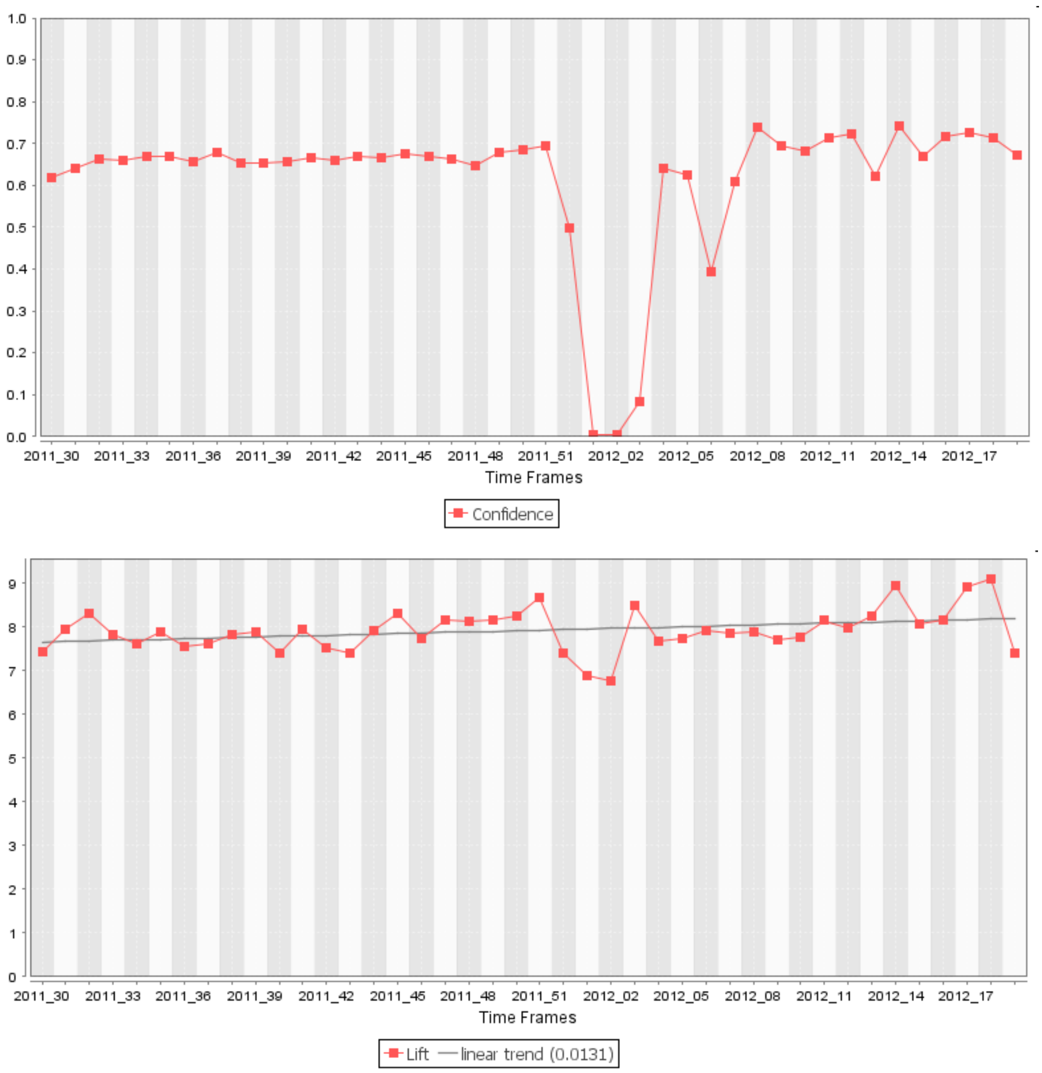


Figure 5.7: *Confidence and lift time series of the rule ① from the bottom chart of Figure 5.6. The rapid dip of the confidence around the turn of the year amounted for the high membership degree to one term of the linguistic concept "increasing local confidence and stable global lift". The trend line on the lift chart at the bottom shows that it clearly contributed to the high membership degree as the trend can be characterized as mainly unchanged with respect to the supplied fuzzy partition (see Figure B.9).*

6 Evaluation

In this chapter I will apply and evaluate the methods of the framework that were proposed and introduced in the last chapters. I will start out with an artificial data set to highlight the applicability of the linguistic filtering. After that, I will go on with analyzing real-world data sets to demonstrate the feasibility and practicality of the application.

6.1 Artificial Data Set

I will use an example scenario that goes back to my early work on visualizing and refining results of data mining algorithms [STEINBRECHER 2006, STEINBRECHER and KRUSE 2008b]. This research was carried out at a large automobile manufacturer and I will apply the linguistic filtering also to some real-world data later (Section 6.2). For quality control, this manufacturer logs for each vehicle its parts configuration and every service incident with respective time stamp and failure code. For this artificial hand-crafted example, five attributes are stored for every car: Time (referring to the time the respective database entry was assessed), Country (to which the car is sold), Engine type, Air condition type and the Class variable. There are five different countries, five air condition types, three engine types and the class variable is indicating a failure by a Boolean value (okay vs. fail). The time has four discrete values mimicking four months where the respective state of cars was assessed. I implemented the following peculiarity: One special type of air condition fails more often in two countries. The average failure rate in the other countries is

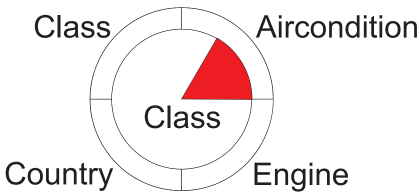


Figure 6.1: *Rule glyph used to visualize the rules induced from the artificial data set. Note, that the* Class *attribute has a reserved segment on the antecedent ring which will never be used because the* Class *is exclusively used as the consequent attribute in this example.*

around 15%. (which was the case in a special real-world example we dealt with.)[1] The failure rates for the two designated countries grow to 30% and 40%, respectively. All given probabilities get added noise with a magnitude of ±4%, that is, the failure rates in the "unaffected" countries may range from 11% to 19%.

Figure 6.2 shows 69 rules at two different time frames (top: first frame, bottom: last frame). The rules were induced according to the workflow introduced in Chapter 4 in theory and Section 5.2 as an implementation. Each rule indicates the above-mentioned class variable in its consequent. One could argue that this rather small rule set does not need any filtering at all. However, it should be obvious that a rule set containing several hundreds of rules—as it may happen to be the case in the real example later on—would be intractable to be assessed manually. The rule glyph assignment is depicted in Figure 6.1. Note that the outer segment belonging to the attribute Class is never filled: As the Class attributes is only represented in the consequent, it never occurs in the antecedent. However, a place must be reserved for it as for general rules any attribute can be in the consequent. For better evaluation, I color-coded only the attribute-value combinations that we intent to detect: The assignments $\mathsf{Class} = \mathsf{fail}$, $\mathsf{Aircondition} = \mathsf{AC1}$, $\mathsf{Country} = \mathsf{OM}$ and $\mathsf{Country} = \mathsf{AEG}$ are drawn in red and orange whereas all other assignments are drawn in gray.

When filtering against the concept "lift is increasing", that is,

$$\xi_1 = \left\langle \Delta_{\text{lift}} \text{ is incr} \right\rangle,$$

with a threshold of 50%, the two remaining rules in the set are exactly the above-mentioned rules:

[1] Note, that this was a special selection of cars which led to this rather large failure rate.

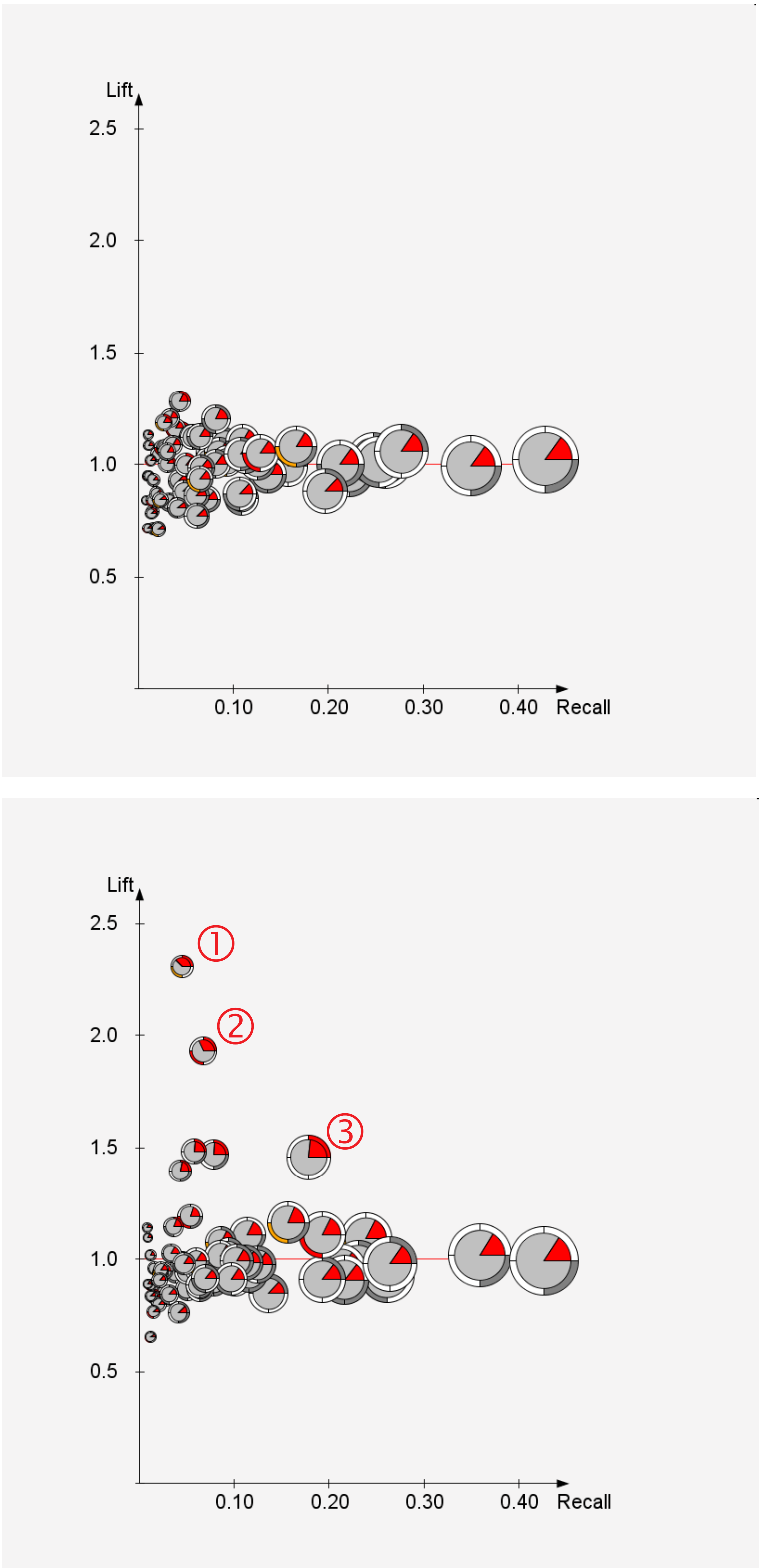


Figure 6.2: *Rule set of 69 rules induced from the artificial data set. Top: Rule set at the first time frame. Bottom: Rule set at the last time frame together with three marked rules that rank highest with respect to the two linguistic concepts discussed in the text.*

① Aircondition = AC1, Country = OM → Class = fail

② Aircondition = AC1, Country = AEG → Class = fail

If we apply the concept “support is increasing and confidence is increasing”, that is,

$$\xi_2 = \left\langle \Delta_{\text{supp}} \text{ is } \mathsf{incr} \text{ and } \Delta_{\text{conf}} \text{ is } \mathsf{incr} \right\rangle,$$

the rules marked ①–③ yield the highest membership degrees. Rules ① and ② represent the oddity that was intentionally implemented into this fictitious data set. Rule ③ is a more general rule (only Aircondition in the antecedent) that has ① and ② as specializations:

③ Aircondition = AC1 → Class = fail

Summary

The previous analysis showed that the proposed methodologies can reveal patterns and dependences exhibiting a certain linguistically describable behavior. To be clear: The filtering approach can only act as a hypothesis generator and reduce the number of rules that need to be assessed manually to a size that is feasible. The threshold (of the minimum membership degree of a rule with respect to the linguistic concept) should be chosen such that e. g. a certain percentage or absolute number of rules remains. Comparing thresholds across different concepts does not make sense. The rule ③ would eventually reappear also for concept ξ_1 by lowering the threshold. However, it would not appear as early (in the third place) as for concept ξ_2.

The next sections present applications of the visualization and linguistic filtering approaches in cooperation with industrial partners and provide empirical evidence of the feasibility of my proposed ideas.

6.2 Car Manufacturer

The increasing complexity of modern automobiles imposes high demands on the quality control. With millions of cars sold, a recall due to a design flaw will cause severe, if not irreparable, damage to the company's reputation and must be avoided by all means. The analysis of vehicle failures reported by service garages is therefore a vital means of quality control. Association rules have proven to be very appealing for this task. Since a vehicle manufacturer needs to detect evolving failure patterns early enough (to fix the problem along with the next inspection instead of risking a mass recall) the technique must be sensible to small subsets of cars. Hence, the minimum support parameter for a frequent pattern induction algorithm will be very small (likely below 0.1%). This in turn leads to a huge number of rules which in addition to that must be evaluated at different time steps.

The data set under analysis is a real-world set of approximately 300,000 tuples that contain 180 attributes. Since this data set was issued by an industrial partner, I am not allowed to provide confidential information such as the meaning of the attribute values or the specific interpretation of the class variable. All I can tell is, that every tuple in the data set represents a unique car that left the production plant of the vehicle manufacturer. Since for every car the time of a failure was logged as well, the full set of tuples can be partitioned into time frames.

I used a preprocessing technique [STEINBRECHER and KRUSE 2008b] based on Bayesian networks [PEARL 1988] to induce a set of attributes that should serve as antecedent attributes of the rules to be visualized.[2]

6.2.1 Analysis 1

For the first analysis, the data set was subdivided into three time frames (beginning, middle and end of the entire data collection period). It was possible to identify a small set of meaningful attributes that were used

[2] See Section 4.5.2 for details.

to generate rules whose temporal trajectories were visualized. Figure 6.3 shows a set of 760 rules at three different times. Note, that in this example the project partner chose the shading of the interior to encode the confidence rather than a pie chart.[3] Two scenarios were to be investigated and were encoded by the following concepts:

1. *Lift is decreasing*
   This pattern could show that some failure pattern is vanishing (e. g., because a countermeasure shows to be effective).
2. *Confidence is increasing and support is increasing*
   Such a pattern would correspond to an emerging problem that needs immediate attention.

6.2.2 Feedback

The analysis actually revealed numerous rules that were interesting to experts. To simplify the assessment, I superposed the rule locations of the second and third chart with the first and indicated the motion with arrows. Four rules that showed an interesting behavior and could be assigned a meaning by experts are numbered in the figure: Rule 1 and 2 had a large membership degree to the first concept: They represent a shrinking sets of cars whose confidence is also dropping. More interesting, however, is the rapidly lessening lift which gave rise to the conjecture that the cause for the failure had been successfully addressed. Contrary, rules 3 and 4 represent sets of cars with increasing failure rate (confidence is increasing indicated by the darkening of the interior of the icons).

6.2.3 Analysis 2

For the second analysis, the data set was subdivided into five time frames. Figure 6.5 shows two ways of presenting an untreated rule set to the user. The left upper chart shows the rule set at the beginning of time (with respect to the underlying data set). The state of the rule set in the middle of

[3] See Section 3.2 starting on page 41.

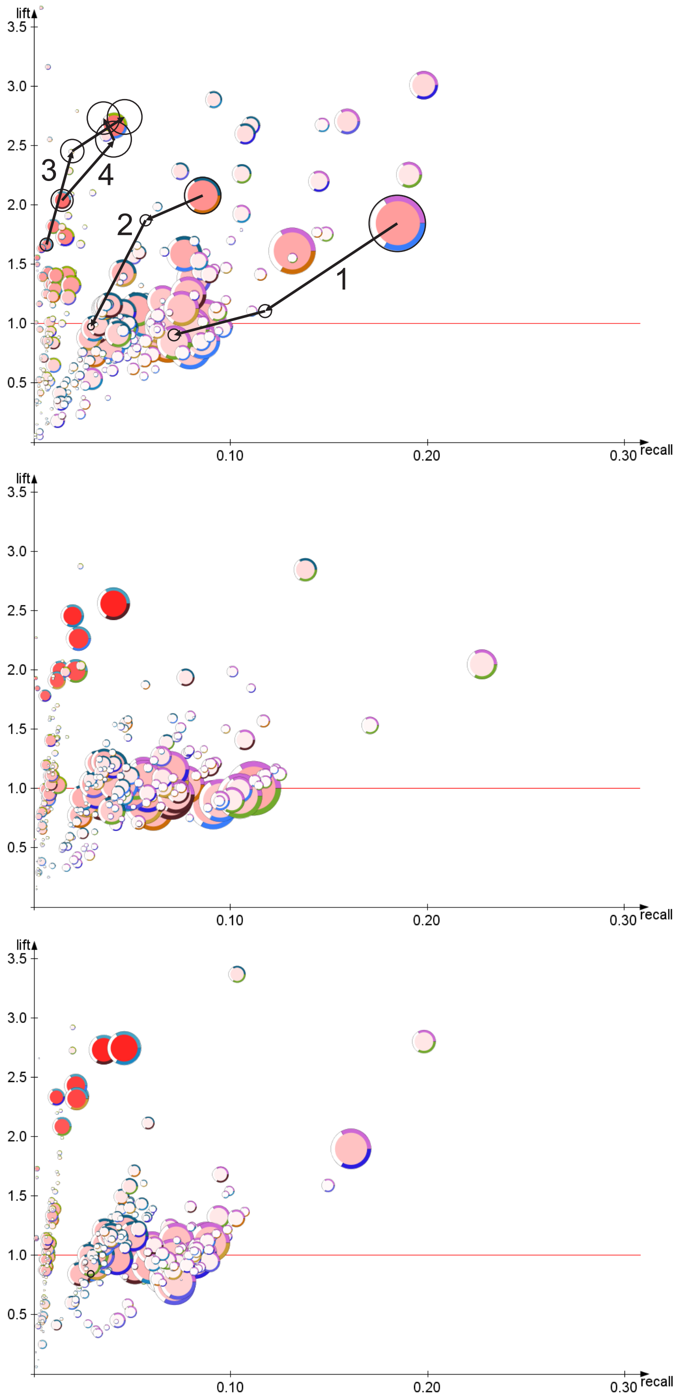


Figure 6.3: *Real-world application of a set of vehicles with a binary class variable: failure and no failure. Only rules indicating a failure are depicted. Two attributes were used to form the rules (hence two filled regions in the outer ring of every rule), therefore no overlapping of covered database cases could occur. The three charts show the rules at the beginning, the middle and the end of the production period. To assess the motion of the rules, I superposed the final locations of the rules with the first image and indicated the corresponding rule with an arrow.*

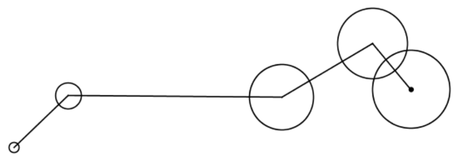


Figure 6.4: *Visualization of an entire rule history over five time points.*

the entire time span is depicted in the right upper chart, whereas the final state at the end of time can be seen in the left lower chart. In this analysis, for every distinct time step for which data is available, a rule is depicted just as a circle to not clutter the figure. Consecutive temporal steps are connected by a straight line (indicating the movement as it would happen in the animation). Additionally, the last location of a rule is marked by a small dot, which serves as an arrow head (which would be to small to recognize as one). An intended trajectory of a single rule can be seen in Figure 6.4.

