A Relevance Feedback System for Association Rules

– Diplomarbeit –

vorgelegt von

cand.inf. Georg Ruß

russ@cs.uni-magdeburg.de

6. September 2006

Erster Gutachter: Prof. Dr. Rudolf Kruse

Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Institut für Wissens- und Sprachverarbeitung
Universitätsplatz 2
D-39106 Magdeburg

Zweiter Gutachter: PD Dr. habil. Detlef D. Nauck

Intelligent Systems Research Centre
British Telecom Research & Venturing
Adastral Park, Martlesham
Ipswich IP5 3RE, UK

Ruß, Georg:
A Relevance Feedback System for
Association Rules
Diplomarbeit,
Otto-von-Guericke-Universität Magdeburg, 2006.

Abstract

Nowadays, businesses, governments, and individuals collect huge amounts of data. It is becoming more and more common to employ data mining on this data to find novel connections and patterns – valuable knowledge – that support a decision-making process. Association rules are one type of the patterns that are frequently mined, albeit often an overwhelming amount of patterns is generated. Finding the most interesting of those patterns is therefore an active research area and a key enabler for businesses and governments.

This thesis deals with time-stamped association rules and addresses the lack of methods to effectively find the most relevant association rules, according to the knowledge and the expectations of a particular user.

Scientific work has proposed to deal with the interestingness of association rules by giving numerous definitions. A literature review reveals that most of the existing work is unable to exactly model the user's expectations. Furthermore, most approaches are expert-based and require time-consuming and tedious work to specify knowledge about the domain.

This thesis contributes a new, more general, definition of similarity-based interestingness in the area of the *interestingness of association rules*. A relevance feedback framework will be shown that integrates this definition. It aims to precisely capture the user's knowledge about a domain, without requiring the user to specify his knowledge manually. It exploits a novel connection between association rules and information retrieval, which is a major contribution of this thesis. Finally, the framework is implemented in a sophisticated user interface, which renders the system usable even for an average user.

Experiments on association rules mined from survey data show the effectiveness of the developed framework in discovering interesting and potentially useful association rules.

Zusammenfassung

Unternehmen, Staaten, und Einzelpersonen sammeln heutzutage immer größer werdende Datenmengen. Darüberhinaus wird es mehr und mehr alltäglich, *Data Mining* anzuwenden, um wertvolles Wissen herauszufiltern. Assoziationsregeln werden häufig generiert und benutzt, um Muster in den Daten zu beschreiben, allerdings entstehen ebensohäufig zu viele dieser Regeln, was der nachgelagerten Datenauswertung zuwiderläuft. Es ist daher wichtig, Methoden zu entwickeln, die es ermöglichen, die interessantesten unter diesen Assoziationsregeln herauszufinden.

Diese Arbeit behandelt Assoziationsregeln und deren Zeitreihen, die zu aufeinanderfolgenden Zeiten aus Momentaufnahmen von Datenbanken generiert wurden. Der Mangel an Methoden, die es ermöglichen, effizient und möglichst automatisch die *interessanten Assoziationsregeln* herauszufinden und dabei die Erwartungshaltung eines Benutzers zu erfüllen, war der Auslöser für diese Arbeit.

In der Literatur sind viele Definitionen präsentiert worden, die sich der Interessantheit von Assoziationsregeln annehmen. Ein Rückblick auf existierende Arbeiten stellt heraus, daß es den meisten dieser Arbeiten nicht gelingt, die sich ändernde Erwartungshaltung des Benutzers korrekt abzubilden. Desweiteren sind die Ansätze meist expertenbasiert und erfordern zeitraubende Arbeit, um das Wissen über ein Gebiet im System zu verankern.

In dieser Arbeit wird eine neue, breiter gefaßte Definition der *subjektiven Interessantheit von Assoziationsregeln* vorgestellt. Davon ausgehend werden die Rahmenbedingungen für ein Relevanz-Feedback-System vorgegeben. Dieses System zielt darauf ab, den Benutzer davon zu entlasten, Wissen manuell vorgeben zu müssen. Eine *neuartige Verbindung* zwischen den Gebieten Assoziationsregeln und Information Retrieval wird hergestellt, die durch das Relevanz-Feedback-System ausgenutzt wird.

Experimente mit einem Satz von zeitreihenbehafteten Assoziationsregeln, die aus Umfragedaten gewonnen wurden, geben Grund zu der Annahme, daß das vorgestellte System es erlaubt, schnell und wirkungsvoll die interessanten Assoziationsregeln herauszufinden.

Acknowledgements

I'd like to thank all those people who were involved in this work. In chronological order: Prof. Dr. Rudolf Kruse for enabling the mind-enhancing stay at the ISRC of BT, Ipswich, UK in the first place; Dr. Detlef D. Nauck, for his organisational and academic support during my work at BT; Mirko Böttcher for the valuable input and the excellent collaboration and guidance during and after my placement; Christian Döring for the numerous corrections, invaluable comments and guidance on this thesis during its creation, Dr. Martin Spott for precious mathematical input. Special thanks go to Stefan Haun, Sebastian Blask, and David Brossard for making my stay in the East end of England more worthwile. Of course, this thesis would not have been possible without my parents' financial and Mimi's and WalTher's emotional support. Thanks to all of you.

“The control of information is something the elite always does, particularly in a despotic form of government. Information, knowledge, is power. If you can control information, you can control people.”

Tom Clancy

“Information technology and business are becoming inextricably interwoven. I don’t think anybody can talk meaningfully about one without talking about the other.”

Bill Gates

Contents

1 Introduction **1**

1.1 A Second-Order Data Mining Problem . . . 2

1.2 Objectives of this Thesis . . . 2

1.3 Thesis Outline . . . 3

2 Basic Concepts **5**

2.1 Association Rules . . . 5

2.1.1 Basic Idea . . . 5

2.1.2 Formal Definition . . . 6

2.1.3 Histories . . . 7

2.2 An Exemplary Association Rule Set . . . 8

3 Interestingness of Association Rules **11**

3.1 Introduction . . . 11

3.1.1 Chapter Structure . . . 12

3.2 Related Work . . . 12

3.2.1 Rule Templates . . . 12

3.2.2 Unexpectedness: Via Contradiction . . . 13

3.2.3 Unexpectedness: Via General Impressions . . . 14

3.2.4 Reliable Exceptions . . . 16

3.2.5 Interestingness Via What Is Not Interesting . . . 17

3.2.6 Surprisingness . . . 18

3.2.7 Actionability . . . 19

3.2.8 Rule Change Mining . . . 20

3.3 Requirements Analysis . . . 23

3.3.1 Literature Summary . . . 23

3.3.2 Some Remarks . . . 24

3.3.3 List of Requirements . . . 25

3.4 System Layout . . . 26

3.5 Similarity-Based Interestingness . . . 28

3.6 Summary . . . 31

4 A Connection to Information Retrieval 33
4.1 Introduction . . . 33
4.2 Linking association rules and information retrieval . . . 33
4.2.1 Querying . . . 33
4.2.2 Filtering . . . 34
4.2.3 Pattern Matching . . . 34
4.2.4 Relevance Feedback . . . 34
4.2.5 Scoring and Ranking . . . 34
4.2.6 User Interface . . . 35
4.2.7 Text Retrieval . . . 35
4.2.8 Conclusion . . . 35
4.3 Choice of an IR Model . . . 36
4.4 Summary . . . 37

5 Preprocessing 39
5.1 From Association Rules to Rule Feature Vectors . . . 39
5.1.1 Calculation of Item Weights . . . 40
5.1.2 Vector Generation . . . 40
5.2 Summary . . . 43

6 Relevance Feedback 45
6.1 Introduction to Relevance Feedback . . . 45
6.2 User Interaction . . . 47
6.2.1 Rule Display . . . 47
6.2.2 Relevance Acquisition . . . 47
6.2.3 Score Update . . . 48
6.3 Initial Ranking . . . 48
6.3.1 Sorting by Existing Orders . . . 48
6.3.2 Feature-Vector-Based Data Processing . . . 48
6.4 Interestingness Scoring . . . 49
6.4.1 Similarity Measures . . . 50
6.4.2 Knowledge Base . . . 50
6.4.3 Rule Set Aggregation . . . 51
6.4.4 Relative Importance of Recent Relevance Choices . . . 54
6.4.5 Similarity between Rule and Rule Set . . . 55
6.4.6 Handling of Relevance Information . . . 56
6.4.7 Score Aggregation . . . 58
6.5 Experimental Results . . . 58
6.5.1 User-Adjustable Weights . . . 58
6.5.2 Weight Settings . . . 58
6.6 Summary . . . 60

7 User Interface 61
7.1 Overview . . . 61
7.2 Rule Browser Module . . . 61
7.2.1 Existing Browser . . . 61
7.2.2 Relevance Selection . . . 61
7.3 Relevance Feedback Module . . . 62
7.3.1 Average User . . . 62
7.3.2 Expert User . . . 64
7.3.3 Relevant and Non-Relevant Rules . . . 64
7.4 Rule Filter Module . . . 66
7.5 Summary . . . 66

8 Conclusions 69
8.1 Summary . . . 69
8.2 Future Work . . . 70

List of Figures 73

List of Tables 75

Bibliography 81

Chapter 1

Introduction

The year 2006 sees us living in a highly networked and connected world. Ever more machines, devices and people are connected to the ever-growing, ubiquitous and pervasive internet. With this, the amount of data which is held by the net constantly increases. It is becoming increasingly common to leave traces of data in everyday life, be it consented in, e.g., a blog or unconsented as in case of wide-ranging surveillance. There lies no exaggeration in saying that the average citizen's (or consumer's) data footprint is growing at a rate unthinkable a few years ago. With ever cheaper data storage and the growing interests of businesses which demand more and more information about their (potential) customers, the collection of data is not likely to cease. Not only do businesses collect valuable data, governments aim to provide safety to their citizens by analysing large databases of citizens' behaviour, their profiles, their overall data footprint. Yet this accumulation of data clearly demonstrates the key problem of finding surprising patterns of behaviour, hidden connections in the data and constellations that could be interesting from the companies', governments' or individuals' point of view.

The process of finding these patterns has been termed *data mining* by [Fayyad et al., 1996] and is formally referred to as 'finding valid, novel, potentially useful, and ultimately understandable patterns in data'. Over the past ten years since the term's establishment, significant efforts have been devoted to automate the process of information extraction and knowledge distillation by the research community. Methods, ideas and algorithms from as diverse fields as artificial intelligence, statistics, soft computing and machine learning are involved, to name but a few. Clustering algorithms, neural networks, variance analysis, trend detection and association rules depict just a small fraction of the research areas incorporated into data mining.

As mentioned, the amount of data to be analysed is voluminous and growing. Nevertheless, methods exist to condense the data to fractions of the original, making it accessible to humans. Necessarily, these methods and tools developed to cope with this data require expert guidance. Eventually the user decides which information is *relevant* to him and which is not. Thus, an important part in the data mining process is the human-machine interaction, presenting a great challenge as well as outstanding chances. Accounting for this chance of acquiring human knowledge and expectations will render any devised data mining software much more useful. Effective software, in turn, can be used advantageously by companies' users to benefit their business: explaining sequences from the past, assisting with present decisions and guiding future directions. Therefore, this thesis will focus on devising new ways to take advantage of the knowledge gathered by human-machine interaction to identify and report information which is *interesting* to the user.

1.1 A Second-Order Data Mining Problem

Corporate and governmental data warehouses have grown to considerable sizes, unmanageable for the information-seeking human. One of the key enablers to explore acquired data is the concept of *association rule mining*, which was developed by [Agrawal et al., 1993] in the early 1990's. It was originally targeted at market basket data analysis, where each transaction consists of a set of purchased items. Its goal is to detect all those items which occur together frequently and to form rules predicting the co-occurrence of these items. Aside from data stores containing transactions, this concept can be applied to arbitrary relational databases. An exemplary association rule could predict that 10% of female customers using a new broadband tariff will cancel their contract within six months[1]. The probability attached to this rule is called confidence, whereas the fraction of transactions that contain these three items (female, broadband, cancellation) is termed support. A superior property of association rule mining is the completeness of its results: it finds the set of all patterns which exceed specified minimum confidence and support thresholds. This, in turn, generates a rather detailed description of the data's structure, leading to a new dataset of association rules which is likely to be large. Rule sets consisting of thousands of rules are to be expected. These rules have already been distilled from the relational database and will contain very valuable business information. Since the large amount of rules results from a data mining step, this dilemma is also called a *second-order data mining problem*. The process is unlikely to be dealt with automatically, so the user is involved and in need of machine support to fulfill his task. Software that aids in extracting the interesting rules is very desirable as well. Dealing with those quantities of rules is therefore a very active research area, which this work contributes to.

1.2 Objectives of this Thesis

This thesis intends to develop a framework of methods that allows users to navigate and explore large sets of patterns. The framework is required to aid the user in identifying those patterns which are *interesting* or *relevant*. To achieve this overall goal of an interestingness assessment, the user's knowledge and his feedback shall be taken into account.

Association rule mining has been applied to survey data and has created a data mining problem of second order. A large set of patterns in the form of association rules exists that has to be explored manually and which the proposed framework will

[1] More examples of association rules can be found in Section 2.2

handle. Furthermore, the association rules have been generated from time-stamped data, which yielded histories of numerical values for each association rule. These two features, the rules and their time series, are available to the framework. The framework is supposed to be integrated into an existing association rule browser, extending the current implementation.

In addition to the association rules, the assumptions and expectations of the user which interacts with the framework are available. Ultimately, the user decides which information is interesting to him and which is not. The framework should therefore account for the human knowledge during the interestingness assessment of association rules.

Different types of users with different skill levels should be accounted for, such as a distinction between an "average" and an "expert" user. A user interface has to be designed that accounts for these types and integrates into the existing implementation of an association rule browser.

The foundations for a novel connection between association rules and information retrieval will be laid and built upon during this thesis, which is one of the main contributions to the field of the interestingness of association rules.

An analysis of the task of *discovering interesting association rules* reveals three key issues that will have to be solved: first, the concept of interestingness has to be defined; second, given the large set of association rules and their histories, a suitable way of preprocessing has to be found that enables us to deal with the rules appropriately; third, suitable methods have to be developed or adopted that will take care of the acquisition of the user's knowledge who interacts with the framework.

In the literature, numerous approaches exist to assess the interestingness of association rules, which is the first key issue above. However, an analysis will reveal their general drawbacks, but will also enable to specify requirements which should be met by the suggested framework. This thesis aims to establish new and improved methods for the interestingness assessment, which will partly be based on the ideas of existing approaches.

1.3 Thesis Outline

Chapter 2 describes the *basic concepts* used throughout this thesis, specifically association rules and the idea of their histories. To better grasp those concepts a prototypical set of association rules is shown, which will be discussed in detail and used for demonstration purposes along the remaining chapters.

Chapter 3 lays its emphasis on the problem statement of *the interestingness of association rules* Related work on the notion of interestingness of association rules and their histories is presented and evaluated. Based on those evaluations, requirements are specified, which lead to a framework based on relevance feedback. The concept of similarity-based interestingness is developed, solving the first key issue.

Chapter 4 establishes and substantiates the novel link between association rules and information retrieval. A data processing model is chosen.

Chapter 5 discusses the necessary preprocessing measures taken to adapt the association rules to the model chosen before, solving the second key issue. A feature-vector-based representation for the association rules is defined.

Chapter 6 deals with the details of *relevance feedback*. Assumptions about the user's preferences are transformed into a mathematical representation. An *interestingness score* will be introduced. It captures the interestingness of a particular association rule after the user has made choices on the relevance of rules. It solves the third key issue.

Chapter 7 shows implementation details in principle. The user interface of an association rule browser and the developed relevance feedback extensions are discussed.

Chapter 8 summarises this thesis' results and indicates possible future work.

Chapter 2

Basic Concepts

This chapter will first formally define association rules, as they are at the very foundation of this thesis. Their origins will be described and their numerical features will be defined. Time series, so-called histories, of those numerical features will be introduced. Finally, a fictitious association rule set will be described. It will serve as an illustrative example throughout this thesis.

2.1 Association Rules

2.1.1 Basic Idea

Loosely speaking, association rules are yet another way to describe the relationships in data. Originally, they were designed to help store owners and large warehouses improve their respective shop's layout. By analysing collected shopping basket data, so-called *frequent itemsets* could be discovered: sets of items which are frequently bought together and therefore contain valuable information for the store owner. The now famous example is the frequent itemset {nappies, beer}: if this is understood as an *association rule* `nappies` $\Rightarrow$ `beer`, it might lead to the decision to place these items next to or in the vicinity of each other on the shelves, improving the overall profit. On the other hand, the decision could as well be made to place these items as far away from each other as possible, which would increase the amount of aisles that the customer visits. In subsequent years, the concept has been generalised and applied to different domains. An algorithm to mine these rules, termed 'A-Priori', has been proposed by [Agrawal et al., 1993]. More efficient [Agrawal and Srikant, 1994] and more general [Srikant and Agrawal, 1997] algorithms have been developed, among others. Sophisticated A-Priori implementations are available, e.g. by [Borgelt and Kruse, 2002].

The applicability of association rule mining is not limited to shopping cart data, but they can be used to analyse different kinds of databases containing tables with nominal attributes. For example, the particular purpose that association rules have been used for in this thesis' context is survey data. Imagine a large business obtaining feedback about their products by conducting customer satisfaction surveys. Usually, the answers are predefined and have to be selected by the customer. By returning the answers, the company now obtains sets of items where an item consists of a question-answer combination. Given a sufficient amount of survey data and well-designed question-answer combinations to choose from, the company can now improve its service by, e.g., offering special products to certain customer groups, revising their internal workflow, or predicting the outcome of business decisions.

2.1.2 Formal Definition

In general, association rule mining is applied to a set $\mathcal{D}$ of *transactions* $\mathcal{T} \in \mathcal{D}$, where each transaction $\mathcal{T}$ is a subset of $\mathcal{L}$, a set of literals. The literals are commonly called *items* and a subset $\mathcal{X} \subseteq \mathcal{L}$ with size $k = |X|$ is named *k-itemset* or *itemset.* A transaction $\mathcal{T}$ *supports* an itemset $\mathcal{X}$ if $\mathcal{X} \subseteq \mathcal{T}$.

An *association rule* r is an expression $\mathcal{X} \Rightarrow \mathcal{Y}$ where $\mathcal{X}$ and $\mathcal{Y}$ are itemsets, $|\mathcal{Y}| > 0$ and $\mathcal{X} \cap \mathcal{Y} = \varnothing$. Following the arguments from section 2.1.1, its explanation is quite natural: given the database $\mathcal{D}$ from which the rules have been mined, an association rule conveys that whenever $\mathcal{X} \subseteq \mathcal{T}$ holds, $\mathcal{Y} \subseteq \mathcal{T}$ is likely to hold as well. The concepts of *generalisation* and *specialisation* are defined as follows: if for two rules $r : \mathcal{X} \Rightarrow \mathcal{Y}$ and $r' : \mathcal{X}' \Rightarrow \mathcal{Y}', \mathcal{X} \subset \mathcal{X}'$ holds, it is denoted as $r \succ r'$; r is said to be a generalisation of r' and consequently r' a specialisation of r. Since this thesis partly extends the work of [Böttcher, 2005], it also holds that I am dealing with association rules containing a 1-itemset as consequent. Therefore, a rule can be written as $\mathcal{X} \Rightarrow y$, with $\mathcal{X} \subset \mathcal{L}$ and $y \in \mathcal{L}$. Given a rule $r : \mathcal{X} \Rightarrow \mathcal{Y}$, $\mathcal{X}$ is called the *antecedent* or *body* of the rule, whereas $\mathcal{Y}$ is termed *consequent* or *head* of the rule.

