TUM

INSTITUT FÜR INFORMATIK

Quantitative Associaton Rules Based on Half-Spaces: An Optimization Approach

Ulrich Rückert, Lothar Richter, and Stefan Kramer

TUM-I0412
August 04

TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM-INFO-08-I0412-0/1.-FI

Druck: Institut für Informatik der
Technischen Universität München

Quantitative Associaton Rules Based on Half-Spaces: An Optimization Approach

Ulrich Rückert, Lothar Richter, and Stefan Kramer

Institut für Informatik, Technische Universität München
D-80290 München

Abstract

We tackle the problem of finding association rules for quantitative data. Whereas most of the previous approaches operate on hyperrectangles, we propose a representation based on half-spaces. Consequently, the left-hand side and right-hand side of an association rule does not contain a conjunction of items or intervals, but a weighted sum of variables tested against a threshold. Since the downward closure property does not hold for such rules, we propose an optimization setting for finding locally optimal rules. A simple gradient descent algorithm optimizes a parameterized score function, where iterations optimizing the first separating hyperplane alternate with iterations optimizing the second. Experiments with two real-world data sets show that the approach is feasible and in fact finds meaningful patterns. We therefore propose quantitative association rules based on half-spaces as an interesting new class of patterns with a high potential for applications.

1 Introduction

Soon after the introduction of association rules for itemsets, researchers began to realize that association rules would also be useful for quantitative data [11]. Most of the generalizations and extensions of association rules to quantitative data either require a discretization of the numerical attributes or a characterization of the numerical attributes in the right-hand side by their means and standard deviations. The discretization process, however, leads to a loss of information in the data set. In the following we present a novel approach that works directly on the continuous data, without the need for any discretization or the calculation of statistical moments. It derives quantitative association rules of the form "if the weighted sum of some variables is greater than a threshold, then a different weighted sum of variables is with high probability greater than a second threshold". For instance, consider a table with wind strength, temperature and the wind chill index. Approaches so far applied to this data would approximate the relationship among the variables by a bundle of quantitative association rules. In contrast, the approach proposed here would find a weighted sum of wind strength and temperature on the left-hand side and the wind-chill index on the right-hand side. Thus, it allows for the discovery of non-axis-parallel regularities and can account for cumulative effects of several variables. Since the downward closure property frequently used in conventional association rule mining does not hold for this type of rule, we cast the problem of finding such rules as an optimization problem. The aim is to find rules that are locally optimal with respect to a parameterized score function. Consequently, the user can adjust the parameters of the presented algorithm to obtain association rules that match her individual interests. For instance, it is possible to specify target values for certain parameters, such that the algorithm attempts to find rules near the target (penalizing rules that are too far off), while simultaneously optimizing the rules' confidence. The whole framework is very flexible in several directions and can easily be adapted to incorporate user constraints. In summary, the paper has two main contributions: Firstly, the *representation* of quantitative association rules based on half spaces, and secondly, the *optimization setting* for finding such rules.

The paper is organized as follows: section 2 introduces the representation of quantitative association rules based on half-spaces. Section 3 elaborates on the optimization setting for finding such rules. A scoring function is defined, and an optimization algorithm is sketched. Finally, we present the results of some experiments in section 4, discuss related work in section 5, before we conclude in section 6.