As easily can be seen, there is a demand of thinning out the rule set because it is not practical to assess the full set manually since it produces an overwhelmingly large number of moving objects even when dealing with relatively small rule sets.

Example 1

The global evaluation method using Bayesian networks [STEINBRECHER and KRUSE 2008a] found the attributes RoadType and Temperature (of the area where the car was last used) to have major impact on the class variable.

For the first analysis, we are interested in rules that exhibit a rather constant lift but become more probable over time (that is, their confidence is increasing):

$$\left\langle \Delta_{\text{lift}} \text{ is unch and } \Delta_{\text{conf}} \text{ is incr} \right\rangle$$

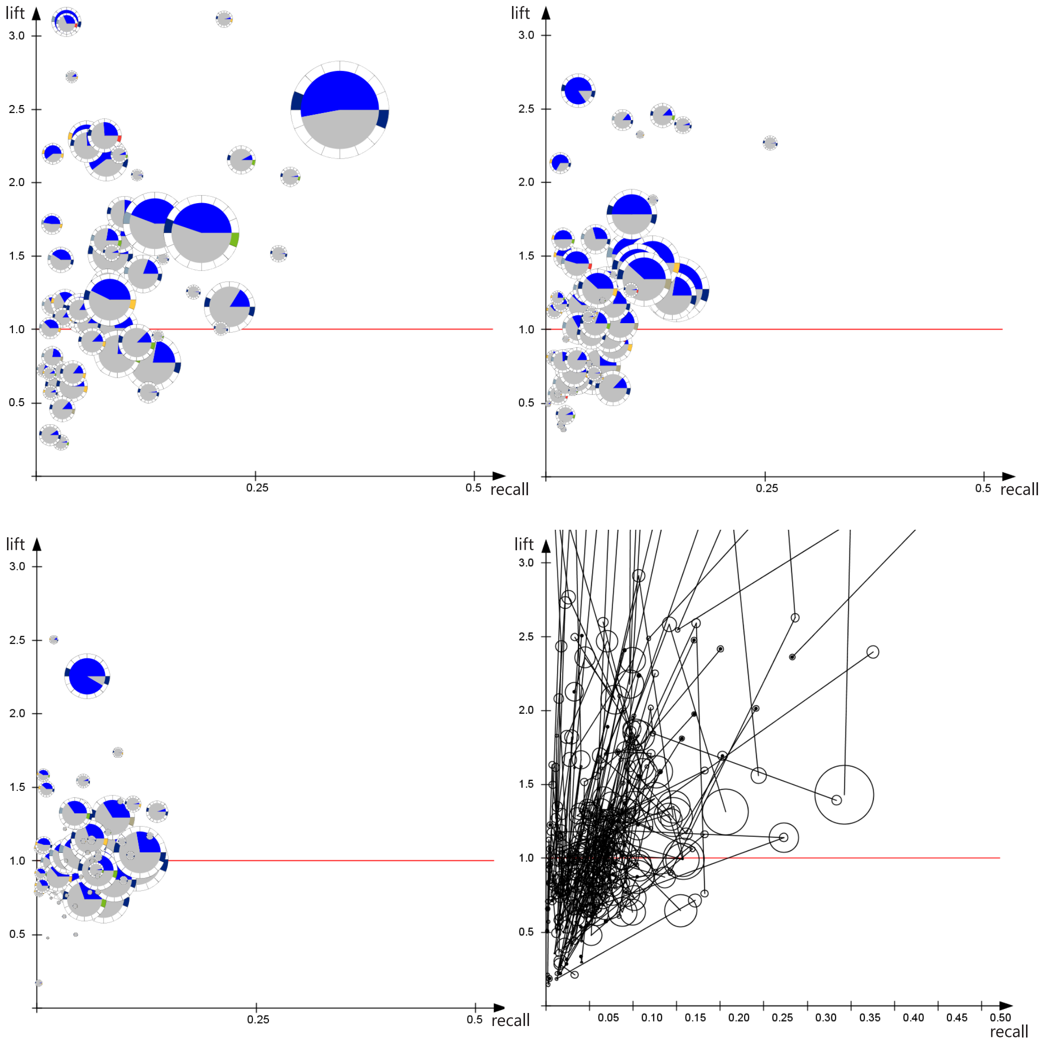


Figure 6.5: *This figure displays the complete temporal evolution of the full set of association rules which were induced from a Bayesian network (see Section 4.5.2). The upper two and the leftmost bottom chart depict the rule set at three different times of the entire time span of the data set, namely the beginning (up left), the middle (up right) and the end (bottom right). The location of every rule is determined by its recall and lift value as these combination have been found more intuitive in that setting. The horizontal line at* $y = 1$ *indicates a lift value of* 1 *(lift-1 line). A rule located on this line has the property that its antecedent does not have an effect on the consequent probability. It is fairly obvious that a manual assessment of every single rule will become tedious. The chart at the bottom right depicts the entire rule's trajectories over the complete time span, where every rule is only represented by its outer border. The lines indicate the trajectory of a single rule. The spot at the end of a trajectory marks the end of the time span, thus serving as a substitute for arrow heads as these would be irrecognizable in the print. The lines ranging out of the top border depict rule movements that leave the viewport used for a consistent presentation. Obviously, none of these representation is of much use unless the number of rules is considerably reduced.*

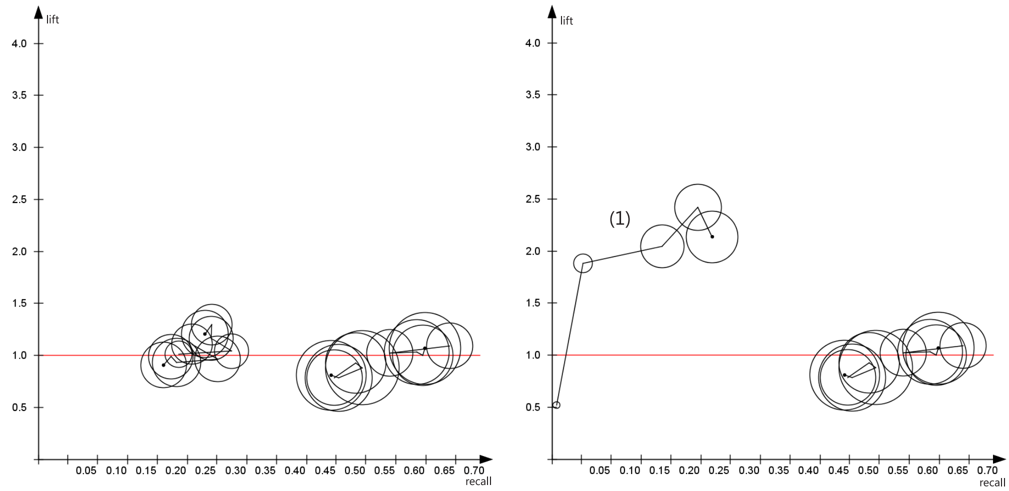


Figure 6.6: *Filtering a rule set of 95 rules. The left chart depicts the result of matching the complete rule set against the linguistic concept* $\langle\Delta_{\text{lift}}$ is unchanged and Δ_{conf} is incr$\rangle$. *The threshold was set to 25%. The right chart shows rules that match the concept* $\langle\Delta_{\text{supp}}$ is incr$\rangle$ *at least to a degree of 65%. The trajectory marked with (1) shows a good example of the obvious support increase since the circle area represents the support.*

The fuzzy partitions of the evaluation measure domains used in these examples are depicted in Figure 6.7. The left chart of Figure 6.6 depicts the resulting rule subset. We display all rules that have a degree of membership with respect to the linguistic concept of at least 25%. It can easily be seen when compared to Figure 6.5 (which depicted the original, unfiltered rule set) that a consid-

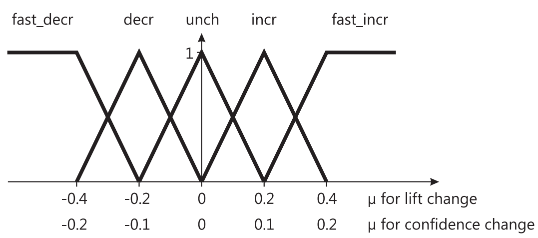


Figure 6.7: *The fuzzy partitions of the domains that describe the lift change and the confidence change rates that are referred to in the linguistic concepts.*

erable reduction was achieved. It can further be stated, that most of the rules exhibit the properties described by the linguistic concept.[4]

Of course, a lift increase from 1 to 1.25 (as it happens to be the case of one rule) can be considered enormous. If so, than this fact has to be reflected in the fuzzy partition of the respective domain.

Example 2

The right chart of Figure 6.6 depicts the result after matching the rule set against the concept

$$\left\langle \Delta_{\text{supp}} \text{ is incr} \right\rangle.$$

The threshold was set to 65% to shrink the number of results. Rule (1) intuitively shows the support increase by the growing size of the circle. The trajectories of the other depicted rules are somewhat indistinguishable but expose the same support-increasing behavior when using the animation visualization instead of the depicted global trajectories.

Example 3

For the third example, I used another underlying rule set. The antecedent attributes are exclusively referring to temperature of the area in which the car was used. The concept to thin out the rule set of 152 rules was the following:

$$\left\langle \Delta_{\text{lift}} \text{ is decr and } \Delta_{\text{supp}} \text{ is decr} \right\rangle$$

The minimum membership degree was set to 15%. The left chart of Figure 6.8 displays the result. This example shows that in linguistic concepts that use more than one criterion one has to select the fuzzy partitions of the respective evaluations measures carefully. As stated in the caption of Figure 6.8, the finding of rule (2) can mainly explained by its rather strong

[4] The lift (the y-coordinate) does not increase or decrease considerably as can be seen from the fact that the rules remain around the lift-1 line. However, the figure does not depict the increase of the confidence which was actually present. The next examples, however, will show only properties that are fully recognizable in the figure.

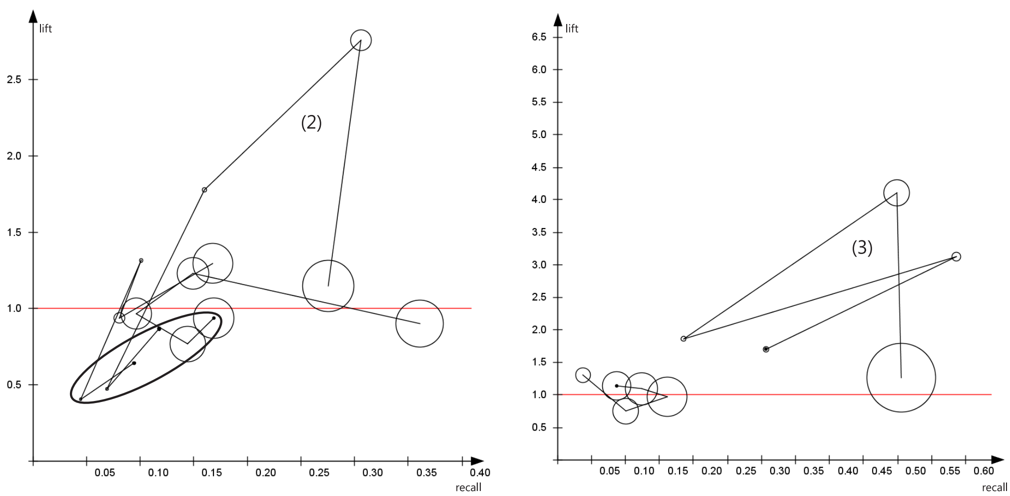


Figure 6.8: *The left chart depicts the result of example 3 with the linguistic concept* $\langle \Delta_{\text{lift}}$ is decr and Δ_{supp} is decr$\rangle$ *and threshold 15%. The resulting rules match the concept intuitively well, however, one has to admit that the support decrease (note that at the end of time there were hardly any cars covered as indicated by the tiny circles which are marked by the ellipse for clarity) governs the concept and the decrease of lift has not a big influence. This can be seen from the fact that the rule (2) has a non-linear lift history but is still covered by the concept. The right chart depicts an example where the concept matching fails due to a non-linearity of the lift history. The trajectory (3) is intuitively not characterized by an "unchanged lift".*

decrease in support. It outweighs the temporary lift increase such that it still matches the linguistic concept.

Example 4

The last example is used to provide evidence that non-linearities in the trajectories may have counterintuitive results. The same rule set as in example 3 was used and matched against the following concept:

$$\left\langle \Delta_{\text{lift}} \text{ is unch} \right\rangle$$

The threshold was raised to 90% to just result in two rules as depicted in the right chart of Figure 6.8. Obviously, rule (3) does not follow the intuition behind an "unchanged lift".

6.3 Second Life Online Community

Another application stems from a cooperation with a company creating content for the 3D online community Second Life [WWW: SL]. In order to evaluate whether online content such as buildings mimicking online stores or museums function as intended, a set of virtual sensors are deployed in that environment. Whenever a player passes by such a sensor (within a specified vicinity) a visit event is logged. I analyzed such a log file for 100 sensors that had been logged for six months. The data set was then discretized into frames of one month length each. A common representation of such a log file is a so-called cooccurrence graph. The nodes comprise the sensors whereas the edges represent that the sensors connected by that edge have been visited by some set of common users (within a certain time span). Edge weights are used to denote the cardinality of such sets. Figure 6.9 shows the sum graph of the data set, that is, the cooccurrences of six months among 100 locations.

I will present two analyses of the same data set: The first will apply only the linguistic filtering whereas the second analysis will discuss how to use the rule visualization to display changes inside the graph structure.

6.3.1 Analysis 1

This data set contains player contacts at certain locations within a 3D environment over a time period of six months. The edge weights indicate some subgraphs that are worth looking closer at. The threshold θ has been chosen to be 1,000 to induce the candidate node sets.[5]

I will match two linguistic concepts against these graph candidates: first, the interest lies on decay, that is, in graphs that show kind of a dissolving behavior, translating into a decreasing completeness and decreasing total weight. In the sample data this might indicate locations whose attractiveness is diminishing. A second concept that is going to be assessed is that of an establishing pattern. An increasing average edge weight and devia-

[5] See Section 4.5.3 for details.

tion (of edge weight) might point out a phase of initial apparent random visiting of multiple locations which accumulates into a strong favored visiting pattern.

Concept 1: Decreasing completeness and decreasing weight

In order to evaluate the membership degrees to the linguistic concept

$$\left\langle\ \Delta_{\text{comp}} \text{ is } \mathsf{decr} \text{ and } \Delta_{\text{wght}} \text{ is } \mathsf{decr} \right\rangle,$$

we need to declare a fuzzy partition on the change rate domains of the functions comp and wght. Figure 6.10 displays all used fuzzy partitions. I am using three fuzzy sets. Note the asymmetric slopes of the borders. This setup has proven to be useful in this context since “unchanged” has a more strict semantic to users than the adjectives “decreasing” and “increasing”. I am deliberately departing from this well-known convention of the area of fuzzy control to use fuzzy partitions with unity membership degree sum across the entire domain (as it is e. g. the case in Figure 6.7). This constraint eases the construction of smoothly operating fuzzy controllers but does not contribute to the concept formulation. The respective values that determine the particular fuzzy partition can be read from the four different horizontal scales. These values have been selected with respect to the data set since the quantity that renders a slope to be highly decreasing or increasing differs, of course, from data set to data set.

If we apply the linguistic concept to the candidate graphs and select the one with the highest membership degree (ignoring the remaining ones here for brevity), the subgraph whose history is depicted in the upper row in Figure 6.11 scores 71%. Most of the high degree can be attributed to the rapid loss of visits in the last two months. The membership degree was evaluated via

$$\min\left\{\mu^{(\mathsf{decr})}_{\Delta_{\text{comp}}}(C_1),\ \mu^{(\mathsf{decr})}_{\Delta_{\text{wght}}}(C_1)\right\} = \min\{0.71, 0.84\} = 0.71,$$

with C_1 being the set containing the five nodes. Data inspection revealed a newly set up structure which was heavily frequented shortly after opening, followed by abating excitement.

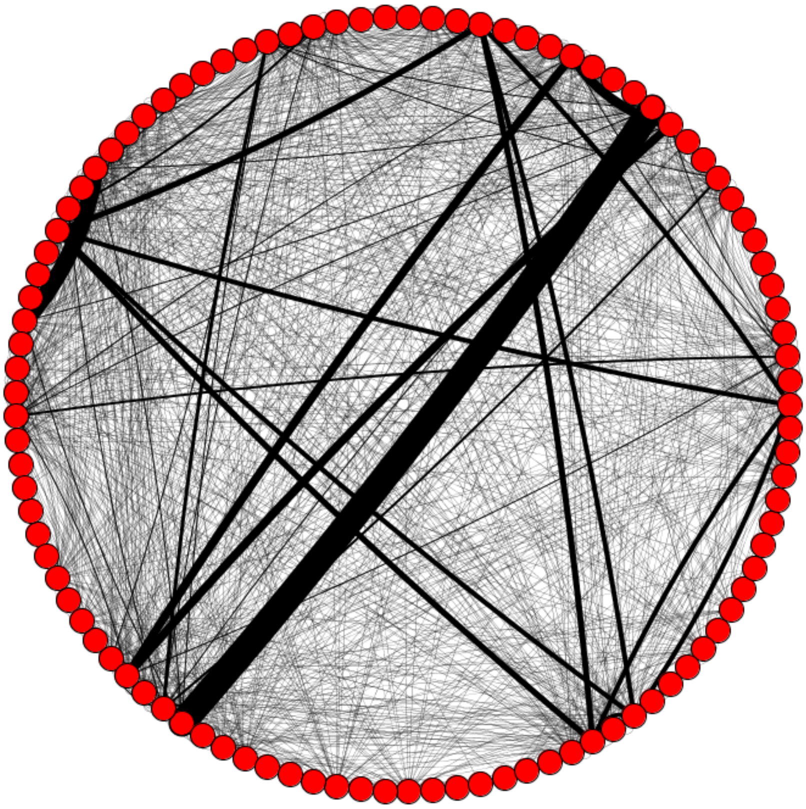


Figure 6.9: *The sum graph of six months of visiting history of players in a 3D environment. I will match the major components (extracted via a user-specified threshold) against two linguistic concepts.*

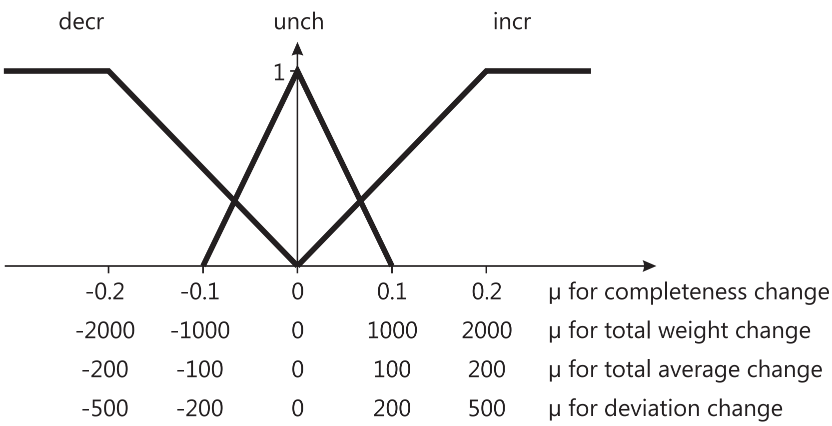


Figure 6.10: *Fuzzy partitions for the four graph measures used in the two analyses.*

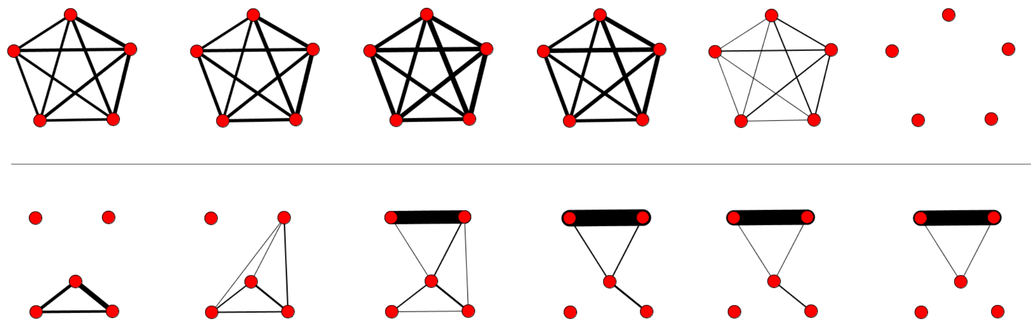


Figure 6.11: *The histories of the two subgraphs that scored highest in the two analyses. The upper row shows a six-month development of five locations that were heavily visited but declined rather rapidly towards the end. It yielded a membership degree of 71% to the concept "completeness is decreasing and total weight is decreasing". The lower row depicts the best-scoring subgraph of the second experiment resulting in a membership degree of 83% to the concept "average is increasing and deviation is increasing."*

Concept 2: Increasing average and increasing deviation

I follow the same procedure to find the subgraph that scores best on the concept

$$\left\langle \Delta_{\text{avg}} \text{ is incr and } \Delta_{\text{dev}} \text{ is incr} \right\rangle.$$

The lower part of Figure 6.11 shows the resulting graph with a score of

$$\min\left\{\mu^{(\text{incr})}_{\Delta_{\text{avg}}}(C_2),\ \mu^{(\text{incr})}_{\Delta_{\text{dev}}}(C_2)\right\} = \min\{0.83, 0.89\} = 0.83,$$

The graph shows an establishing link between two nodes in parallel with a weakening in the remaining edges thus rendering the graph history becoming more unbalanced.