Association rules inevitably have some numerical features attached to them. A rule's *confidence* conf(r) measures the predictive ability or the reliability of a rule. Given a rule $r : \mathcal{X} \Rightarrow \mathcal{Y}$, it is defined as the ratio of transactions that contain $\mathcal{Y}$ in addition to $\mathcal{X}$ in relation to the number of transactions that contain $\mathcal{X}$ exclusively:

$$\text{conf(r)} := \frac{|\{\mathcal{T} \in \mathcal{D} | \mathcal{X} \cup \mathcal{Y} \subseteq \mathcal{T}\}|}{|\{\mathcal{T} \in \mathcal{D} | \mathcal{X} \subseteq \mathcal{T}\}|} \tag{2.1}$$

The *support* measures a rule's significance. Given a rule $r : \mathcal{X} \Rightarrow \mathcal{Y}$, the support is defined as the proportion of transactions that contain $\mathcal{X} \cup \mathcal{Y}$:

$$\text{supp(r)} := \frac{|\{\mathcal{T} \in \mathcal{D} | \mathcal{X} \cup \mathcal{Y} \subseteq \mathcal{T}\}|}{|\mathcal{D}|} \tag{2.2}$$

Rules with very low support values probably represent outliers or very small numbers of transactions that are unlikely to be of interest; nevertheless they can be valuable. An additional concept of *antecedent support* is sometimes used. It is defined as the fraction of transactions which contain $\mathcal{X}$:

$$\text{asupp(r)} := \frac{|\{\mathcal{T} \in \mathcal{D} | \mathcal{X} \subseteq \mathcal{T}\}|}{|\mathcal{D}|} \tag{2.3}$$

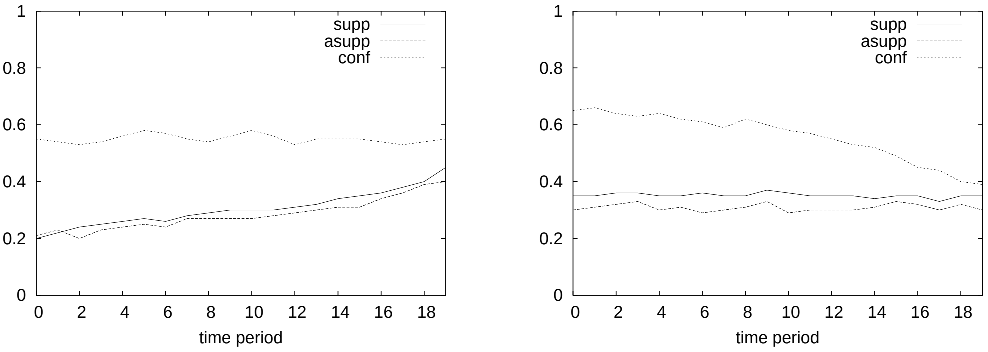
Based on these definitions, the algorithms which generate association rules from database transactions can efficiently restrict their search space to rules with minimum support and/or confidence. The algorithm details are outside the scope of this work and can be found in [Agrawal et al., 1993] (A-Priori). Other approaches to find association rules exist, such as an equivalence class-based approach by [Zaki, 2000] (Eclat) or [Savasere et al., 1995] (Partition). A sophisticated implementation of the A-Priori and Eclat algorithms has been given by [Borgelt, 2003]. Depending on the domain, further specialised algorithms to mine rules exist, e.g. in [Lu et al., 2005], or [Lee et al., 2001]

2.1.3 Histories

Association rules are mined from database snapshots at certain points in time. As more and more data is collected, the databases that are used to generate association rules might change drastically over time. Therefore an association rule's numerical properties (such as confidence, support, antecedent support) are also likely to change over time. Those numerical values can be collected in time series, which are called *histories*, for obvious reasons. On the one hand, this contradicts the common assumption of association rule mining that the considered domain is stable over time. On the other hand, if one considers the descriptive power of the time series information, those histories are definitely an association rule's feature that can distinguish between an interesting and a less interesting rule.

The generation of the rule histories is simple, though computationally heavy: database snapshots are taken regularly and rules are extracted from this data. Their confidence, support and antecedent support are measured and stored in a new *rule database* in a way such that every rule now has several histories attached to it: one each for confidence, support, and antecedent support.

Figure 2.1 shows two examples for the time series attached to an association rule: 2.1(a) shows a rule with an upward trend in support and antecedent support, and stability for confidence; 2.1(b) shows a downward trend in confidence, and stability for the support values.



(a)

(b)

Figure 2.1: Plots of two association rules' support and confidence values

2.2 An Exemplary Association Rule Set

To better grasp the concept of association rules and their descriptive power regarding large datasets, a simple, small set of rules is presented below. Let us assume that a telecommunications provider supplies internet access to its end customers via different access technologies[1]. Surveys about customer satisfaction have been conducted and their results have been stored in a database. According to the definition from Section 2.1 the attributes with their realisable values will be explained first, structured in an ATTRIBUTE=VALUE1, VALUE2, ... way.

TEC = ADSL, CABLE, UMTS, WIMAX Describes the broadband internet access technology provided to the customer by the telecommunications company; Asymmetric Digital Subscriber Line, Internet over TV cable, Universal Mobile Telecommunications System, Worldwide Interoperability for Microwave Access.

VOL = NONE, 1GB, 10GB, FLAT Describes the data transfer volume paid for in advance, included in the customer's bills; no volume included, 1 Gigabyte, 10 Gigabytes, all volume included.

AGE = 18-35, 36-50, 51-65, 66+ Divides the customers into age groups.

SEX = M, F Denotes the sex of the customer; male or female.

SAT = VSAT, SAT, DISSAT Denotes one of the key indicators for businesses, describing overall customer satisfaction; very satisfied, satisfied, dissatisfied.

WCC = YES, NO Describes whether the customer would be willing to extend his contract with the provider; yes or no.

EQS = GOOD, BAD Denotes the experience of customers with the quality of service; good or bad.

Starting with transactions that contain the above attribute-value combinations, association rule mining is applied to the database that contains these transactions. More specifically, the rule mining is applied once for each database snapshot. These database snapshots have been created in regular intervals. For each association rule in each snapshot, confidence, support, and antecedent support values are calculated. With these values, the respective histories can be generated. Trend and stability detection can then be applied to the histories. For ease of presentation, the table below shows the time series as a simplified version, for rules with detected trends and/or stabilities only. Table 2.1 shows a small number of association rules that may have been mined, among numerous others.

[1] For reasons of data protection it is not possible to disclose any data which might be related to British Telecommunications plc or its customers. All given examples in this thesis as well as interpretations thereof, are completely fictitious and therefore not related to British Telecommunications plc or its customers. The given samples are for demonstration and contextualisation purposes only.

	rule symbolic		*time series*	
#	body/antecedent	head/consequent	support	confidence
1	TEC=ADSL	WCC=YES	stable	stable
2	SEX=M, TEC=WIMAX	VOL=FLAT	up	up
3	AGE=66+	VOL=NONE	down	down
4	SEX=F, AGE=18-35, EQS=BAD	WCC=YES	up	down
5	TEC=ADSL, AGE=36-50	WCC=NO	stable	stable
6	AGE=51-65, VOL=10GB	SAT=VSAT	up	up
7	AGE=66+	VOL=10GB	up	up
8	AGE=18-35, VOL=FLAT	SAT=VSAT	up	up
9	TEC=UMTS, VOL=1GB	WCC=YES	down	down
10	AGE=66+, TEC=CABLE	SAT=SAT	stable	stable

Table 2.1: Exemplary set of association rules

Chapter 3

Interestingness of Association Rules

3.1 Introduction

Once the association rules and their histories have been mined, depending on the input dataset, the second-order data mining problem mentioned in Section 1.1 is likely to be encountered: The large number of association rules mined inhibits manual exploration. The naive approach, to reduce the number of association rules in the mining step by changing support and confidence thresholds, is unsuitable since important rules might have low support and confidence values. Therefore a different approach has to be taken, laying its emphasis on the notion of *interestingness*. This chapter aims to achieve a basic understanding of this concept.

The association rules which are considered here have two fundamentally different properties: firstly, their *symbolic representation*, as defined in Section 2.1.2. Secondly, each rule has a *time series representation* of the confidence, support, and antecedent support values mined at different points in time (Section 2.1.3). These two, semantically different, features are available to assess the interestingness of association rules. Regarding the symbolic representation, [Hilderman and Hamilton, 2003] have performed a classification of interestingness measures for discovered knowledge. They have identified numerous measures for interestingness in the association rule model. Regarding the time series representation, there is fewer literature available that deals with the interestingness of an association rule's histories. A good overview can be obtained by the review of approaches to *rule change mining* in [Böttcher, 2005].

Certainly the most basic interestingness measures for a set of association rules are the support and confidence values already presented in Section 2.1.2. In addition to those, there are numerical properties that can be derived from them. An example, that is sometimes called an objective measure of interestingness, is the *lift* value:

$$\text{lift}(\mathcal{X} \Rightarrow \mathcal{Y}) := \frac{\text{supp}(\mathcal{X} \cup \mathcal{Y})}{\text{supp}(\mathcal{X}) \cdot \text{supp}(\mathcal{Y})} \tag{3.1}$$

This, as well as the confidence, support, and antecedent support values, might be sufficient to define a basic ordering on the set of rules, but I assume they do not take the concept of interestingness into account as it is conceived by a human. Interestingness is highly subjective and is unlikely to be captured correctly by objective measures, therefore those measures are not considered here. Numerous other measures are specialised to work in the rule mining process, such as the 'strength' developed by [Gray and Orlowska, 1998], different metrics as assessed in [Roberto J. Bayardo and Agrawal, 1999] or incorporating Bayesian networks as background knowledge [Jaroszewicz and Simovici, 2004]. However, this thesis deals with a process where the association rules have already been mined, therefore those concepts can not be taken into consideration here.

3.1.1 Chapter Structure

An overview of the wide-spread field of interestingness assessment of association rules can be obtained by the surveys of [Hilderman and Hamilton, 1999] and [McGarry, 2005]. I am going to consider seven of the approaches which are mentioned in those reviews in this thesis, in the following overview (Sections 3.2.1 to 3.2.7). I will present those that are relevant to my work and which contain ideas that are worth further consideration. In addition, an approach that deals with several interestingness measures for association rules' time series will be summarised in Section 3.2.8. A short evaluation will be given for each approach. It will include the categorisation into subjective or objective interestingness, which is a distinction that has been agreed upon throughout the literature. Roughly speaking, the subjective- or objectiveness of a measure depends on whether the user is involved in the discovery process (subjective or user-driven) or whether the machine computes interestingness automatically (objective or data-driven).

After the literature has been evaluated, the requirements of an interestingness assessment system will be derived directly from the reviewed work (Section 3.3). A system layout that captures the requirements will be proposed in Section 3.4. It will be based on a notion of similarity between association rules and their temporal features, defined in Section 3.5, which will conclude this chapter.

3.2 Related Work

3.2.1 Rule Templates

Inspired by the work of [Hoschka and Klösgen, 1991], [Klemettinen et al., 1994] have proposed a novel concept to address the problem of second-order data mining as encountered in my work. They assume that the user's knowledge is invaluable to influence the interestingness assessment and therefore require him to optionally specify a descendant of *regular expressions for association rules*. Since regular expressions can also be seen as some kind of template that has to be matched, their idea is called *rule templates*. As in the approach by [Liu et al., 2000] in Section 3.2.3, a hierarchy of possible attributes is generated manually, adding relative information to the database. After classes of attributes and their hierarchies have been created, the user has to specify *inclusive* templates that an interesting rule must match. Once matching rules have been found (or even before that step), the user can specify rule *restrictive* templates that an interesting rule must not match. After the specification step, the system applies the rule templates to the rule set and filters accordingly. Seen from a different perspective, their system represents a filter much like a specialised text processor that handles association rules and works solely on a syntax-level. The proposed filter is complemented by an enhanced visualisation interface, which allows for easy click-and-drag of rule templates, as well as for rule browsing and a sophisticated rule visualisation by means of a rule graph.

Evaluation

Developed in 1994, the rule templates approach obviously must have its limitations due to computer hardware restrictions, posing a challenge to researchers when computing time was more limited than nowadays. Nevertheless, it certainly touches the important aspect of *filtering redundant or common knowledge* by specifying attributes or classes of attributes. If a user is definitely not interested in finding out about what he expects he already knows, he can choose to reduce the rule set accordingly. The drawback is that he might miss important information because it has been filtered, since he did not know about its existence. However, the basic assumption that prefiltering can be highly useful still holds. Therefore, a way of filtering association rules should be given in any association rule mining system, along with an easy-to-use interface, if only for an expert user. This approach obviously falls into the category of subjective interestingness measures since the user specifies the inclusive and exclusive templates.

3.2.2 Unexpectedness: Via Contradiction

The work of [Padmanabhan and Tuzhilin, 1998] represents another approach that has to be considered in this review. They connect belief models with patterns, whereas the latter term is used synonymously for association rules. In particular, their initial work assumes that a belief system exists and that it can be used for the rule discovery algorithms. In subsequent works ([Padmanabhan and Tuzhilin, 2000] and [Padmanabhan and Tuzhilin, 2002]) the problem of specifying this initial set of beliefs is handled by having experts specify those beliefs manually. They assume that rules and beliefs are defined as association rules of the form $X \rightarrow A$, where X is an itemset and A is an item. This complies with the kind of association rules that is dealt with in this thesis. Further assumptions hold as defined in Section 2.1. Based on this definition of rules and beliefs, a rule $A \rightarrow B$ is defined to be *unexpected* with respect to the belief $X \rightarrow Y$ on the rule database D if the following conditions hold:

(a) B and Y logically contradict each other ($B \wedge Y \models \text{FALSE}$);

(b) $A \wedge X$ holds on a "large" subset of tuples in D;

(c) The rule $A, X \rightarrow B$ holds.

Given this definition, in [Padmanabhan and Tuzhilin, 1999] an algorithm *ZoomUR* is proposed which discovers the set of unexpected rules regarding a specified set of beliefs. The algorithm itself consists of two different discovery strategies: *ZoominUR* discovers all unexpected rules that are refinements (or specialisations). On the other hand, *ZoomoutUR* discovers all unexpected rules that are more general (the definitions of *specialisation* and *generalisation* can be obtained from Section 2.1). Results are presented, e.g. from a marketing application and a web logfile mining application. Furthermore, it is mentioned that some kind of tradeoff has to be made between the generation of too many rules by a-priori (albeit containing all the interesting rules) and an insufficient number of rules by ZoomUR, possibly missing interesting ones.

Evaluation

This is the first of two reviewed definitions of interestingness that use unexpectedness as an interestingness measure. This approach describes an algorithm that works in the rule mining process and can handle large numbers of association rules. It follows from the three logical criteria above that the algorithm will work with the rules already mined as well, since it will then consider a smaller subset of the overall rule set. Here, unexpectedness is defined as *contradicting a set of beliefs* which is quite intuitive. However, the main limitation is the definition of this *set of beliefs* by a domain expert, which is costly, tedious and error-prone. The algorithm is not guaranteed to find every interesting rule available, partly due to the tradeoff between the generation of too few and too many rules. This approach classifies as a subjective interestingness measure since beliefs have to be specified manually.

3.2.3 Unexpectedness: Via General Impressions

[Liu et al., 2000] address the insufficiency of objective interestingness measures by focusing on the unexpectedness of generalised association rules. They assume that taxonomies exist among association rules' attributes; an example taxonomy can be found in Figure 3.1. Human knowledge is recognised to be granular, i.e., with different degrees of certainty or preciseness. Their system allows for three degrees of preciseness, notably *general impressions* ([Liu et al., 1997]), *reasonably precise concepts* and

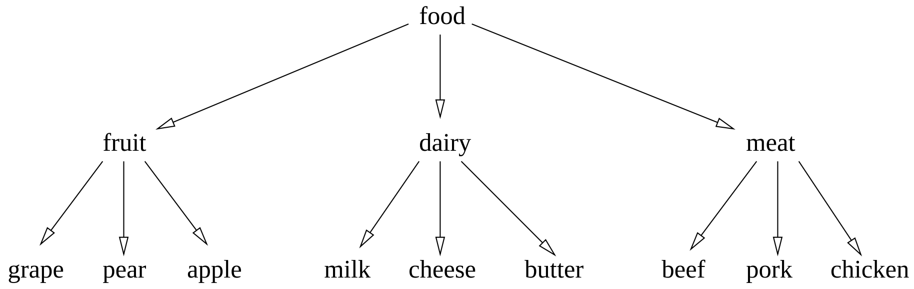


Figure 3.1: An example taxonomy

precise knowledge. It is reasoned that humans often have vague assumptions about connections, for example that (using Figure 3.1) grapes and pears should be associated, but that it is not mandatory. On the other end of the preciseness spectrum, a user may have obtained the precise knowledge that buying bread implies buying milk with a support of roughly 10% and confidence of roughly 70%. The proposed Interestingness Analysis System (IAS) accounts for these differences and uses the gathered knowledge to find rules which are unexpected in regard to the expressed knowledge. IAS works iteratively:

1 the user specifies his knowledge or modifies previously specified knowledge, supported by the specification language;

2 the system analyses the association rules according to conformity and unexpectedness;

3 the user inspects the analysis results (aided by visualisation), saves interesting rules and discards uninteresting rules.

The first step is rather straight-forward once the concept of the specification language has been grasped. Based on the three preciseness categories above, a user can express his knowledge, with constraints for each category's contents for syntax as well as confidence and support values. In the second step, the IAS uses this information by performing a syntax-based analysis to find unexpected rules, i.e., those which do not conform to the knowledge. Again, all possible combinations of features are analysed to identify unexpected rules. Since each rule consists of an antecedent and a consequent with boolean conformity (*matches*, *does not match* the specified knowledge), the four resulting possibilities are exploited to determine unexpected rules by calculating a degree of match. A hierarchy is created on top of the four possibilities; using the degrees of match a re-ranking of the rules is calculated. Finally, a visualisation is used to present the results to the user.

Evaluation

A very interesting and straight-forward approach itself, the IAS addresses some key issues that I encounter as well: the interestingness of association rules is broken down to *unexpectedness based on syntax*; the *insufficiency of objective measures of interestingness* is pointed out, and the *human knowledge is explicitly accounted for*; finally, a visualisation is used to present the rules. The main drawback of the approach is the explicit specification of a user's knowledge using yet another language which needs practice and experience. Even if the language has been explicitly constructed to support this task, it still is a tedious and error-prone job to specify knowledge from memory. Furthermore, the categorisation of a user's knowledge is expected, presenting another burden to the user. Finally, the visualisation as shown in [Liu et al., 2000] seems rather less intuitive than what a user would expect or what he is already used to. As with the approach presented in Section 3.2.3, a certain knowledge has to be specified first by the user, therefore this approach classifies as subjective interestingness measure as well.

3.2.4 Reliable Exceptions

[Liu, Lu, Feng, and Hussain, 1999] start with a three-fold categorisation of patterns into strong, weak and random patterns. The first category represents regularities for numerous objects - that is what conventional data mining techniques are designed to find. They focus on the second category, weak patterns, in their case called *reliable exceptions*, which represent a relatively small number of objects. Based on the work of [Suzuki, 1997], they discover that current data mining techniques as well as intuitive extensions cannot help finding those weak patterns. Depending on the bias of the rule induction algorithm, there could be either too many or not enough of them. In their approach, they skip searching for strong patterns altogether. Instead, deviation analysis is used to find exceptional instances in the original data and mine weak patterns from them. The algorithm works in four consecutive steps:

Rule induction and focusing As in 'normal' rule mining approaches, this phase obtains the strong patterns. The following steps require some rules (or some knowledge) that they can focus on to find reliable exceptions in regard to these patterns.

Contingency table and deviation The algorithm now focuses on some attribute in a rule and uses the additional class attribute to build a two-way contingency table which allows for calculating deviations.

Positive, negative, and outstanding deviations Based on their definition of deviation, there are three possible types: positive, zero, or negative. Positive deviation marks consistency with the strong patterns, zero deviation marks norm, negative deviation marks inconsistency with the strong patterns. Depending on a user-specific threshold, negative deviations can be outstanding and therefore interesting.

Reliable exceptions Once outstanding negative deviations have been found, they will be used to obtain every instance that contains those attribute-value and class pairs, therefore reducing the dataset size considerably and improving mining performance. Further filtering according to support and confidence values should be applied.

Evaluation

This approach categorises the expected patterns in the dataset into three parts: strong, weak, random. It is reasoned that the weak patterns only can be the interesting ones that have to be found. Based on deviation analysis, the weak patterns can be identified and mined. However, this advanced approach still has to rely on conventional data mining in its algorithm's first step; this originates from the term 'exception' which inherently assumes that standards are known to find deviations from it. Consequently, the first step gathers those standards whereas the rest of the algorithm builds upon the found standards and uses them to find exceptions. This usage of conventional techniques might represent a drawback, but the basic idea to *find exceptions in regard to a set of strong patterns* will be remembered. However, a full understanding of deviation analysis requires a strong background in statistics. Therefore the rest of this approach should be left to an expert user. Nevertheless, this approach can be classified as an objective interestingness measure since the algorithm requires no user-specific input and works automatically.

3.2.5 Interestingness Via What Is Not Interesting

A different approach to assess interestingness has been proposed by [Sahar, 1999] and improved in [Sahar, 2002]. They refer to the distinction between objective and subjective interestingness measures examined by [Silberschatz and Tuzhilin, 1996] and concentrate on subjective interestingness. The subjectivity requires the acquisition of the user's knowledge, either by having a domain expert specify his knowledge or by having the user classify each rule. Since both of these approaches are expected to be very costly, if not infeasible, their novel idea consists of reducing the number of rules by eliminating those which are *not interesting.* As in the measurement of unexpectedness presented in Section 3.2.3, all possible combinations of features are evaluated. This yields four possible combinations of features. Here, the classification task consists of judging two features of a rule: first, if it is true in general; second, if the user is interested in the family of the classified rule. The first choice is used to construct a knowledge base and to exclude common knowledge, whereas the second choice characterises the family of the rule. Each of the resulting four classification combinations is accounted for in the algorithm. It works iteratively by generating a *best candidate*, which is presented to and judged by the user. The best-candidate selection process also relies on whether a rule's presentation would eliminate a sufficient, considerable amount of related rules. Based on the user's choice, the algorithm eliminates those rules which it deems uninteresting. Experiments have shown that a decrease in problem size of about 50% could be achieved within a few iterations.