2 Quantitative Association Rules Based on Half-Spaces

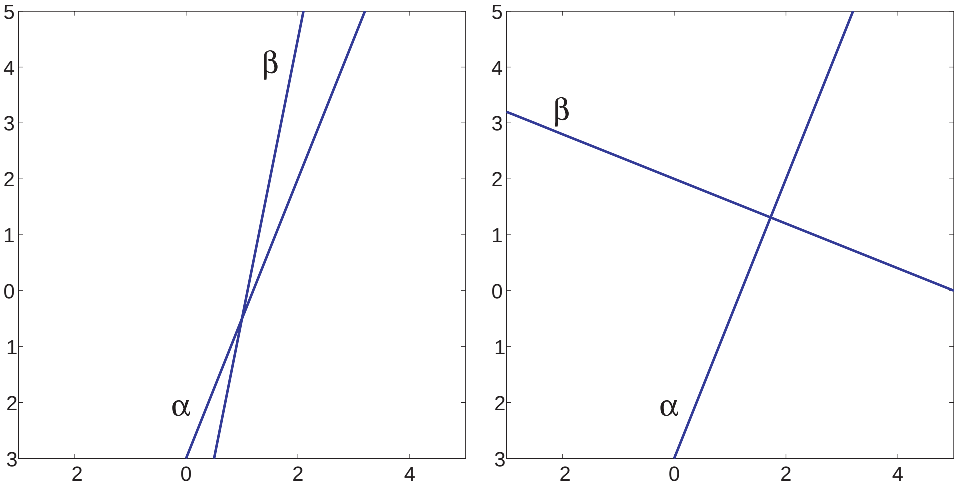
As outlined above, the aim of this paper is to extend the association rule framework to quantitative data. In general, an association rule is an implication of the form "if the left-hand side condition is true for an instance, then, with high probability, a right-hand side condition is also true". In the traditional setting, the conditions on the right-hand side and left-hand side are based on hyperrectangles of discrete attributes. To extend association rules to continuous data, we therefore need to decide which kind of "conditions" the quantitative association rules should be based on.

Of course, there are lots of different ways to impose conditions on numerical data. At the core we would expect from a useful condition that it separates the instance space in two subspaces, the space of instances that meets the condition, and the one that does not. The border between those two subspaces can then be conveniently expressed by some *separation function*. For numerical data, it makes sense to select a smooth separation function to minimize the error that is caused by random noise or measurement errors in the data. In this paper we will focus on *hyperplanes*, a particularly simple, but powerful class of separation functions. However, large parts of this paper also apply to more complex separation functions. From a geometrical perspective, a hyperplane α is given by a vector $\bar{\alpha}$ and an intercept α_0. An instance x is then assigned to one half-space, if the scalar product $\bar{\alpha} \cdot x + \alpha_0$ is positive and to the other half-space, if it is negative. In figure 1 (b), the one-dimensional hyperplane α (i.e. a line) separates the two-dimensional space into two half-spaces, one left of α, the other right of α.

In the case of association rules, the use of hyperplanes as conditions boils down to testing a weighted sum of variables against a threshold; i.e. an instance x in an n-dimensional space meets the condition $\alpha \in \mathbb{R}^{n+1}$, if

$$\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n \geq -\alpha_0 \tag{1}$$

With this, one could build an association rule such as $x_1 \geq 31 \rightarrow 0.9x_5 + 1.2x_6 \geq 250$. In a particular medical application this association rule might be interpreted as "if the body mass index is greater than or equal to 31, then the weighted sum of the systolic and diastolic blood pressure is greater than or equal to 250". Obviously, there are many cases where finding such a quantitative association rule might lead to valuable insights into the structure of the data at hand.



(a) (b)

Figure 1: Two non-perpendicular hyperplanes α and β (a), and two perpendicular hyperplanes α and β, separating the instance space into four subspaces (b).

Of course, it is quite easy to generate a large number of trivial association rules with high confidence. For example, the association rule $1.5x_1 \geq 5 \rightarrow 2x_1 \geq 4$ has confidence 100%, but does not give any new insight. More generally, situations like the one in figure 1 (a) are problematic: we have two hyperplanes α and β in a two-dimensional space, that define an association rule $\alpha_1 x_1 + \alpha_2 x_2 \geq -\alpha_0 \rightarrow \beta_1 x_1 + \beta_2 x_2 \geq -\beta_0$. The problem is that α and β are highly correlated. If an instance is left of the α hyperplane, it is very likely to be left of the β hyperplane as well, simply because the space that is right of β, but left of α is much smaller than the space left of α and left of β[1]. Because of that, the association rule has high confidence even for randomly generated data; it does not give much information about the data at hand. For our purposes it is therefore essential, that α and β are uncorrelated, i.e. they have to be perpendicular as in figure 1 (b). This is, of course, the case, if the scalar product is zero:

$$\sum_{i=1}^{n} \alpha_i \cdot \beta_i = 0 \tag{2}$$

Note that this requirement does not prevent using the same variables on the left-hand and the right-hand side of an association rule. For example, the association rule $x_1 + x_2 > 2 \rightarrow x_1 - x_2 > 0.2$ is perfectly valid, because $\binom{1}{1} \cdot \binom{1}{-1} = 0$. At first sight this might seem to be a strange finding. However, it is often easy to come up with a reasonable explanation for such a rule. For instance, if we know, that in the example above the x_1 and x_2 values are always positive, we could (loosely) interpret the rule as “if x_1 and x_2 are sufficiently large, then x_1 is larger than x_2 by a margin of at least 0.2”. This might be a valuable insight in the structure of the data set at hand. In the next section we describe an algorithm that is able to find such quantitative association rules.

[1] Of course, in a strict mathematical sense it does not make sense to compare the “sizes” of subspaces, because all subspaces are infinite anyway. A more formal justification would demand that the resulting probability distributions are independent for uniform data.

3 Quantitative Association Rule Mining

The main problem with finding good quantitative association rules is that the space of rules is uncountably infinite and therefore not suited to an enumeration strategy as employed by APriori [1]. In particular, the downward closure property does not hold for such rules, and thus we have to abandon the idea of generating the complete set of solutions. However, we can adopt an optimization approach, where the user can specify clearly the sort of rules she is looking for, and the algorithm returns locally optimal solutions. While this may seem unusual for association rule mining, it is common practice in other areas, for instance clustering (e.g, K-means clustering) and Bayesian learning (e.g., the EM algorithm).

In the following we describe one particular algorithm for mining quantitative association rules in this setting. First, we define a score function to assess the "interestingness" of an association rule. Then, we sketch a simple optimization algorithm searching for association rules with a low score. Before we go into further detail, however, we need to introduce the basic setting and some notational conventions.

For mining quantitative association rules we are given a *data set* X containing m *instances*. Each instance is given as a vector of n real values, i.e. $x \in \mathbb{R}^n$, so that $X \subset \mathbb{R}^n$. We are now looking for association rules that are defined by two hyperplanes $\alpha := (\alpha_0, \alpha_1, \dots, \alpha_n)^T$ and $\beta := (\beta_0, \beta_1, \dots, \beta_n)^T$. The α hyperplane specifies the condition on the left-hand side of the association rule, the β hyperplane specifies the right-hand side. Both hyperplanes are given in Hessian normal form: the α_0 value of a hyperplane α is the *intercept*, i.e. the hyperplane's distance to the origin. The *direction vector* $\bar{\alpha} := (\alpha_1, \dots, \alpha_n)^T$ specifies the slope of the hyperplane. As a notational shortcut we use $\bar{\alpha}$ to denote the direction vector part of α and α_0 to denote the intercept of α. Usually, the direction vector is normalized so that $|\bar{\alpha}| = 1$ and the distance between an instance x and α is simply $\bar{\alpha}^T x + \alpha_0$. However, to allow for an efficient optimization procedure, we will sometimes allow non-normalized direction vectors. In this case we have to use a modified distance function $\delta(\alpha, x)$ to calculate the distance between a hyperplane and an instance:

$$\delta(\alpha, x) := \frac{\bar{\alpha}^T x}{|\bar{\alpha}|} + \alpha_0 \tag{3}$$

This distance δ simply normalizes the direction vector before calculating the actual distance. For the optimization procedure, we also need the derivatives of δ:

$$\frac{\partial \delta(\alpha, x)}{\partial \alpha_0} = 1; \quad \frac{\partial \delta(\alpha, x)}{\partial \alpha_j} = \frac{1}{|\bar{\alpha}|}\Big(x_j - \frac{\bar{\alpha} \cdot x}{|\bar{\alpha}|^2}\alpha_j\Big) \tag{4}$$