6.3.2 Analysis 2

Even though I have proposed methods to deal with temporal changes within series of cooccurrence graphs [STEINBRECHER and KRUSE 2010, 2009a], we can extract more information when we induce rules from these graphs and apply the proposed visualization. This is advantageous because of the following reasons: A rule of the form "If sensor X was visited, sensor Y was also visited in p% of all cases" is more meaningful than just an edge with an associated weight (which corresponds to the absolute support). The above rule depicted with our visualization immediately reveals how many visitors compared to other rules are covered and how that number compares to the general visiting activity of a sensor. Further, a certain rule set will have a fixed visual representation whereas the corresponding graph needs to be laid out which can create different impressions based on the layout algorithms. However, coordinates from any graph representation cannot carry the information of the rule icons.

Figure 6.12 shows the rules induced from the visiting events of the abovementioned log file at three different time steps. Rules have been induced that covered at least 1% of all visits and had a minimum confidence of 1%. Note that this low confidence value is not unusual as we can expect a meaningful lift value since this is the ratio between confidence and consequent probability. Note also, since a rule represents an edge, the antecedent contains exactly one attribute. Hence, I have omitted the border of the rule icons.

I discuss the findings based on two subsets of rules. The subset marked by the ellipse in Figure 6.12 corresponds to a triangle in the original graph. Note, as the graph is undirected, it is possible that two rules are induced for a single edge if they match the specified criteria. Hence, a triangle might be represented by up to six different rules. All rules show a lift increase to the middle of the time period after which it decreased again. Semantically, this corresponds to an upraise and loss of predictability of the sensors in that subset: A high lift tells that given a visit at one sensor, we can conclude a more likely visit at the sensor described by the rule consequent. Of course, not necessarily by the same user! An inquiry at the company confirmed that a certain modification had been undertaken at the 3D content in the vicinity of these sensors during the middle of the log period. The rules marked by the rectangle in Figure 6.12 correspond to a single edge that was heavily visited as can be seen from the rule icon size. The lift is decreasing, which is not surprising in the underlying 3D setting. The two sensors were placed along a frequented pathway that more and more users were accepting to use. If the general usage increases (and thus the consequent probability) the lift will decrease, as we observe here.

6.4 European Telecommunications Provider

The last and most recent evaluation was carried out with data obtained from a cooperation with a large European telecommunications service provider. This company has a high incentive to address and resolve any support issues in the shortest time possible. As the product and service portfolio is developing, so are the key performance indicators (KPIs), metrics and attributes that are used for assessment.

The specific cooperation with the company's research department gave rise to look into support inquiry data in order to check for conspicuities. The data set consisted of a collection of support inquiry descriptions collected between July 2011 and May 2012.[6] Each support inquiry was described with ten descriptive and one time attribute. I subdivided the data set into time frames of one week each, resulting in 43 time frames. The workflow introduced in Section 5.2 was applied in order to induce the final rule set of 1585 rules. Since there is no dedicated class variable, each of the ten attributes can occur in the consequent.

The following subsections reiterate the analysis with some of the identified rules. A conclusive assessment is given at the end of this section. It turned out that for most concepts there are clusters of rules that exhibit a very similar temporal behavior. For the sake of brevity, I will illustrate one representative of each cluster in greater detail. For each rule, I will give the full time series of the antecedent support, consequent support, joint support, confidence and lift. Figure 6.13 shows the attribute assignment

[6] For reasons of anonymization, the data set was transformed in order not to reveal the true absolute supports of rules as these may give insight into the company's business secrets. The data set was augmented via rescaling the weights of each tuple (that is, basically copying it). Doing so, obviously preserves the relative frequencies and allows to talk about rule evaluation measures. The only figures where this fact has to be taken into consideration are the top charts of figures 6.15–6.20. The augmented data set under analysis consisted of some 4.3 million tuples.

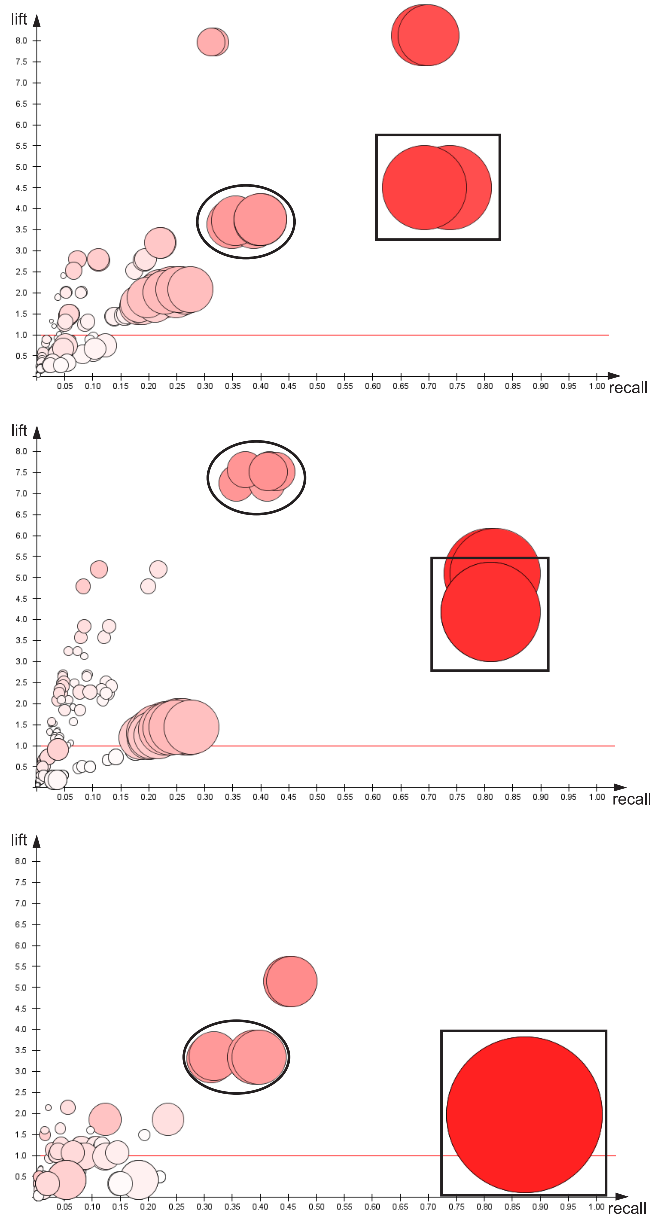


Figure 6.12: *Rules induced from a cooccurrence graph at three different time steps. The two rule sets that are marked by the ellipse and rectangle were identified to correspond to modified artifacts inside the 3D world: The rule set marked by the ellipse showed a collective lift increase towards the middle of the logging period and diminished after. The rules marked by the rectangle exhibit a lift decrease and support increase.*

to the rule glyph for all rules in this analysis. An asterisk ($*$) in the rule description denotes a wildcard, and therefore represents a set of rules. Figure 6.14 depicts all rules in one image at the first time frame. Clearly, some filtering is needed in order to inspect the underlying process.

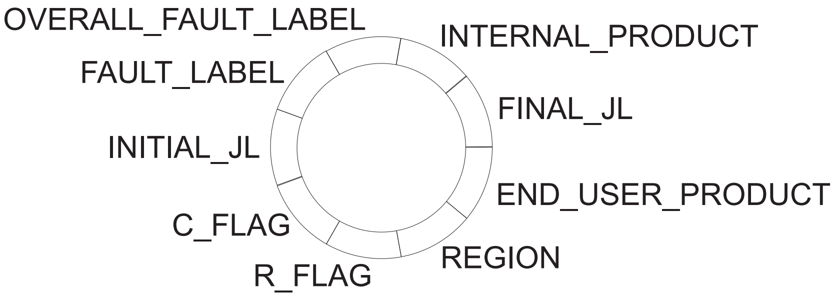


Figure 6.13: *Attribute assignment to the rule glyphs.*

6.4.1 Concept 1: Local lift is increasing

Checking the rules for increasing local lift is a kind of a standard first task as it can reveal information about both changing confidence and support. Two groups of rules are immediately identified when raising the concept membership threshold:

- $\mathsf{OVERALL\_FAULT\_LABEL} = \mathsf{OFL4} \wedge * \rightarrow \mathsf{FAULT\_LABEL} = \mathsf{FL3}$
- $\mathsf{FAULT\_LABEL} = \mathsf{FL3} \wedge * \rightarrow \mathsf{OVERALL\_FAULT\_LABEL} = \mathsf{OFL4}$

Both rule types are obviously complementing each other. Figure 6.15 shows the time series for the following specific rule:

$$\mathsf{FAULT\_LABEL} = \mathsf{FL3} \wedge \mathsf{R\_FLAG} = \mathsf{N} \rightarrow \mathsf{OVERALL\_FAULT\_LABEL} = \mathsf{OFL4}$$

The rule's support time series reveal a drastic drop and increase around the turn of year: The joint support drops from to almost zero before it rises again to roughly the initial level. Since the confidence practically remains constant throughout the year, the huge peak of the lift is its consequence.

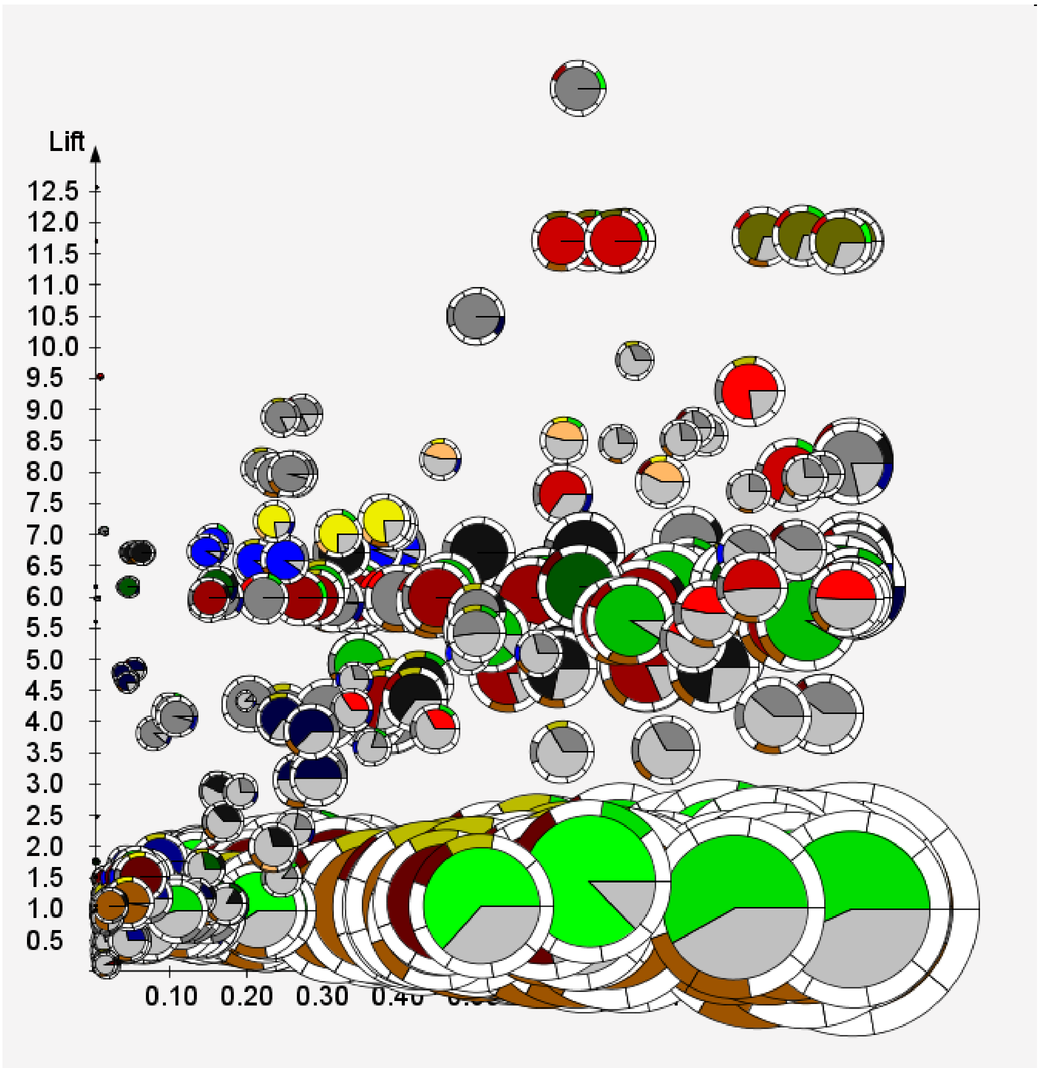


Figure 6.14: *Full rule set with 1585 rules in the first time frame.*

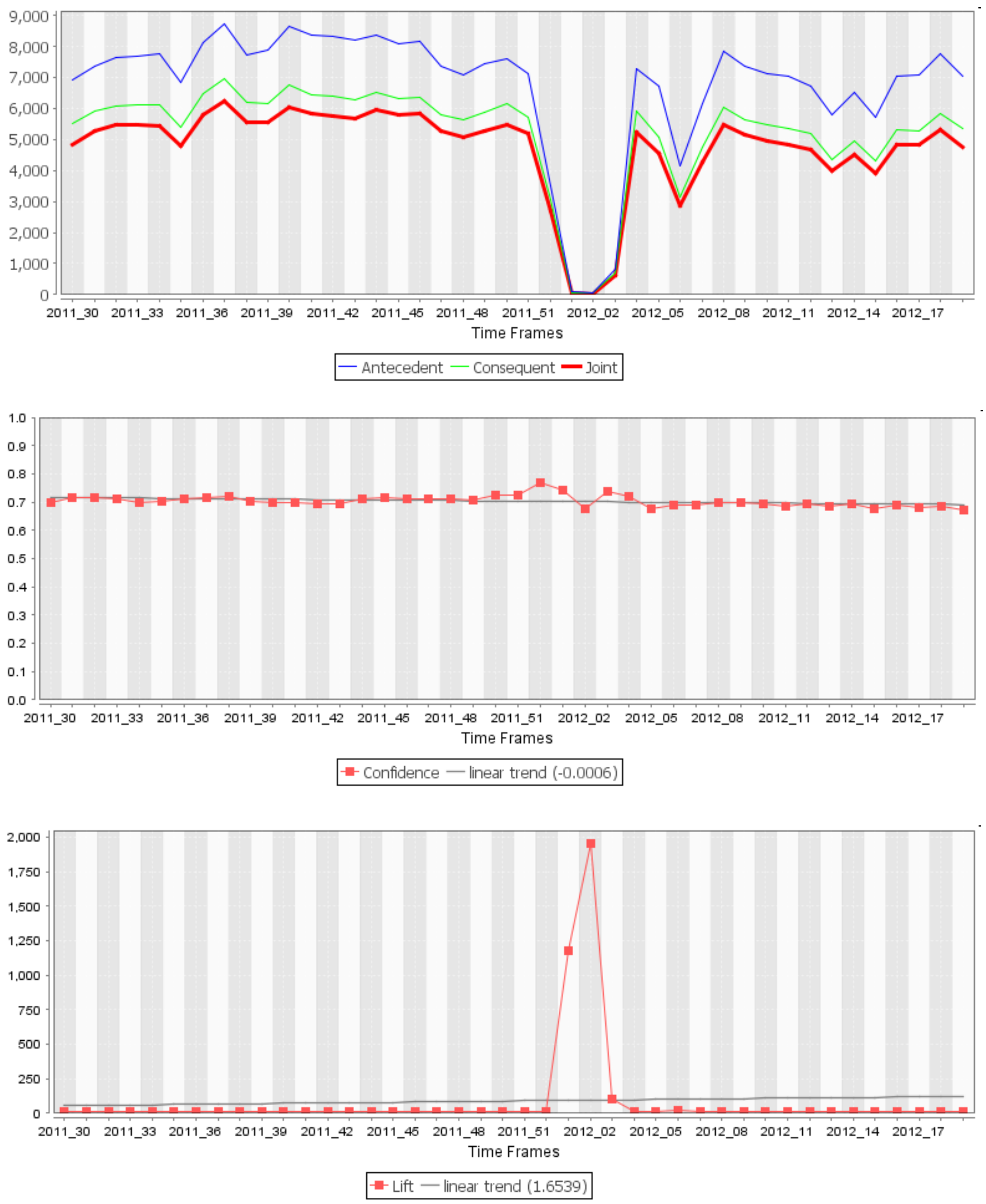


Figure 6.15: *Time series for rule:* FAULT_LABEL = FL3 ∧ R_FLAG = N → OVERALL_FAULT_LABEL = OFL4.

6.4.2 Concept 2: Local support and global confidence are increasing

We just saw that there might be drastic changes in the support, so a deeper look into such rules might be a good idea. In addition to that, I also constrain the confidence to be globally increasing. A stable confidence as in the section before may not be expressive.

The following group of rules is revealed:

$$\mathsf{INTERNAL\_PRODUCT} = \mathsf{P2} \wedge * \rightarrow \mathsf{FINAL\_JL} = \mathsf{NULL}$$

The time series of one instance of this group are depicted in Figure 6.16. Again, things change tremendously around the turn of year: The confidence shoots from around 10% up to 100%. Likewise, the joint support of the rule increases from around 1,500 to values between 17,000 and 20,000.[7] We will bear this temporal location "turn of the year" in mind in order to constrain the following concepts.