Evaluation

In comparison to the work of [Liu et al., 2000], this work is quite similar: every single possibility of feature combinations is analysed to assess interesting combinations of rule features which the user has to choose from. The system then uses these choices to generate one best candidate. Credit has to be given for the simple user interaction, which can be performed by the end user and does not require a domain expert. Since *only one rule is presented and two choices have to be made*, the system rates high in usability. The intuitiveness should therefore be kept in mind. Nevertheless, drawbacks exist: even if their algorithm succeeds at eliminating half of the rules from an association rule database, there will still be a large chunk of remaining rules that would have to be examined. Furthermore and much worse, rules which have been eliminated once will never be shown as a follow-up, not even if the user's knowledge or expectations change in the future, therefore possibly interesting knowledge may be deleted. As the title of this approach suggests, it is a subjective interestingness measure since the user has to specify what he deems uninteresting.

3.2.6 Surprisingness

[Freitas, 1998] presents a series of additions and generic ideas on the concept of rule surprisingness. He agrees with the general assumption that surprisingness inherently has to be a subjective measure of interestingness ([Liu et al., 1997]). Therefore he aims to complement the existing surprisingness measures by ideas that should be considered in the design of an objective interestingness measure.

His first aspect considers the *surprisingness of small disjuncts*. A disjunct is the set of items of an association rule. He considers rules with an antecedent whose number of items m is relatively small. It is assumed that a small disjunct is considered as surprising knowledge if it predicts a different consequent than its minimum generalisations. Minimum generalisations[1] are generated by leaving out one of the items of the rule. This process is repeated for each of the m items and produces m new disjuncts, each covering a superset of the items covered by the original disjunct. The class distribution of the items covered by the new disjunct can be significantly different from the class distribution of the items covered by the original disjunct. Let C be the consequent predicted by the original disjunct. C_k is the consequent predicted by the k-th minimum generalisation of the original disjunct ($k = 1...m$). C is compared against each C_k and the number of times C differs from C_k is counted. This integer number is defined as the raw surprisingness of the original disjunct. An exemplary calculation for a simple association rule can be found in Figure 3.2. Since this measure is significantly dependent on the number of items in the disjunct, it can naturally be normalised by dividing it by m. It is to be noted that the measure is still inherently biased towards rules with fewer items in the consequent and that the applicability of this measure depends on the application domain.

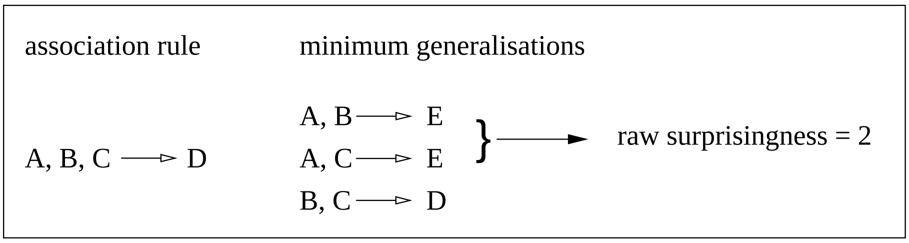


Figure 3.2: Example for the calculation of raw surprisingness

The author's second definition presents the concept of an *attribute's information gain*. It is based on the observation that other interestingness measures only capture the antecedent as a whole, instead of acquiring interestingness on a per-attribute basis. This, in turn, is derived from the observation that attributes, when considered individually, are quite irrelevant for classification and therefore uninteresting. Nevertheless, attribute interactions can turn an otherwise irrelevant attribute into a relevant one. This phenomenon is intuitively associated with rule surprisingness.

[1] Simplified here because of the non-existence of continuous attributes in the association rule sets covered by this thesis.

Given the notations as follows: the goal attribute G (the attribute-value combination in each rule's consequent); $\mathrm{Info(G|A_i)}$ is the information of the goal attribute G given predicting attribute A_i; A_{ij} denotes the j-th value of attribute A_i; G_j denotes the j-th value of the goal attribute G; $Pr(x)$ denotes the probability of X; $Pr(X|Y)$ the conditional probability of X given Y; logarithms are in base 2; index j varies from $1 \dots n$ with n the number of goal attribute values; index k varies from $1 \dots m$ with m the number of values of the predicting attribute A_i. The information gain of each predicting attribute in the rule antecedent, $\mathrm{InfoGainA_i}$, is now calculated as follows:

$$\mathrm{InfoGain(A_i) = Info(G) - Info(G|A_i)}, \tag{3.2}$$

where

$$\mathrm{Info(G)} = -\sum_{\mathrm{j=1}}^{\mathrm{n}} \mathrm{Pr(G_j)\log Pr(G_j)}, \tag{3.3}$$

and

$$\mathrm{Info(G|A_i)} = \sum_{\mathrm{k=1}}^{\mathrm{m}} \mathrm{Pr(A_{ik})}(-\sum_{\mathrm{j=1}}^{\mathrm{n}} \mathrm{Pr(G_j|A_{ik})\log Pr(G_j|A_{ik})}) \tag{3.4}$$

It is argued that rules containing attributes with low information gain are more surprising than rules containing attributes with high information gain, all other variables equal. The surprisingness of an attribute AttSurp is defined as follows:

$$\mathrm{AttSurp} = \frac{1}{\sum_{\mathrm{i=1}}^{\mathrm{\#att}} \frac{\mathrm{InfoGain(A_i)}}{\mathrm{\#att}}} \tag{3.5}$$

where #att is the number of attributes occurring in the rule antecedent. The larger the value of AttSurp, the more surprising is the rule.

Evaluation

The author of this work aims to complement the (necessarily) subjective measures of interestingness by his work. He adds suggestions that should be considered in the design of objective measures. Numerous other approaches towards objective interestingness use antecedent and consequent as a whole, whereas this work splits an association rule into its items. His ideas do not require any user input apart from the association rules, therefore the presented work classifies as an objective measure of interestingness. The author himself states that *interestingness requires subjectivity.* The presented objective measures can only complement subjective measures, so they will not be used directly in this thesis. The inspection of small disjuncts or an attribute's information gain might be insufficient or unsuitable, but the basic idea to *examine each item separately* should be exploited.

3.2.7 Actionability

The last approach to be reviewed for the symbolic part of association rules is based on the work of [Piatetsky-Shapiro and Matheus] and contains the aspect of *actionability.*

In [Silberschatz and Tuzhilin, 1995], a pattern is classified as interesting if the user can do something about it, that is, he can use it to his advantage. The measure is rated as an important subjective measure of interestingness because users are mostly interested in the precise knowledge that permits them to take action in response to the newly discovered knowledge. What is more interesting, unexpectedness and actionability seem to be unrelated to each other since some patterns can be actionable *and* unexpected at the same time, whereas some actionable patterns can still be expected and some unactionable patterns can be unexpected. However, it can be expected that the majority of unexpected patterns are actionable.

Subsequent work of the authors ([Silberschatz and Tuzhilin, 1996]) identifies actionability as a good approximation for unexpectedness and unexpectedness as a good approximation for actionability. Actionability is assumed to be the key concept that is of main interest to business people. Unfortunately, it is hard to grasp formally for the following reasons: first, the space of all patterns would have to be partitioned into a finite set of equivalence classes which would have to have an action assigned to them. This task would probably be infeasible since in many cases the space of all patterns is unknown. Furthermore, the partitioning would probably be complicated. Secondly, even if the first task succeeded, actions and the mapping of actions to patterns may often change over time and would have to be reassigned manually. Therefore it is quite difficult to capture actionability formally. However, since this concept is related to unexpectedness it can be defined by addressing interestingness via unexpectedness. To assess unexpectedness, beliefs will be defined first via belief systems; they are split into 'hard beliefs' which are consistent over time and 'soft beliefs' that may change when new knowledge is acquired. A Bayesian, Frequency and Statistical belief model are shown, among others, and a comparison of their suitability to defining soft beliefs is presented.

Evaluation

Stripping this work from the formal, mathematical definitions which have been given, it is quite interesting to see how concepts relate to each other. Some ideas by other researchers build upon this work (especially in Sections 3.2.2 and 3.2.3) Nevertheless, the concept of actionability seems to be expressible via different terms. It is a highly subjective descriptor of interestingness. From my point of view, actionability is situated at the top of a hierarchy of interestingness measures: below it, at the same level, there are the concepts of surprisingness, contradiction, and unexpectedness. Furthermore, as it has been presented here, it represents yet another notion of unexpectedness. Again unexpectedness requires a set of beliefs that has to be defined by an expert user who is skilled in handling the chosen belief system. Therefore this approach is unsuitable to be used directly in my work.

3.2.8 Rule Change Mining

In [Böttcher, 2005], a framework for *rule change mining* is proposed. It deals with the discovery of useful, interesting, and interpretable patterns in the change of support and confidence of association rules over time. Here, the approach consists of a three-layer framework, which covers the following consecutive steps:

(1) In the mining layer, association rules are discovered from timestamped datasets. In a straightforward approach, database snapshots are taken, and the respective rule mining algorithm is applied to each snapshot. The discovered association rules are stored, along with their confidence and support histories.

(2) In the detection layer, change patterns in the histories are discovered; After applying noise reduction techniques to the rule histories, *trend* and *stability* detection methods are applied. A trend is present, e.g., if steady upward growth or a downward decline are shown when considering a sequence of those values. On the other hand, stability is roughly achieved if the values' mean level and variance are constant over time and in addition the variance reasonably small. For trend detection the Mann-Kendall test [Mann, 1945] and the Cox-Stuart test [Cox and Stuart, 1955] have been evaluated, for stability detection the χ^2 test has been used. Detected trends or stabilities are stored with each rule, which adds valuable feature-like information to the rules.

(3) In the evaluation layer, post-processing is applied that aims to support the user in the identification of the most interesting patterns, and therefore – since histories are attached to a certain association rule – the most interesting association rules. The interestingness of change patterns is assessed by a collection of measures, which, in turn, are used to generate an *interestingness ranking* for the association rules. The interestingness measures for trend histories are described below. For stable histories, the author states that 'a stable history itself has very few significant properties' and that it is somewhat consistent with the assumption about the stability of the considered domain. It can be summarised by the mean of its values, which can be used as an objective interestingness measure.

For trend histories, the author clearly states the predicament between subjective and objective measures of interestingness. Subjective measures would be desirable, but it is questionable if the user has enough expertise and time to specify those trend descriptions that he is interested in. Most trends have complex shapes that are difficult to express formally and would make the task too complicated for a user. The opposite extreme would be the simplistic approach to have the user specify his assumption about the existence of an upward or downward trend. This would condense the complex shape of a trend into a Boolean property, which discards valuable details of a trend. For objective measures, on the other hand, there are a large number of trend properties available due to the numerical nature of a time series.

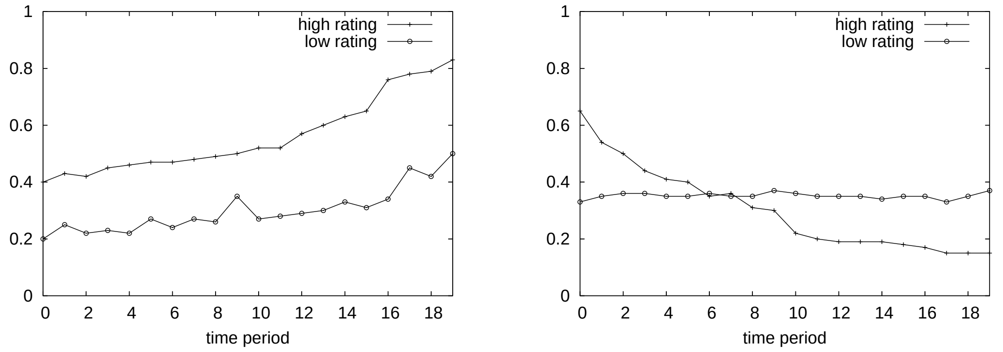
It is noted that only trends which exhibit a salient feature will gain the user's attention. Nevertheless, it is hard to state whether a feature is salient without a reference point. A user's naive assumptions can provide this point of reference: histories with a trend become more interesting with increasing inconsistency between its features and the user's assumptions. Three of those assumptions are noted, and the following interestingness measures for trends are described:

Clarity This metric assesses the certainty that a detected trend indeed exists. Maximum clarity for an upward trend is achieved if each value of the time series is greater than its predecessor; for a downward trend it is achieved if each value is smaller than its predecessor. Figure 3.3(a) shows an example for this measure.

Pronouncedness This describes the deviation of a trend from stability over the complete time series, where stability can be understood as the mean line. The larger the deviation, the higher the trend's pronouncedness is rated. Figure 3.3(b) gives an example of a high- and a low-rated trend according to this measure.

Dynamic A trend's dynamic represents the rate of its incline or decline. It measures the change rate for the n most recent values of a history. A regression line is fitted to those n values; the larger the slope of the regression line, the higher the trend history rates in the dynamic measure. Figure 3.4(a) shows two different trends; if observed over the last six time periods only, the upper curve exhibits a stronger dynamic.

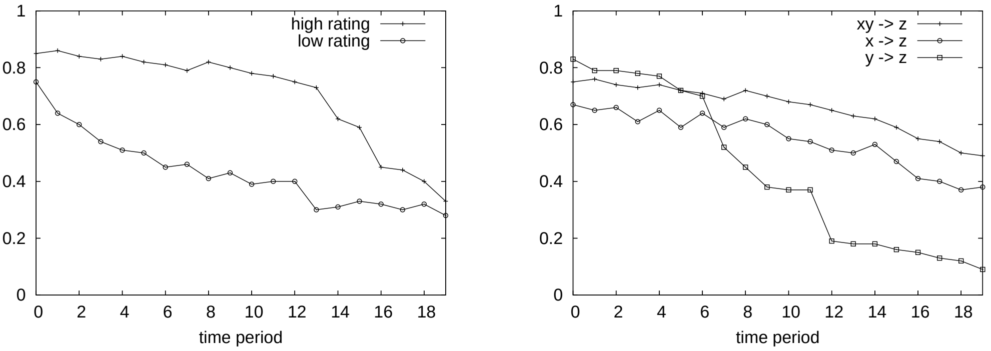
Homogeneity The homogeneity measure targets the assumption that a user is likely to be interested in those subpopulations that behave differently from the population to which they belong. Here, the support and confidence histories of an association rule should be compared to the histories of each more general rule: the more inconsistencies with those histories are discovered, the more the homogeneity assumption is violated and the higher the interestingness of the history (and the associated rule) is. Figure 3.4(b) illustrates this relationship between a rule $x, y \Rightarrow z$ and its more general rules $x \Rightarrow z$ and $y \Rightarrow z$.



(a) Clarity

(b) Pronouncedness

Figure 3.3: Interestingness measures for trends (1)



(a) Dynamic over last six periods

(b) Homogeneity

Figure 3.4: Interestingness measures for trends (2)

Evaluation

As one of the few approaches that address the interestingness of association rules by considering their histories, [Böttcher, 2005] describes a complete framework that handles the mining of the rules and their time series, the detection of trends and stabilities, and the evaluation of the interesting change patterns among those which were detected. He takes the user's assumptions into account during the psychological considerations that lead to the choice of four interestingness measures for a rule's time series: clarity, pronouncedness, dynamic, and homogeneity of its time series. Those values can be used to generate a new ranking on the rules, where the user can choose according to which interestingness measure the rules shall be sorted. The association rules which are dealt with consist of the same two parts that are given in this thesis. His assessment of an association rule's interestingness is entirely based on the interestingness of the associated histories. However, this is the major drawback: both the symbolic and the time series part of an association rule carry valuable information that should be exploited in the analysis of a rule's interestingness. Nevertheless, the basic idea to *generate a new ranking* on the association rules – by means of an interestingness score – shall be utilised.

3.3 Requirements Analysis

3.3.1 Literature Summary

Important aspects from the literature that are worth further consideration have been pointed out in the respective evaluation sections of the presented approaches. The reviewed work is summarised in Table 3.1. The table also shows the categorisation into subjective or objective measure of interestingness. Except for the approach in Section 3.2.7, each of the reviewed papers brings forward one or more aspects that lead directly to requirements for the interestingness assessment of association rules.

Section	*Work*	*subjective*	*objective*	*Summary of important aspects*
3.2.1	[Klemettinen et al., 1994]	x		filtering redundant or common knowledge
3.2.2	[Padmanabhan and Tuzhilin]	x		set of beliefs as background knowledge, contradiction of this set of beliefs
3.2.3	[Liu et al., 2000]	x		unexpectedness based on syntax, acquisition of human knowledge
3.2.4	[Liu et al., 1999]		x	exceptions to a set of strong patterns
3.2.5	[Sahar, 1999]	x		user classifies rules, best candidate is identified, uninteresting rules are deleted
3.2.6	[Freitas, 1998]		x	separate examination of items, splitting of antecedent
3.2.7	[Silberschatz and Tuzhilin]	x		very high-level concept, highly subjective; not directly usable; expert user required
3.2.8	[Böttcher, 2005]		x	interestingness of rules via the trends of their histories

Table 3.1: Literature Summary

3.3.2 Some Remarks

According to the Oxford English Dictionary, *measure* is defined as ‘a standard unit used to express size, amount, or degree’. Therefore, any measure of interestingness defines its own standard, based on the application scenario and the context of the considered rules. Furthermore, an important distinction between *objective* and *subjective* interestingness has been agreed upon, e.g., in [Hilderman and Hamilton, 2003]. Unfortunately, interestingness is user-dependent as it strongly relies on the user’s beliefs, his knowledge and his experience. It is also agreed upon that objective measures of interestingness do not seem to correctly capture this highly psychological notation. On the other hand, the presented subjective measures are domain- and application-specific, adapted to the needs of the particular user. In conclusion, systems that handle post-processing of association rules would have to be built from scratch for each application. As in the presented approaches, the generation of such a system is time-consuming and error-prone. This effort should be reduced, which this thesis aims to achieve. A generic relevance feedback framework will be proposed that integrates ideas from the objective and the subjective aspects, taking the user’s expectations into account.

Another important aspect is that the evaluated approaches consider the user’s expectations as static. It is assumed that the user has a fixed interestingness concept in mind – which is then formalised by the different approaches. From my point of view, this only captures a snapshot of the human expectation. An improved framework should account for the constant change of a user’s knowledge and assumptions while he is exploring the presented association rules and interacting with the system.

The concept of *relevance* is strongly related to interestingness, as mentioned in, e.g., the work of [Freitas, 1998]. In most cases, one can say that an association rule is relevant if it is interesting and vice versa; the same holds for non-relevance, respectively. Once this connection has been drawn, ideas from the *relevance feedback* world are likely to be associated with it. In particular, the possibility of simple user interaction to acquire a knowledge base will be exploited in a subsequent step of this thesis.

3.3.3 List of Requirements

In the following, the requirements of an interestingness assessment system for association rules are listed. They originate directly from either the literature or the business task that this thesis aims to solve.

A definition of interestingness Every single approach in the literature review gives its specialised definition of interestingness, which is mostly adapted to the particular application scenario. In contrast, this thesis aims to provide a rather generic definition, which will make use of the association rules' symbolic and time series features. The definition should not be adapted to the specifics of survey data, although it may be illustrated using the fictitious survey data from the exemplary rule set shown on page 9.

Avoid explicit specification of expert knowledge Although expert knowledge is invaluable to any data mining or knowledge discovery process, the user should not be expected to be an expert. Differently from the approaches in Sections 3.2.2, 3.2.3, 3.2.7, and partly 3.2.4, he should therefore not be treated as one by the system. Knowledge specification languages should be avoided or left to said experts. Instead, knowledge should be acquired implicitly by monitoring the user and his behaviour, such as in Section 3.2.5. This will also keep the approach more flexible. This requirement is related to the user interface, which must also account for an expert and an average user.

An advanced filter The existence of a filter for association rules can be seen as a classical dichotomy. On the one hand, there is the expert user: he is a definite professional in the domain that the association rules have been mined from, and he is perfectly familiar with the association rules that he is handling. Such an expert should be provided with some kind of advanced, semi-automatic filtering to possibly eliminate rules, attributes or items which he knows will not be of interest to him. From a corporation's point of view, this approach should be implemented and reserved for an expert user since it can discard many interesting rules. On the other hand, an average user should not have the possibility to remove association rules before any further rule processing is applied. This complies with the fact that during the knowledge discovery process, expectations and directions are very likely to change, therefore association rules should not be discarded per se.

An implementation of a rule filter would roughly be based on the rule templates approach (Section 3.2.1). It is a pre-filtering step and should be applied before any other processing steps are taken, since it reduces the size of the rule set. The user interface of this filter will be mentioned shortly in Chapter 7.

An inclusion of the time series Although there is relatively few literature available that deals with the histories of association rules, they contain valuable information and should be included in the interestingness assessment. In contrast to the work in Section 3.2.8, the time series should not be treated separately from the symbolic representation, but should instead be integrated, possibly in a preprocessing stage.

A ranking system It has been noted earlier that association rules should not be discarded from the rule set (except by an expert user). A possible and widely used solution (e.g. in Section 3.2.8) to this is a ranking or scoring subsystem. This would fulfill the task of marking the currently interesting association rules, by sorting the rules according to their interestingness score.

A knowledge base via relevance feedback Since the system should not be limited to an expert user, the acquisition of knowledge by the system should be done implicitly. Furthermore, in Section 3.2.6 it was mentioned that interestingness inherently requires subjectivity. As in Section 3.2.5, a possible way to do this is to have the user select relevant or non-relevant[2] association rules. The system would use this feedback internally to adapt its ranking accordingly. This would also account for the changing interestingness assumptions of the user while he is exploring the rules.