For later use, we denote the derivative along the jth axis by δ'_j:

$$\delta'_j(\alpha, x) := \frac{\partial \delta(\alpha, x)}{\partial \alpha_j} \tag{5}$$

The score function is a combination of four different criteria: *confidence*, *coverage*, *contrast* and *sparseness*. It is designed to be differentiable to make it amenable to optimization approaches. Confidence is defined as for ordinary association rules, while coverage reflects the number of instances covered by the left-hand side of a rule. Contrast is a measure of how evenly the instances are distributed if the left-hand side does *not apply*. Thus, it measures whether the left-hand side makes any

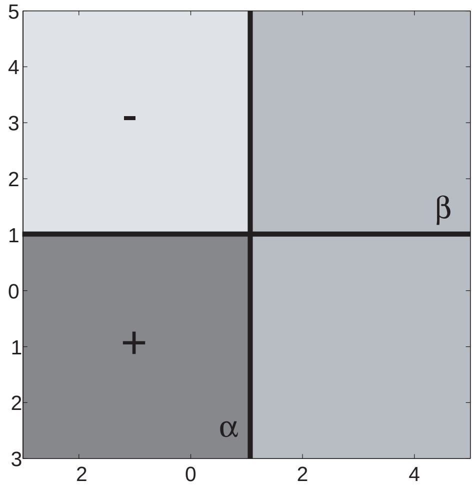


Figure 2: Confidence is optimal if the distribution is uneven left of α, while contrast is optimal if it is even right of α.

difference. Sparseness is a term penalizing overly complex rules. If only a few variables occur in a quantitative association rule, then its sparseness value will be small.

To summarize, the setting is as follows: The user specifies a target coverage and a target sparseness of the rules. Then the optimization algorithm is run until a local optimum with respect to the score function is found. The algorithm optimizes the confidence, under the constraint that the contrast is balanced, targeting at user-specified values for the rule's coverage and sparseness. Random restarts are usually performed to return several of those locally optimal association rules. In the following, we will formally define the components of the score function.

3.1 Confidence

We are mainly interested in association rules with high *confidence*, i.e. the fraction of instances in X, that fulfill both conditions α and β divided by the fraction of instances that fulfill only the α condition should be as high as possible. Figure 2 illustrates this idea: we consider only instances that fulfill the α condition, i.e. that are left of the α hyperplane. If an instance x is located left of α and below β, it contributes to a high confidence score. If it is located left of α, but above β, it decreases the confidence measure. Thus, we have the following maximization problem at hand:

$$\max_{\alpha,\beta} \sum_{x \in X} \mathbf{I}[\delta(\alpha, x) \geq 0] \operatorname{sgn}(\delta(\beta, x)) \tag{6}$$

where $\mathbf{I}[\dots]$ denotes the indicator function that is one if the condition in the brackets is fulfilled and zero otherwise.

Unfortunately, this "confidence score" has two disadvantages. First of all, it is not differentiable and thus not suited for standard numerical optimization techniques. Second, and more importantly, it assigns the same weight to all instances, independent of the actual distance of the instance to the hyperplanes. In practice, most values are not known exactly, but only up to a certain measurement error. Thus, it makes sense to regard an instance that is left of the α hyperplane, but very close to it,

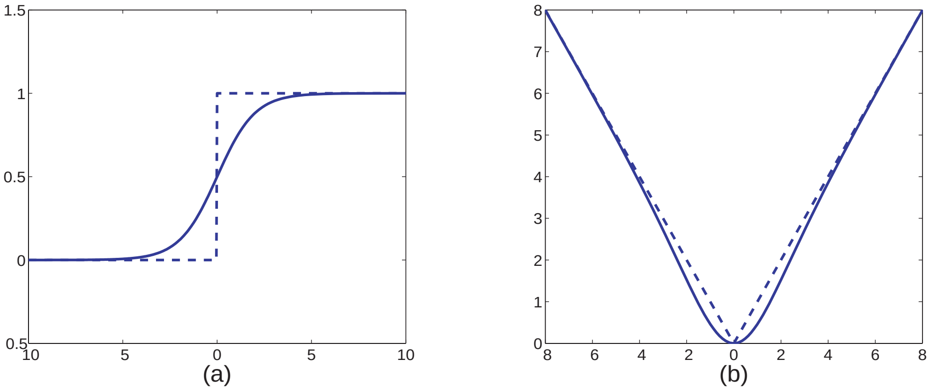


Figure 3: The sigmoid function as a replacement for the step function (a), and the τ function replacing the absolute value function (b).