6.4.3 Composite Patterns

The two distinguished patterns of the last two concepts (peak pattern in the lift series of Figure 6.15 and the plateau-like pattern of the confidence and lift series of Figure 6.16) give rise to check the rule set for similar patterns. I will discuss both with an example each in the following two subsections.

Peak Patterns

Checking for peak patterns follows the descriptions of Section 4.3.2.[8] The population was initialized with 15 random chromosomes each of which representing the borders of the two flanks of a peak pattern. The genetic algorithm ran for at most 300 iterations and was run on each time series.

[7] Remember, that these absolute values do not reveal the actual numbers for the sake of anonymization! Nevertheless, the ratio between them is the same.

[8] See pages 69 ff.

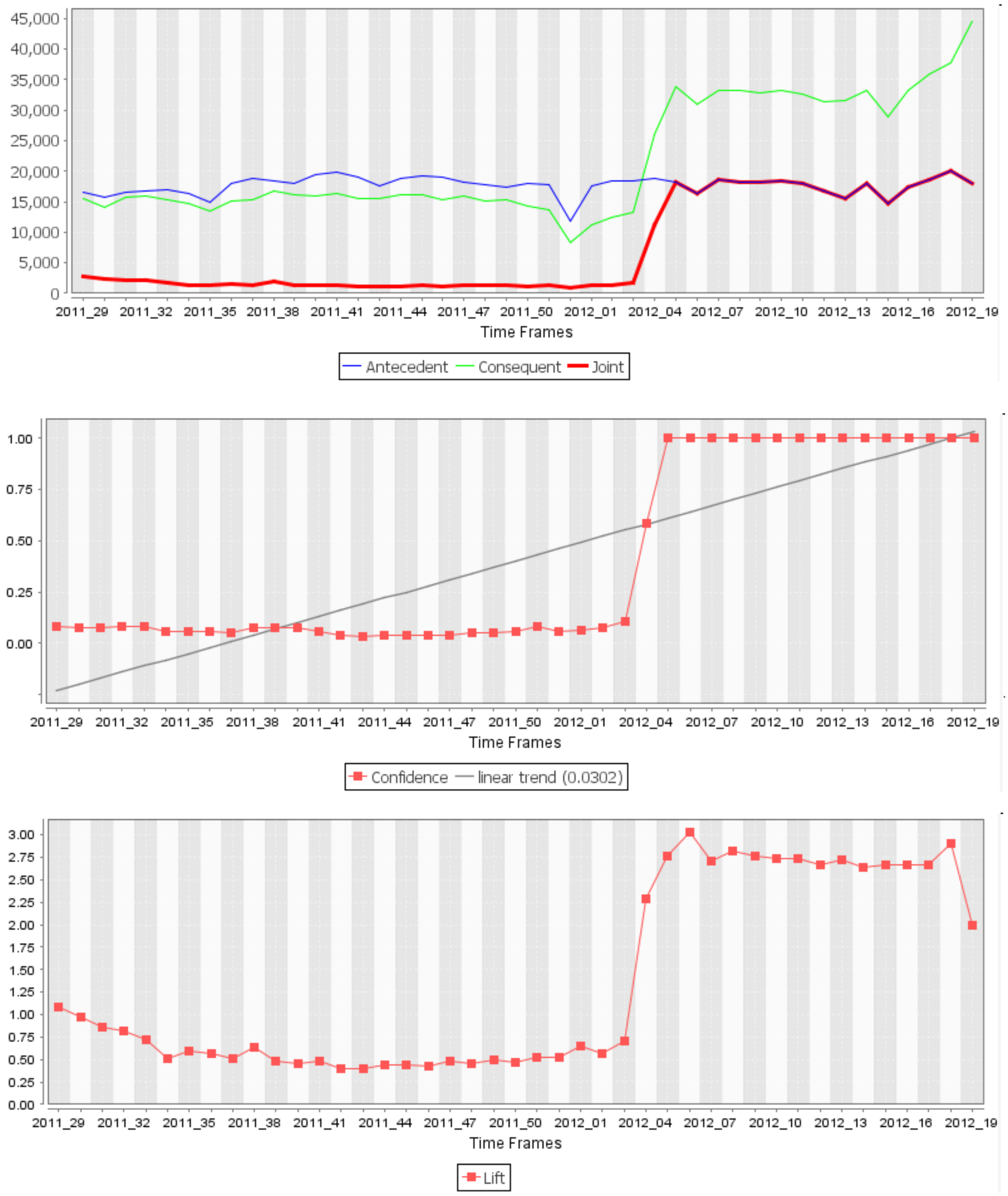


Figure 6.16: *Time series for rule:* INTERNAL_PRODUCT = P2 ∧ C_FLAG = N → FINAL_JL = NULL.

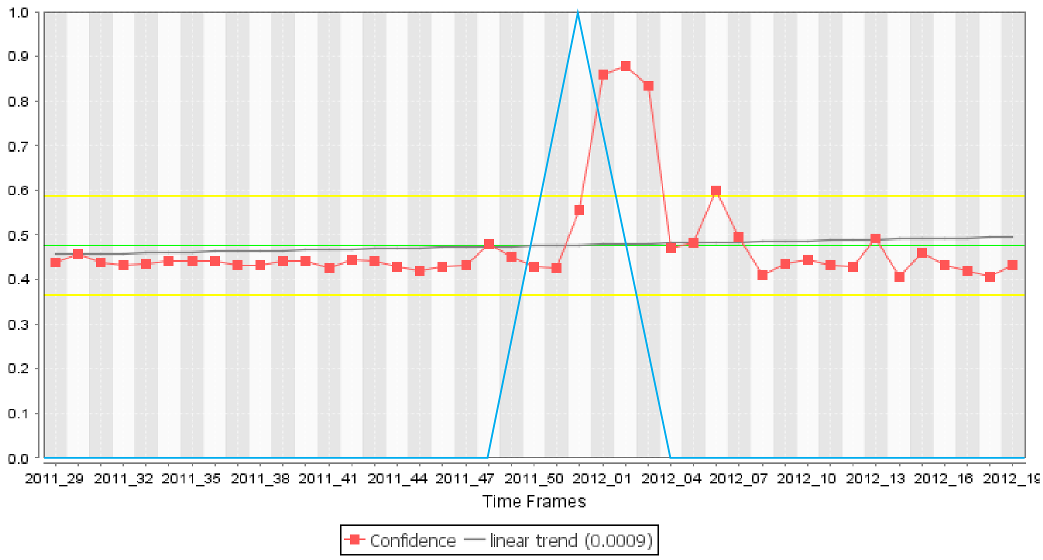


Figure 6.17: *Time series for rule:* INTERNAL_PRODUCT = P9 → OVERALL_FAULT_LABEL = OFL9. *It was identified as the winner time series matching best against the linguistic concept "confidence peak around turn of year". The blue superimposed fuzzy set represents the term "around turn of year".*

I chose to check for peaks in the confidence series since a manual inspection showed that this measure brought up the most diverse patterns and motifs and hence would put a higher challenge on the filtering approach. The average running time of the described composite pattern detection was around 100 ms.[9] Further, the composite pattern was constrained to be present at the "turn of the year" as motivated in the last section. The fuzzy set used for this temporal constraint is shown superimposed as the blue graph in Figure 6.17. The time series in red shows the winner confidence time series for which the linguistic concept "peak around the turn of year" had the highest membership degree.

[9] I refrain from giving more detailed performance measurements here for two reasons: First, the running time for a single rule's time series of 100 ms is short enough to be applied to all rules in the set and still get feedback quick enough. And second, the reported average duration was measured while the application was running in debug mode which lead to larger variation. Controlled tests would have much likely lead to even better results. However, the obtained numbers were found to be good enough to focus on pattern investigation rather than ungrounded optimization.

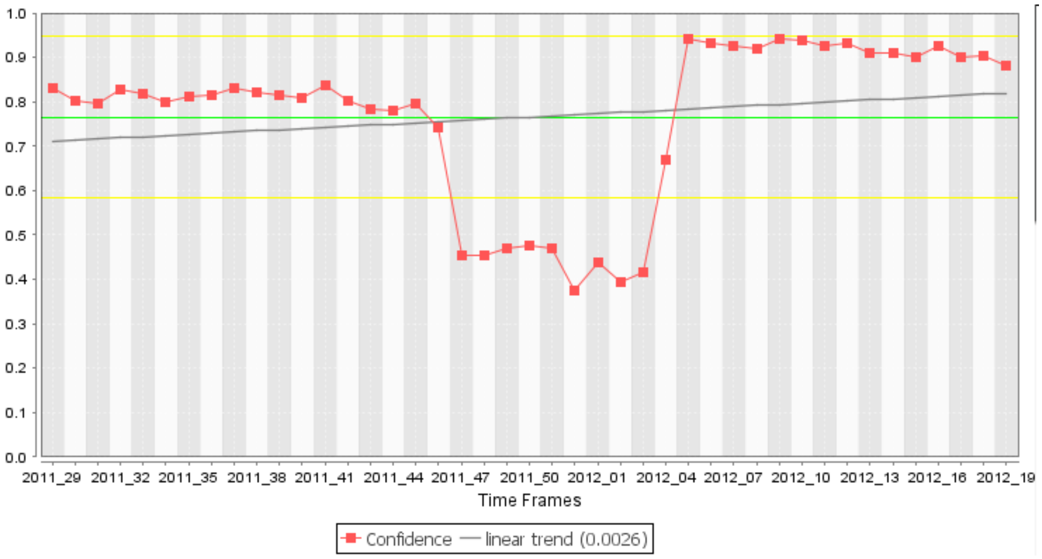


Figure 6.18: *Time series for rule:* FINAL_JL = OK ∧ REGION = R2 → INTERNAL_PRODUCT = P9. *The time series was detected by matching against the linguistic concept "confidence with ditch".*

Ditch Patterns

Investigating the data set revealed another pattern that arose quite often: A ditch pattern where after a decreasing flank and some stable part a steep increase followed (similar to an inverted plateau). In order to find those patters, I modified the peak membership computation function[10] in the following way: First, the proximity constraint of the two flanks (line 14) was removed and second, the two fuzzy sets in line 19 were swapped. I did not apply a temporal constraint here as it was not clear whether the patterns cumulated at a special point in time. The settings of the genetic algorithm were the same as in the peak detection step. Figure 6.18 shows one of the detected time series. A multitude of rules with such confidence patterns was identified; one is depicted here as a representative.

[10] See Algorithm 4 on page 70.

6.4.4 Concept 3: Local support and local lift are increasing

This concept combines elements from the above two analyses (concept 1 and 2). As we will see, we could have also used local lift instead of local confidence. The following group of rules can be revealed (together with the group from concept 1 above):

$$\mathsf{FAULT\_LABEL} = \mathsf{FL3} \land * \rightarrow \mathsf{FINAL\_JL} = \mathsf{JL1}$$

Figure 6.19 shows one rule of that group. Clearly, the turn of year is again playing a dominant role here as the articulated drop of confidence and lift at CW 2/2012 indicates.

6.4.5 Concept 4: Local confidence and global lift are decreasing

Let us now filter for rules exhibiting decreasing trends in their time series. Again, with focus on the turn of year. Matching against this concept, the following single rule can be identified:

$$\mathsf{INITIAL\_JL} = \mathsf{JL2} \land \mathsf{END\_USER\_PRODUCT} = \mathsf{P2} \rightarrow \mathsf{FINAL\_JL} = \mathsf{JL2}$$

Up to CW 3/2012 the rule represents an inclusion (confidence of 100%, also visible as the antecedent and joint supports are equal). All of a sudden, this changes and the rule does not have a meaning any more: Its support drops to and stays at zero. Figure 6.20 depicts this situation.

6.4.6 Feedback

The findings of the previous filtering runs can be summarized as follows: There are conspicuous effects visible around the turn of the year: Be it support, confidence or lift time series, they are likely to change dramatically at this time span. Further, most rules share one attribute in common: FINAL_JL. I compiled an extensive set of rules (containing the above-mentioned rules) and handed the findings over to the company's research staff. An internal assessment showed that the attribute FINAL_JL was only

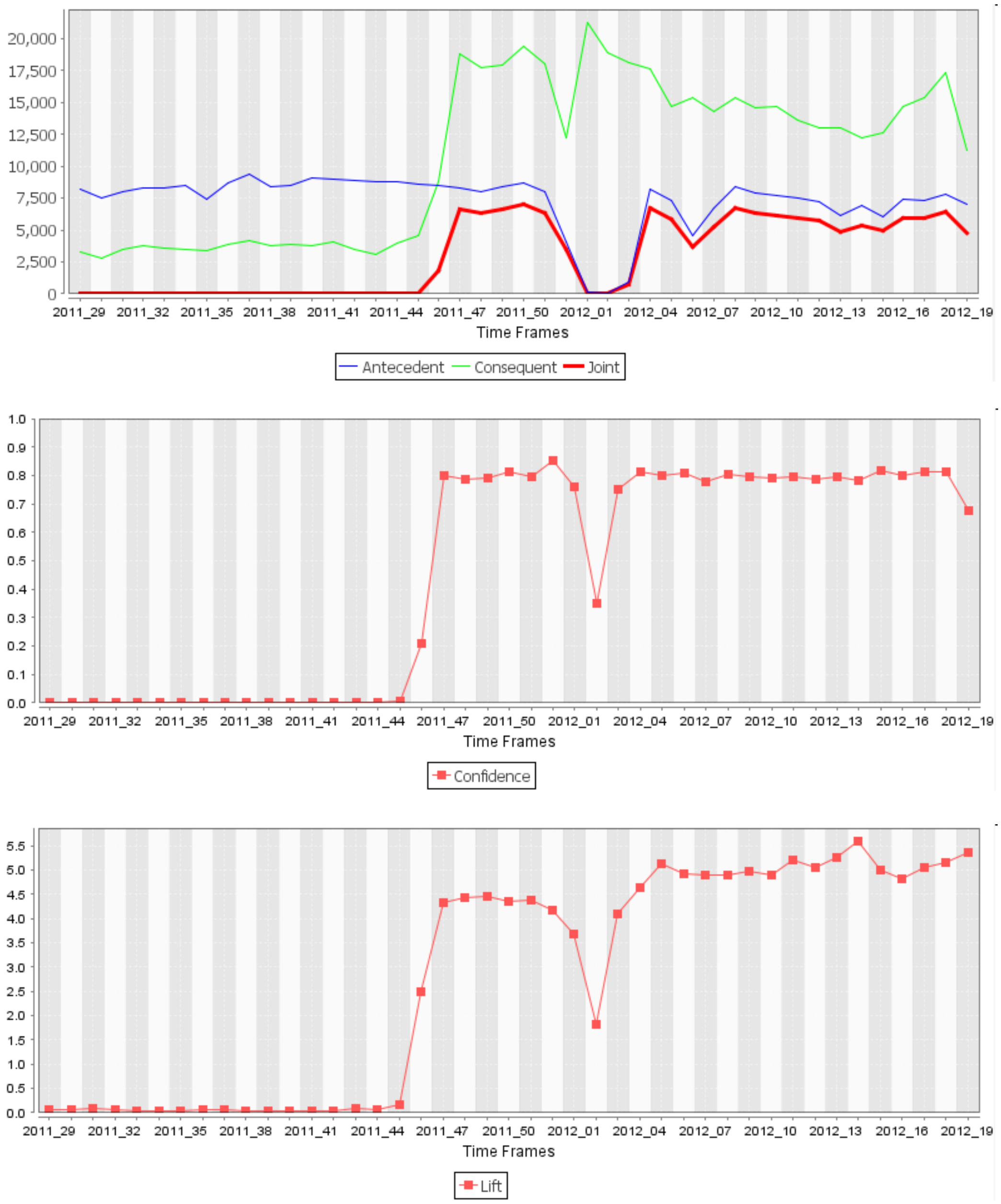


Figure 6.19: *Time series for rule:* INTERNAL_PRODUCT = P9 ∧ FAULT_LABEL = FL3 → FINAL_JL = JL1.

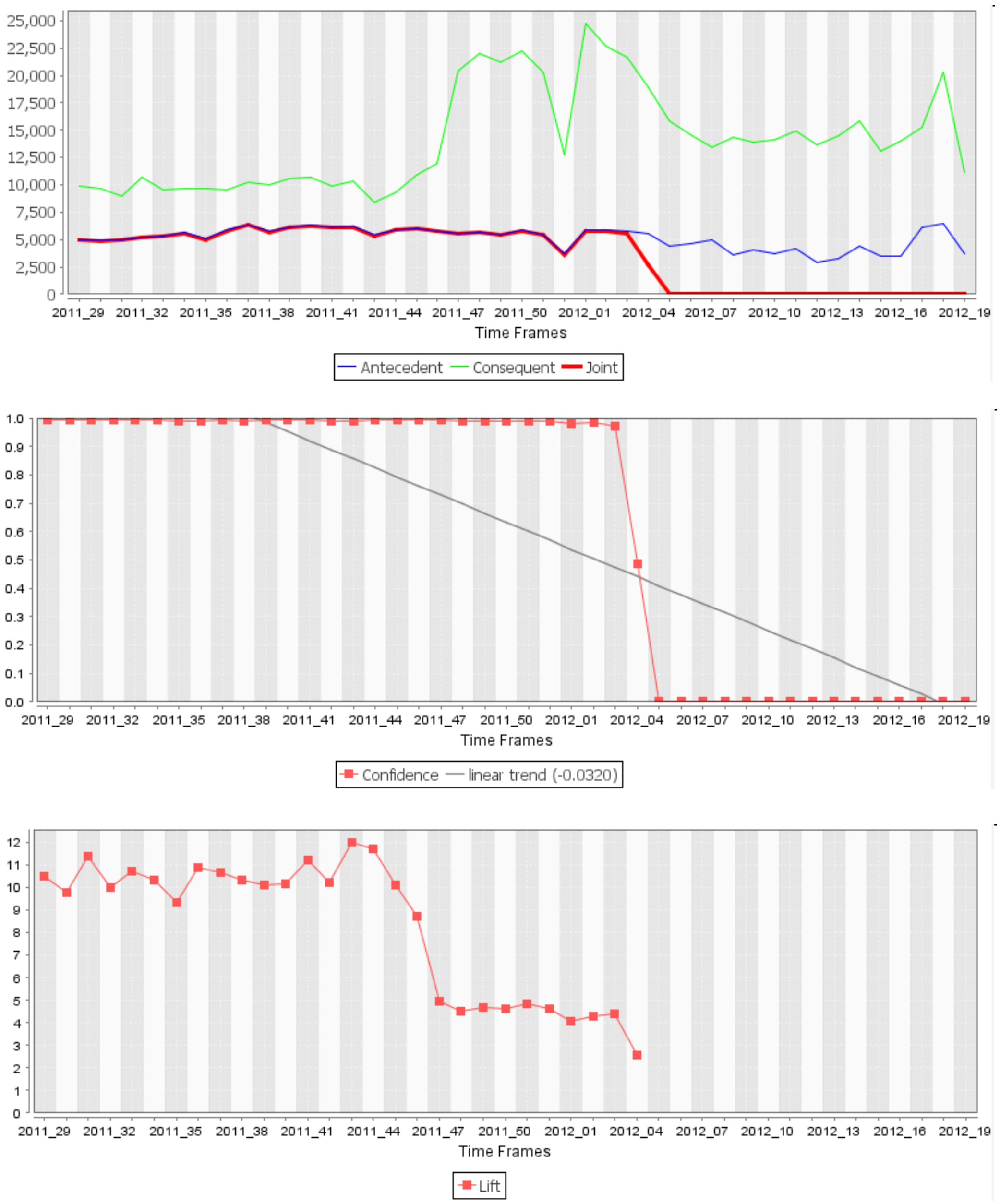


Figure 6.20: *Time series for rule:* INITIAL_JL = JL2 ∧ END_USER_PRODUCT = P2 → FINAL_JL = JL2. *The lift cannot be computed as the support dropped to zero at CW 5/2012.*

recently removed from the data set as it did not deliver reliable semantics. My proposed analysis approaches could have accelerated this detection. In addition to that, the peculiar patterns that occur around the turn of year are related to changes in business processes. The patterns detected by my proposed filtering technique could have earlier led to the insight that these changes need to be complemented by respective adjustments in the data acquisition and analysis processes.