An easy-to-use interface Finally, the above requirements should be captured by a system that can be controlled easily via a graphical user interface. This should be intuitive or at least quickly to learn. The user's choices should be simple, but not limiting. Different user skill levels should be accounted for, such as the expert/average user differentiation that has been mentioned above.

3.4 System Layout

A framework that covers the above requirements is shown in Figure 3.5. The numbers in the following description correspond the the respective numbers in the relevance feedback framework overview.

(1) Rule database The system starts from the database that contains the association rules and their time series.

(2) Rule filter Optionally, the rules can be filtered by an expert, based on their attributes and/or items.

[2] The linguistic connotation of "irrelevant" implies a certain finality of a decision. Since a user is likely to change his mind during a relevance feedback process, the term "non-relevant" will be used instead throughout this thesis.

(3) Interestingness definition The parameterised interestingness definition will be based on rule similarity, which requires preprocessing of the association rules.

(4) Preprocessing The preprocessing steps transform the association rules into an internal representation that allows these similarity calculations. Furthermore, the time series information is included in this internal representation. A novel connection to information retrieval will be established in the next chapter, which the preprocessing will be based on.

(5) Relevance feedback The relevance feedback engine represents a cycle that consists of three steps: Firstly, the ranking algorithm is applied to the association rules. Based on the interestingness definition and the knowledge base, a score is assigned to each rule. Secondly, the rules can be sorted by this "interestingness score" and shown in the system's user interface. The user can select relevant and/or non-relevant association rules. Third, the system collects this relevance information to construct a knowledge base. The cycle is restarted, with an updated knowledge base, and possibly different parameters. The user can stop at any point and examine the knowledge that he has collected.

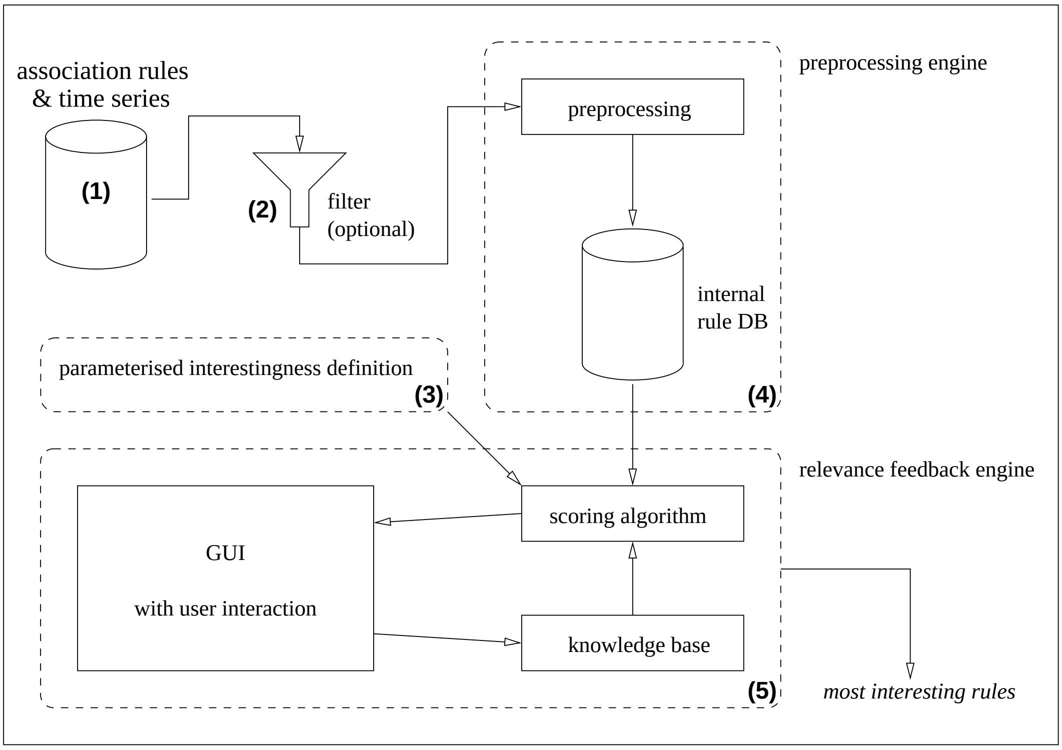


Figure 3.5: Relevance Feedback Framework Overview

3.5 Similarity-Based Interestingness

The different notions of interestingness that have been presented in Section 3.2 have in common that some knowledge has to be provided or is collected. It is then used to identify rules which are interesting *with respect to this knowledge*. Moreover, a common notion of similarity between rules is assumed to be applicable, be it in syntactic parts of rules or complete rules. Consider that most of the reviewed interestingness measures can be broken down to similarity or dissimilarity of parts of or whole rules:

(i) a rule is unexpected if it is very dissimilar to existing knowledge (Section 3.2.3);

(ii) a rule is an exception if it is very dissimilar to or "far away" from common knowledge (Section 3.2.4);

(iii) a rule is surprising in regard to an existing rule if a different (or dissimilar) consequent follows from the same (or similar) antecedent (Section 3.2.6);

(iv) a rule is contradicting in regard to an existing rule if the negation (or strong dissimilarity) of the consequent is yielded by a similar antecedent (Section 3.2.2);

(v) a rule is interesting with regard to its time series if its history is inconsistent (i.e., dissimilar) with the history of a more special rule (Section 3.2.8).

Based on these observations, the issue of how to define similarity on parts of association rules and their temporal features arises. Similarity should be captured by a mathematical model. The choice of this model influences the similarity measure which and will be discussed in Chapter 4. The necessary preprocessing will be discussed in Chapter 5. For now, I assume that rule similarity can be calculated easily. With this in mind, the extended definition of interestingness based on the similarities between an association rule's parts is shown below. As mentioned before (see Sections 2.1.2 and 2.1.3), in the given application scenario, an association rule consists of two different components: the "rule" itself, consisting of items; and the time series for the numeric properties like confidence and support over multiple periods of time. The first part can be split up into antecedent and consequent for separate treatment. According to these rule features, six criteria for finding interesting rules against a knowledge base can be identified in Table 3.2. Each of those criteria will later contribute to an aggregated *interestingness score* for each rule.

The numbers in the table correspond to the interesting combinations listed below. This table reads as follows: A rule is interesting in regard to a rule from the knowledge base if it has the following properties:

(1) similar antecedent but different consequent If rules exist that have similar conditions but result in a different consequence, this would extend the user's knowledge about the data. For example, if he selected the rule $A, B \to C$ and there exists a rule $A, B \to D$ (with different confidence/support), this could explain business changes in the past, e.g., when the latter rule's confidence is rising and the first one's is falling. In the exemplary rule set shown in Table

dissimilar *similar*	consequent	antecedent	time series	rule
consequent	-	(4)	(5)	-
antecedent	(1)	-	(6)	-
time series	-	-	-	(2)
rule	-	-	(3)	-

Table 3.2: Interestingness Matrix

2.1, given rule #1 as knowledge, rule #5 would be found (see below). If the telecommunications provider wants to push a certain technology into the market (here: ADSL), he might be interested in the fact that a certain age group (here: 36-50) is not likely to accept this technology and would probably have to be targeted more specifically.

	rule symbolic		*rule trend*	
#	body/antecedent	head/consequent	support	confidence
1	TEC=ADSL	WCC=YES	stable	stable

5	TEC=ADSL, AGE=36-50	WCC=NO	stable	stable

(2) similar time series - dissimilar rule Dissimilar rules having similar time series' features could be related to business decisions in the past, although being much less obvious or expected than the ones selected. They should be reported, to give the user indications about possible relationships in the data. As an example, given rule #2 as knowledge, rules #6,7,8 would be of interest as well since they exhibit the same trend. This is a broader criterion since it can lead to large numbers of rules. It should be used in combination with other criteria.

2	SEX=M, TEC=WIMAX	VOL=FLAT	up	up

6	AGE=51-65, VOL=10GB	SAT=VSAT	up	up
7	AGE=66+	VOL=10GB	up	up
8	AGE=18-35, VOL=FLAT	SAT=VSAT	up	up

(3) similar rule - dissimilar time series Rules whose symbolic representation is similar but whose time series are dissimilar are expected to be of interest. Example: $A, B \rightarrow C$ with an upward trend in confidence and $A \rightarrow C$ with a downward trend in confidence. Again using the exemplary rule set, given rule #3 as knowledge, rule #7 is of high interest, which is quite intuitive here: the customer group is shifting its tariff from VOL=NONE to VOL=10GB. This could enable business actions to support or counteract this migration.

3	AGE=66+	VOL=NONE	down	down

7	AGE=66+	VOL=10GB	up	up

(4) same consequent - dissimilar antecedent This heuristic is more about furthering the user's knowledge: If he selected a rule $A, B \rightarrow C$ and there are other rules $D, E \rightarrow C$ (having the same consequence), those might reveal previously unknown connections and patterns in the underlying data. In the exemplary rule set, given rule #6, rule #8 would receive a high score for this criterion; in this case the colloquialised question would be to 'find the combinations that lead to high customer satisfaction' which is indeed a highly useful business objective.

6	AGE=51-65, VOL=10GB	SAT=VSAT	up	up

8	AGE=18-35, VOL=FLAT	SAT=VSAT	up	up

(5) same consequent - dissimilar time series This combination is a special case of the third combination above and might be encountered less frequently in practice. Rules having the same consequence but with dissimilar time series could indicate changes in the business' directive in the past and possibly predict the outcome of business decisions: A selected rule $A, B \rightarrow C$ with an upward support trend and a found rule $D \rightarrow C$ with very little support might be a combination worth investigating since the found rule could become important after business changes. From Table 2.1, given rule #9 as input knowledge, rules #1 and 4 would be found. Rule #9 might show that a business decision to market a new UMTS broadband tariff with 10GB leads to a decline in the 1GB variant, but has no influence on the ADSL market, expressed in a higher score of rule #1.

9	TEC=UMTS, VOL=1GB	WCC=YES	down	down

1	TEC=ADSL	WCC=YES	stable	stable
4	SEX=F, AGE=18-35, ESQ=BAD	WCC=YES	up	down

(6) similar antecedent - dissimilar time series This condition is another special case of the third combination of rule features and might also be encountered less frequently in practice. Rules with similar antecedent, but dissimilar time series might be interesting since they could contradict the user's expectations. For example, if he selected the rule $A, B, C \rightarrow D$ and the rule's confidence has an upward trend then he would probably expect the rule $A, B \rightarrow E$ to have an upward trend in terms of confidence as well. This approach is also related to the concepts of generalisation and specialisation of association rules. Starting from rule #10 as knowledge, rules #3 and 7 could be considered relevant to the user. Here, the target could be the customer age group (66+), whose consumption patterns are to be examined.

10	AGE=66+, TEC=CABLE	SAT=SAT	stable	stable

3	AGE=66+	VOL=NONE	down	down
7	AGE=66+	VOL=10GB	up	up

3.6 Summary

At the end of this chapter, *interestingness* has been defined, resolving the first out of three key issues identified in Section 1.2. A literature review has been conducted, and the approaches to assess the interestingness of association rules have been evaluated. Certain key ideas that appeared throughout the literature led directly to a list of requirements. A system layout that captures the requirements has been proposed. It is based on a novel view of similarity-based interestingness. Based on the presented step-by-step interestingness definition, a rule's interestingness can be assessed quite easily once **(a)** the knowledge base has been established that is required to apply similarity calculations and **(b)** of course, the underlying similarity measure has been defined. Similarity measures, in turn, are linked to, among other areas, information retrieval – the following chapter will point out further links to this area.

Chapter 4

A Connection to Information Retrieval

4.1 Introduction

According to [Baeza-Yates and Ribeiro-Neto, 1999], Information Retrieval (IR) deals with the representation, storage, organisation of, and access to information items. The representation and organisation of the information items should provide the user with "easy access to the information in which he is interested". IR consists of a multitude of areas to deal with the information needs of a user. Among those areas, there are surprisingly many ideas that can be applied to the scenario of *finding the most interesting association rules*. In the following, the requirements specified in the preceding chapter will be linked to the respective areas of IR, establishing a novel connection between the hitherto conceptually different areas of association rules and information retrieval. It will be reasoned that association rules can be dealt with effectively using IR methods. An IR model will be chosen according to the specifics of association rules and their histories.

4.2 Linking association rules and information retrieval

4.2.1 Querying

Querying is the most basic operation that IR systems deal with. Historically, queries of an IR system have often been expressed in special query languages, such as SQL (Structured Query Language) for database operations or XQuery for XML (Extended Markup Language) data. The user interaction is based on queries: they are formulated by a user, who must be able to express his information need precisely in the provided query language, to gain access to the information he is interested in. IR systems regularly deal with vast amounts of data and aim to answer those queries. In the association rule scenario, the basic issue is quite similar: a system holds a large number of association rules with their histories and a user queries the system (implicitly or explicitly) to find those rules that he is interested in. The user must therefore be able to express his information need – an issue that, again, is addressed in IR.

4.2.2 Filtering

The first of the requirements in Section 3.3 is the development of an advanced, syntax-based filter for association rules. If association rules are regarded as combinations of textual items or as *documents*, filtering can be seen as a special text operation that has already been available in IR for quite a long time. Document preprocessing and text compression are related issues from IR – filtering of association rules can therefore be performed easily using their textual representation. More on the textual rule representation can be found in Section 4.2.7.

4.2.3 Pattern Matching

The interestingness definition in the preceding chapter is based on a notion of similarity between association rules, their histories, or parts thereof. The similarity of two or more patterns, in turn, can be expressed via a degree of match between them: the better they match, the more similar they are. Pattern matching is a very active research area in IR and has recently attracted some attention in regard to association rules, e.g., in [Wang et al., 2003], where a definition of a *degree of match* between association rules is given and used to find unexpected rules in a new rule mining algorithm.

4.2.4 Relevance Feedback

A cornerstone of the framework for assessing similarity-based interestingness devised in Section 3.5 is the acquisition of a knowledge base. Since the final system should not be limited to an expert user who is willing and capable of learning a query specification language, the knowledge base should be constructed at run-time in the background. This can be achieved via the user's explicit relevance feedback about the association rules that have been presented to him. However, relevance feedback is one of the fundamental methods in IR and has widely been used for query adaptation. Its roots are located in the 1970s, with early work of [Rocchio, 1971] and [Elliott and Cashman, 1973]. A more detailed description can be found in Chapter 6.

4.2.5 Scoring and Ranking

During the work with the association rules, none of them should be discarded as ultimately not interesting – a user's background knowledge and his expectations are likely to change as he sees new association rules. Therefore, a scoring-and-ranking approach has been suggested in the requirements in Section 3.3. Association rules shall be assigned an interestingness score (based on the interestingness definition) and can be ranked according to this score. Again, scoring and ranking are well-known topics in IR: most of nowadays' web search engines employ this basic idea to present their results to the user. The web surfer enters a query into a search engine's interface; the search engine processes the query, assigns relevance scores to the web sites and presents the results, ranked by this score, to the user. Although this is highly simplified, it shows that the scoring-and-ranking approach is employed regularly in IR.

4.2.6 User Interface

Since the framework for similarity-based interestingness of association rules shall ultimately be usable for an average end user, the conception and design of a user interface have to be tackled. Unsurprisingly, as IR systems are used by average people as well, the user interface has been well-researched in IR: from simple collection browsing interfaces over dialogs and wizards to the more sophisticated and computationally heavy visualisations and, ultimately, natural language queries, there is a large range of techniques available. The user interface also addresses the issue of a starting point for the search – more about this topic can be found in Chapter 6.

4.2.7 Text Retrieval

The previous sections 4.2.1 to 4.2.6 have established a series of links between association rules and IR. However, IR generally assumes that the data it deals with can be treated to fit into one of its models. In the following it is shown conceptually how association rules can be fitted to the IR requirements, the preprocessing details can be found in Chapter 5.

From a certain perspective, the items of an association rule set are plain text values or *strings*. For example, rule #2 from the exemplary association rule set (Section 2.2):

$$\text{SEX=M, TEC=WIMAX} \Rightarrow \text{VOL=FLAT}$$

If this representation is interpreted as a set of sets of strings, then it may be represented as follows:

$$\{\{\text{SEX=M, TEC=WIMAX}\},\{\text{VOL=FLAT}\}\}$$

Back in the IR area, these textual representations are well-known and have been researched for a long time, e.g., from document retrieval in reply to a user's query. Documents that consist of textual items (words or strings) are preprocessed to suit a predefined model where retrieval operations can be performed easily and with a multitude of options. As the prerequisite for the aforementioned links, this idea is at the very foundation of this thesis. Association rules will be regarded as sets of strings, like text documents, and can therefore be treated in very much the same way as text documents. This is, however, the first such approach, and limitations are likely to exist and will be mentioned, where appropriate.

4.2.8 Conclusion

In the previous sections 4.2.1 to 4.2.7, several links between association rules and the area of IR have been established. Numerous key areas that IR deals with can be applied to association rules and their histories. It is therefore justified to assume that IR models can cope with the task of finding the most interesting association rules. Naturally, preprocessing of the rules will be necessary and depends on the used IR model, which will be chosen in the following.

4.3 Choice of an IR Model

From information retrieval, three classic models are well-known and will be described shortly in the following. These classic models assume that each document can be described by a set of representative keywords, so-called index terms. Considering association rules, those index terms would be the items of the rule's body and head. IR would deal with user queries which are to be specified according to one of the models and, when executed, return an answer set of relevant association rules. In the framework devised in Section 3.5, this query does not have to be formalised explicitly, but will be constructed from the relevance information supplied by the user.

The Boolean Model This model is based on set theory and the Boolean algebra. It considers that an association rule's item is either present in the rule or it is not, marked in a feature representation with 1 or 0, respectively. Queries can be specified as Boolean expressions, which are precise - this is also the model's main drawback. In the standard model that is considered here, there exists no grading scale and no partial match; the degree of similarity between two association rules would be defined as 1, if the rules or the compared parts thereof are the same, and 0 otherwise.

The Probabilistic Model This model assumes that, given a user query, there is an ideal set of association rules that contains the relevant rules. If the description of this set were given, there would be no problems in retrieving these rules. Therefore, the query process can be thought of as an interactive procedure of specifying the properties of this optimal rule set. In a first step, some attributes would have to be guessed. In subsequent steps, the first guess would have to be refined by the user. In each step, the system tries to maximise the overall probability of relevance of the rules in the answer set that is returned by the algorithm.

The Vector Model It recognises that the use of binary weights (as in the Boolean model) limits the possibilites and defines a framework that enables partial matching. It accomplishes this task by assigning non-binary weights to index terms of the association rules. In other words, each item of an association rule's textual representation is seen as a feature. Therefore, a feature vector can be calculated for each association rule. Those feature vectors would enable the computation of a degree of similarity between different rules' components: similarity can be understood as the distance between their respective vectors. The smaller the distance between two feature vectors, the larger their similarity is. Algorithms that implement different distance measures between vectors are readily available.

The probabilistic model can be ruled out, simply because of its inherent assumption that an ideal set of association rules exists that contains the relevant rules. In a relevance feedback framework, the user's assumptions are likely to change as he sees new association rules, therefore the assumption of a static answer set that is known in advance does not hold. Furthermore, due to the large number of mined association rules, a first guess of the distribution between a small set of relevant rules and an overwhelmingly large set of non-relevant rules would be hard to obtain. As a side issue, the integration of the rules' histories would be hard to perform.

The Boolean model, though conceptually better suited to treat association rules due to a *feature vector representation*, has the major drawback of the binary 'similarity value' that can be computed. The model would, in its standard form, not be much more sophisticated than the rule templates approach from Section 3.2.1. The rule histories would have to be integrated using a different representation, except if the change patterns were used – a strong and lossy simplification. The Boolean model will therefore not be used on association rules.

The vector model also employs the idea of a feature vector, albeit with the important difference that the components are not limited to binary values, but that index term weights can be used. These term weights can be used to compute the *degree of similarity* between association rules' components. The computed similarity value can be processed further to lead to an interestingness score, as suggested in Section 3.5. Furthermore, the interestingness definition from the last chapter requires the integration of the time series information into the knowledge discovery process. Feature vectors consist of a set of numerical weights that represent the association rule, which makes the time series easy to incorporate. For those reasons, the vector model will be used – association rules will have to be preprocessed into a feature vector representation with index term weights as the vector components.

Based on those standard models, there are more sophisticated set-theoretic, algebraic, or probabilistic models available, but those are outside the scope of this work and can be used during future improvements.

4.4 Summary

In this chapter, the *novel connection* between association rules and information retrieval has been established and substantiated. Requirements from the previous chapter have been linked to the respective areas in IR that deal with the same or similar issues. It is therefore justified to apply key methods from IR to the association rules. The feature vector model has been chosen from the three standard IR models, since it is expected to cope best with the specifics of association rules and their histories. However, preprocessing is necessary and will be explained in the following chapter.

Chapter 5

Preprocessing

In the preceding chapter, a novel connection between association rules and information retrieval has been established. The feature vector model from IR has been chosen to cope best with the association rules and their time series. The definition of interestingness given in Section 3.5 is based on the assumption that a similarity calculation between components of association rules can be performed. This chapter will deal with the preprocessing steps that will enable those similarity calculations. Association rules and their time series will be transformed into a feature vector representation using text retrieval methods from IR. A detailed, formal description of the preprocessed data will be given.

