

Relevance Feedback for Association Rules by Leveraging Concepts from Information Retrieval

Georg Ruß

Institute for Knowledge and
Language Engineering
University of Magdeburg
Germany

Mirko Böttcher

Intelligent Systems
Research Centre, BT Group
Ipswich, United Kingdom

Detlef Nauck

Intelligent Systems
Research Centre, BT Group
Ipswich, United Kingdom

Rudolf Kruse

Institute for Knowledge and
Language Engineering
University of Magdeburg
Germany

Abstract

The task of detecting those association rules which are interesting within the vast set of discovered ones still is a major research challenge in data mining. Although several possible solutions have been proposed, they usually require a user to be aware what he knows, to have a rough idea what he is looking for, and to be able to specify this knowledge in advance. In this paper we compare the task of finding the most relevant rules with the task of finding the most relevant documents known from Information Retrieval. We propose a novel and flexible method of relevance feedback for association rules which leverages technologies from Information Retrieval, like document vectors, term frequencies and similarity calculations. By acquiring a user's preferences our approach builds a repository of what he considers to be (non-)relevant. By calculating and aggregating the similarities of each unexamined rule with the rules in the repository we obtain a relevance score which better reflects the user's notion of relevance with each feedback provided.

1 Introduction

Association rule mining [1, 2] originally has been developed for market basket data analysis, where each basket, also referred to as a transaction, consists of a set of purchased items. The goal of association rule mining is to detect all those items which frequently occur together and to form rules which predict the co-occurrence of items. However, association rule mining is not just bound to this specific purpose. It can be applied, for example, to every relational database.

Nowadays, the discovery of association rules is a relatively mature and well-researched topic. Many algorithms have been proposed to ever faster discover and maintain association rules. However, one of the biggest problems of association rules still remains unresolved. Usually, the number of discovered associations will be immense, easily in the thousands or even tens of thousands. Clearly, the large numbers make rules difficult to examine by a human user. Moreover, many of the discovered rules will be obvious, already known, or not relevant to a user. For this reason several methods have been proposed to assist a user in detecting the most interesting or relevant ones. The vast majority of these approaches either calculate a relevance score or determine rules that contradict a user's prior knowledge based on Boolean logic.

In this paper we argue that such approaches only insufficiently reflect the way a user searches for relevant rules because a user's perception of relevance is not a static but rather a dynamic process due to several reasons: firstly, when a user starts to explore a set of discovered association rules he only has a very vague notion about which rules might be relevant to him. Secondly, while seeing more rules his knowledge about the domain of interest changes, some aspects might gain while others might lose importance. His notion of relevance depends on these changes and thus changes too, almost always becoming clearer. The more rules a user examines, the more knowledge he gathers about the domain of interest. This knowledge then helps him to decide for newly encountered rules whether they are (non-)relevant for him, for example, because they are kind-of similar to previously seen (non-)relevant ones.

The importance of user dynamics and incremental knowledge gathering in assessing the relevance of data mining results only recently gained attention in the research community [3, 4]. However, it is a rather well-researched topic in the field of information retrieval where it is known for a long time that a user cannot express his information need from scratch. For example, when using a internet search engine to search documents about a non-trivial topic most users start with a rather simple query. By analysing the search results they gain more knowledge about what they actually look for and thus are able to further refine their initial query, i.e. to express their notion of relevance more clearly. To support a user in this process techniques like relevance feedback based on document similarities have been developed.

In fact, the way a user builds up his internal notion of relevancy when searching for the most relevant association rules described above is very similar to the models of user behaviour used in information retrieval (cp. [5]). Based on these similarities we present a new approach to the problem of finding the most relevant rules out of a large set of association rules which is inspired by ideas from information retrieval. Our approach uses relevance feedback to acquire users' preferences and to build a knowledge base of what he considers to be relevant and non-relevant, respectively. By calculating the (dis-)similarity of each unexamined rule with the rules in the knowledge base and aggregating the scores we obtain a relevance score which—with each feedback provided—better reflects the user's notion of relevance.