7 Conclusions and Outlook

This thesis represents my attempt to simplify a particular data analysis task and reduce the effort to arrive at meaningful results. Section 7.1 summarizes the previous chapters, complements the announcements of the introductory chapter and highlights my contributions before I close my elaborations with prospects for further development in Section 7.2.

7.1 Summary and Contributions

The introductory Chapter 1 concretized the data mining setting in which I embedded by proposed analysis and visualization techniques. I repeat the quote from page 7 here to help summarize the accomplished tasks:

> This thesis dealt with the ② *identification* and ③ *visualization* of ① *patterns* that exhibit a certain ④ *temporal behavior* that is considered ⑤ *interesting to the user.*

① Pattern Types

The main pattern type was chosen to be association rules for reasons described in Section 1.3. I later transferred the proposed concepts to other models such as decision trees, Bayesian networks and cooccurrence graphs (see Section 4.5). The applicability was illustrated for cooccurrence graphs in the evaluation in Chapter 6 (Section 6.3). All model types were theoretically covered in Chapter 2 and Section 4.5.

② Identification

The notion of identification was assigned two aspects: First, to provide a graphical user interface (GUI) that enables the user to intuitively interact with the data and patterns and thus to come up with hypotheses that can yield new insights. Such a software was implemented and introduced in Chapter 5. The presented Information Miner 2.0 not only was used to scientifically validate the concepts. Parts of it were also used in real-world industrial settings. Results of the latter were presented in Chapter 6. The second aspect of identification as I see it, is the intuitive description of what is considered interesting to the user. I suggested the linguistic filtering approach using fuzzy concepts in Chapter 4. The theoretical ideas were both successfully implemented (Chapter 5) and validated (Chapter 6).

③ Visualization

Chapter 3 presented glyph visualizations that are capable of encoding high-dimensional nominal and continuous data with little redundancy. These glyphs are used to visually represent association rules (and also—with some restrictions—edges in cooccurrence graphs) and thus provide a means of displaying entire rule sets to the user. The glyph visualization was implemented and tested in the above-mentioned Information Miner 2.0 software stack (Chapter 5).

④ Temporal Behavior and ⑤ User Interestingness

Following the motivation from the introductory chapter, I proposed an approach in Chapter 4 that allows to specify fuzzy descriptions that are matched against the time series of the rules' properties in order to filter out the most relevant rules. The user specifies fuzzy descriptions that are defined over the domains of interest, here: the domain of change rates of selected rule evaluation measures. By changing the fuzzy partitions over these domains the user is able to relax or tighten the strictness of his concepts. Since rules can have a membership degree between zero and one to every concept, the resulting rule set (rules that have a positive membership degree) can be ordered to focus on rules that match best the user concept. The software platform (Chapter 5) was used to evaluate and validate (Chapter 6) these proposals.

7.2 Future Work

This last section collects ideas that might extent the contributions of this thesis. I also address shortcomings that might limit a data analysis.

7.2.1 Time Frame Discretization

In this thesis I used implicit time stamps from the underlying database management system that were discretized into equidistant time frames based on the user's choice (week-wise or month-wise). Week-wise frames can lead to quite a large number of frames to animate and analyze which might become tedious to look at. Further, global trends might lose their expressiveness over such a large number of frames. In contrast to this, too few time frames might not reveal subtle changes in the patterns trajectories. Therefore, it can be beneficial to allow for time frames of different widths. In [STEINBRECHER and KRUSE 2009a] I sketched a way how that could look like in the realm of cooccurrence graphs: The data set was initially subdidived into equidistant subsets of short width. Then, consecutive frames were merged if certain structural criteria were met. Some of the frames were merged, some remained unchanged, thus introducing differently sized frames that helped to reduce the analysis complexity. The above-mentioned criteria need to be rephrased in the area of association rules in order to come up with similar complexity reductions.

7.2.2 Seasonal Aspects

When it comes to time series analysis, seasonal changes require special attention. I did not take them into account here for mainly one reason: Seasonal aspects need to be treated with respect to a variable (such as a purchase amount, order numbers and the like). However, when inducing general association rules (that is, without restricting the consequent to a single (class) attribute), we would need to treat potential seasonal aspects for each consequent attribute separately. In no case I had enough information about such effects for each attribute to be able to cater for seasonal aspects. However, for analyses where a single class attribute is fixed for the consequent (see e. g. Section 6.2) such investigations might increase the explanatory power of the entire analysis.

7.2.3 Address Odd Trajectories

We have seen in Example 3 (Section 6.2.3) that oddly shaped trajectories such as depicted in the right chart of Figure 6.8 can be misleading in global trend filtering. Two options arise: If such rapidly changing trajectories are of interest, they must be made available within the filtering framework. Regression methods will then be too limited, calling for other assessments that better cover the shape of the trajectories. Or, if such oddly shaped trajectories are unwanted, they must be filtered out such that they do not match against trend filters as in the above example.

7.2.4 Explain Similar Rules

The evaluation of Section 6.4 revealed that in some circumstances there might be groups of similar rules exhibiting a similar temporal behavior. Similarity with respect to rules means that they share all but one antecedent or consequent attribute/item. In such cases it might be very helpful to replace such a group by just one visual placeholder or to describe the overall behavior of all these rules in a linguistic manner. This would actually call for the opposite procedure: Derive linguistic (human-readable textual) descriptions from given time series. Such techniques are studied (e. g. [KACPRZYK and WILBIK 2010]) and could be used to infer fuzzy concepts that then match against entire sets of rules (namely the above-mentioned groups that share items and behavior). Another way to deal with similar rules is to find condensed representations of the item sets they are extracted from [BÖTTCHER et al. 2005, 2009]. A combined approach of filtering a rule set that was created from a previously reduces set of item sets might prove to be an interesting and promising undertaking.

A
Related Work

This appendix summarizes and sketches works from other authors referred to in the visualization and linguistic filtering chapter. Figures and verbatim quotations are taken from the respective reference specified in the section title.

A.1 Rule Visualization

A.1.1 Visualizing Association Rules for Text Mining [WONG, WHITNEY and THOMAS 1999]

The paper proposes an visualization that displays an item-to-rule relation which is encoded in a two-dimensional matrix. Each column represents a rule whereas each row stands for a single item. If an item is present in the antecedent or consequent of a rule, the respective matrix entry is colored (with different colors for antecedent and consequent, of course). The matrix is actually drawn in three-dimensional space in order to place two rows of bar chart columns next to it, representing the support and confidence values of the respective rule. Figure A.1 shows a set of 45 rules that are described on nine items.

The authors used their method to visualize text documents. The item set consisted of keywords extracted from a corpus and the rules marked the relation between these words. Clearly, the system is feasible only for a small number of items and rules. Dealing with hundreds of rules and

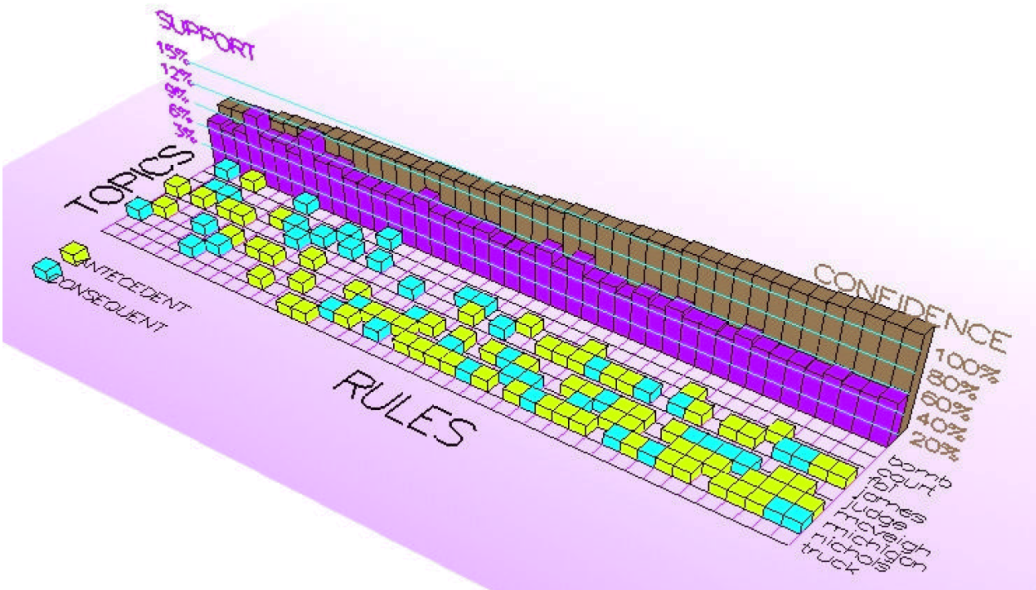


Figure A.1: *A set of 45 association rules together with their support and confidence values. The colors have been inverted in order to increase readability.*

items would result in a huge and sparse matrix that would be inconceivable for a human interaction. Depending on the angle of view, it can be hard to observe all association rule measures assigned to the bar charts.

Comments

One objective of my visualization method was to deliver a representation that contains as little arbitrariness as possible. In Figure A.1, the items are ordered alphabetically and the rules are sorted ascendingly by confidence. The latter choice proves to be arbitrary when one is tempted to encode temporal behavior, that is the change in support or confidence. The bar charts are likely to change and the order of the rules becomes random. In addition, the third dimension is merely used as a workaround to possible overlapping in the bar charts and does not carry any additional information about the rules.

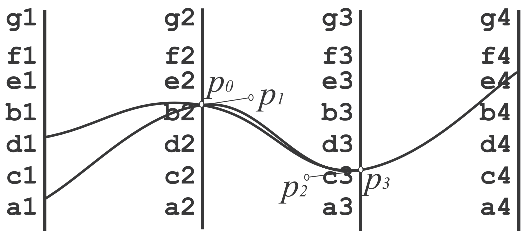


Figure A.2: *Visualization of the two association rules $db \rightarrow ce$ and $ab \rightarrow ce$ using Bézier curves to distinguish both lines. The p_i denote the Bézier control points [*BARTELS *et al. 1998] that are used to create non-overlapping curves.*

A.1.2 Visual Exploration of Frequent Itemsets and Association Rules [YANG 2005, 2008]

The authors use parallel coordinates [INSELBERG 1985, 2009] to visualize frequent item sets and association rules. Every coordinate (that is, every vertical axis) contains all the items that are referred to by the rules. The items contained in the antecedent and consequent are connected by a Bézier curve in order to be able to distinguish rules with common items and thus overlapping straight lines. Figure A.2 shows an example of two rules. If a taxonomy (that is, a hierarchical structure) of the item set is available, it can be used to replace the parallel coordinate axes as shown in Figure A.3.

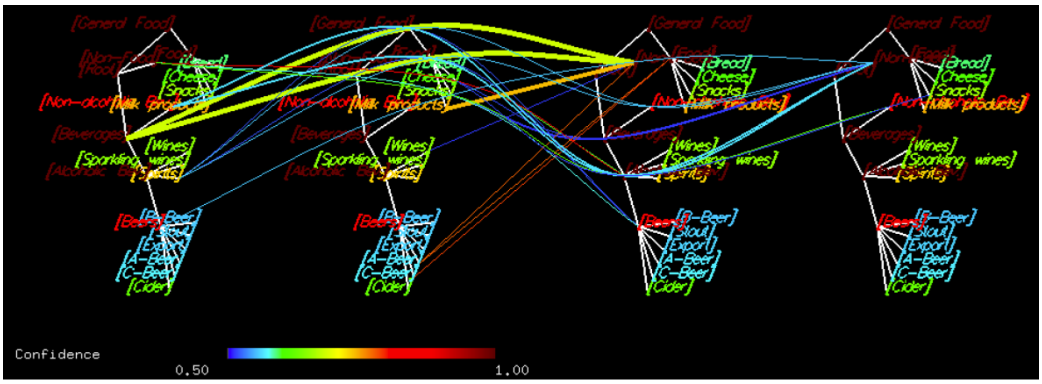


Figure A.3: *Associations rules drawn on the item categories* Beers *and* Foods.

Comments

I rejected to follow an alike approach for association rule visualization. First and foremost, none of the evaluation measures are represented spatially[1] which in turn disqualifies the approach for a further extension towards temporal change. Second, the order of the parallel axes is arbitrary and thus complicates the visual assessment. Lastly, the visualization is not capable of displaying a large number of rules properly as the overlap—even though it might be reduced by using appropriate Bézier curves—renders this task hard to impossible.

A.1.3 Visualizing Association Rules with Interactive Mosaic Plots [HOFMANN, SIEBES and WILHELM 2000]

The paper motivates mosaic plots [HARTIGAN and KLEINER 1981] as an intuitive visualization for association rules. Figure A.4 shows an example of 8 association rules (actually, 16, if we also count the rules with negated conclusion). A set of items is chosen to be the pool for antecedent items. From this pool, all item subsets with nonzero support are created. In the example, the antecedent pool is $\{\mathsf{heineken}, \mathsf{coke}, \mathsf{chicken}\}$. All 8 possible subsets are created and depicted by a visual indicator formed by stacked bars at the bottom of Figure A.4. Black represents a present item, which a missing one. Each antecedent is assigned a vertical bar with the width of it representing the relative antecedent support. For each antecedent, an association rule is formed that has an additional previously-chosen item (here: sardines) as its consequent. The confidence of this rule is depicted by filling the bar up to the respective percentage level. Figure A.4 depicts one rule with a confidence of almost 100%: $\{\mathsf{heineken}, \mathsf{coke}, \mathsf{chicken}\} \to \mathsf{sardines}$.

Comments

The visualization by mosaic plots (or more precise: double decker plots, to stress the antecedent encoding) resembles the visualization of conditional probabilities used in [STEINBRECHER 2006, STOBER et al. 2009]. It is an appealing approach for assessing single or few association rules. In-

[1] One measure is encoded by color, though.

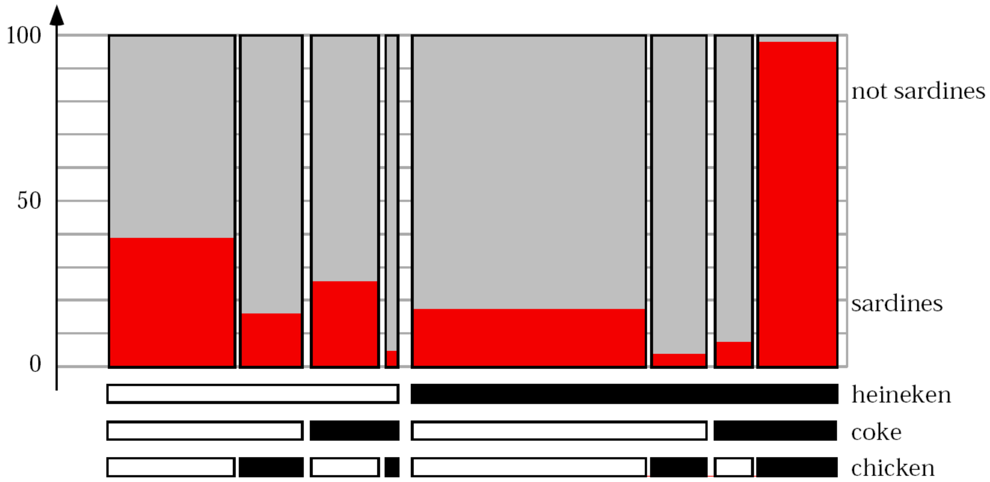


Figure A.4: *Eight vertical bars each representing an association rule with the antecedents set to all subsets of* {heineken, coke, chicken} *and the consequent being* sardines. *The bar width encodes the antecedent support while the filled area in red denotes the confidence of the respective rule. The rightmost rule clearly stands out by its extremely high confidence.*

deed, the authors use the plots to check which antecedent items have the highest impact on the confidence and thus apply their visualization on only a few rules simultaneously. Therefore, I did not consider an alike approach for visualizing large sets of association rules.

A.1.4 Visual post-analysis of association rules [Bruzzese and Davino 2001, 2008]

The authors use *multiple correspondence analysis* [Roux and Rouanet 2004] to visualize the dependences encoded between the items of the rules' antecedents and consequents. Given the rule set $\{\rho_1, \dots, \rho_n\}$ and the underlying item set $\{a_1, \dots, a_N\}$ the starting point is an $(n \times N + 2)$-

matrix that contains for every rule and item a binary value whether the item is contained in that particular rule:[2]

$$
\begin{array}{c}
\\ \rho_1 \\ \vdots \\ \rho_j \\ \vdots \\ \rho_n
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} a_1 & \cdots & a_i & \cdots & a_N & s & c \end{array} \\
\begin{pmatrix}
[\![a_1 \in \rho_1]\!] & \cdots & [\![a_i \in \rho_1]\!] & \cdots & [\![a_N \in \rho_1]\!] & \mathrm{supp}(\rho_1) & \mathrm{conf}(\rho_1) \\
\vdots & & \vdots & & \vdots & \vdots & \vdots \\
[\![a_1 \in \rho_j]\!] & \cdots & [\![a_i \in \rho_j]\!] & \cdots & [\![a_N \in \rho_j]\!] & \mathrm{supp}(\rho_j) & \mathrm{conf}(\rho_j) \\
\vdots & & \vdots & & \vdots & \vdots & \vdots \\
[\![a_1 \in \rho_n]\!] & \cdots & [\![a_i \in \rho_n]\!] & \cdots & [\![a_N \in \rho_n]\!] & \mathrm{supp}(\rho_n) & \mathrm{conf}(\rho_n)
\end{pmatrix}
\end{array}
$$

Two columns are added to contain the supports and confidences of all the rules. The multiple correspondence analysis yields a dimension reduction which then allows for the visualization of the items or the rules or both. Figure A.5 shows both items and rules of an example data set from the well-known UCI Machine Learning Repository [WWW: UCI].[3] The circles represent items (with the support being proportional to the area) whereas gray boxes denote rules. The closer the item circles are, the larger is the number rules that contain those items in either antecedent or consequent. Hence, rules that are depicted in close proximity (are likely to) share more items that rules farther apart.

Comments

One of the requirements I stated in Chapter 3 was to ensure any dimension used for locating a graphical artifact (such as a circle for an item or box for a rule) has a strict meaning conveying an exact rule or item measure. Here instead, the high-dimensional matrix was flattened introducing axes that are formed of linear combinations of original axes. The x- and y-coordinate of an artifact alone does not convey much information, it only allows conclusions based on distances. I therefore do not consider approaches where any means of dimension reduction is applied.

[2] The expression $[\![\cdot]\!]$ represents the (truth) value of the specified statement.
[3] This data set contained 101 rows of 15 boolean attributes.