5.1 From Association Rules to Rule Feature Vectors

The similarity-based interestingness definition in the preceding chapter is based on similarities between four rule components: body, head, their combination, and a rule's time series. A *feature vector $\vec{r}$ of an association rule* r will therefore contain these four basic components, see Equation 5.1. The remainder of this chapter will discuss the components of this vector.

$$\vec{r} = (\underbrace{\overbrace{r_1, \ldots, r_b}^{\text{body}}, \overbrace{r_{b+1}, \ldots, r_{b+h}}^{\text{head}}}_{\text{symbolic}}, \underbrace{r_{b+h+1}, \ldots, r_{b+h+t}}_{\text{timeseries}}) \tag{5.1}$$

The different components can be seen as a projection of $\vec{r}$ and will be referred to as follows:

$$\vec{r}_{\text{body}} = (r_1, \ldots, r_b) \tag{5.2}$$

$$\vec{r}_{\text{head}} = (r_{b+1}, \ldots, r_{b+h}) \tag{5.3}$$

$$\vec{r}_{\text{sym}} = (r_1, \ldots, r_{b+h}) \tag{5.4}$$

$$\vec{r}_{\text{time}} = (r_{b+h+1}, \ldots, r_{b+h+t}) \tag{5.5}$$

5.1.1 Calculation of Item Weights

Since I am considering the association rules as text documents, the notations and conventions regarding text retrieval in IR will be applied here as well. Standard term weighting approaches have been summarised in [Salton and Buckley, 1987]. Similar to the TF-IDF approach shown there, term weights will be calculated here. The *document terms* are the association rules' *items*. The TF-IDF approach weights terms according to their appearance in a document and in the overall document collection. A high term weight, which is correlated with a high importance of that particular term, is achieved if the term appears frequently in the document (term frequency, TF) but much less frequently in the document collection (inverse document frequency, IDF). This approach filters out commonly used terms and tries to capture the perceived relevance of certain terms.

Both parts of the TF-IDF calculation can be applied on association rules. The term frequency is, however, binary: an item is or is not contained in an association rule. Therefore, the TF of an item λ in an association rule r is calculated as follows:

$$tf(\lambda, r) = \begin{cases} 1 & \text{if } \lambda \in r, \\ 0 & \text{otherwise.} \end{cases} \tag{5.6}$$

The inverse document frequency can be adapted to association rules: an item is supposed to have the lowest weight if it is contained in every rule and therefore bears no information. It is supposed to be of the highest weight if it appears in only one rule out of the rule set. Certainly, this approach might suffer from noise in the data, but it has shown to be effective in practice. The inverse document frequency *idf* of an item λ in an association rule r and in regard to a rule set R is now calculated as follows:

$$idf(\lambda, R) = 1 - \frac{\ln |r : r \in R \wedge \lambda \in r|}{\ln |R|} \tag{5.7}$$

Equation 5.7 ensures that item weights are from the interval $[0 \ldots 1]$ and meet the assumption about highest and lowest weights above.

5.1.2 Vector Generation

To give a schematic overview, the preprocessing layout can be obtained from Figure 5.1. It shows the transition from the textual representation of association rules on the one hand, and their time series on the other hand, towards a feature vector representation for each association rule. The figure shows the programmer's point of view and also illustrates, in principle, how the preprocessing has been implemented.

Part 1: Symbolic Representation

Different from some of the approaches from Section 3.2, I do not impose a taxonomy or a hierarchy on the association rules' attributes and items, neither are any (possible) relations between them taken into consideration. I consider every attribute and item

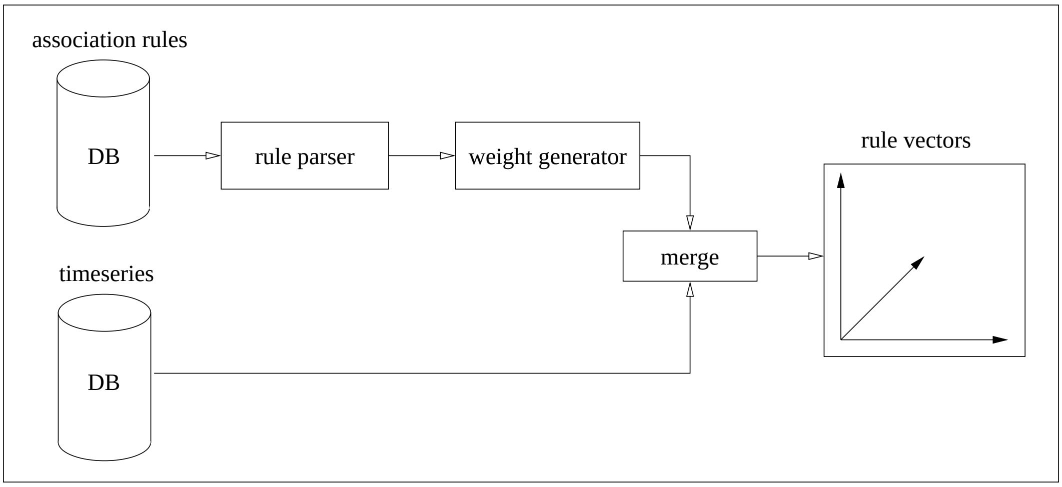


Figure 5.1: Overview of the preprocessing

to be independent of any other attributes and items. In the given application scenario, an attribute's appearance in an association rule's body or head is expected to be binary: it exists or it does not exist. This also means that an attribute can not appear more than once in body or head, respectively.

To generate rule vectors, a series of steps has to be performed. First, the rule set R that consists of association rules $r : \mathcal{X} \Rightarrow \mathcal{Y}$ (see the definition in Section 2.1.2) is parsed. For body and head separately, a set of items is generated as follows:

$$I_{body} = \{\lambda_1, \ldots, \lambda_b\} \tag{5.8}$$

$$I_{head} = \{\lambda_1, \ldots, \lambda_h\} \tag{5.9}$$

where the λ_i are the items that occur in body or head of the association rules in R, respectively. Each item of these sets is assigned exactly one vector dimension in $\vec{r}_{\text{body}}$ or $\vec{r}_{\text{head}}$. Hence, the values for b and h in Equation 5.1 are the cardinalities of the respective itemsets:

$$b = |I_{body}| \tag{5.10}$$

$$h = |I_{head}| \tag{5.11}$$

Referring to the exemplary rule set on page 9, the theoretical maximum dimensions of each rule vector's symbolic part would be 42 dimensions, thereof 21 each for body and head. This number results directly from the possible attribute-value combinations. However, with the low number of association rules given, the numbers of existing items amount to 14 for rule body and 7 for rule head. Therefore, the dimensionality of the rule vector's symbolic part would sum up to 21 dimensions.

The symbolic part of the feature vector of an association rule r will contain TF-IDF values. Let λ_i the i-th item of the alphabetically ordered set I_{body}. Then, the part for the rule's body in the feature vector is filled as follows:

$$r_i = tf(\lambda_i, r) \cdot idf(\lambda_i, R), \quad i = 1, \ldots, b \tag{5.12}$$

$\vec{r}_{\text{head}}$ is treated in the same way, except that λ_j is the j-th item of the alphabetically ordered set I_{head}

$$r_{b+j} = tf(\lambda_j, r) \cdot idf(\lambda_j, R), \quad j = 1, \ldots, h \tag{5.13}$$

The equations above describe the construction of the feature vector representation of the rules' symbolic part. Most of the entries in the feature vector are 0, since association rules only consist of a small number of items of the respective itemset. The entries of a rule's feature vector are different from 0 only if the respective item is contained in the rule. In this case, they correspond to the *idf* of the item in the rule set. In summary, the preprocessing of the symbolic part yields sparsely populated feature vectors, much like the feature vectors of text documents in IR, which was the intended connection between association rules and IR.

Furthermore, if the Boolean retrieval model with its binarisation had been chosen, the feature vectors would be sparsely populated as well. In fact, where a feature vector in the current model would contain an IDF weight, the feature vector of the Boolean model would contain a 1; the rest of the symbolic part of the feature vector would be the same.

Part 2: Rule Time Series

For every rule property such as support, antecedent support, and confidence, the numerical values of each of these properties have been stored (as described in Section 2.1.3). For every period of these values, an additional dimension is introduced into the feature vector. It is assumed that the time series of the association rules in a rule set are homogenous, i.e., of the same length. Otherwise, a similarity operator between feature vectors of different length would have to be defined. If one assumes that the number of time periods is p, the time series part of the feature vector for the association rule r will be filled as follows:

$$\vec{r}_{\text{time}} = (\underbrace{conf_{r,1}, \ldots, conf_{r,p}}_{\text{confidence}}, \underbrace{supp_{r,1}, \ldots, supp_{r,p}}_{\text{support}}, \underbrace{asupp_{r,1}, \ldots, asupp_{r,p}}_{\text{antecedent support}}) \tag{5.14}$$

where $conf_{r,i}$, $supp_{r,i}$, and $asupp_{r,i}$ are the numerical values of confidence, support, or antecedent support of the association rule r at time period i.

In the exemplary rule set, the association rules have been observed and mined at 20 different database snapshots. 20 different values each for support, antecedent support, and confidence would result - altogether 60 additional dimensions to introduce into the rule vector. This part of the rule vector can easily be adapted to contain more time periods or different numerical values.

5.2 Summary

In this chapter, a definition has been given that shows how the rules *and* their time series can be preprocessed into a *combined feature vector representation* (Equation 5.1). It is based on ideas from IR, namely to consider the association rules as text documents and treat them accordingly. The resulting, sparsely populated, feature vectors and parts thereof will be used for similarity calculations. This way, the second key issue of preprocessing that was identified in Section 1.2 could be resolved. However, the preprocessing is an intermediate step to enable the use of relevance feedback, which will solve the third key issue and will be discussed in the next chapter.

Chapter 6

Relevance Feedback

In the preceding chapters, a relevance feedback system layout for association rules has been devised. The interestingness of association rules has been defined, based on a user's relevance judgements. The rules have been preprocessed into a feature vector representation to enable similarity calculations. The last part of the framework shown in Figure 3.5 on page 27 will now be developed in this chapter, solving the third key issue identified in Section 1.2, which was the development of suitable relevance feedback methods. According to the interestingness definition and based on the relevance judgement of the user, an *interestingness score* for each rule will be calculated and updated after each relevance decision. A rule browser will show the list of association rules, ranked by their interestingness score.

6.1 Introduction to Relevance Feedback

Relevance feedback is an intuitive technique that has been introduced in the mid-1960s [Salton, 1971]. It is a controlled, semi-automatic, iterative process for query reformulation, that can greatly improve the usability of an IR system [Jaakkola and Siegelmann, 2001]. The basic idea is quite simple, and it extends the basic query process that is used to interact with an IR system:

(1) To start the relevance feedback cycle, an initial set of documents has to be retrieved that the user can select from. There are basically two possibilities to obtain this set. First, the user formulates an initial query just as he would query nowaday's web search engines. This is an expert-based approach that should be avoided. Second, an initial set is presented using existing rankings or an otherwise existing order on the documents. Data aggregation techniques, such as clustering, could be applied here as well, to give an overview about the documents.

(2) The feedback cycle starts. In consecutive iteration steps, the ranked results of querying the IR system are presented. In each iteration, the result set is expected to be more relevant to the user, since the query is refined each time.

(3) The user examines the results. At least two basic variants of relevance feedback are possible. Firstly, the user specifies explicitly which documents are relevant to him and which are not. This version will be used in the association rule feedback system. Secondly, the feedback is acquired implicitly by monitoring the user and his behavior when he examines the rules; a survey on implicit feedback has been conducted, e.g., by [Kelly and Teevan, 2003]. This solution could be used in a future development stage of this framework. In both cases, a set of relevant and a set of non-relevant documents are returned.

(4) The system uses the two returned sets of documents to reformulate the initial query. The query is executed and the cycle continues with step 2. The user can stop whenever he wishes.

An early implementation that uses the vector space model and modifies the original query has been described by [Rocchio, 1971]. One of the earlier comparisons of different relevance feedback techniques has been provided by [Elliott and Cashman, 1973]. A more recent publication by [Salton and Buckley, 1990] clearly states the three main advantages of using relevance feedback in conjunction with IR queries:

Encapsulation It shields the user from the details of the query formulation process, and permits the construction of useful search statements without intimate knowledge of collection make-up and search environment.

Segmentation It breaks down the search operation into a sequence of small search steps, designed to approach the wanted subject area gradually.

Control It provides a controlled query alteration process designed to emphasise some terms and to deemphasize others, as required in particular search environments.

These three properties make relevance feedback suitable for the purpose of finding interesting association rules. The user does not have to know about the internal calculations of the system nor must he learn yet another query language; and the user can explore and discover his own and new knowledge gradually by altering the query implicitly.

In information retrieval, two concepts that rate the performance of the returned documents are known: first, *precision* is the proportion of retrieved and relevant documents to all the retrieved documents; second, *recall* is the proportion of relevant documents that are retrieved to all relevant documents available. It should be noted that it is desirable to be able to use such ratings on the given relevance feedback system, to compare its performance to other approaches. However, it is hard to obtain a set of “benchmark” association rules and define in advance which rules are relevant and which are not relevant. Therefore, there can be no such objective measures that evaluate its quality.

The generic relevance feedback definition above has to be adapted to the specifics of association rules. Figure 6.1 illustrates this adaptation. The figure also shows the two major parts that this chapter will deal with: first, an initial ranking has to be obtained, and second, the query refinement with the update of the ranking has to be tackled. However, these parts have to be integrated into a user interface, therefore it will be explained first.

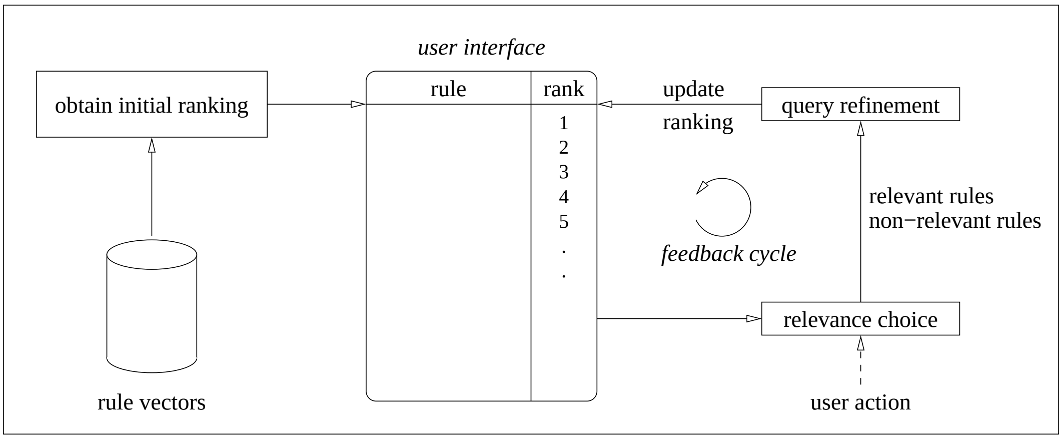


Figure 6.1: Overview of the relevance feedback

6.2 User Interaction

6.2.1 Rule Display

One of the requirements in Section 3.3 was an intuitive user interface. An implementation of an association rule browser was available and could be extended, see Section 7.2 in the following chapter for details. In this chapter an *interestingness score* will be introduced, which will be updated after relevance choices have been made. The rules in the rule browser can then be sorted by this interestingness score.

There are two types of users that should be treated differently: for a beginner, the basic choice steps are sufficient. However, an expert will expect more options that support and enhance his browsing behavior. Different sorting steps, a search for items and attributes, and fine-tuning of the relevance feedback parameters might be required. Those parameters will be developed in the rest of this chapter, whereas the user interface that allows the setting of those parameters will be covered in Chapter 7.

6.2.2 Relevance Acquisition

The user's choices within the relevance feedback cycle are simple: *browse* the association rules, *select* an association rule and *mark* it as either relevant or non-relevant. Whenever a rule has been marked as relevant, it is added to the set of relevant rules R_{rel} . Whenever a rule is marked as non-relevant, it is added to the set of non-relevant rules R_{nrel} . For each added rule, the cycle iteration number in which it has been chosen is remembered.

6.2.3 Score Update

After each selection, the rulesets are passed to the score update algorithm, which updates the interestingness score of each association rule in the complete rule set. The user interface is updated and shows the new scores, which are based on the relevance choices made before – therefore the top-ranked rules are expected to be more interesting in each step.

6.3 Initial Ranking

There are at least two options to generate an initial ranking of the association rules. The first option is a standard rule browser view, that uses existing orders on the association rules to give an overview. The second, more advanced option, is to process the association rules and show an aggregated overview. Depending on the used processing or aggregation techniques, the second version is likely to increase the user's satisfaction, but is also likely to require a different user interface. From my point of view, the simple, yet effective first version suffices as a starting point for relevance feedback, since sorting and search operations are available. Furthermore, once the relevance feedback has been started, an interestingness score for each rule will impose a new order on the association rules. The rule browser view will only require slight modifications to incorporate this interestingness score. Nevertheless, the more advanced version with an adapted user interface could be used at a later stage.

6.3.1 Sorting by Existing Orders

Association rules and their properties already supply a large variety of options to sort them. The three groups are as follows:

Alphabetical order The rules are split up into antecedent and consequent. Those can be sorted alphabetically. A string-based search for specific items or attributes can be conducted.

Numerical order Support, antecedent support, and confidence values can be used to sort the rules accordingly, ascending or descending order included. Furthermore, interestingness measures for trends can be used to sort the rules.

Trend grouping Rules can be aggregated into groups that exhibit certain trends or stability. This grouping is quite coarse, but should be sufficient for an overview.

6.3.2 Feature-Vector-Based Data Processing

Since the association rules are accessible via a feature vector representation, the main prerequisite for a multitude of processing options from information retrieval is accomplished. One of those techniques that could be applied to obtain an initial overview of the association rules is clustering. Since some experiments with rule clustering were conducted, it will be explained in principle in the following.

Clustering

An advanced approach to obtain an initial ranking that can be refined by relevance feedback could be to cluster the association rules. Here, clustering would be used as a grouping step, for the purpose of understandability of the association rules and their relations. If a large number of association rules is to be presented, clustering would help in the data reduction, making the association rules better accessible, especially for an average user.

The cluster hypothesis is that "closely associated documents [here: association rules] tend to be relevant to the same requests" [Rijsbergen, 1979]. Clustering would find closely associated feature vectors and group them together into one cluster. It could be based on any combination of the $\vec{r}_{\text{body}}$, $\vec{r}_{\text{head}}$, $\vec{r}_{\text{sym}}$, $\vec{r}_{\text{time}}$ feature vector parts. If clustering were based on $\vec{r}_{\text{sym}}$, a cluster would contain those rules that describe a group of similar attributes or items; it would then also be related to the concepts of generalisation and specialisation of association rules, as defined in Section 2.1.2. If clustering were based on $\vec{r}_{\text{time}}$, clusters would contain rules that have similar time series.

If a decision to cluster any parts of the association rules were agreed upon, some choices would have to be made. First of all, a clustering algorithm would have to be chosen that fits the sparsely populated feature vectors which are encountered here. An overview of clustering algorithms can be found, e.g., in [Jain et al., 1999]. Secondly, clustering parameters would have to be set, among which is the choice of a similarity measure for distance calculations. Thirdly, the parameter settings would have to be integrated into the user interface of the relevance feedback system.

From a certain point of view, clustering would be a different approach to help in browsing large document [rule] collections, as described in [Cutting et al., 1992] or [Hearst and Pedersen, 1996]. A general evaluation for document clustering in IR has been given in [Leuski, 2001]. The author shows that clustering can be a very effective way to directing a user towards relevant documents. In the given association rule environment, clustering could also be applied in addition to the scoring and re-ranking approach that is shown in this thesis.

Experimental results that were conducted using hierarchical agglomerative clustering on the rule symbolic's feature vectors showed that a good overview of related attributes could be obtained by this grouping step. The large set of association rules could be reduced to a set of a few clusters which could then be examined manually to identify relevant or non-relevant rules, albeit at a high computational cost. Nevertheless, this task exceeded the timeframe of this work and is therefore left for future approaches.

6.4 Interestingness Scoring

The interestingness definition given in Section 3.5 is based on relations between different rule parts: body, head, their combination, and rules' time series. The preprocessing has led to a vector representation for the association rules, to enable similarity calculations. There are some additional issues that have to be resolved, before an interestingness score for each rule can be determined: a similarity measure has to be defined, the knowledge base has to be described, rule sets have to be aggregated for similarity calculations, and more recent relevance choices have to be given a higher weight than former choices. Finally, an interestingness score for each association rule will combine these calculations.