as “probably left of α”. Such an instance should not contribute the same weight to the optimization problem as an instance that is very far from α and thus “certainly left of α”. One common approach to address those considerations is to use the *sigmoid* function. The sigmoid function is given by

$$\sigma(x) := \frac{1}{1 + e^{-x}} \tag{7}$$

and plotted in figure 3 (a). In its basic form, it is a “smoother” version of the step function $\mathbf{I}[x \geq 0]$. Since it assigns intermediate values in the vicinity of the origin, it takes the distance of an instance to one of the hyperplanes better into account than the sharp step function. The sgn function is just a rescaled version of the step function: $\operatorname{sgn}(x) = 2\mathbf{I}[x \geq 0] - 1$. Consequently we can use the sigmoid function for a better handling of instances near the β hyperplane as well.

With this, and by reformulating the maximization problem as a minimization problem, we get the optimization problem $\min_{\alpha,\beta} l(\alpha, \beta, X)$, where l is defined as follows:

$$l(\alpha, \beta, X) := -\sum_{x \in X} \sigma(\delta(\alpha, x)) \cdot (2\sigma(\delta(\beta, x)) - 1) \tag{8}$$

For later use, we also give the derivatives with respect to α and β:

$$\frac{\partial l(\alpha, \beta, X)}{\partial \alpha_j} = -\sum_{x \in X} (2\sigma(\delta(\beta, x)) - 1) \cdot \sigma'(\delta(\alpha, x)) \cdot \delta'_j(\alpha, x) \tag{9}$$

$$\frac{\partial l(\alpha, \beta, X)}{\partial \beta_j} = -\sum_{x \in X} \sigma(\delta(\alpha, x)) \cdot 2\sigma'(\delta(\beta, x)) \cdot \delta'_j(\beta, x) \tag{10}$$

where σ' is the derivative of the sigmoid function:

$$\sigma'(x) := \frac{e^{-x}}{(1 + e^{-x})^2} = e^{-x}\sigma^2(x) \tag{11}$$

3.2 Coverage

A second criterion for the interestingness of an association rule is its *coverage*. The coverage is simply the fraction of instances in the data set that satisfy the left-hand side condition. Together with the confidence, the coverage determines the support of the association rule, i.e. the fraction of instances that fulfills both conditions. Unfortunately, the coverage of interesting rules is not clear a priori. If the coverage of an association rule is very large, the rule is true for almost the whole data set. Such rules often express trivial dependencies in the data. On the other hand, if the coverage of a rule is very small, the pattern describes a very local phenomenon that might just be a random fluctuation instead of a structural property of the underlying data. Thus, the coverage values for interesting association rules are somewhere in between, depending on the data at hand and the knowledge about the data. In practice, the desired coverage (or equivalently, the support) is often determined empirically.

To take these considerations into account, we design a parameterized coverage interestingness function $c(\alpha, g, t, X)$, that leaves the choice of the "target coverage" as parameter t to the user. The function should be low, if the coverage of the association rule given by α is near the desired target coverage t and high otherwise. In this way the user can adjust the scoring function according to his perception about which coverage is interesting. The following function fulfills this requirement. It is zero, if the coverage is equal to t, and increases linearly with the factor g as α departs from this optimum:

$$c(\alpha, g, t, X) := g \cdot \Big| \sum_{x \in X} \big(\operatorname{sgn}(\delta(\alpha, x)) - (2t - 1) \big) \Big| \tag{12}$$

Using the factor g the user can fine-tune the importance of the desired coverage in relation to the interestingness score for the confidence. If g is set to one, confidence and coverage are treated equally. If g is set to a lower value, confidence is weighted higher than coverage during the optimization process.