The remainder of this paper is organised as follows: Section 2 gives the

background on association rules, Section 3 shows the related work that is most relevant to our topic. Section 4 will further elaborate the link between information retrieval and interestingness assessment of association rules. Section 5 introduces a novel notion of association rules based on features vectors which are inspired by document vectors from information retrieval. This representation is closely related to our notion of rule similarity explained in Section 6 and Section 7. The relevance scoring metric will be derived in Section 8 before Section 9 concludes the paper.

2 Association Rules

Formally, association rule mining is applied to a set \mathcal{D} of *transactions* $\mathcal{T} \in \mathcal{D}$. Every transaction \mathcal{T} is a subset of a set of items \mathcal{L} . A subset $\mathcal{X} \subseteq \mathcal{L}$ is called *itemset*. It is said that 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 $\mathcal{X} \cap \mathcal{Y} = \emptyset$. Its meaning is quite intuitive: Given a database \mathcal{D} of transactions the rule above expresses that whenever $\mathcal{X} \subseteq \mathcal{T}$ holds, $\mathcal{Y} \subseteq \mathcal{T}$ is likely to hold too. If for two rules $r : \mathcal{X} \rightarrow \mathcal{Y}$ and $r' : \mathcal{X}' \rightarrow \mathcal{Y}$, $\mathcal{X} \subset \mathcal{X}'$ holds, then it is said that r is a *generalization* of r' . This is denoted by $r' \prec r$.

As usual, the reliability of a rule $r : \mathcal{X} \rightarrow \mathcal{Y}$ is measured by its *confidence* $\text{conf}(r)$, which estimates $P(\mathcal{Y} \subseteq \mathcal{T} \mid \mathcal{X} \subseteq \mathcal{T})$, or short $P(\mathcal{Y} \mid \mathcal{X})$. The statistical significance of r is measured by its *support* $\text{supp}(r)$ which estimates $P(\mathcal{X} \cup \mathcal{Y} \subseteq \mathcal{T})$, or short $P(\mathcal{X}\mathcal{Y})$. We also use the support of an itemset \mathcal{X} denoted by $\text{supp}(\mathcal{X})$.

3 Related Work

The strength of an association rule learner to discover all patterns is likewise its weakness. Usually the number of discovered associations can be immense, easily in the thousands or even tens of thousands. Clearly, the large numbers make rules difficult to examine by a human user. Therefore significant research has been conducted into methods which assess the relevance, or interestingness, of a rule. Studies concerning interestingness assessment can roughly be divided into two classes. The first class are objective measures. These are usually derived from statistics, information theory or machine learning and assess numerical or structural properties of a rule and the data to produce a ranking [6]. Objective measures do not take any background information into account and are therefore suitable if an unbiased ranking is required, e.g. in off-the-shelf data mining tools. The second class are subjective measures which incorporate a user's background knowledge. In this class a rule is considered interesting if it is either *actionable* or *unexpected*.

Actionability of a rule means that the user "can act upon it to his advantage" [7]. Their focal point is on rules that are advantageous for the user's goals. The actionability approach needs detailed knowledge about the current

goals and also about the cost and risks of possible actions. Systems that utilise it are hence very domain specific, like the *KEFIR* system described in [8].

A rule is unexpected if it contradicts the user's knowledge about the domain. Systems that build upon this approach require the user to express his domain knowledge – a sometimes difficult, long and tedious task. The methods are usually based on pairwise comparison of a discovered rule with rules representing the user knowledge. This comparison can be logic-based [9, 10, 11] or syntax-based [12]. In logic-based systems a contradiction is determined by means of a logical calculus, whereas in syntax-based systems a rule contradicts if it has a similar body but a dissimilar head.

In [9, 10, 11] the authors connect belief models with association rules. In particular, they assume that a belief system has been provided by the user whereby beliefs are defined as association rules. Based on this definition they provide a set of conditions to verify whether a rule $\mathcal{X} \rightarrow y$ is *unexpected* with respect to the belief $\mathcal{X} \rightarrow z$ on the rule database D . They propose an algorithm *ZoomUR* 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.

In [12] the authors 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. In subsequent work [13], human knowledge is recognised to have different degrees of certainty or preciseness. Their system allows for three degrees, notably *general impressions*, *reasonably precise concepts* and *precise knowledge*. The approach they propose accounts for these degrees and uses the gathered knowledge to find rules which are unexpected in regard to the expressed knowledge. The approach works iteratively: first, the user specifies his knowledge or modifies previously specified knowledge, supported by the specification language; second, the system analyses the association rules according to conformity and unexpectedness; and third, the user inspects the analysis results (aided by visualisation), saves interesting rules and discards uninteresting rules.