6.4.1 Similarity Measures

The first issue is the choice of a suitable similarity measure. The vector space model provides measures like the Euclidean distance or the Mahalanobis distance. These distance calculations can be used to assess similarity: the closer the vectors are, the more similar the association rules are. The Euclidean distance between two n-dimensional vectors r and s would be calculated as follows:

$$d(\vec{r}, \vec{s}) = \sqrt{\sum_{i=1}^{n} (r_i - s_i)^2} \tag{6.1}$$

However, in IR the most commonly used measure for assessing the similarity is the Cosine similarity. It calculates the angle between two n-dimensional vectors r and s as follows:

$$sim(\vec{r}, \vec{s}) = \frac{\sum_{i=1}^{n} r_i s_i}{\sqrt{r_i^2}\sqrt{s_i^2}} \tag{6.2}$$

The latter Cosine measure has been used successfully in a large number of IR applications, especially in text retrieval. The association rules have been considered as text documents and preprocessed as such, leading to sparsely populated feature vectors. Since this representation is similar to the one of text documents in IR, it is assumed that Cosine similarity can be used to assess the similarity between rule vectors or parts thereof. Moreover, experimental results on association rules from the survey data set in [Böttcher, 2005] exhibited the expected behaviour when using the Cosine measure: rules with similar items or similar time series received a higher similarity, when using the appropriate feature vectors for the calculation. Equation 6.2 yields values in $[0, 1]$. The dissimilarity of rules can be expressed via the similarity as follows:

$$dissim(\vec{r}, \vec{s}) = 1 - sim(\vec{r}, \vec{s}) \tag{6.3}$$

6.4.2 Knowledge Base

The second issue concerns the specification of a knowledge base. Here, the system's knowledge base is synonymous to the user's interests. Those are expressed via his relevance choices. The system collects the user's choices into the sets R_{rel} and R_{nrel}. The user's assumptions and his interests are likely to change over the course of the relevance feedback. To account for this, the system also stores the time step at which a certain relevance choice has been made. This information will be used to influence the scoring accordingly, which will be described in Section 6.4.4. In short, more recent choices will have a larger impact on the interestingness score.

6.4.3 Rule Set Aggregation

The third issue arises in the similarity calculation: the knowledge base consists of *rule sets*, but a similarity measure has only been defined between single rule vectors or parts thereof (Equation 6.2). Therefore, a suitable aggregation of R_{rel} and R_{nrel} has to be found. In the literature, similar problems of "score aggregation" between vectors have been encountered (e.g., [Fagin and Wimmers, 2000], [Singitham et al., 2004]). They could usually be solved by calculating every single similarity and aggregating those values accordingly. If one adopts this procedure, the most basic approach to aggregate the similarities would be to average them to obtain one similarity value.

An Example

However, the user's assumptions could be different: in one application scenario, he might want to prefer finding more similar rules; in a different application scenario, he might want to prefer finding dissimilar rules, to the ones he has already marked as relevant (or non-relevant), respectively. Consider the following example from the exemplary rule set: from Table 6.1, the user has marked rules #1, 3, 5, and 7 as relevant. It is assumed that he is interested in rule #10, based on the antecedent of the rules, so this rule should receive a high interestingness score.

The similarities between the rule's bodies are computed and aggregated. If an averaging were performed, rule #10 would receive an intermediate score (Figure 6.2a). If the higher similarities between rules #3 and 7 and rule #10 were emphasised, rule #10 would receive a high score (Figure 6.2b). Finally, if the lower similarities between #1,5 and #10 were emphasised, rule #10 would receive a low score (Figure 6.2c), all other parameters equal. An aggregation operator should not only account for at least these different user assumptions, but should also be able to be handled by an average user.

	rule symbolic		*rule trend*	
#	body/antecedent	head/consequent	support	confidence
1	TEC=ADSL	WCC=YES	stable	stable
3	AGE=66+	VOL=NONE	down	down
5	TEC=ADSL, AGE=36-50	WCC=NO	stable	stable
7	AGE=66+	VOL=10GB	up	up
10	AGE=66+, TEC=CABLE	SAT=SAT	stable	stable

Table 6.1: OWA aggregation

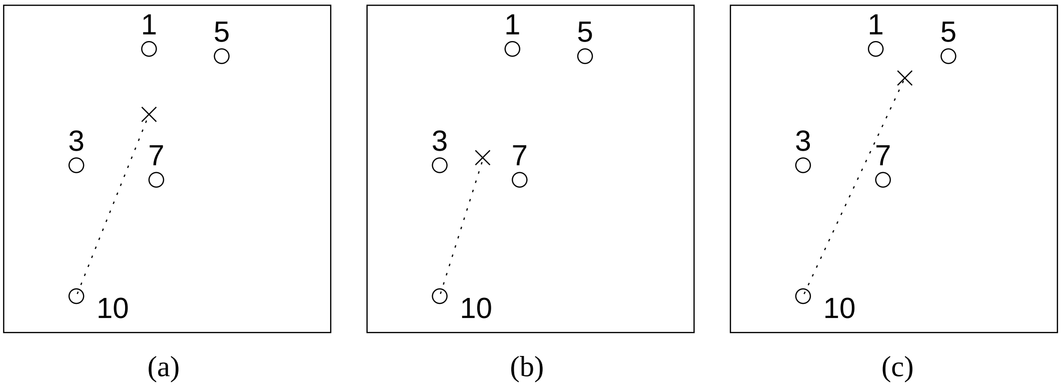


Figure 6.2: Different similarity aggregations

Figure 6.2 shows the difference between similarity aggregations during the relevance feedback cycle. The multi-dimensional rule space has been projected into 2D, and the Euclidean distance measure has been used. Circles denote the feature vectors $\vec{r}_{body}$ of association rules. 6.2(a) shows the averaging of similarities, whereas in (b) and (c) the higher or lower similarities are emphasised, respectively.

An Aggregation Operator

The above example showed that a more influencable aggregation is desirable that accounts for the different application scenarios which a user could be working in. In the following, a more sophisticated aggregation operator will be introduced, that aims to capture that requirement.

Advanced aggregation operators exist in the Fuzzy Domain, that could be used to handle this aggregation task. There were three reasons that led to the choice of an *ordered weighted averaging* (OWA) operator, which all rely on the OWA's weights:

Parameterisation The user should be able to influence the more flexible aggregation of similarities. An OWA operator is parameterised by a weight vector and can serve this purpose. Furthermore, the parameters can make the operator usable even to an average user.

Emulation By setting the weights of the OWA operator accordingly, simple aggregation operators like average, min, max, or median can be obtained. A user can switch between these concepts or set the weights as required. This also allows for the distinction between an expert and an average user.

Implementation The weights also influence the implementation of the OWA operator. In the overall relevance feedback framework, for every different aggregation operator, a new implementation would have to be written. Using the OWA operator, there is only one operator that can fulfill different aggregation concepts, by setting the weights accordingly. This simplifies the implementation and avoids errors during programming.

Those aggregation operators have been introduced by [Yager, 1988]. They are strongly related to the concepts of linguistic quantifiers, such as *many, a few, most.* They were originally introduced to aggregate the importance levels of the alternatives in a decision making problem. Nevertheless, [Yager, 1988] presented the connection to linguistic quantifiers, by explaining how the weights that appeared in the OWA expression could be obtained by using the membership function of any quantifier. In subsequent work ([Yager, 1992]), OWA operators are presented as a way to compute the accomplishment of linguistic quantifiers when used in conjunction with imprecise properties such as:

The most similar relevant rules should influence the interestingness score.
Rules which are dissimilar to the selected relevant rules are preferred.

Definition

An OWA operator Ω is a mapping $\Omega : S \rightarrow \mathbf{R}$, where S is a set of numerical values s_i with $S \neq \varnothing$ and $|S| = n$. Ω has an associated weighting vector $W = (w_1, w_2, \ldots, w_n)^T$, in which

(1) $w_j \in [0, 1]$ and

(2) $\sum_{j=1}^{n} w_j = 1$,

where

$$\Omega(\{s_1, s_2, \ldots, s_n\}) = \sum_{j=1}^{n} w_j b_j \quad , \tag{6.4}$$

with b_j being the j-th largest of the s_i.

The most important feature of this operator is the ordering of the arguments by value. The OWA operator is in a way very general in that it allows different conventional aggregation operators. This is achieved by appropriately setting the weights in W – different arguments can be emphasised based upon their position in the ordering. If most of the weights are placed near the beginning of W, the higher similarities are emphasised. If most of the weights are placed near the end of W, the lower similarities are emphasised. Figure 6.3 illustrates this issue, showing the weights for the example described at the beginning of this section.

The following special cases, described by [Yager, 1997], illustrate the generality of the OWA operator. Consider that $w_1 = 1$ and $w_j = 0$ for all $j \neq 1$. In this case, the *max* operator is obtained:

$$\Omega(\{s_1, s_2, \ldots, s_n\}) = \mathrm{max_j[s_j]} \tag{6.5}$$

If the weights are $w_n = 1$ and $w_j = 0$ for $j \neq n$, the *min* operator is obtained:

$$\Omega(\{s_1, s_2, \ldots, s_n\}) = \mathrm{min_j[s_j]} \tag{6.6}$$

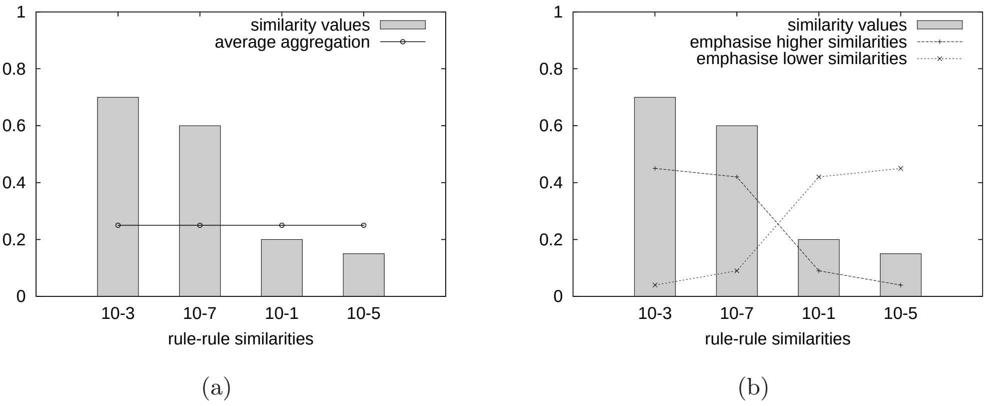


Figure 6.3: Example weights for the OWA operator

Figure 6.3 shows weight settings for the different similarity aggregations depicted in Figure 6.2. The average aggregation is shown in 6.3(a), with all OWA weights equal. The emphasising weights are shown in 6.3(b), where the higher or lower similarities are emphasised, respectively.

If the weights are calculated such that $w_j = \frac{1}{n}$ for all j, then the *average* operator is returned:

$$\Omega(\{s_1, s_2, \ldots, s_n\}) = \frac{1}{n} \sum_{j=1}^{n} s_j \tag{6.7}$$

The *median* operator can also be obtained. If n is odd, then $w_{\frac{n+1}{2}} = 1$ and $w_j = 0$ for $j \neq \frac{n+1}{2}$. If n is even, then $w_{\frac{n}{2}} = w_{\frac{n+1}{2}} = \frac{1}{2}$ and $w_j = 0$ for all other terms. Those special cases show the flexibility of the ordered weighted averaging operator, hence it has been used here. Another reason for its usage is the convenience in its implementation, since there is now only one operator to "emulate" conventional aggregation operators *and* to integrate user-specific aggregations. By setting the weights accordingly, the user can influence the interestingness score to suit the needs of his particular application scenario.

It should be noted that, as the sets of relevant and non-relevant rules grow, the weight vector of the OWA operator has to grow accordingly. If none of the above special cases of the OWA operator is used, an appropriate weight distribution should be computed. This could be done using a concept similar to a probability density function where mean and variance are specified by the user, according to which of the similarities he would like to emphasise.

6.4.4 Relative Importance of Recent Relevance Choices

The retrieval of interesting association rules is a consecutive, iterative process. The user's knowledge, his beliefs and assumptions change during the relevance feedback cycle as he sees new data. Therefore, the user's latest choices should be considered as having a higher priority over the first, relatively uninformed, relevance choices. This concept can be captured as the *decay of a relevant or non-relevant* rule's importance over time. An equation similar to, e.g., radioactive decay can be used. Let t the *age* of a relevant or non-relevant association rule: t is the number of feedback cycles that have been performed since the rule's selection as relevant or non-relevant, where a newly selected rule receives $t = 0$. Let $\delta \in [0, 1]$ a decay constant that controls the speed of decay. Then two possible equations for the *time-weighted importance* τ are as follows:

$$\begin{aligned} \tau_{exp} &= (1-\delta)^t && (6.8) \\ \tau_{lin} &= max(1 - t \cdot \delta, 0) && (6.9) \end{aligned}$$

with Equation 6.8 for an exponential type of decay and Equation 6.9 for a linear decay down to a minimum of zero. This concept can also be described as a kind of *memory* of the relevance feedback engine. The higher the decay factor δ, the faster the system forgets what has been chosen in an earlier step. If we set $\delta = 1$ then the system would only consider the user's latest relevance decision in its interestingness score calculation. The value of $\delta = 0$ would deactivate the decay completely. Values of δ in between those bounds activate a gradual decay.

6.4.5 Similarity between Rule and Rule Set

In Section 6.4.1, similarity between rule vectors or their respective parts has been defined. However, the relevance decisions of a user are collected in rule sets. The interestingness score calculation relies on the similarity between a rule and a rule set, therefore this operation will have to be defined. The prerequisites have been given above, namely the OWA aggregation operator Ω and the time-weighted importance τ. Let R be a set of feature vectors (association rule vectors) and $\vec{v}$ one of the following feature vectors $\vec{r}_{\text{body}}, \vec{r}_{\text{head}}, \vec{r}_{\text{sym}}, \vec{r}_{\text{time}}$ (see Equations 5.1 to 5.5 on page 39). The similarity sim_{rs} between $\vec{v}$ and the *respective* feature vector of the rules in R is now defined as follows:

$$sim_{rs}(\vec{v}, R) = \Omega(\{(\tau \cdot sim(\vec{v}, \vec{s_1})), \ldots, (\tau \cdot sim(\vec{v}, \vec{s_m}))\}) \qquad (6.10)$$

where $m = |R|$ and the dimension of the weight vector W of the OWA operator is equal to m. Equation 6.10 calculates the pairwise similarity between the feature vector $\vec{v}$ and each of the respective feature vectors $\vec{s} \in R$. Each of the resulting similarities is weighted by τ. The set of weighted similarities is then aggregated using the OWA aggregation operator Ω with an appropriate weight vector W. Since the sum of the OWA operator's weights is defined to be 1 and the rule similarities are in the interval $[0, 1]$, the dissimilarity between a rule and a rule set can be defined as follows:

$$dissim_{rs}(\vec{r}, R) = 1 - sim_{rs}(\vec{r}, R) \qquad (6.11)$$

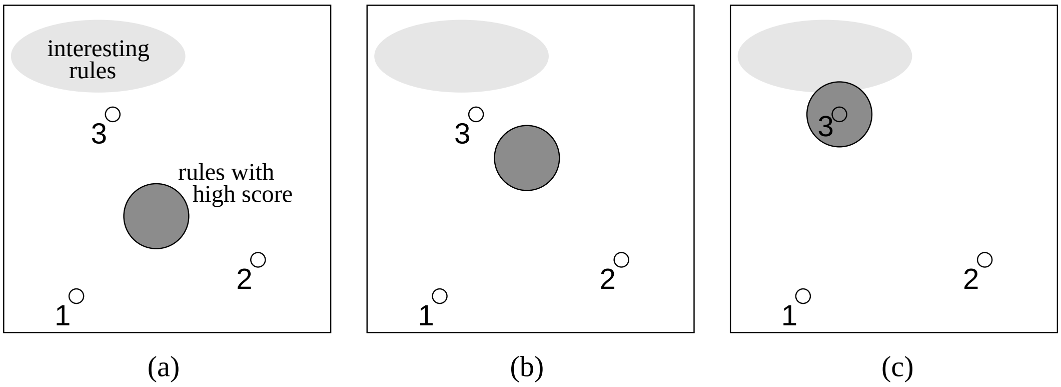


Figure 6.4: Decay of relevant rules' importance

Figure 6.4 shows the influence of the decay factor in the relevance feedback cycle. The multi-dimensional rule space has been projected into 2D, and the Euclidean distance measure has been used, along with an exponential decay. The grey ellipse depicts the interesting rules that are supposed to be found via relevance feedback. Three relevant rules (circles 1, 2, and 3) have been chosen before the current decision, in ascending order. The dark grey circle shows the area in which the rules with the highest interestingness score are to be expected. The decay constant is set to (a) 0, (b) 0.5, (c) 1. In (a), the rules with the highest score are located in the middle of the three selected rules, since every rule has the same influence on the score. In (b), only the rules 2 and 3 have an influence on the interestingness score, whereas in (c) only the latest choice, rule 3, influences the interestingness score. It can be seen that the decay constant affects how much of the rule space the user can explore if he relies solely on the interestingness score.

6.4.6 Handling of Relevance Information

After each user interaction that provides relevance information, the interestingness score of the association rules has to be updated. Both possible relevance choices bear information that must be treated differently.[1] The interestingness score equation consists of two parts for relevant and for non-relevant rules.

Relevant Rules

The prerequisites for the calculation of the interestingness score have been given above. The interestingness matrix in its short form is shown again in Table 6.2, slightly altered to include the respective weights from Equation 6.12. In accordance to the definition given in Section 3.5, the score $\Phi(\vec{r}, R_{\text{rel}})$ for an association rule vector $\vec{r}$ in regard to a set of relevant rules R_{rel} is calculated as follows:

[1] The feedback engine does not check for the consistency in a user's choices, i.e., he can select any rule available and can also have similar rules in R_{rel} and R_{nrel} at the same time.

similar \ *dissimilar*	head	body	time series	symbolic
head	-	ω_4	ω_5	-
body	ω_1	-	ω_6	-
time series	-	-	-	ω_2
symbolic	-	-	ω_3	-

Table 6.2: Modified Interestingness Matrix

$$\begin{aligned}\Phi(\vec{r}, R_{\text{rel}}) = {} & \omega_1 \cdot sim_{rs}(\vec{r}_{\text{body}}, R_{\text{rel}}) \cdot dissim_{rs}(\vec{r}_{\text{head}}, R_{\text{rel}}) \\ & + \omega_2 \cdot sim_{rs}(\vec{r}_{\text{time}}, R_{\text{rel}}) \cdot dissim_{rs}(\vec{r}_{\text{sym}}, R_{\text{rel}}) \\ & + \omega_3 \cdot sim_{rs}(\vec{r}_{\text{sym}}, R_{\text{rel}}) \cdot dissim_{rs}(\vec{r}_{\text{time}}, R_{\text{rel}}) \\ & + \omega_4 \cdot sim_{rs}(\vec{r}_{\text{head}}, R_{\text{rel}}) \cdot dissim_{rs}(\vec{r}_{\text{body}}, R_{\text{rel}}) \\ & + \omega_5 \cdot sim_{rs}(\vec{r}_{\text{head}}, R_{\text{rel}}) \cdot dissim_{rs}(\vec{r}_{\text{time}}, R_{\text{rel}}) \\ & + \omega_6 \cdot sim_{rs}(\vec{r}_{\text{body}}, R_{\text{rel}}) \cdot dissim_{rs}(\vec{r}_{\text{time}}, R_{\text{rel}})\end{aligned} \tag{6.12}$$

I assume that, depending on the user, he might be willing to specify the importance of certain interestingness assumptions, therefore the ω_i weights have been introduced in Equation 6.12. The weights' indices correspond to the numbers in Table 6.2.

In this equation, the similarities and dissimilarities of the different feature combinations are aggregated into an interestingness score. Let us assume that, without loss of generality, only one of the ω_i is set to 1, i.e., only one of the six score parts is "activated". The general assumption now is that the interestingness score is highest if both concepts (similarity and dissimilarity) are satisfied to their full extent, i.e., are assigned a value close to 1; the score is assumed to be lowest if none of the concepts is satisfied. For this purpose, *sim* and *dissim* could be averaged, which would satisfy those extreme cases. However, some type of linear score would result: for example, if $sim = 0.95$ and $dissim = 0.05$, the score would be 0.5 and indicate a certain interestingness, although one of the combinations clearly indicates uninterestingness. A multiplication of the *sim* and *dissim* values better grasps this concept: here, the score for the example would be 0.0475 and indicate the expected uninterestingness, whereas interesting rules are exposed. Finally, the six weighted score parts are added, which keeps the equation understandable; the weights can be seen as the percentages that a certain score part should influence the overall score with. In summary, the equation awards a higher score to the rules that meet those interestingness criteria that the user has specified via the weights.

Non-Relevant Rules

Non-relevant rules contain information that differs from the relevant rules. I assume that the non-relevant rules specify a subspace of the rule space where more non-relevant rules are located. To direct the user away from this subspace, rules that are far away from it will receive a higher score, whereas those in the vicinity will receive a low score. Therefore, the score $\Psi(\vec{r}, R_{\text{nrel}})$ for an association rule vector $\vec{r}$ in regard to a set of non-relevant rule vectors R_{nrel} is calculated as follows:

$$\Psi(\vec{r}, R_{\text{nrel}}) = dissim(\vec{r}, R_{\text{nrel}}) \tag{6.13}$$

6.4.7 Score Aggregation

With the prerequisites above, an interestingness score for each association rule vector $\vec{r}$ of a rule vector set R at a timestep t, depending on the set of relevant rules R_{rel} and the set of non-relevant rules R_{nrel} can now be calculated as follows:

$$F(\vec{r}, R, R_{\text{rel}}, R_{\text{nrel}}, t) = w_{\text{rel}}\Phi(\vec{r}, R_{\text{rel}}) + w_{\text{nrel}}\Psi(\vec{r}, R_{\text{nrel}}) \tag{6.14}$$

In Equation 6.14, the score parts, which result from the relevance and non-relevance decisions, are aggregated. Similar to the weighted sum of Equation 6.12, the score parts are added here. The user can choose which of his choices are more important to him – relevance or non-relevance – by setting the weights w_{rel} and w_{nrel} appropriately.