Again, one can argue that instances very close to the hyperplane should be weighted lower than instances far from the hyperplane. As above, this is achieved by replacing the sgn function with a scaled sigmoid function. The absolute value function $|x|$ is not differentiable at $x = 0$. This might be a problem during the optimization process, because the optimization procedure might get stuck in the induced "peak" optimum, even though nearby settings for α might have slightly worse coverage, but better confidence scores. To avoid this problem, we replace the absolute value with a modified sigmoid function $\tau(x)$:

$$\tau(x) := \frac{2x}{1 + e^{-x}} - x \tag{13}$$

As can be seen in figure 3 (b), this function resembles the absolute value function, except for the area around the origin, where $\tau(x)$ is slightly lower. This leads to the following coverage interestingness function:

$$c(\alpha, g, t, X) := g \cdot \tau\Big(\sum_{x \in X} (2\sigma(\delta(\alpha, x)) - 2t) \Big) \tag{14}$$

With this we can estimate the interestingness of an association rule with regard to confidence and coverage. Again, we give the derivatives of c regarding to α:

$$\frac{\partial c(\alpha, g, t, X)}{\partial \alpha_j} = g \cdot \tau' \Big(\sum_{x \in X} (2\sigma(\delta(\alpha, x)) - 2t) \Big) \cdot \sum_{x \in X} (2\sigma'(\delta(\alpha, x)) \cdot \delta'(\alpha, x)) \tag{15}$$

where $\tau'(x)$ is the derivative of $\tau(x)$:

$$\tau'(x) := 2\sigma(x) + 2x\sigma'(x) - 1 \tag{16}$$

3.3 Contrast

The confidence and coverage scores determine what the optimization algorithm is looking for on the left side of α in figure 2: confidence requires that the lower left subspace contains more instances than the upper left subspace, while coverage determines the fraction of instances that are left of α. Just like in traditional association rule mining, there is no constraint regulating the distribution of instances on the right side of α. For quantitative association rule mining, this can be a problem: one can simply move the β hyperplane upwards until it is located above all instances. While this achieves maximal confidence, the resulting association rule is not very interesting, because the right-hand side condition is true for all instances anyway. One way to overcome this problem is to regulate the distribution of instances that are right of α with regard to β. One might be tempted to demand that most instances right of α should be located above β. However, this would generate association equivalences of the form "$\bar{\alpha}^T x \geq -\alpha_0 \leftrightarrow \bar{\beta}^T x \geq -\beta_0$" instead of implications in one direction as in traditional association rules. For our purposes it seems to be more sensible to ask for an even distribution of instances above and below β.

We call this criterion *contrast*. The rationale is that the "contrast" between the instance above and below β should be as low as possible on the right side of α in figure 2. This ensures that the optimization algorithm searches for *local* patterns, that is, patterns that hold for the specified subset of patterns, but not for the reverse or general case. The following contrast scoring function $r(\alpha, \beta, X)$ formalizes the idea. It is zero, if the number of instances in the upper right subspace in figure 2 is equal to the number of instances in the lower right subspace and increases linearly otherwise.

$$r(\alpha, \beta, X) := \Big| \sum_{x \in X} \mathbf{I}[\delta(\alpha, x) < 0] \operatorname{sgn}(\delta(\beta, x)) \Big| \tag{17}$$

Again, it makes sense to replace the sgn and the absolute value function with differentiable counterparts to accommodate for noisy data and avoid unwanted local optima. This yields:

$$r(\alpha, \beta, X) := \tau\Big(\sum_{x \in X} \sigma(-\delta(\alpha, x))(2\sigma(\delta(\beta, x)) - 1) \Big) \tag{18}$$