How to incorporate user dynamics into the relevance assessment has been studied in [3]. They propose an approach based on two models which a user has to specify prior to any analysis: a model of his existing knowledge and a model of how he likes to apply this knowledge. The degree of unexpectedness of each discovered rule is calculated with respect to these two models. Their approach is based on what they call the See-and-Know assumption. Once a user has seen a rule, the rule itself and similar rules are not of interest anymore. Our approach, in contrast, uses two classes of seen rules, relevant and non-relevant ones. The ranking is calculated by aggregating the (dis-)similarity of a rule with respect to rules in both classes. Our approach also does not require a user to specify any kind of prior model of a his knowledge.

4 Using Concepts from Information Retrieval

Existing approaches to assess the relevance of association rules strongly require a user to explicitly specify his existing knowledge in advance. This leads to two major drawbacks. In the first place, when specifying their existing knowledge, domain experts often forget certain key aspects or may not remember others which come into play under rarer circumstances. This problem can be termed ‘expert dilemma’ and has already been observed by designers of expert systems in the 1980s [14]. Secondly, at the beginning of an analysis session a user can only very vaguely specify what he considers to be relevant. His notion of relevance only becomes clearer the more rules he examines. This problem, that a user is incapable of specifying his information need from scratch, is very well-known in the field of information retrieval [5] where it led to the development of relevance feedback methods.

Relevance feedback is an intuitive technique that has been introduced to information retrieval in the mid-1960s [15]. In information retrieval it is a controlled, semi-automatic, iterative process for query reformulation, that can greatly improve the usability of an information retrieval system [16]. Relevance feedback allows a user to express what he considers to be relevant by marking rules as relevant and non-relevant, respectively. Whenever a rule has been marked as relevant, it is added to the set of relevant rules R_r . Whenever a rule is marked as non-relevant, it is added to the set of non-relevant rules R_n . For simplicity, we will assume that in each feedback cycle exactly one rule is marked.

After each feedback cycle the remaining rules are compared with the set of annotated rules and a new relevance score is calculated. The set of annotated rules, in turn, can be seen as a representation of the user’s notion of relevance. Hence it also provides a solution to the first of the above-mentioned drawbacks by supporting an iterative, easy way for a user to specify his knowledge about a domain. For example, he may annotate rules that are already known as non-relevant and some novel rules as relevant.

In order to develop a feedback system for association rules the following questions need to be answered:

- How do we represent association rules for the purpose of relevance feedback?
- How do we score the likely relevance of a rule in relation to a rule already marked as (non-)relevant?
- How do we aggregate those scores to an overall relevance score?

We will provide answers to these questions in the subsequent sections. In particular we are aiming at adapting established methods from information retrieval.

5 Rule Representation

To be the core building block of a relevance feedback approach it is necessary to transform the rules into an equivalent representation. In particular, such a representation should have a couple of properties. Firstly, rather than relying on generalisation and specialisation relationships among rules as a key to rule similarity it should support a less crisp and thus more flexible definition. For example, rules that have the same head and share items in their body should be regarded as similar to a certain degree. Secondly, items have a different importance to a user. For example, an item that is contained in almost every rule does not contribute much towards a user’s understanding of the domain, whereas an item that is only contained in a few rules can contribute considerably. This importance should be reflected in the rule representation. Thirdly, it should be easy to extend the rule representation by further numeric properties of a rule. For example, recently there has been an increasing interest into the change of a rule’s support and confidence values (e.g. [17]) as a key to rule interestingness. In this scenario the rule representation should incorporate the timeseries of support or confidence in order to enable similarity calculations based on rule change. To illustrate the usage of further information about rules for relevance feedback we will use the example of rule change throughout this paper.