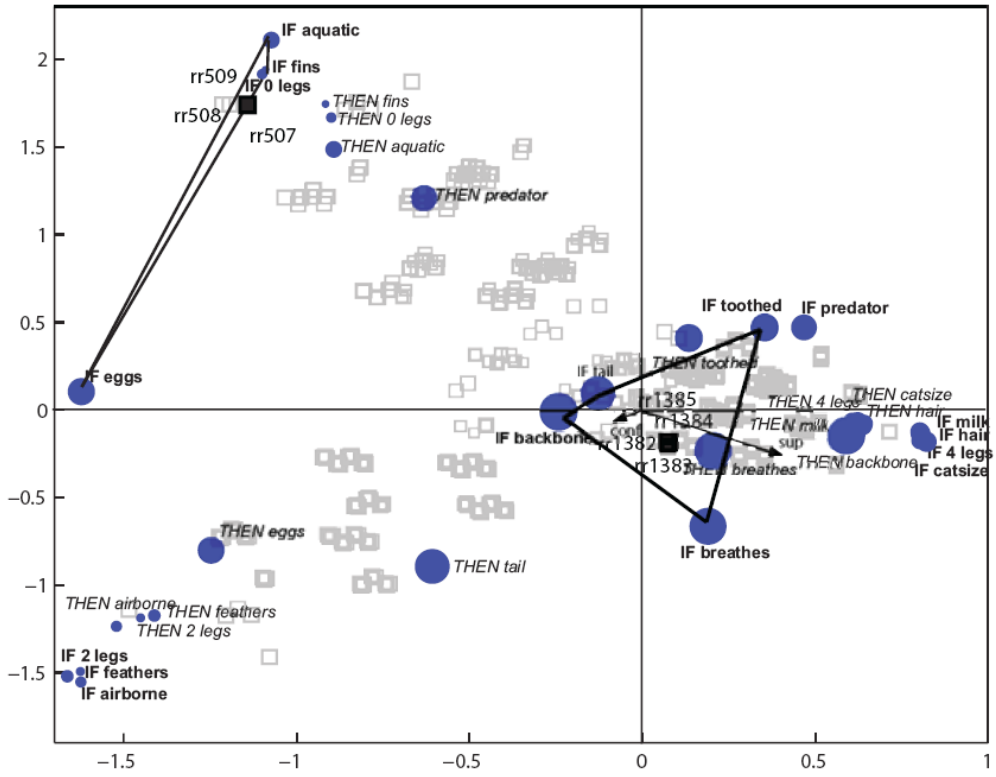


Figure A.5: *Circles represent items, boxes depict rules. The closer items are, the more rules contain these items and consequently the closer rules are, the more items they have in common.*

A.1.5 A 2D-3D Visualization Support for Human-centered Rule Mining [BLANCHARD, GUILLET and BRIAND 2003]

The authors introduce a three-dimensional glyph-based visualization for association rules. The underlying idea resembles the widely known Chernoff faces [CHERNOFF 1973] where dimensions and proportions of facial elements (such as eyes size, nose size, eyes distance, etc.) are used to represent a high-dimensional data set in two-dimensional space.

Figure A.6 (left) shows such a glyph consisting of a sphere on top of a cone. Several glyph properties (such as sphere size and color, cone size and color, cone apex angle) can be assigned user-specified rule evaluation measures. The glyphs of an entire rule set are laid out in the shape of an arena as can be seen in the main part of Figure A.6.

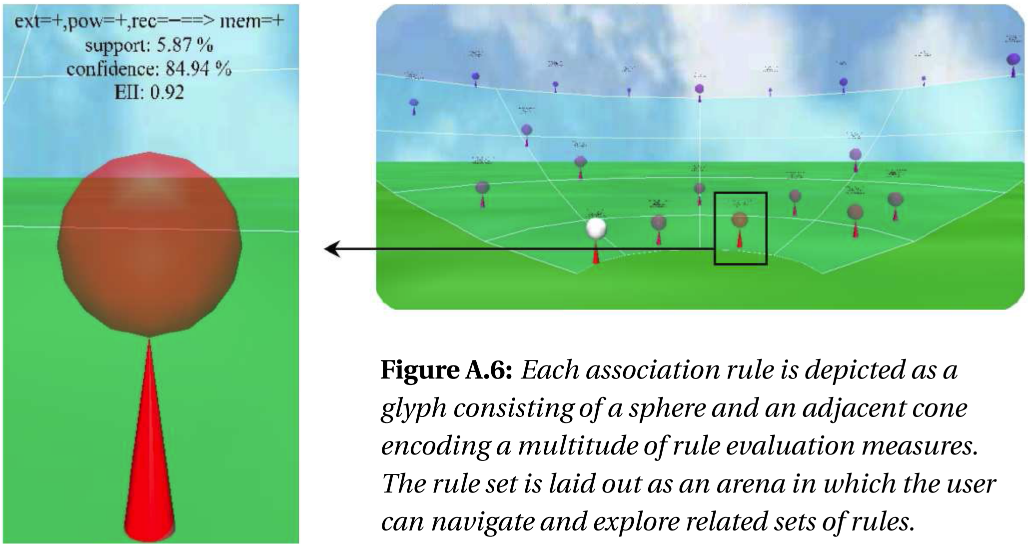


Figure A.6: *Each association rule is depicted as a glyph consisting of a sphere and an adjacent cone encoding a multitude of rule evaluation measures. The rule set is laid out as an arena in which the user can navigate and explore related sets of rules.*

Comments

I do not consider three-dimensional visualizations since the approach suggested in this thesis already adds one dimension in terms of time. The user would have to navigate through a changing three-dimensional space which I reject in order not to overstrain the user. The idea, however, to use glyphs exposing several properties that get assigned evaluation measures influenced the proposed methods of this thesis.

A.2 Temporal Aspects of Rules

There are multiple approaches to take the dimension *time* into account when inducing or post-processing frequent item sets or association rules. This section sketches selected approaches that use different ideas to represent temporal aspects.

A.2.1 An approach to discovering temporal association rules [Ale and Rossi 2000]

This paper merely conveys an elaborated idea for temporal rule induction rather than a sound and evaluated framework. The authors use the normal notion of frequent items and assign to each item an interval in which it is valid. This is done by assigning to each transaction in the database a time stamp. The validity interval of an item is then defined as the minimum and maximum of the union of all timestamps of transactions that contain the respective item.

A temporal rule is algebraically of the same type as an ordinary association rule, just with a validity interval assigned with respect to which the well-known rule evaluation measures are calculated.

Comments

As we are dealing with constant (or user-specified) time frame widths, it is not necessary to come up with an adapted Apriori algorithm that takes into account validity intervals that are connected with any item in the transaction database. In addition to that, we allow the user to run a rule induction algorithm at his choice, disburdening him further.

A.2.2 On Mining General Temporal Association Rules in a Publication Database [Lee, Lin and Chen 2001]

The authors address a problem that is a common issue when the transaction database is not fixed but rather grows constantly: recent item sets are less likely to exceed the minimum support than older ones, simply because they had not much time to occur over and over again. Even though the authors do not mention the notion *stream mining*, the problem falls exactly into this field.

The paper modifies the original association rule definition by assigning to it a so-called *maximum common exhibition period* which is basically an validity interval not unlike to that of [ALE and ROSSI 2000] sketched above: it is the smallest interval in which all items of the rule occur in the database. The authors then introduce a modified Apriori algorithm [AGRAWAL and SRIKANT 1994] that allows to directly generate rules of that type.

Comments

In line with the requirements specified in Chapter 4, I do not intend to change the algebraic definition of an association rule. Therefore, methods in that direction are not considered any further.

A.2.3 Mining Changes in Association Rules: A Fuzzy Approach [AU and CHAN 2005]

Au and Chan use a given set of association rules combined with collections of time series of rule evaluation measures for each rule to induce a new set of higher-order rules. These higher-order rules describe (temporal) relationships of the temporal behavior of some of the evaluation measures. The output of the algorithm might be a rule like the following:

> **If** the *change in support* in time frame i is *fairly decreasing*,
> **then** the *change in support* in time frame $i+1$ is *highly decreasing*.

The time series of support and confidence (other measures are not considered) are split into several equidistant time frames. For each time frame, the increase or decrease of both measures is assigned a fuzzy membership degree thus forming a new sequence of temporally ordered linguistic terms (the one belonging to the fuzzy set leading to the highest membership degree in the step before). To induce higher-order rules of the above-mentioned type, a fuzzy decision tree [JANIKOW 1998] is induced.

An example is shown in Figure A.7. A path from the root node to a child node can then be interpreted as a higher-order rule describing the temporal behavior of the original rule set.

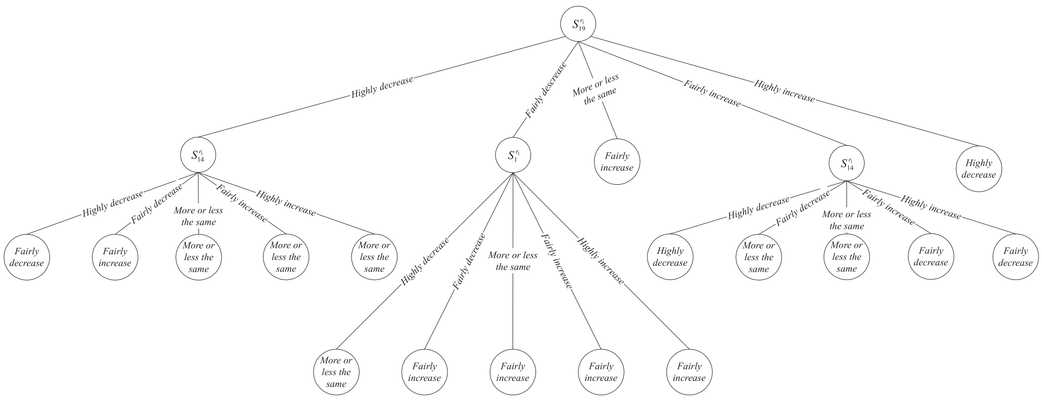


Figure A.7: *Fuzzy decision tree induced from the temporal behavior of the support of a single rule. Starting at the root node and continuing to a leaf, one can extract a rule that describes a part of the support time series of the underlying association rule.*

Comments

The underlying principle to assess (parts of) time series with fuzzy sets is the key concept that also underlies the framework presented in Chapter 4. However, I pursue a different goal: to simplify the set of association rules rather than introducing new sets of (fuzzy) rules. Further, inducing a set of rules from the evaluation measure time series of a single rule provides only weak means of generalization (if one is not willing to run another rule induction algorithm on the fuzzy rules expecting general patterns among the fuzzy rules).

B

Charts and Screenshots

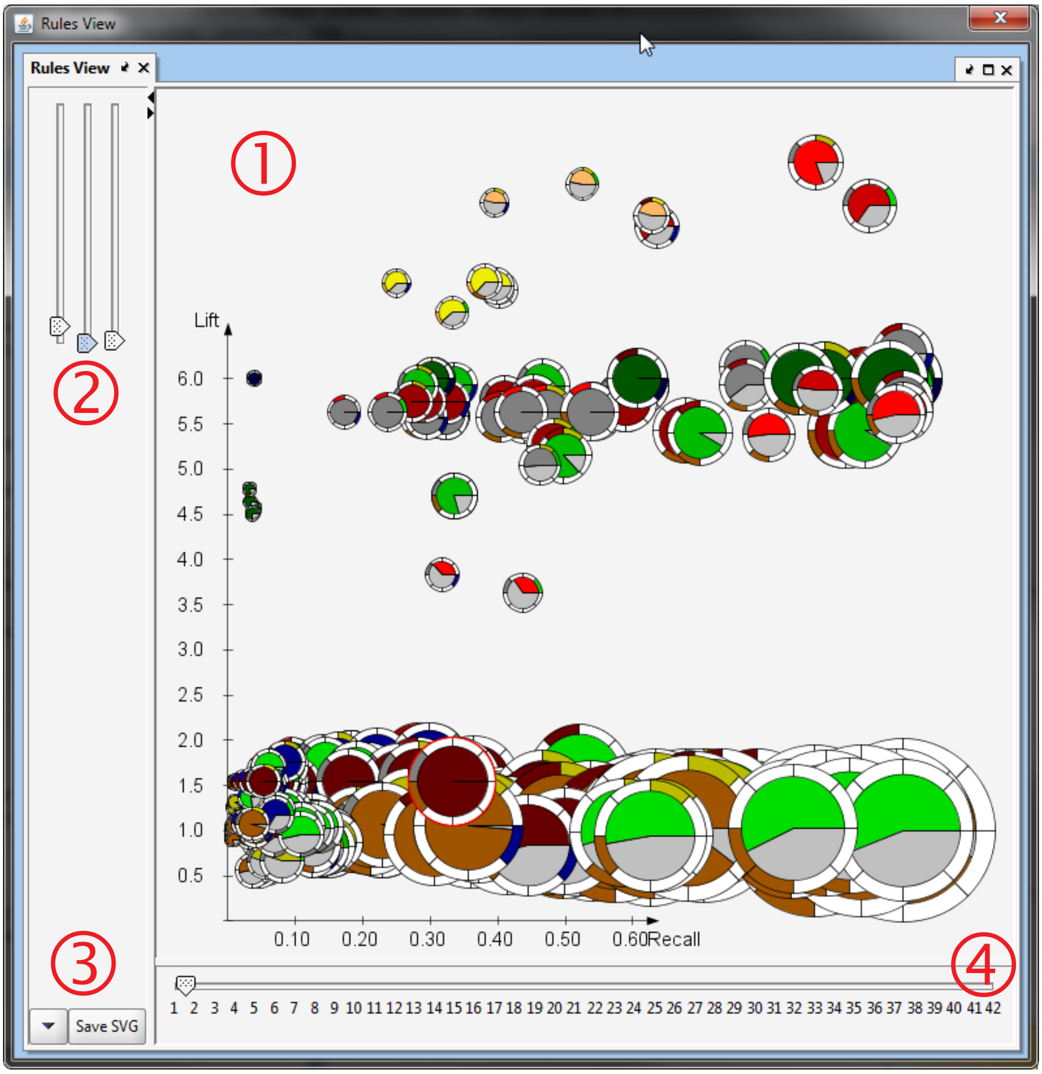


Figure B.1: *Rules view. ①: main part depicting the glyphs of all rules, ②: sliders to (re)scale the axes, ③: drop-down menu for evaluation measure selection (see Figure B.2), ④: time slider to flick through all time frames.*

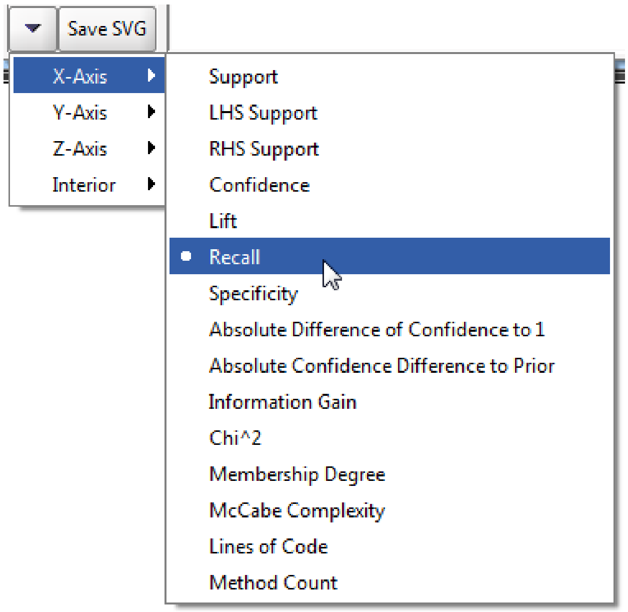


Figure B.2: *Set of evaluation measures that can be chosen to determine the location and size of each rule glyph in the rules view.*

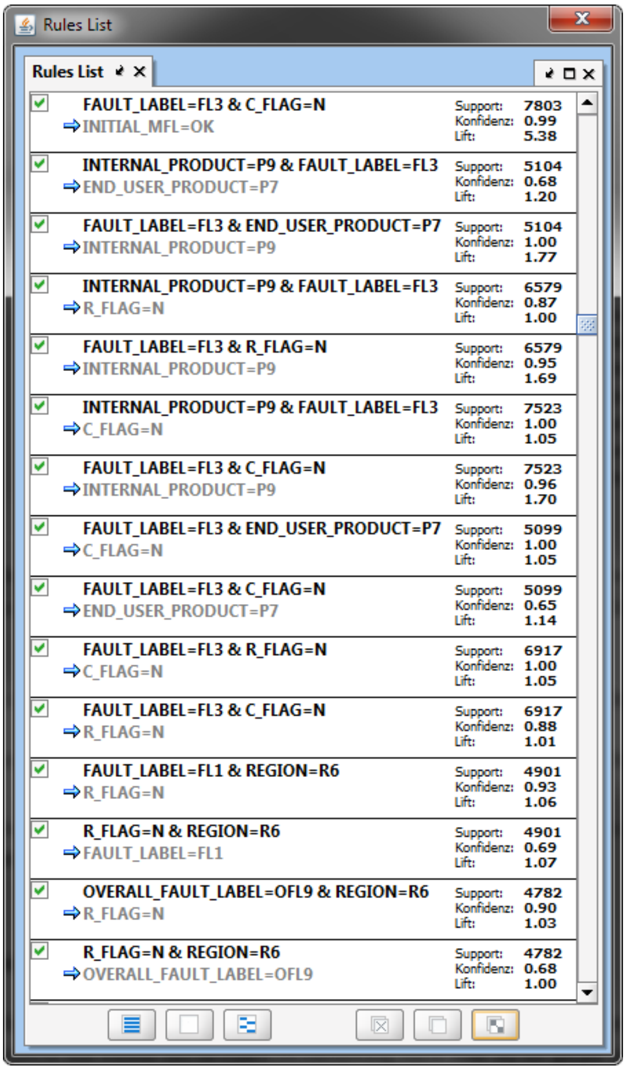


Figure B.3: *Rule list view. Contains all rules as a list. Each entry encodes the textual rule representation as well as visibility indicators and the support, confidence and lift of the first time frame. The controls at the bottom can be used to (de)select subsets of rules or to finetune filtering results.*

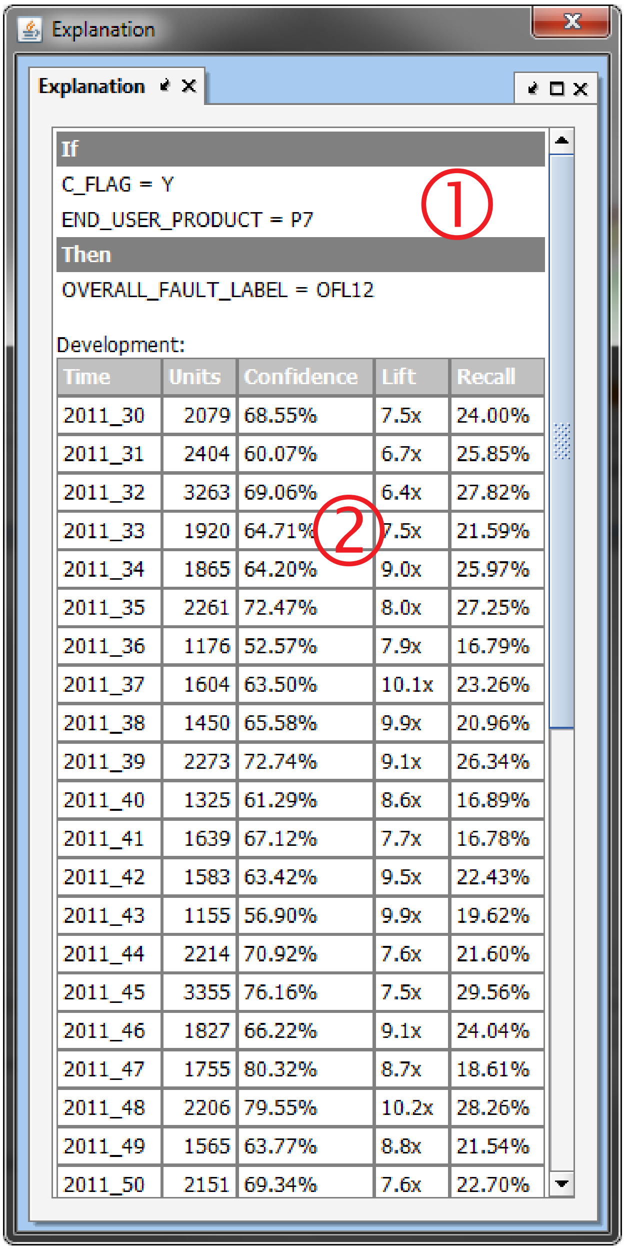


Figure B.4: *Rule details view. ①: textual rule representation, ②: support, confidence, lift and recall time series of the selected rule.*