For the optimization algorithm in section 3.5 we also need the derivatives of r:

$$\frac{\partial r(\alpha, \beta, X)}{\partial \alpha_j} = \tau'\Big(\sum_{x \in X} \sigma(-\delta(\alpha, x))(2\sigma(\delta(\beta, x)) - 1) \Big) \cdot \sum_{x \in X} (2\sigma(\delta(\beta, x)) - 1)\sigma'(-\delta(\alpha, x))(-\delta'_j(\alpha, x)) \tag{19}$$

$$\frac{\partial r(\alpha, \beta, X)}{\partial \beta_j} = \tau'\Big(\sum_{x \in X} \sigma(-\delta(\alpha, x))(2\sigma(\delta(\beta, x)) - 1) \Big) \cdot \sum_{x \in X} \sigma(-\delta(\alpha, x)) 2\sigma'(\delta(\beta, x))\delta'_j(\beta, x) \tag{20}$$

Algorithm 1 The frame algorithm, where iterations optimizing the first separating hyperplane alternate with iterations optimizing the second

```
procedure QAR($t, g, h, X$)
    $\alpha \leftarrow$ a random vector
    $\beta \leftarrow$ a random vector
    repeat
        $y \leftarrow L(\alpha, \beta, t, g, h, X)$
        $\alpha \leftarrow$ LineSearch1($\alpha, \beta, t, g, h, X$)
        $\beta \leftarrow$ LineSearch2($\alpha, \beta, t, g, h, X$)
    until $|L(\alpha, \beta, t, g, h, X) - y| \leq 0.1$
    return $(\alpha, \beta)$
end procedure
```

3.4 Sparseness

The three preceding scoring functions give sufficient information to identify values of α and β that are unusual or interesting enough to justify a further analysis. However, for humans who have to interpret the resulting association rules, there is one more pragmatic criterion: the components of the α and β vectors of an quantitative association rule with low confidence, coverage and contrast scores are usually not zero. This means that the resulting association rule contains n addends on both sides of the implication. It is hard and cumbersome work to identify which coefficients contribute significantly to the confidence, coverage and contrast of the association rule, and which coefficients can be omitted without changing the scores too much. Usually, the user prefers finding *sparse* association rules, i.e. rules where most coefficients are zero and only the relevant coefficients are given. Those rules are shorter and thus easier to interpret and validate.

To account for these pragmatic considerations, one can add a term to penalize non-sparse association rules. Both, α and β are normalized, so that the sum of the components is one. Thus, to receive sparse vectors we only need to increase the variance in the components, so that we have many very low (ideally zero-valued) components and a small number of large components. The following function $a(\alpha, h)$ expresses this property formally. It is zero, if one coefficient is one and the others are zero. In the worst case, where all components have value $\sqrt(1/n)$, the score is maximal at $hm - \frac{hm}{n}$.

$$a(\alpha, h) := hm - hm \sum_{i=1}^{n} \Big(\frac{\alpha_i}{|\bar{\alpha}|}\Big)^4 \tag{21}$$

The parameter h determines how large this penalty should be in relation to the other scores. If sparseness is very important to the user, she should set it to a high value near one. Again, we give the derivative for later use in the optimization procedure:

$$\frac{\partial a(\alpha, h)}{\partial \alpha_j} = 4hm\Big(\sum_{i=1}^{n} \frac{\alpha_i^4 \alpha_j}{|\bar{\alpha}|^6} - \frac{\alpha_j^3}{|\bar{\alpha}|^4}\Big) \tag{22}$$

If one incorporates this penalty function during the search for association rules, the induced rules will have many coefficients near zero. Unfortunately, they are not necessarily exactly zero. However, one can set those coefficients to zero in a post-processing step without changing confidence, coverage, and contrast too much. This post-processing step is explained in section 3.6.