As a representation that fulfills all of the above requirements we define a *feature vector* \vec{r} of an association rule r whose elements are numerical values and which consists of three components: a representation of the rule’s body, a representation of the rule’s head and a rule’s time series. The latter component can easily be replaced by other numeric features of a rule or completely omitted. Formally, a feature vector thus is defined as

$$\vec{r} = \underbrace{(r_1, \dots, r_b)}_{\text{symbolic}} \underbrace{(r_{b+1}, \dots, r_{b+h})}_{\text{timeseries}} \underbrace{(r_{b+h+1}, \dots, r_{b+h+t})}_{\text{timeseries}} \quad (1)$$

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

$$\vec{r}_{\text{body}} = (r_1, \dots, r_b) \quad (2)$$

$$\vec{r}_{\text{head}} = (r_{b+1}, \dots, r_{b+h}) \quad (3)$$

$$\vec{r}_{\text{sym}} = (r_1, \dots, r_{b+h}) \quad (4)$$

$$\vec{r}_{\text{time}} = (r_{b+h+1}, \dots, r_{b+h+t}) \quad (5)$$

To calculate the *item weights* r_i we adapted the well-known TF-IDF approach [18] from information retrieval. 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.

This method, carried over to association rules, means that items that appear in the vast majority of rules will get a very low weight whereas items that are rather infrequent will get a rather high weight. Since item appearance in rules is linked to item appearance in a data set this also means that infrequent attribute values in the data set will receive a high weight.

The term frequency tf of an item x in an association rule r is calculated as follows:

$$tf(x, r) = \begin{cases} 1 & \text{if } x \in r, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

The inverse document frequency idf of an item x in an association rule r and in regard to a rule set R is calculated as follows:

$$idf(x, R) = 1 - \frac{\ln |r : r \in R \wedge x \in r|}{\ln |R|} \quad (7)$$

To generate \vec{r}_{body} and \vec{r}_{head} , a series of steps has to be performed. For body and head separately, a set of items is generated: $I_{\text{body}} = \{x_1, \dots, x_b\}$ and $I_{\text{head}} = \{x_1, \dots, x_h\}$ where the x_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}_{body} or \vec{r}_{head} , respectively. Hence, the values for b and h in (1) are the cardinalities of the respective itemsets: $b = |I_{\text{body}}|$ and $h = |I_{\text{head}}|$

The part of the feature vector of an association rule r which covers body and head consists of TF-IDF values. Let x_i the i -th item of the alphabetically ordered set I_{body} and let r_i be the i -th component of \vec{r}_{body} . Then, \vec{r}_{body} is defined as follows:

$$r_i = tf(x_i, r) \cdot idf(x_i, R), \quad i = 1, \dots, b \quad (8)$$

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

$$r_{b+j} = tf(x_j, r) \cdot idf(x_j, R), \quad j = 1, \dots, h \quad (9)$$

6 Pairwise Similarity

A relevance feedback system must have the ability to compare unrated rules, or features of those, with rules previously rated as (non-)relevant. Instead of utilizing the generalisation and specialisation relationships among rules we choose a more flexible approach based on a notion of similarity among rules. As a similarity measure we have chosen the cosine similarity. It calculates the cosine of the angle between two n -dimensional vectors r and s as follows:

$$sim(\vec{r}, \vec{s}) = \frac{\sum_{i=1}^n r_i s_i}{\sqrt{r_i^2} \sqrt{s_i^2}} \quad (10)$$

Since the cosine measure yields values in $[0, 1]$, the corresponding dissimilarity measure therefore is:

$$dissim(\vec{r}, \vec{s}) = 1 - sim(\vec{r}, \vec{s}) \quad (11)$$

The cosine similarity compared to other similarity measures, like ones based on the Euclidean distance, has the advantage that it does not take missing items in a rule into account. For example, when measuring the similarity between a rule $\mathcal{X}y \rightarrow z$ and its more general rule $\mathcal{X} \rightarrow z$ only the item weights contained in both rules (i.e. \mathcal{X} and z) contribute towards the similarity measure. This property of the cosine measure is also the reason why it is frequently used in information retrieval systems. When comparing, for example, a query with a document it is desirable only to take the actual words contained in the query into account and not each of the many words the user did not specify.