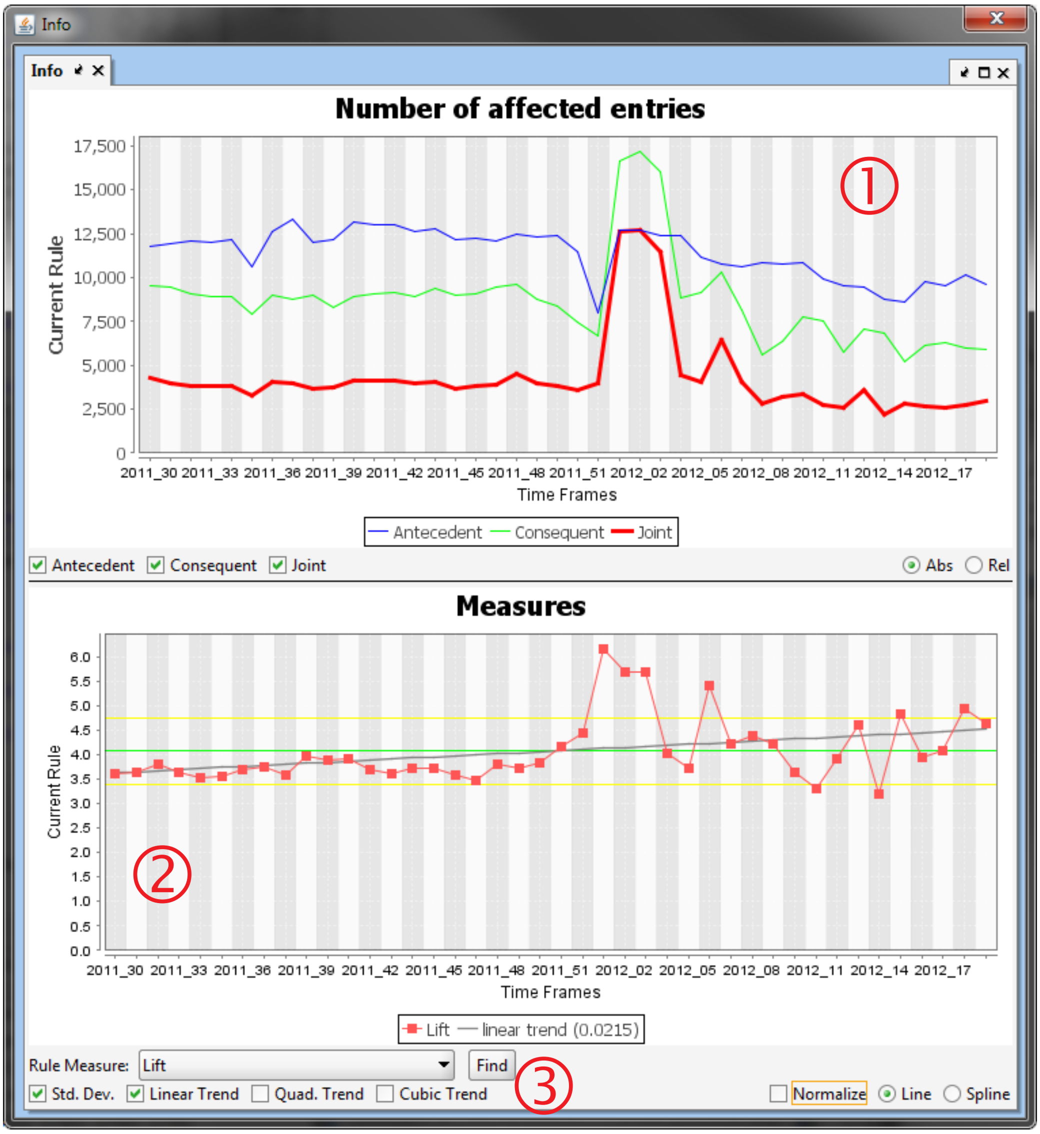


Figure B.5: *Rule Time Series View. ①: time series of antecedent support, consequent support and joint support of the selected rule, ②: time series and trend lines of a user-selectable evaluation measure, ③: list of available evaluation measures and trend line fittings.*

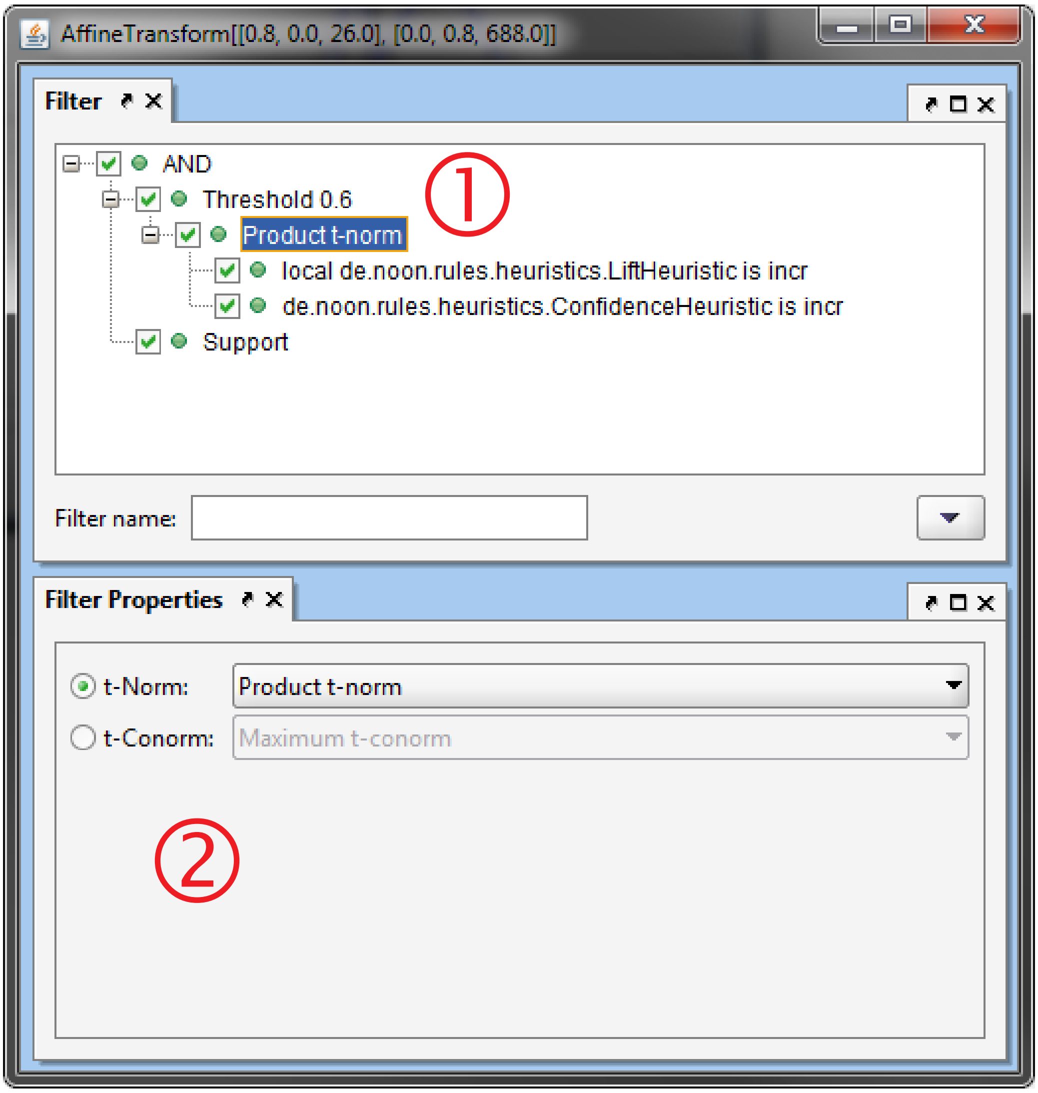


Figure B.6: *Filter concept editor. ①: expression tree of the linguistic concept, ②: properties of the selected nodes of the linguistic concept.*

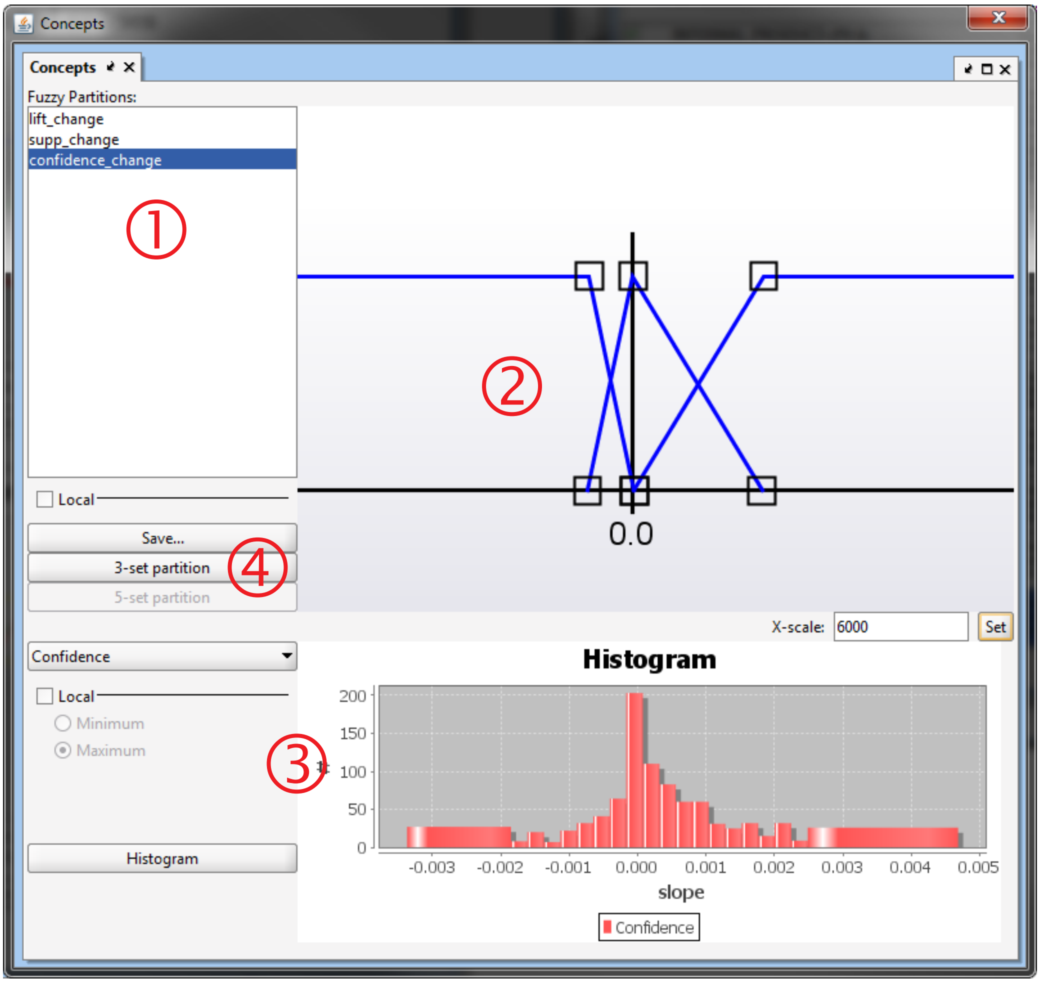


Figure B.7: *Fuzzy partition editor.①: list of available fuzzy partitions, ②: graphical editor for the sets of a fuzzy partition, ③: histogram display for change rates, ④: controls for fuzzy partition induction*

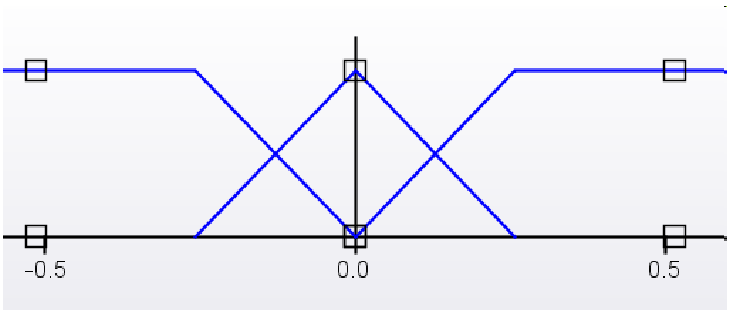


Figure B.8: *Induced fuzzy partition for the confidence change rate domain with respect to maximal local changes.*

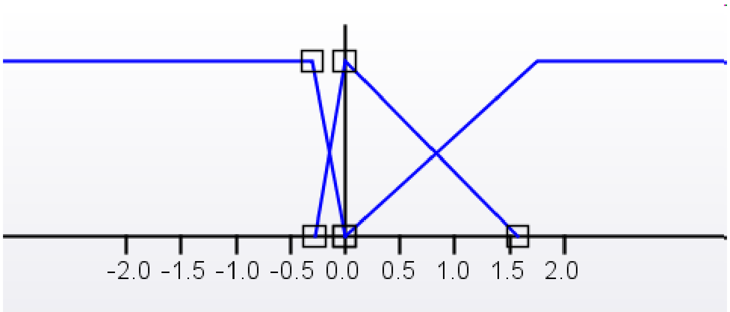


Figure B.9: *Induced fuzzy partition for the lift change rate domain with respect to global changes. The histogram of all 904 lift slopes is shown in Figure B.10 which explains the asymmetric shape.*

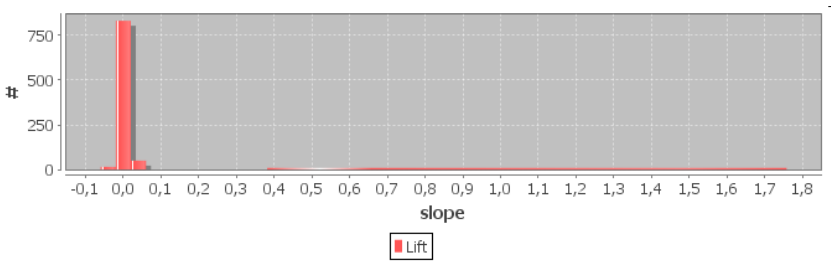


Figure B.10: *Histogram of global lift change rates, that is, the histogram of all 904 slopes of the linear trend in each rule's lift time series.*

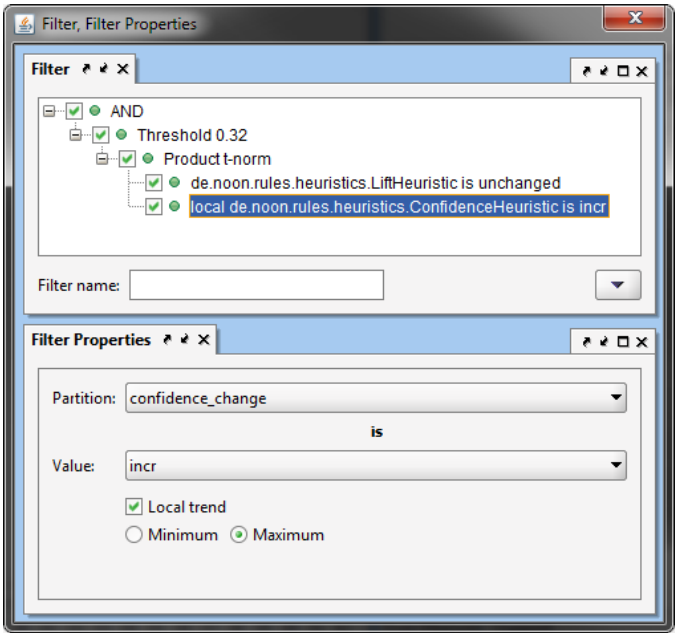


Figure B.11: *Linguistic concept "confidence is locally increasing and lift is globally unchanged".*

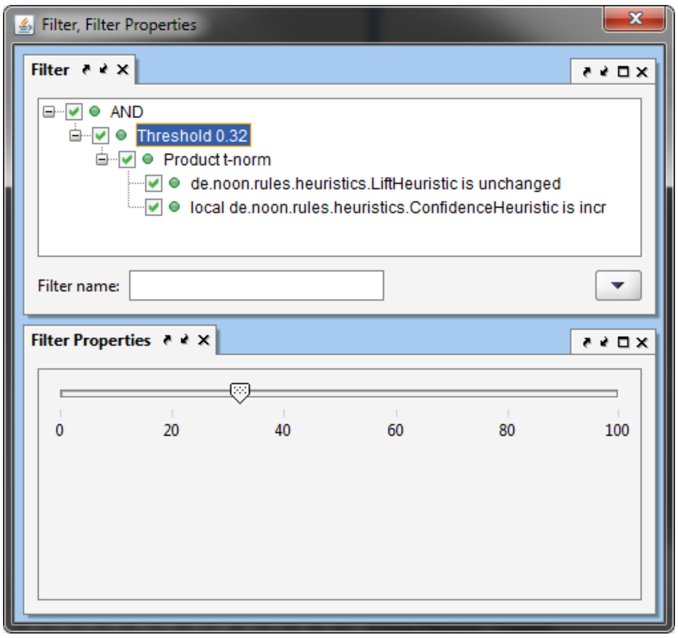


Figure B.12: *Selecting the threshold node allows to adjust the minimum membership degree above which the rule glyphs are drawn, thus filtering the rule set by hiding all the other glyphs.*

C Bibliography

[AGRAWAL et al. 1993] R. Agrawal, T. Imielinski and A. N. Swami. **Mining Association Rules between Sets of Items in Large Databases**. In: P. Buneman and S. Jajodia, Eds., Proc. ACM SIGMOD Int. Conf. on Management of Data, Washington, D.C., May 26-28, 1993, pp. 207–216. 1993, ACM Press.

[AGRAWAL and SRIKANT 1994] R. Agrawal and R. Srikant. **Fast Algorithms for Mining Association Rules**. In: J. B. Bocca, M. Jarke and C. Zaniolo, Eds., Proc. 20th Int. Conf. Very Large Data Bases, VLDB, pp. 487–499. 1994, Morgan Kaufmann.

[ALE and ROSSI 2000] J. M. Ale and G. H. Rossi. **An approach to discovering temporal association rules**. In: SAC '00: Proceedings of the 2000 ACM symposium on Applied computing, pp. 294–300. 2000, ACM, New York, NY, USA.

[ALLEN 1983] J. F. Allen. **Maintaining knowledge about temporal intervals**. Commun. ACM, Vol. 26(11):832–843, 1983.

[ARANDA and XAMBÓ 2010] J. Aranda and S. Xambó, Eds. **WCCI 2010 Plenary and Invited Lectures**. Institute of Electrical and Electronics Engineering, Inc., 2010.