A summary of the introduced weights is presented in Section 6.5.1. After each relevance choice, the score for each association rule is updated. The scores at each timestep do not depend on the scores that have been calculated before. Finally, the rule that receives the highest score is assumed to be the most interesting to the particular user, after he has supplied relevance choices. The rules in the rule browser can be sorted by this interestingness score, which concludes the relevance feedback.

6.5 Experimental Results

6.5.1 User-Adjustable Weights

Numerous sets of weights have been introduced in this chapter, therefore a summary of the influencable parameters and their constraints is given in Table 6.3. The effects of setting certain parameters are described in the following section.

6.5.2 Weight Settings

The weight settings could be tested on the association rule set which was mined from survey data and is described in [Böttcher, 2005]. It will be referred to as "survey data set". The effects of the parameters and suggested settings are listed below.

δ **- decay parameter** This parameter controls how fast the relevance feedback engine forgets the relevance decisions that have been made by the user. Setting

parameter	*constraints*	*impact*
δ	$\in [0,1]$ exponential/linear	decay factor of relevance decisions, “memory” of the relevance feedback
OWA	as defined in Section 6.4.3, weight vector W	ordered weighted averaging operator, aggregation of similarities
$\omega_1 \dots \omega_6$	$\sum_i \omega_i = 1$ and $\omega_i \geq 0$	allocation of interesting feature combinations
$w_{\text{rel}}, w_{\text{nrel}}$	$w_{\text{rel}} + w_{\text{nrel}} = 1$ and $w_{\text{rel}}, w_{\text{nrel}} \geq 0$	balance between the relevance and non-relevance decisions

Table 6.3: Summary of introduced weights

this parameter to its maximum 1 will cause the system to only consider the latest relevance choice in its interestingness computation. Setting it to 0 will switch off the decay, hence, every relevance decision will have the same impact. When tested on the survey data set, the value 0 led to a stagnation of the interestingness score after a relatively low number of about five to seven relevance choices, which was expected. On the other hand, setting it to 1, the interestingness score became quite volatile, but did not start to alternate during consecutive steps. It is suggested to start with a low value of 0.2 to 0.3 and the linear decay and increase it in subsequent data analysis steps to explore a larger rule space.

W **- OWA operator weights** This parameter controls the aggregation of the similarities, which is computed when assessing the similarity between a rule feature vector and a set of feature vectors. It allows to determine which of the relevant or non-relevant rules should influence the calculation of the interestingness score: the similar ones or the dissimilar ones. The special cases of min, max, and average weights have been implemented and tested on the survey data set. Good results could be obtained using the max and min operators, which also yielded the expected different search strategies by generating different interestingness scores. Before setting W manually, the special cases should therefore be experimented with.

$\omega_1, \dots, \omega_6$ **- feature balance** This set of parameters is used to determine which of the possibly interesting features identified in Section 3.5 should be used to which degree in the computation of an interestingness score. The sum of the weights is restrained to be 1 and each weight must be positive: these assumptions model the idea of a proportion between the features. The user can set those which are interesting to him to a high percentage and neglect the remaining features. To get used to the relevance feedback system, it is advisable to study the effects of setting the parameters for at most two features, i.e., at most two weights should have a value that differs from 0 at the beginning. In subsequent steps, more weights can be set. Comprehensible results could be obtained on the survey data set: when setting only one weight to 1, the association rules with a high interestingness score exactly matched the assumptions from the interestingness definition.

$\omega_{rel}, \omega_{nrel}$ **- relevance balance** In the final aggregation step of the interestingness score assessment, the score contributions that derive from the separate treatment of relevant and non-relevant rules are added. ω_{rel} and ω_{nrel} are similar to the ω_i above in that they determine a balance between the influence that R_{rel} and R_{nrel} have on the interestingness score. However, since the calculation in the part of the non-relevant rules is quite simple, the interestingness score is dominated by the part for the relevant rules. The interestingness definition is also mostly based on the relevant rules. Good results on the survey data set could be achieved by setting the weights to 0.8 and 0.2 for ω_{rel} and ω_{nrel}, respectively. As with the ω_i above, it is also advisable to begin with a setting of 1.0 and 0 or vice versa, to study the effects of the parameter in the specific application scenario.

6.6 Summary

This chapter has given a brief overview of relevance feedback and its advantages under the given circumstances. The application of this technique in the association rule scenario has been motivated. Three problems could be solved: firstly, an initial ranking of the association rules had to be obtained; secondly, the user's knowledge had to be collected in terms of relevance decisions. Thirdly, based on this knowledge, the interestingness definition given in Chapter 3 and the preprocessing shown in Chapter 5, a relevance feedback engine has been devised. A necessary similarity measure has been obtained from IR literature and a sophisticated aggregation operator could be applied. The change of a user's assumptions and knowledge during the relevance feedback iterations has been accounted for by means of the memory of the relevance feedback engine.

Finally, the third key issue from Chapter 1.2 could be resolved: after each relevance feedback step, the user is presented with an updated ranking of the association rules, based on his relevance choices. The list can be sorted by an interestingness score, which enables the user to find the subjectively most interesting rules. This ranking can be influenced by an average as well as by an expert user by setting a comprehensive set of parameters. This chapter concludes the description of the *relevance feedback system for association rules.* The following chapter will give an overview of the implementation of the association rule browser and also show how the relevance feedback could be integrated.

Chapter 7

User Interface

7.1 Overview

This chapter will give an overview and suggestions on the user interface of the relevance feedback system for association rules. The existing rule browser will be described. The extensions to incorporate the relevance feedback and its parameters will be addressed. In addition, the user interface of an item-based rule filter will be shown. The resulting overall user interface consists of four parts in principle and is shown in Figure 7.1.

7.2 Rule Browser Module

7.2.1 Existing Browser

The principle components of the existing association rule browser are shown in Figure 7.2. The central idea is the browsable list of association rules with their numeric features. The rules can be grouped by trend via the two controls in the top row. A string-based search is implemented in the top left corner. Rules can be selected and their histories can be examined via the rule time series viewer at the bottom. Numerical values have been left out of this illustration of the rule browser for reasons of simplicity. Rules can be sorted according to support, confidence, and further numeric values, such as the different interestingness measures for trends.

7.2.2 Relevance Selection

The existing rule browser could be extended to allow the relevance selection. Since rules could already be sorted according to numeric values, the interestingness score could be incorporated easily. In each relevance feedback step, the rules are sorted by interestingness score and the user can examine the top-ranked rules and still browse the remaining ones. A user that might already have been working with the basic rule browser interface will hence be very familiar with it. When a user wants to mark an association rule as relevant or non-relevant, the browser collects this choice in an extra window. Figure 7.3 illustrates this process.

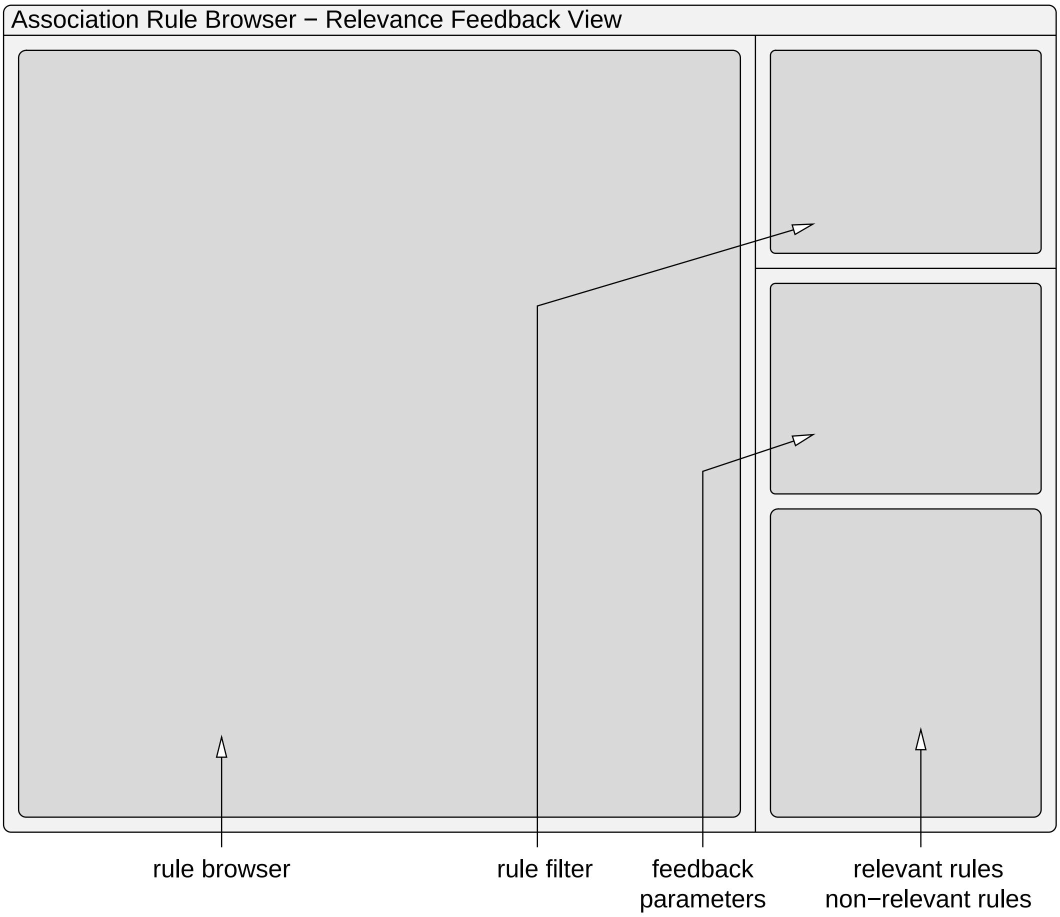


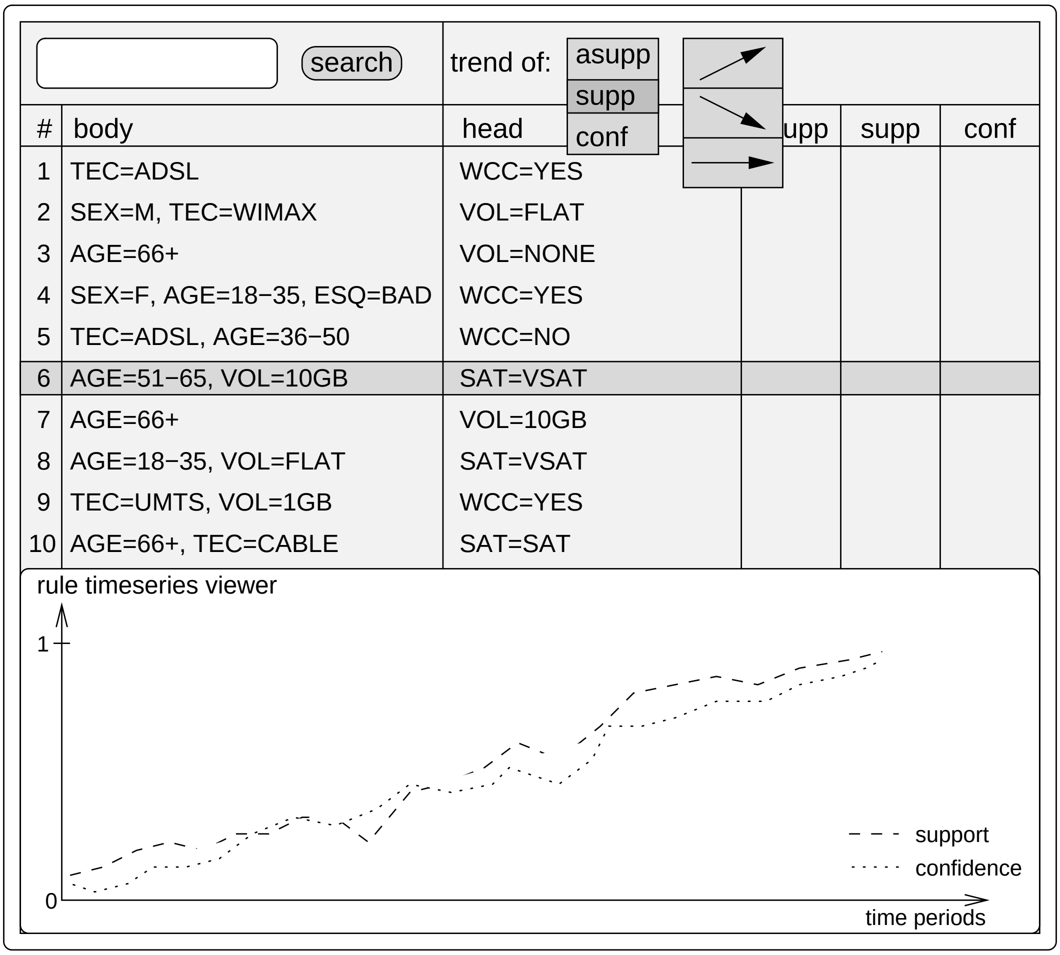
Figure 7.1: Overview of the association rule browser with relevance feedback

7.3 Relevance Feedback Module

Since the software is still in a prototypical state, the suggestions for the setting of relevance parameters have not been implemented. Nevertheless, they can be set in a straightforward way. From Section 6.5.1 the user-influentiable weights can be obtained, which have to be accounted for. It has been argued that the software should be usable both for an average user as well as an expert user. For this purpose, two different possible parameter settings modules will be explained.

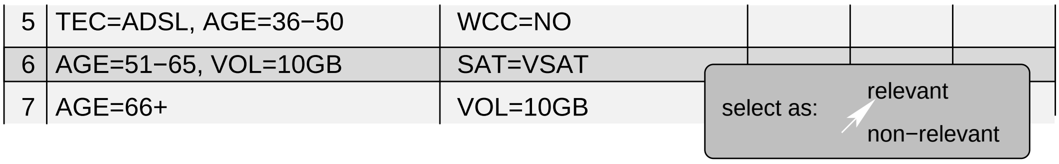
7.3.1 Average User

The average user receives basic settings, his choices are shown in Figure 7.4. Since he is not assumed to be too familiar with the internal workings of the system, he should not be bothered with a too exact setting of the parameters. His choices are therefore slightly limited. In the top control, he can adjust how long he wants the relevance feedback engine to memorise his relevance decisions, the decay factor. The



#	body	head	upp	supp	conf
1	TEC=ADSL	WCC=YES			
2	SEX=M, TEC=WIMAX	VOL=FLAT			
3	AGE=66+	VOL=NONE			
4	SEX=F, AGE=18−35, ESQ=BAD	WCC=YES			
5	TEC=ADSL, AGE=36−50	WCC=NO			
6	AGE=51−65, VOL=10GB	SAT=VSAT			
7	AGE=66+	VOL=10GB			
8	AGE=18−35, VOL=FLAT	SAT=VSAT			
9	TEC=UMTS, VOL=1GB	WCC=YES			
10	AGE=66+, TEC=CABLE	SAT=SAT			

Figure 7.2: Association rule browser



5	TEC=ADSL, AGE=36−50	WCC=NO			
6	AGE=51−65, VOL=10GB	SAT=VSAT			
7	AGE=66+	VOL=10GB			

Figure 7.3: Association rule browser, relevance selection

second control adjusts the OWA weights according to which search strategy the user follows. The third control selects *one* of the six interesting combinations, which will keep the interestingness score more understandable. The balance between the sets of relevant and non-relevant rules concludes the parameter settings for the average user. In the implementation, sliders would be used to relieve the user from entering numerical values.

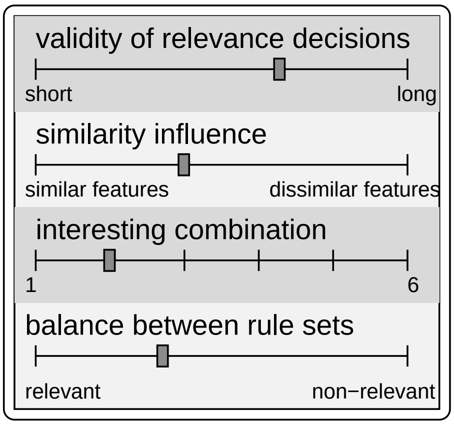


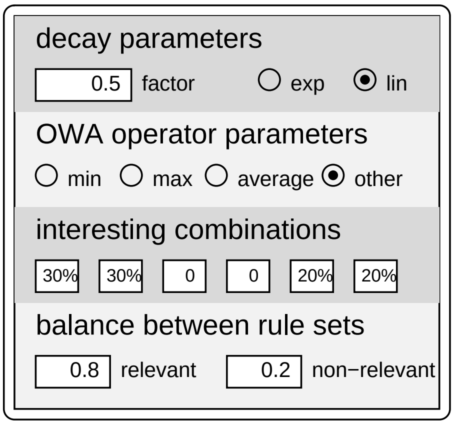
Figure 7.4: Relevance feedback parameters for average users

7.3.2 Expert User

The expert user is likely to know more about the rules and the relevance feedback system's internal workings, he therefore also receives more influence and more parameter precision on it. His options are shown in Figure 7.5. The top parameter influences the decay factor directly, and the type of decay can be chosen. The second parameter adjusts the weights of the OWA operator. The three special cases max, min, and average can be chosen directly. In addition, the option “other” can be chosen, which would invoke an additional settings window where the OWA weights can be set individually. The third parameter, interesting combinations, weights the combination of interesting features accordingly. Fine-granular settings are possible due to numerical entry fields. Those fields also account for the balance between the rule sets, which is the fourth parameter.

7.3.3 Relevant and Non-Relevant Rules

A simple and less interactive part of the relevance feedback module, the collected rule sets are displayed in a list. The rules can be sorted and should always be displayed, to guide the user in his relevance decisions and remind him which rules have been chosen. The selected relevant and non-relevant rules are shown in Figure 7.6.



decay parameters

0.5 factor ○ exp ● lin

OWA operator parameters

○ min ○ max ○ average ● other

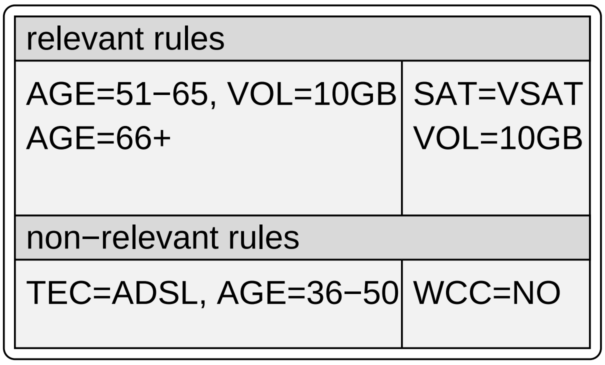
interesting combinations

30% 30% 0 0 20% 20%

balance between rule sets

0.8 relevant 0.2 non−relevant

Figure 7.5: Relevance feedback parameters for expert users



relevant rules	

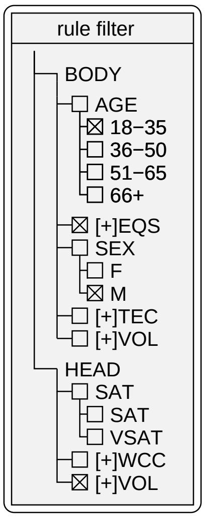
AGE=51−65, VOL=10GB
AGE=66+
	SAT=VSAT
	VOL=10GB

non−relevant rules	
TEC=ADSL, AGE=36−50	WCC=NO

Figure 7.6: Collected rule sets during relevance feedback

7.4 Rule Filter Module

The rule filter, as a module of the program that does not touch the relevance feedback, is shown in Figure 7.7. It allows to select certain attributes or items; the association rules which contain those values will be removed from the rule set. Attributes and items are shown in a tree-like structure, divided into body and head, which is quite intuitive. The filter lists only the items that exist in the rule set: if an attribute has five possible values and only three of them occur in the rule set, the filter will list these three. The rule filter has been implemented and can be used.