The similarity between rules' bodies or rules' heads can be calculated straightforwardly using the cosine measure, yielding $sim(\vec{r}_{\text{body}}, \vec{s}_{\text{body}})$ and $sim(\vec{r}_{\text{head}}, \vec{s}_{\text{head}})$, respectively. By averaging both we obtain the similarity of a rule \vec{r}_{sym} with regard to a rule \vec{s}_{sym} :

$$sim(\vec{r}_{\text{sym}}, \vec{s}_{\text{sym}}) = 0.5sim(\vec{r}_{\text{body}}, \vec{s}_{\text{body}}) + 0.5sim(\vec{r}_{\text{head}}, \vec{s}_{\text{head}}) \quad (12)$$

The cosine measure is also suitable as a measure of time series similarity $sim(\vec{r}_{\text{time}}, \vec{s}_{\text{time}})$ which we use in this paper as an example of further information about rules embedded into the rule vector. For time series the cosine measure has the advantage only to reflect the magnitude of the angle between two vectors but—compared with other distance measures (e.g. Euclidean distance)—to ignore the magnitude difference between the two vectors. This means, it is robust w.r.t. different variation ranges of the time series. It is, however, not robust w.r.t. shifts of the time series mean value. Nevertheless, robustness can be achieved by subtracting from both time series their respective mean value prior to similarity calculation.

7 Similarity Aggregation

So far, we have discussed how to calculate pairwise similarities between vectors which represent certain features of a rule like its head, body or a time series of rule measures. For the purpose of relevance feedback it is necessary to measure the similarity of a feature of an unrated rule r relative to the features contained in the elements of a rule set R which may represent relevant and non-relevant rules. Generally, we define the similarity of a vector \vec{r} relative to a set $R = \{\vec{s}_1, \dots, \vec{s}_m\}$ as

$$sim_{rs}(\vec{r}, R) = \Omega(\{sim(\vec{r}, \vec{s}_1), \dots, sim(\vec{r}, \vec{s}_m)\}) \quad (13)$$

whereby Ω denotes a suitable aggregation operator which we will describe in the next section. As in Section 6, the dissimilarity of a vector relative to a set is defined as

$$dissim_{rs}(\vec{r}, R) = 1 - sim_{rs}(\vec{r}, R) \quad (14)$$

7.1 The OWA Operator

Our choice of the aggregation operator Ω is guided by two requirements: firstly, the user should be able to influence the aggregation operator, either implicitly or explicitly. Secondly, to obtain comparable results, the aggregation operator should be able to represent also simple aggregation operators like min, max or median. These two requirements are met by the family of OWA operators, which originate in the Fuzzy Domain and have been introduced by [19]. An OWA operator Ω is a mapping $\Omega : S \rightarrow \mathbf{R}$, where S is a set of numerical values s_i with $S \neq \emptyset$ and $|S| = n$. The OWA operator Ω has an associated weighting vector $W = (w_1, w_2, \dots, w_n)^T$ with $w_j \in [0, 1]$ and $\sum_{j=1}^n w_j = 1$. It is defined as

$$\Omega(\{s_1, s_2, \dots, s_n\}) = \sum_{j=1}^n w_j b_j \quad , \quad (15)$$

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.

Min, *max*, *mean*, and *median* are special cases for the OWA operator and were described by [20]. They illustrate the generality and flexibility of the OWA operator. By setting the weights accordingly, the user can influence the relevance score to suit the needs of his particular application scenario. For example, $(1/n, 1/n, \dots, 1/n)^T$ yields the mean, whereas $(1, 0, \dots, 0)^T$ yields the maximum operator.

Furthermore, the OWA operator is strongly related to the concept of linguistic quantifiers, such as *many*, *a few*, *most*. In [19] the connection to linguistic quantifiers is presented by explaining how the weights of the OWA expression can be obtained by using the membership function of any linguistic quantifier.

7.2 Relative Importance of Recent Relevance Choices

The retrieval of relevant association rules is a consecutive, iterative process. The user's knowledge, his beliefs and assumptions change during the relevance feedback cycle as he sees more rules. Therefore, the user's latest choices should be considered as having a higher priority over the first, relatively uninformed ones. This concept can be captured as the *decay of a relevant or non-relevant rule's importance over time*. The similarity aggregation should account for this and thus should weight recently selected rules higher than older ones.