[AU and CHAN 2005] W.-H. Au and K. C. C. Chan. **Mining Changes in Association Rules: a Fuzzy Approach**. Fuzzy Sets and Systems, Vol. 149(1):87–104, 2005.

[BARTELS et al. 1998] R. H. Bartels, J. C. Beatty and B. A. Barsky. An Introduction to Splines for Use in Computer Graphics and Geometric Modelling, Chapter **Bézier Curves**, pp. 211–245. 1998. Morgan Kaufmann, San Francisco, CA, USA.

[BAUMANN et al. 2009] S. Baumann, J. J. Burred, A. Nürnberger and S. Stober, Eds. **Proceedings of the 3rd Workshop on Learning the Semantics of Audio Signals (LSAS)**. Graz, Austria, 2009.

[BEER et al. 2010] M. Beer, R. L. Muhanna and R. L. Mullen, Eds. **Proceedings of the 4th International Workshop on Reliable Engineering Computing (REC 2010)**. Professional Activities Centre, National University of Singapore, Research Publishing Services, Singapore, 2010.

[BERTHOLD et al. 2010] M. R. Berthold, C. Borgelt, F. Höppner and F. Klawonn. **Guide to Intelligent Data Analysis**, Vol. 42 of Texts in Computer Science. Springer-Verlag, Berlin, Germany, 1st Edn., 2010.

[BLANCHARD et al. 2003] J. Blanchard, F. Guillet and H. Briand. **A User-driven and Quality-oriented Visualization for Mining Association Rules**. In: Proceedings of the Third IEEE International Conference on Data Mining, ICDM '03, pp. 493–496. 2003, IEEE Computer Society, Washington, DC, USA.

[BORGELT 2005] C. Borgelt. **Keeping Things Simple: Finding Frequent Item Sets by Recursive Elimination**. In: Workshop Open Source Data Mining Software (OSDM'05, Chicago, IL), pp. 66–70. 2005, ACM Press, New York, NY, USA.

[BORGELT et al. 2009] C. Borgelt, M. Steinbrecher and R. Kruse. **Graphical Models — Representations for Learning, Reasoning and Data Mining**. John Wiley & Sons, United Kingdom, 2nd Edn., 2009.

[BÖTTCHER et al. 2009] M. Böttcher, M. Spott and R. Kruse. **A Condensed Representation of Itemsets for Analyzing their Evolution over Time**. In: 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD2009), Lecture Notes in Artificial Intelligence (LNAI). 2009, Springer.

[BÖTTCHER et al. 2005] M. Böttcher, M. Spott and D. Nauck. **Detecting Temporally Redundant Association Rules**. Fourth International Conference on Machine Learning and Applications (ICMLA '05), pp. 397–403, 2005.

[BREIMAN et al. 1984] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone. **Classification and Regression Trees**. The Wadsworth & Brooks. Wadsworth, 1984.

[BRUZZESE and DAVINO 2001] D. Bruzzese and C. Davino. **Statistical Pruning of Discovered Association Rules**. Computational Statistics, Vol. 16:387–398, 2001.

[BRUZZESE and DAVINO 2008] D. Bruzzese and C. Davino. **Visual Mining of Association Rules**. In: [SIMOFF et al. 2008], pp. 103–122.

[BURDICK et al. 2003] D. Burdick, M. Calimlim, J. Flannick, J. Gehrke and T. Yiu. **MAFIA: A Performance Study of Mining Maximal Frequent Itemsets**. 2003, In: Proc. Workshop on Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL) CEUR Workshop Proceedings 90, TU Aachen, Germany.

[CHAPMAN et al. 2000] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer and R. Wirth. **CRISP-DM 1.0 Step-by-step data mining guide**. Tech. rep., The CRISP-DM consortium, 2000.

[CHEN et al. 2008] C.-h. Chen, W. Härdle and A. Unwin, Eds. **Handbook of Data Visualization**. Springer Handbooks of Computational Statistics. Springer Berlin/Heidelberg, 2008.

[CHERNOFF 1973] H. Chernoff. **The Use of Faces to Represent Points in K-Dimensional Space Graphically**. Journal of the American Statistical Association, Vol. 68:361–368, 1973.

[COOK et al. 2009] D. J. Cook, J. C. Augusto and V. R. Jakkula. **Ambient intelligence: Technologies, applications, and opportunities**. Pervasive and Mobile Computing, Vol. 5(4):277 – 298, 2009.

[DARWIN 1859] C. Darwin. **On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life**. John Murray, London, United Kingdom, 1859.

[DURAND-GUERRIER 2003] V. Durand-Guerrier. **Which notion of implication is the right one? From logical considerations to a didactic perspective**. Educational Studies in Mathematics, Vol. 53(1):5–34, 2003.

[EDDY 1981] W. F. Eddy, Ed. **Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface**. Springer-Verlag, New York, 1981.

[FAYYAD et al. 1996] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy. **Advances in Knowledge Discovery and Data Mining**. AAAI Press and MIT Press, Menlo Park and Cambridge, MA, USA, 1996.

[FECHNER 1860] G.-T. Fechner. **Elemente der Psychophysik**. Breitkopf und Härtel, Leipzig, 1860.

[FINK et al. 2009] A. Fink, B. Lausen, W. Seidel and A. Ultsch, Eds. **Advances in Data Analysis, Data Handling and Business Intelligence, Proceedings of the 32nd Annual Conference of the Gesellschaft für Klassifikation e. V., Joint Conference with the British Classification Society (BCS) and the Dutch/Flemish Classification Society (VOC), Helmut-Schmidt University, Hamburg, July 16-18, 2008**, Studies in Classification, Data Analysis, and Knowledge Organization. Springer Verlag, 2009.

[GEBHARDT et al. 2003] J. Gebhardt, H. Detmer and A. L. Madsen. **Predicting Parts Demand in the Automotive Industry — An Application of Probabilistic Graphical Models**. 2003, In: Proc. International Joint Conference on Uncertainty in Artificial Intelligence (UAI 2003), Bayesian Modelling Applications Workshop, Acapulco, Mexico, 4.-7. August 2003.

[GOLDBERG 1989] D. E. Goldberg. **Genetic Algorithms in Search, Optimization and Machine Learning**. Addison-Wesley Pub. Co., 1989.

[HAN et al. 2000] J. Han, J. Pei and Y. Yin. **Mining Frequent Patterns without Candidate Generation**. In: W. Chen, J. Naughton and P. A. Bernstein, Eds., 2000 ACM SIGMOD Intl. Conference on Management of Data, pp. 1–12. 2000, ACM Press.

[HARTIGAN and KLEINER 1981] J. Hartigan and B. Kleiner. **Mosaics for contingency tables**. In: [EDDY 1981], 1981, pp. 268–273.

[HOFMANN et al. 2000] H. Hofmann, A. P. J. M. Siebes and A. F. X. Wilhelm. **Visualizing association rules with interactive mosaic plots**. In: KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 227–235. 2000, ACM, New York, NY, USA.

[HOLLAND 1975] J. H. Holland. **Adaptation in Natural and Artificial Systems**. The MIT Press, 1975.

[INSELBERG 1985] A. Inselberg. **The plane with parallel coordinates**. The Visual Computer, Vol. 1:69–91, 1985.

[INSELBERG 2009] A. Inselberg. **Parallel Coordinates: Interactive Visualisation for High Dimensions**. In: [JAIN et al. 2009], pp. 1–30.

[JAIN et al. 2009] L. Jain, X. Wu, R. Liere, T. Adriaansen and E. Zudilova-Seinstra, Eds. **Trends in Interactive Visualization**. Advanced Information and Knowledge Processing. Springer London, 2009.

[JANIKOW 1998] C. Z. Janikow. **Fuzzy Decision Trees: Issues and Methods**. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 28(1):1–14, 1998.

[KACPRZYK and WILBIK 2010] J. Kacprzyk and A. Wilbik. **Temporal Linguistic Summaries of Time Series Using Fuzzy Logic**. 2010.

[KEIM et al. 2010] D. Keim, J. Kohlhammer, G. Ellis and F. Mansmann, Eds. **Mastering the Information Age: Solving Problems with Visual Analytics**. Eurographics Association, 2010.

[KOLMOGOROV 1933] A. N. Kolmogorov. **Grundbegriffe der Wahrscheinlichkeitsrechnung**. Springer, Berlin, 1933.

[KRUSE et al. 2011] R. Kruse, C. Borgelt, F. Klawonn, C. Moewes, G. Ruß and M. Steinbrecher. **Computational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze**. Computational Intelligence. Vieweg+Teubner, Wiesbaden, Germany, 2011.

[KRUSE and STEINBRECHER 2010] R. Kruse and M. Steinbrecher. **Visual data analysis with computational intelligence methods**. Bulletin of the Polish Academy of Sciences, Vol. 58(3), 2010.

[KRUSE et al. 2010a] R. Kruse, M. Steinbrecher and M. Böttcher. WCCI 2010 Plenary and Invited Lectures, Chapter **Temporal Aspects in Data Mining**, pp. 1–22. 2010. In: [ARANDA and XAMBÓ 2010].

[KRUSE et al. 2010b] R. Kruse, M. Steinbrecher and C. Moewes. **Data Mining Applications in the Automotive Industry**. In: [BEER et al. 2010], 2010, pp. 23–40.

[KRUSE et al. 2010c] R. Kruse, M. Steinbrecher and C. Moewes. **Temporal pattern mining**. In: Proceedings of the International Conference on Signals and Electronic Systems (ICSES 2010), pp. 3–8. Institute of Electronics, Silesian University of Technology, 2010, IEEE Press, Piscataway, NJ, USA.

[LAURITZEN and SPIEGELHALTER 1988] S. L. Lauritzen and D. J. Spiegelhalter. **Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems**. Journal of the Royal Statistical Society, Series B, Vol. 2(50):157–224, 1988.

[LEE et al. 2001] C.-H. Lee, C.-R. Lin and M.-S. Chen. **On Mining General Temporal Association Rules in a Publication Database**. In: In Proceedings of ICDM2001, pp. 337–344. 2001, IEEE Computer Society.

[MELIN et al. 2007] P. Melin, O. Castillo, L. T. Aguilar and W. Pedrycz, Eds. **Foundations of Fuzzy Logic and Soft Computing, Proceedings of the**

12th International Fuzzy Systems Association World Congress, IFSA 2007, Cancun, Mexico, June 18-21, Vol. 4529 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007.

[MOEWES and KRUSE 2009] C. Moewes and R. Kruse. **Zuordnen von linguistischen Ausdrücken zu Motiven in Zeitreihen**. at-Automatisierungstechnik, Vol. 57(3):146–154, 2009.

[OGASAWARA] **R Graphic Manual**. `http://rgm2.lab.nig.ac.jp/RGM2/`. Accessed April 2012.

[PEARL 1988] J. Pearl. **Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference**. Morgan Kaufmann, San Mateo, California, 1988.

[PLATTNER and ZEIER 2011] H. Plattner and A. Zeier. **In-Memory Data Management, An Inflection Point for Enterprise Applications**. Spinger-Verlag Berlin Heidelberg, 2011.

[PROKHOROV 2008] D. Prokhorov, Ed. **Computational Intelligence in Automotive Applications**, Vol. 132 of Studies in Computational Intelligence. Springer Berlin / Heidelberg, 2008.

[QUINLAN 1986] J. R. Quinlan. **Induction of Decision Trees**. Machine Learning, Vol. 1:81–106, 1986.

[RÁCZ et al. 2005] B. Rácz, F. Bodon and L. Schmidt-Thieme. **Benchmarking Frequent Itemset Mining Algorithms: From Measurement to Analysis**. In: Proc. Open Source Data Mining on Frequent Pattern Mining Implementations (OSDM 2005 at ACM SIGKDD 2005, Chicago, IL), pp. 36–45. 2005, ACM Press, New York, NY, USA.

[ROUX and ROUANET 2004] B. L. Roux and H. Rouanet. **Geometric Data Analysis—From Correspondence Analysis to Structured Data Analysis**. Kluwer Academic Publishers, 2004.

[RÜGHEIMER and KRUSE 2005] F. Rügheimer and R. Kruse. **Information Miner – a Data Analysis Platform**. In: Proc. of the Joint 4th EUSFLAT

11th LFA Conference 2005. 2005, Universitat Politènica de Catalunya, Barcelona, Spain.

[SENO and KARYPIS 2001] M. Seno and G. Karypis. **LPMiner: An Algorithm for Finding Frequent Itemsets Using Length Decreasing Support Constraint**. In: Proc. 1st IEEE Int. Conf. on Data Mining (ICDM 2001). 2001, IEEE Press, Piscataway, NJ, USA.

[SHENOI 2005] B. A. Shenoi. **Introduction to Digital Signal Processing and Filter Design**. Wiley-Interscience, 2005.

[SIMOFF et al. 2008] S. Simoff, M. H. Böhlen and A. Mazeika, Eds. **Visual Data Mining—Theory, Techniques and Tools for Visual Analytics**, Vol. 4404 of Lecture Notes in Computer Science. Springer Berlin/Heidelberg, 2008.

[SPENCE and LEWANDOWSKY 1991] I. Spence and S. Lewandowsky. **Displaying Proportions and Percentages**. Applied Cognitive Psychology, Vol. 5:61–77, 1991.

[STEINBRECHER 2006] M. Steinbrecher. **Lernen Graphischer Modelle zur Produktbewährung und Schadensanalyse in der Automobilindustrie**. Diploma thesis, Otto-von-Guericke-Universität Magdeburg in cooperation with DaimlerChrysler Research and Technology, 2006.

[STEINBRECHER and KRUSE 2007a] M. Steinbrecher and R. Kruse. **Visualization of Possibilistic Potentials**. In: [MELIN et al. 2007], 2007, pp. 295–303.

[STEINBRECHER and KRUSE 2007b] M. Steinbrecher and R. Kruse. **Visualizing Interesting Rules through Belief Network Inspection**. In: V. Köppen and R. M. Müller, Eds., Business Intelligence: Methods and Applications, Vol. 23 of Studien zur Wirtschaftsinformatik, Essays in Honor of Prof. Dr. Hans-J. Lenz **Visualizing Interesting Rules through Belief Network Inspection**, pp. 95–101. 2007. Verlag Dr. Kovač, Hamburg.

[STEINBRECHER and KRUSE 2008a] M. Steinbrecher and R. Kruse. **Identifying Temporal Trajectories of Association Rules with Fuzzy Descrip-**

tions. In: Proc. Conf. North American Fuzzy Information Processing Society (NAFIPS 2008), 2008, pp. 1–6.

[STEINBRECHER and KRUSE 2008b] M. Steinbrecher and R. Kruse. **Visualization of Local Dependencies of Possibilistic Network Structures**. Vol. 224 of Studies in Fuzziness and Soft Computing, pp. 93–104. 2008, Springer Berlin / Heidelberg.

[STEINBRECHER and KRUSE 2009a] M. Steinbrecher and R. Kruse. **Assessing the Strength of Structural Changes in Cooccurrence Graphs**. In: B. Mertsching, M. Hund and Z. Aziz, Eds., KI 2009: Advances in Artificial Intelligence, 32nd Annual German Conference on AI, Paderborn, Germany, Vol. 5803 of Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence, pp. 476–483. 2009, Springer Verlag.

[STEINBRECHER and KRUSE 2009b] M. Steinbrecher and R. Kruse. **Clustering Association Rules with Fuzzy Concepts**. In: [FINK et al. 2009], 2009, pp. 197–206.

[STEINBRECHER and KRUSE 2009c] M. Steinbrecher and R. Kruse. **Fuzzy Descriptions to Identify Temporal Substructure Changes of Cooccurrence Graphs**. In: Proceedings of 2009 IFSA/EUSFLAT, 2009, pp. 1177–1182.

[STEINBRECHER and KRUSE 2010] M. Steinbrecher and R. Kruse. **Visualizing and fuzzy filtering for discovering temporal trajectories of association rules**. Journal of Computer and System Sciences, Vol. 76(1):77–87, 2010.

[STEINBRECHER et al. 2008] M. Steinbrecher, F. Rügheimer and R. Kruse. Computational Intelligence in Automotive Applications, Chapter **Application of Graphical Models in the Automotive Industry**, pp. 79–88. 2008. In: [PROKHOROV 2008].

[STOBER et al. 2009] S. Stober, M. Steinbrecher and A. Nürnberger. **A Survey on the Acceptance of Listening Context Logging for MIR Applications**. In: [BAUMANN et al. 2009], 2009, pp. 45–57.

[TUFTE 2001] E. R. Tufte. **The Visual Display of Quantitative Information**. Graphics Press, 2nd Edn., 2001.

[WARD 2002] M. O. Ward. **A taxonomy of glyph placement strategies for multidimensional data visualization**. Information Visualization, Vol. 1:194–210, 2002.

[WONG et al. 1999] P. C. Wong, P. Whitney and J. Thomas. **Visualizing Association Rules for Text Mining**. IEEE Symposium on Information Visualization (InfoVis 1999, San Francisco, CA, USA), pp. 120–123, 1999.

[WWW: BORGELT 1] **Christian Borgelt: Apriori - Association Rule Induction / Frequent Item Set Mining**. `http://borgelt.net/apriori.html`. Accessed November 2012.

[WWW: BORGELT 2] **Christian Borgelt: Support of an Association Rule**. `http://borgelt.net/doc/apriori/apriori.html#supprule`. Accessed March 2012.

[WWW: HDD] **Cost of Hard Drive Storage Space**. `http://ns1758.ca/winch/winchest.html`. Accessed September 2012.

[WWW: SAP] **SAP Leads Next Wave of Computing With In-Memory Innovation**. `http://www.sap.com/press.epx?pressid=13293`. Accessed August 2012.

[WWW: SL] **Second Life**. `http://secondlife.com/`. Accessed September 2012.

[WWW: UCI] **UCI Machine Learning Repository**. `http://archive.ics.uci.edu/ml/`. University of California, Irvine, School of Information and Computer Sciences, Accessed August 2012.

[YANG 2005] L. Yang. **Pruning and Visualizing Generalized Association Rules in Parallel Coordinates**. IEEE Transactions on Knowledge and Data Engineering, Vol. 17:60–70, 2005.

[YANG 2008] L. Yang. **Visual Exploration of Frequent Itemsets and Association Rules**. In: [SIMOFF et al. 2008], pp. 60–75.

[ZADEH 1965] L. Zadeh. **Fuzzy Sets**. Information and Control, Vol. 3(8):338–353, 1965.

[ZAKI et al. 1997] M. J. Zaki, S. Parthasarathy, M. Ogihara and W. Li. **New Algorithms for Fast Discovery of Association Rules**. In: D. Heckerman, H. Mannila, D. Pregibon, R. Uthurusamy and M. Park, Eds., In 3rd Intl. Conf. on Knowledge Discovery and Data Mining, pp. 283–296. 1997, AAAI Press.

D
Lists

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

Acronym	Meaning
API	Application Programming Interface
DBMS	Database Management System
CSV	Comma-Separated Values
CW	Calendar Week
DM	Data Mining
JDBC	Java Database Connectivity
KDD	Knowledge Discovery in Databases
KPI	Key Performance Indicator
MDS	Multi-Dimensional Scaling
PCA	Principal Component Analysis
POJO	Plain Old Java Object
SDK	Software Development Kit
SQL	Structured Query Language
XML	Extensible Markup Language

E Index

Selbständigkeitserklärung

Ich, Matthias Steinbrecher, erkläre hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Insbesondere habe ich nicht die Hilfe eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Potsdam, November 2012

Matthias Steinbrecher