rule filter

BODY
- ☐ AGE
  - ☒ 18–35
  - ☐ 36–50
  - ☐ 51–65
  - ☐ 66+
- ☒ [+]EQS
- ☐ SEX
  - ☐ F
  - ☒ M
- ☐ [+]TEC
- ☐ [+]VOL

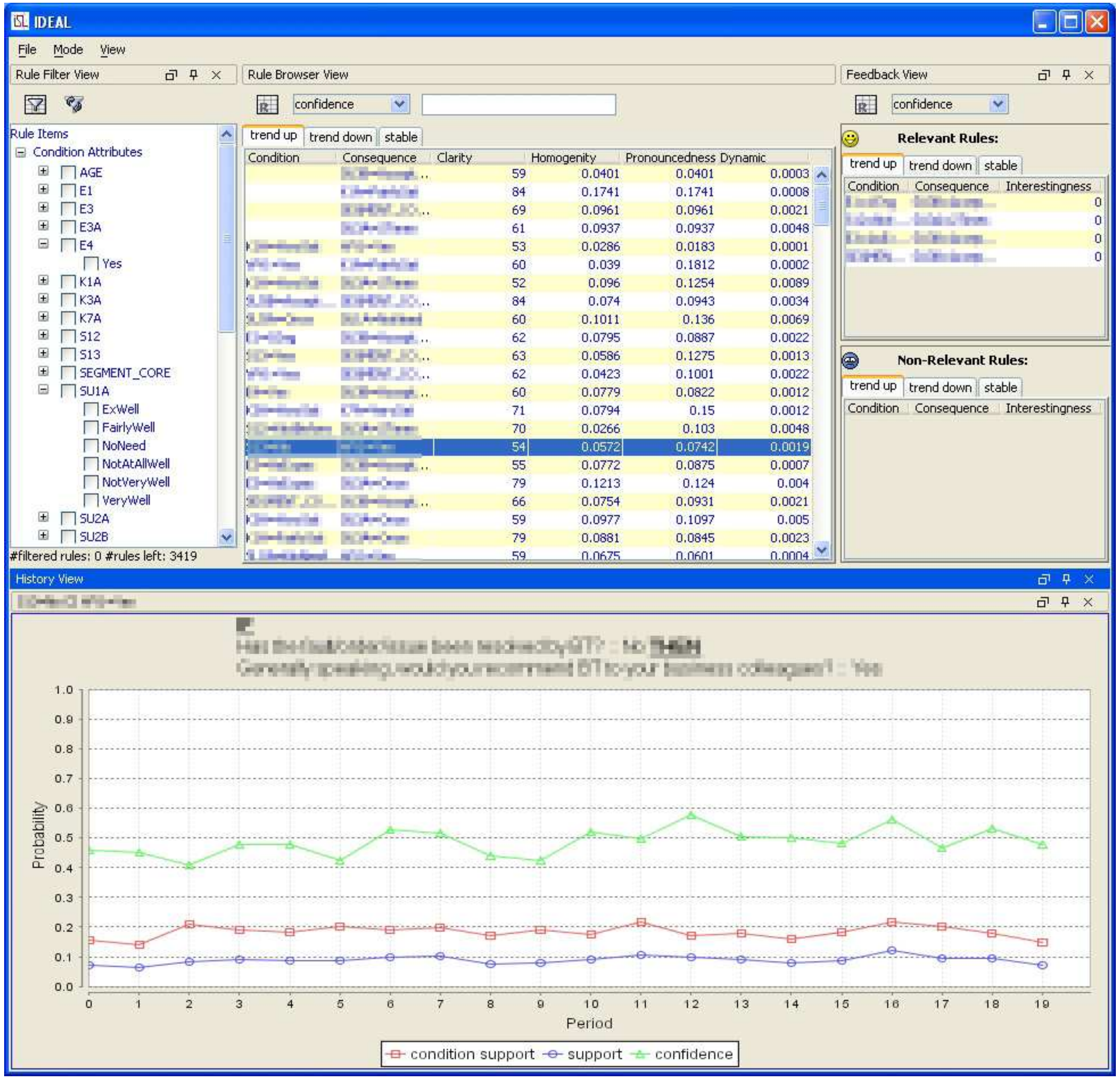
HEAD
- ☐ SAT
  - ☐ SAT
  - ☐ VSAT
- ☐ [+]WCC
- ☒ [+]VOL

Figure 7.7: Association rule filter

7.5 Summary

This chapter gave a brief summary of the existing implementation and the user interface of the association rule browser. The browser could be extended to incorporate the relevance feedback engine devised in the preceding chapters. The interestingness score could be integrated in a very convenient way for the end user. The rule filter, as a step that can also be performed before the relevance feedback, has been implemented. Suggestions for setting the relevance feedback parameters have been given, for an average as well as for an expert user. Since the software is still at a prototypical stage, the parameter settings modules have not been implemented into the user interface. For the same reason, a user evaluation, albeit desirable, could not be performed. Nevertheless, the underlying methods are readily available, and the user interface can be easily adapted to the changing needs of the software's users. A screenshot of the overall JAVA implementation is shown[1] in Figure 7.8.

[1] The attributes and values shown in Figure 7.8 are completely fictitious and have been used for illustration purposes only. Association rules have been scrambled.



Rule Items
- Condition Attributes
  - □ AGE
  - □ E1
  - □ E3
  - □ E3A
  - □ E4
    - □ Yes
  - □ K1A
  - □ K3A
  - □ K7A
  - □ S12
  - □ S13
  - □ SEGMENT_CORE
  - □ SU1A
    - □ ExWell
    - □ FairlyWell
    - □ NoNeed
    - □ NotAtAllWell
    - □ NotVeryWell
    - □ VeryWell
  - □ SU2A
  - □ SU2B

#filtered rules: 0 #rules left: 3419

confidence

trend up | trend down | stable

Condition	Consequence	Clarity	Homogenity	Pronouncedness	Dynamic
	[redacted]	59	0.0401	0.0401	0.0003
	[redacted]	84	0.1741	0.1741	0.0008
	[redacted]	69	0.0961	0.0961	0.0021
	[redacted]	61	0.0937	0.0937	0.0048
[redacted]	[redacted]	53	0.0286	0.0183	0.0001
[redacted]	[redacted]	60	0.039	0.1812	0.0002
[redacted]	[redacted]	52	0.096	0.1254	0.0089
[redacted]	[redacted]	84	0.074	0.0943	0.0034
[redacted]	[redacted]	60	0.1011	0.136	0.0069
[redacted]	[redacted]	62	0.0795	0.0887	0.0022
[redacted]	[redacted]	63	0.0586	0.1275	0.0013
[redacted]	[redacted]	62	0.0423	0.1001	0.0022
[redacted]	[redacted]	60	0.0779	0.0822	0.0012
[redacted]	[redacted]	71	0.0794	0.15	0.0012
[redacted]	[redacted]	70	0.0266	0.103	0.0048
[redacted]	[redacted]	54	0.0572	0.0742	0.0019
[redacted]	[redacted]	55	0.0772	0.0875	0.0007
[redacted]	[redacted]	79	0.1213	0.124	0.004
[redacted]	[redacted]	66	0.0754	0.0931	0.0021
[redacted]	[redacted]	59	0.0977	0.1097	0.005
[redacted]	[redacted]	79	0.0881	0.0845	0.0023
[redacted]	[redacted]	59	0.0675	0.0601	0.0004

confidence

Relevant Rules:

trend up | trend down | stable

Condition	Consequence	Interestingness
[redacted]	[redacted]	0
[redacted]	[redacted]	0
[redacted]	[redacted]	0
[redacted]	[redacted]	0

Non-Relevant Rules:

trend up | trend down | stable

Condition	Consequence	Interestingness

[redacted]

[redacted]

Figure 7.8: Association rule browser with relevance feedback, JAVA implementation

Chapter 8

Conclusions

8.1 Summary

Nowadays, large businesses and governments collect ever-growing volumes of data. Methods of data mining, namely association rule mining, aim to cope with this data, since it is crucial for businesses and governments to find the interesting information and act upon it. Hence, there is a significant need for data mining approaches that deal with the *interestingness of time-stamped association rules*.

I designed and implemented a relevance feedback system for association rules that allows to find the association rules that are most interesting to a particular user, out of a large set of association rules.

During the design phase, three key problems could be identified, which had to be resolved in order to arrive at the final relevance feedback system.

The first of those problems is a *definition of interestingness*: if interesting association rules were to be discovered, a mathematical model that assesses this interestingness had to be developed. I gave a thorough literature review that evaluated existing approaches and identified their advantages and their drawbacks. Based on those evaluations, I derived the requirements of an interestingness assessment system. This system employs a similarity-based interestingness definition that takes the specific properties of association rules and the user's assumptions into account. I developed a relevance feedback framework that captures those requirements.

As an important contribution, this thesis established a *novel connection between association rules and information retrieval*. This link could be used to provide methods from information retrieval and use them on association rules, which leads to the second issue: a suitable way of *preprocessing* the association rules had to be found. In the link to IR I gave a short evaluation of three different standard IR models and chose the feature-vector-based model. I gave a definition of the preprocessing, which resulted in the combined feature vector representation of an association rule's components.

The third issue that derived from the relevance feedback framework was the *definition of the relevance feedback engine* itself: I gave a short introduction and motivated its use in the given context. It consists of several iterative steps, in which the user can interactively explore the set of association rules. His feedback provides the system with invaluable knowledge about what the user is interested in. I answered several questions that concern the generation of an *interestingness score* for each association rule; I had to choose a similarity measure, the knowledge base had to be described, a rule set aggregation had to be found, and finally, an equation for the calculation of the interestingness score could be given. The relevance feedback can be controlled via high-level parameters that I introduced. This greatly improves the usability of the relevance feedback system, since a user does not have to be an expert to handle it. Nevertheless, expert users are accounted for in the parameter settings.

An existing implementation of an association rule browser could be easily extended to incorporate the relevance feedback that I proposed. The parameters can be adjusted easily in a graphical user interface.

The proposed relevance feedback system can be applied to any association rule mining system where interesting rules have to be discovered out of a large set of rules by a user. Yet, the association rules must have histories, i.e. be time-stamped, to use the interestingness definition to its full extent. Apart from these requirements, there are several advantages: Firstly, the system works independently from the domain under consideration, i.e. special features of the association rules in this thesis (which were mined from survey data) are not considered. Secondly, no expert user is needed to explore the association rules, which is an advantage over many data mining approaches where the expert knowledge is an implicit input to the system. Finally, an existing rule browser could be extended to incorporate the relevance feedback system proposed in this thesis.

8.2 Future Work

The novel link between association rules and IR has been proposed in this thesis. Nevertheless, a user evaluation should be carried out, to possibly improve the software prototype's user interface. Future work could aim to extend the developed relevance feedback framework in several areas:

Interestingness Definition The given interestingness definition relies on a notion of similarity between three different rule components: body, head, and time series. Six of sixteen possible combinations of the similarity and the dissimilarity of those components have been incorporated into the interestingness definition. However, in future association rule environments, other combinations could become interesting as well. The shown framework allows an easy integration of those combinations.

Time Series Split Related to the adaptation of the interestingness definition above, the time series could be used to a greater extent by making use of the different properties of support, antecedent support, and confidence. In the system presented in this thesis, the time series are used as a combined representation, but the prerequisites have been given to use each of support, antecedent support, and confidence separately in similarity calculations.

Initial Ranking of Relevance Feedback In Chapter 6, that deals with the details of the relevance feedback engine, an initial ranking is shown that will be refined by the user's relevance choices. The initial ranking serves as a starting point for the user – an overview of the rule set can also be achieved via other means, using feature-vector-based data clustering. An advanced example has been given in Section 6.3.2.

Obtain Feedback Implicitly Instead of having the user specify relevant and non-relevant rules explicitly, the system could employ advanced user monitoring techniques to construct the internal knowledge base. A basic approach would monitor input devices, whereas advanced approaches could monitor the user's body actions, such as his eye movements [Salojärvi et al., 2003].

Collaborative Relevance Choices During the relevance feedback, a user's relevance choices could be collected and stored in a database. If a sufficiently large number of those "relevance histories" has been collected, some way of collaboration could be integrated into the system: once a user selects a rule as relevant or non-relevant, the system could show the choices of previous users and give useful insights to new users.

List of Figures

2.1 Plots of two association rules' support and confidence values 7

3.1 An example taxonomy 14
3.2 Example for the calculation of raw surprisingness 18
3.3 Interestingness measures for trends (1) 22
3.4 Interestingness measures for trends (2) 23
3.5 Relevance Feedback Framework Overview 27

5.1 Overview of the preprocessing 41

6.1 Overview of the relevance feedback 47
6.2 Different similarity aggregations 52
6.3 Example weights for the OWA operator 54
6.4 Decay of relevant rules' importance 56

7.1 Overview of the association rule browser with relevance feedback . . . 62
7.2 Association rule browser 63
7.3 Association rule browser, relevance selection 63
7.4 Relevance feedback parameters for average users 64
7.5 Relevance feedback parameters for expert users 65
7.6 Collected rule sets during relevance feedback 65
7.7 Association rule filter 66
7.8 Association rule browser with relevance feedback, JAVA implementation 67

List of Tables

2.1 Exemplary set of association rules 9

3.1 Literature Summary 24
3.2 Interestingness Matrix 29

6.1 OWA aggregation 51
6.2 Modified Interestingness Matrix 57
6.3 Summary of introduced weights 59

Bibliography

Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, New York, NY, USA, 1993. ACM Press. ISBN 0-89791-592-5. 2, 5, 7

Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 12–15 1994. ISBN 1-55860-153-8. 5

Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999. ISBN 0-201-39829-X. 33

C. Borgelt and R. Kruse. Induction of association rules: A priori implementation, 2002. 5

Christian Borgelt. Efficient implementations of apriori and eclat, 2003. 7

Mirko Böttcher. Discovering interesting temporal changes in association rules. Master's thesis, Otto-von-Guericke-Unversität Magdeburg, Germany, 2005. 6, 11, 20, 23, 24, 50, 58

David Cox and A Stuart. Some quick sign tests for trend in location and dispersion. *Biometrika*, 42:80–95, 1955. 21

Douglass R. Cutting, Jan O. Pedersen, David Karger, and John W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–329, 1992. 49

Roger W. Elliott and Lee E. Cashman. An experimental comparison of relevance-feedback techniques. In *ACM'73: Proceedings of the annual conference*, pages 256–261, New York, NY, USA, 1973. ACM Press. 34, 46

Ronald Fagin and Edward L. Wimmers. A formula for incorporating weights into scoring rules. *Theoretical Computer Science*, 239(2):309–338, 2000. 51

Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining.* AAAI/MIT Press, 1996. ISBN 0-262-56097-6. 1

Alex Alves Freitas. On objective measures of rule surprisingness. In *PKDD '98: Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, pages 1–9, London, UK, 1998. Springer-Verlag. ISBN 3-540-65068-7. 18, 24

Brett Gray and Maria E. Orlowska. Ccaiia: Clustering categorial attributed into interseting accociation rules. In *PAKDD '98: Proceedings of the Second Pacific-Asia Conference on Research and Development in Knowledge Discovery and Data Mining*, pages 132–143, London, UK, 1998. Springer-Verlag. ISBN 3-540-64383-4. 12

Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 76–84, Zürich, CH, 1996. 49

Robert J. Hilderman and Howard J. Hamilton. Knowledge discovery and interestingness measures: A survey. Technical Report CS 99-04, 1999. 12

Robert J. Hilderman and Howard J. Hamilton. Measuring the interestingness of discovered knowledge: A principled approach. *Intell. Data Anal.*, 7(4):347–382, 2003. 11, 24

Peter Hoschka and Willi Klösgen. A support system for interpreting statistical data. In *Knowledge Discovery in Databases*, pages 325–346. 1991. 12

Tommi Jaakkola and Hava Siegelmann. Active information retrieval. In *Advances in Neural Information Processing Systems 14*, pages 777–784. MIT Press, 2001. 45

Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999. 49

Szymon Jaroszewicz and Dan A. Simovici. Interestingness of frequent itemsets using bayesian networks as background knowledge. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 178–186, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-888-9. 12

Diane Kelly and Jaime Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003. ISSN 0163-5840. 46

Mika Klemettinen, Heikki Mannila, Pirjo Ronkainen, Hannu Toivonen, and A. Inkeri Verkamo. Finding interesting rules from large sets of discovered association rules. In Nabil R. Adam, Bharat K. Bhargava, and Yelena Yesha, editors, *Third International Conference on Information and Knowledge Management (CIKM'94)*, pages 401–407. ACM Press, 1994. 12, 24

Chang-Hung Lee, Cheng-Ru Lin, and Ming-Syan Chen. Sliding-window filtering: an efficient algorithm for incremental mining. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 263–270, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-436-3. 7

Anton Leuski. Evaluating document clustering for interactive information retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 33–40, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-436-3. 49

Bing Liu, Wynne Hsu, and Shu Chen. Using general impressions to analyze discovered classification rules. In *Knowledge Discovery and Data Mining*, pages 31–36, 1997. 14, 18

Bing Liu, Wynne Hsu, Shu Chen, and Yiming Ma. Analyzing the subjective interestingness of association rules. *IEEE Intelligent Systems*, 15(5):47–55, 2000. 12, 14, 15, 17, 24

Huan Liu, Hongjun Lu, Ling Feng, and Farhad Hussain. Efficient search of reliable exceptions. In *PAKDD '99: Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*, pages 194–203, London, UK, 1999. Springer-Verlag. ISBN 3-540-65866-1. 16, 24

Nan Lu, Jing-Zhou Zhou, Wang Zhe, and Chun-Guang Zhou. Research on association rules mining algorithm with item constraints. In *CW '05: Proceedings of the 2005 International Conference on Cyberworlds*, pages 325–329, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2378-1. 7

Henry Mann. Nonparametric tests against trend. *Econometrica*, 13(3):245–259, 1945. 21

Ken McGarry. A survey of interestingness measures for knowledge discovery. *Knowl. Eng. Rev.*, 20(1):39–61, 2005. ISSN 0269-8889. 12

Balaji Padmanabhan and Alexander Tuzhilin. A belief-driven method for discovering unexpected patterns. In *Knowledge Discovery and Data Mining*, pages 94–100, 1998. 13, 24

Balaji Padmanabhan and Alexander Tuzhilin. Unexpectedness as a measure of interestingness in knowledge discovery. *Decis. Support Syst.*, 27(3):303–318, 1999. ISSN 0167-9236. 14

Balaji Padmanabhan and Alexander Tuzhilin. Small is beautiful: discovering the minimal set of unexpected patterns. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 54–63, New York, NY, USA, 2000. ACM Press. ISBN 1-58113-233-6. 13

Balaji Padmanabhan and Alexander Tuzhilin. Knowledge refinement based on the discovery of unexpected patterns in data mining. *Decis. Support Syst.*, 33(3):309–321, 2002. ISSN 0167-9236. 13

G. Piatetsky-Shapiro and C. Matheus. The interestingness of deviations. In *KDD-94*, pages 25–36. 19

C. J. van Rijsbergen. *Information Retrieval, 2nd edition.* Dept. of Computer Science, University of Glasgow, 1979. 49

Jr. Roberto J. Bayardo and Rakesh Agrawal. Mining the most interesting rules. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 145–154, New York, NY, USA, 1999. ACM Press. ISBN 1-58113-143-7. 12

J. J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System : Experiments in Automatic Document Processing*, Englewood Cliffs, New Jersey, USA, 1971. Prentice Hall Inc. 34, 46

Sigal Sahar. Interestingness via what is not interesting. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 332–336, New York, NY, USA, 1999. ACM Press. ISBN 1-58113-143-7. 17, 24

Sigal Sahar. On incorporating subjective interestingness into the mining process. In *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, page 681, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7695-1754-4. 17

Jarkko Salojärvi, Ilpo Kojo, Jaana Simola, and Samuel Kaski. Can relevance be inferred from eye movements in information retrieval? In *Proceedings of WSOM'03*, pages 261–266, Helsinki, Finland, 2003. 71

Gerard Salton. *The SMART Information Retrieval System.* Prentice Hall, Englewood Cliffs, NJ, 1971. 45

Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. In *Information Processing and Management*, volume 5, pages 513– 523, Ithaca, NY, USA, 1987. 40

Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41:288–297, June 1990. 46

Ashoka Savasere, Edward Omiecinski, and Shamkant B. Navathe. An efficient algorithm for mining association rules in large databases. In *The VLDB Journal*, pages 432–444, 1995. 7

Abraham Silberschatz and Alexander Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Knowledge Discovery and Data Mining*, pages 275–281, 1995. 20, 24

Avi Silberschatz and Alexander Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):970–974, 1996. ISSN 1041-4347. 17, 20

Pavan Kumar C. Singitham, Mahathi S. Mahabhashyam, and Prabhakar Raghavan. Efficiency-quality tradeoffs for vector score aggregation. In *VLDB*, pages 624–635, 2004. 51

Ramakrishnan Srikant and Rakesh Agrawal. Mining generalized association rules. *Future Generation Computer Systems*, 13(2–3):161–180, 1997. 5

Einoshin Suzuki. Autonomous discovery of reliable exception rules. In *KDD'97*, pages 159–176. AAAI Press, 1997. 16

Ke Wang, Yuelong Jiang, and Laks V. S. Lakshmanan. Mining unexpected rules by pushing user dynamics. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 246–255, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-737-0. 34

R. R. Yager. Fuzzy quotient operator. In *Proceedings of the Fourth International Conference on Information Processing and Management of Uncertainty*, pages 317–322, Palma de Majorca, Spain, 1992. 53

Ronald R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Trans. Syst. Man Cybern.*, 18(1):183–190, 1988. ISSN 0018-9472. 53

Ronald R. Yager. On the inclusion of importances in owa aggregations. In *The ordered weighted averaging operators: theory and applications*, pages 41–59, Norwell, MA, USA, 1997. Kluwer Academic Publishers. ISBN 0-7923-9934-X. 53

Mohammed J. Zaki. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3):372–390, 2000. ISSN 1041-4347. 7

Selbständigkeitserklärung

Hiermit versichere ich, daß ich die vorliegende Diplomarbeit selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht habe, was aus Arbeiten anderer unverändert oder mit Abänderung entnommen wurde.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form für keine andere Prüfung vorgelegt.

Magdeburg, den 6. September 2006

Georg Ruß