Let $t(r)$ be the *age* of a relevant or non-relevant association rule r . This means, $t(r)$ is the number of feedback cycles that have been performed since the rule r was marked as being (non-)relevant, thereby a newly selected rule

receives $t = 0$. Two possibilities to model such a relevance decay are:

$$\tau_{exp}(r) = (1 - \delta)^{t(r)} \quad (16)$$

$$\tau_{lin}(r) = \max(1 - t(r) \cdot \delta, 0) \quad (17)$$

with (16) for an exponential type of decay and (17) for a linear decay down to a minimum of zero, whereby $\delta \in [0, 1]$ is a decay constant that controls the speed of decay.

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 our approach would only consider the user's latest relevance decision in its relevance score calculation. The value of $\delta = 0$ would deactivate the decay completely. Values of δ in between those bounds activate a gradual decay. Using the time weighted importance we refine our definition of a vector \vec{r} its similarity relative to a set R and yield

$$sim_{rs}(\vec{r}, R) = \Omega(\{\tau(\vec{s}_1)sim(\vec{r}, \vec{s}_1), \dots, \tau(\vec{s}_m)sim(\vec{r}, \vec{s}_m)\}) \quad (18)$$

8 Relevance Scoring

Based on the similarity measure we defined in the last section we can develop a notion of a rule's pairwise score, i.e. its relevance score with respect to a certain rule that was marked as relevant. While in information retrieval it is mostly assumed that those documents which are similar to (non-)relevant ones are (non-)relevant too, we use a slightly different approach.

For rules marked as relevant we assume that once a user has seen such a rule rather than being interested in similar ones his attention is attracted by those which are similar in certain features but dissimilar in others. This means, a user aims for rules which have an element of surprise. For example, a rule could have a very similar antecedent, but a rather dissimilar head when compared to a relevant one. It would therefore be surprising to a user because it is an exception to his previous knowledge. This approach also captures the case of rule contradiction employed by other authors [12, 11], albeit in a fuzzy, less crisp way.

Table 1 shows three of such interesting combinations of rule features. The case discussed above is named C_1 in this table. Another example is C_2 . It assigns a high score to those rules that are very different in their symbolic representation, but exhibit a similar time series. Such a combination can hint at an unknown hidden cause for the observed changes, which in turn are of interest to a user who typically will assume that only similar rules change similarly. The remaining entry C_3 is basically the inversion of the last one. A rule is considered interesting if it is similar to a relevant one, but has a very dissimilar time series

For rules marked as non-relevant we use an approach similar to the one used in information retrieval, i.e. rules that are similar to non-relevant ones are also considered non-relevant.

	<i>similar</i>	<i>dissimilar</i>		
			head	time series
body	C_1	-	-	
time series	-	-	C_2	
symbolic	-	C_3	-	

Table 1: Interestingness Matrix

Based on these considerations our calculation of the overall relevance score is split into two parts: one each for the relevant and non-relevant rules, respectively.

Our definition of the relevance of a rule with regard to the set of relevant rules is rather straightforward and shown in (19),(20) and (21) for the three cases mentioned above. To pick up on our examples from the previous section, using C_1 a rule receives a high relevance score if its body is similar to the rule bodies in R_r and its head dissimilar to the rule heads in R_r . Likewise, the score for C_2 is calculated by multiplying the similarity of the rule/rule set combination for the time series with the dissimilarity of the rule/rule set combination for the symbolic representation.

$$C_1 : \Phi(\vec{r}, R_r) = sim_{rs}(\vec{r}_{\text{body}}, R_r) dissim_{rs}(\vec{r}_{\text{head}}, R_r) \quad (19)$$

$$C_2 : \Phi(\vec{r}, R_r) = sim_{rs}(\vec{r}_{\text{time}}, R_r) dissim_{rs}(\vec{r}_{\text{sym}}, R_r) \quad (20)$$

$$C_3 : \Phi(\vec{r}, R_r) = sim_{rs}(\vec{r}_{\text{sym}}, R_r) dissim_{rs}(\vec{r}_{\text{time}}, R_r) \quad (21)$$

For the non-relevant rules we assume that rules in R_n 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. An unrated rule r should therefore receive a high interestingness score the more dissimilar it is from the set of non-relevant rules, i.e.

$$\Psi(\vec{r}, R_n) = dissim(\vec{r}, R_n) \quad (22)$$

Our final relevance score of an unrated rule r under consideration of the set of relevant and (non-)relevant rules consists of two parts, $\Phi(\vec{r}, R_r)$ and $\Psi(\vec{r}, R_n)$, which are both weighted to give the user more influence on the scoring.

$$F(\vec{r}, R_r, R_n) = w_{\text{rel}}\Phi(\vec{r}, R_r) + w_{\text{nrel}}\Psi(\vec{r}, R_n) \quad (23)$$

After every feedback cycle, i.e. after every update of R_r or R_n , each unrated rule r is being reevaluated whereby a new score $F(\vec{r}, R_r, R_n)$ is assigned. Rules which previously have been ranked as rather non-relevant can now receive a higher score whereas others may lose their relevance.

9 Conclusion

In this paper we have dealt with the well-known issue of finding the most relevant rules within a large set of association rules. By leveraging techniques and concepts from Information Retrieval we have proposed a novel method for association rule relevance feedback which has several advantages compared to existing methods. Firstly, it allows a user to refine his notion of relevancy over time by providing feedback. Secondly, it utilizes a more flexible notion of related rules based on vector similarity. Thirdly, it allows to incorporate further information about rules into the exploration process by turning the symbolic notion of a rule into a numeric feature vector.

Our approach is currently being trialed within BT Group to assist users in exploring association rule changes with the CRM (Customer Relationship Management) domain. So far, the results we obtained are quite promising and underlining the usefulness of our relevance feedback method.

References

- [1] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In *Proc. ACM SIGMOD 1993*, pages 207–216, Washington, DC, 1993.
- [2] 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.
- [3] Ke Wang, Yuelong Jiang, and Laks V. S. Lakshmanan. Mining unexpected rules by pushing user dynamics. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 246–255, 2003.
- [4] Dong Xin, Xuehua Shen, Qiaozhu Mei, and Jiawei Han. Discovering interesting patterns through user’s interactive feedback. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 773–778, New York, NY, USA, 2006. ACM Press.
- [5] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.
- [6] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right objective measure for association analysis. *Information Systems*, 29(4):293–313, 2004.
- [7] Abraham Silberschatz and Alexander Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):970–974, 1996.

- [8] G. Piatetsky-Shapiro and C. J. Matheus. The interestingness of deviations. In *Proceedings AAAI workshop on Knowledge Discovery in Databases*, pages 25–36, 1994.
- [9] Balaji Padmanabhan and Alexander Tuzhilin. Unexpectedness as a measure of interestingness in knowledge discovery. *Decision Support Systems*, 27, 1999.
- [10] Balaji Padmanabhan and Alexander Tuzhilin. Small is beautiful: discovering the minimal set of unexpected patterns. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 54–63, 2000.
- [11] Balaji Padmanabhan and Alexander Tuzhilin. Knowledge refinement based on the discovery of unexpected patterns in data mining. *Decision Support Systems*, 33(3):309–321, 2002.
- [12] Bing Liu, Wynne Hsu, and Shu Chen. Using general impressions to analyze discovered classification rules. In *Proceedings of the 3rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 31–36, 1997.
- [13] Bing Liu, Wynne Hsu, Shu Chen, and Yiming Ma. Analyzing the subjective interestingness of association rules. *IEEE Intelligent Systems*, 15(5):47–55, 2000.
- [14] David B. Fogel. The advantages of evolutionary computation. In D. Lundh, B. Olsson, and A. Narayanan, editors, *Bio-Computing and Emergent Computation*. World Scientific Press, Singapore, 1997.
- [15] Gerard Salton. *The SMART Information Retrieval System*. Prentice Hall, Englewood Cliffs, NJ, 1971.
- [16] Tommi Jaakkola and Hava Siegelmann. Active information retrieval. In *Advances in Neural Information Processing Systems 14*, pages 777–784. MIT Press, 2001.
- [17] Mirko Boettcher, Detlef Nauck, Dymitr Ruta, and Martin Spott. Towards a framework for change detection in datasets. In *Proceedings of the 26th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 115–128. Springer, 2006.
- [18] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 5(24):513– 523, 1987.
- [19] Ronald R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Trans. Syst. Man Cybern.*, 18(1):183–190, 1988.

- [20] 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.