Algorithm 2 The line search algorithm to perform a gradient descent step that maintains perpendicularity. This version keeps β fixed and optimizes for α. LineSearch2 uses the same algorithm, but keeps α fixed and optimizes for β. Note that $\bar{x}$ denotes the direction vector part of x.

```
procedure LineSearch1($\alpha, \beta, t, g, h, X$)
    repeat
        $y \leftarrow L(\alpha, \beta, t, g, h, X)$
        $\lambda \leftarrow \nabla L(., \beta, t, g, h, X)$
        $\mu \leftarrow \lambda$
        $\bar{\mu} \leftarrow \bar{\mu} - (\bar{\mu}^T \bar{\beta})\bar{\beta}$
        $s \leftarrow 0.5$
        while $L(\alpha + s\mu, \beta, t, g, h, X) > y - 0.001s(\mu^T \lambda)$ do
            $s \leftarrow 0.9s$
        end while
        $\alpha \leftarrow \alpha + s\mu$
        $\bar{\alpha} \leftarrow \frac{\bar{\alpha}}{|\bar{\alpha}|}$
    until $|L(\alpha, \beta, t, g, h, X) - y| \leq 0.01$
    return $\alpha$
end procedure
```

3.5 The Algorithm

With the discussion in the previous sections we have a number of criteria to decide whether a particular α and β define an interesting quantitative association rule. The final interestingness scoring function $L(\alpha, \beta, g, t, h, X)$ simply calculates the sum of those scores

$$L(\alpha, \beta, g, t, h, X) := l(\alpha, \beta, X) + c(\alpha, g, t, X) + r(\alpha, \beta, X) + a(\alpha, h) + a(\beta, h) \qquad (23)$$

A high score indicates that the association rule is uninteresting with regard to the selected parameter settings, a low score means we found an interesting rule. As the scoring function is continuous, there usually is a whole subspace of “good” rules and it is easy to modify a rule with a low score to some small extent and obtain a rule with an even lower score. We are therefore aiming at finding association rules with optimally low score, that is, the local optima of the scoring function, subject to the constraint that $\bar{\alpha}^T \bar{\beta} = 0$.

This constrained optimization problem can be tackled using established methods from optimization theory. A standard approach is to introduce a Lagrange multiplier and use one of the many published optimization algorithms to solve the resulting optimization problem with $2n + 3$ variables. This can be a hard optimization problem for large values of n. We take a different approach that alternatingly keeps α fixed while optimizing β and vice versa. In this way one solves a sequence of $n + 1$-dimensional optimization problems. Empirical results in section 4 indicate that only a few iterations are sufficient to find such an optimum.

For the sake of simplicity we use a simple gradient descent method in each iteration. In the following description of the algorithm we hold β fixed and optimize for α. The other case can be derived simply by using β as optimization variable and using the gradient with regard to β as search direction. First, the algorithm calculates the gradient $\bar{\lambda} = \nabla L$ of L with regard to $\bar{\alpha}$ as the locally best descent direction. However, we can not use this direction for the line search, because we might

Algorithm 3 A postprocessing algorithm to derive a sparse version of the found association rule.

```
procedure MakeSparse($\alpha, \beta, t$)
    Replace all values less than $t$ in $\bar{\alpha}$ with 0
    Replace all values less than $t$ in $\bar{\beta}$ with 0
    $\bar{\gamma} \leftarrow \bar{\alpha}$
    for $i = 1$ to $n$ do
        if $\bar{\beta}_i = 0$ then
            $\bar{\gamma}_i \leftarrow 0$
        end if
    end for
    if $|\bar{\gamma}| > 0$ then
        $\bar{\beta} \leftarrow \bar{\beta} - \frac{(\bar{\gamma}^T \bar{\beta})\bar{\gamma}}{|\bar{\gamma}|^2}$
    end if
    return $(\frac{\bar{\alpha}}{|\bar{\alpha}|}, \frac{\bar{\beta}}{|\bar{\beta}|})$
end procedure
```

end up with a vector that is not perpendicular to $\bar{\beta}$. We therefore calculate $\bar{\mu} = \bar{\lambda} - (\bar{\lambda}^T \bar{\beta})\bar{\beta}$, that is, the vector component of $\bar{\lambda}$, that is perpendicular to $\bar{\beta}$. The line search is then performed in the $\bar{\mu}$ direction. Thus, after the line search we have the new value $\bar{\alpha}' = \bar{\alpha} + s\bar{\mu}$ for some scalar s that yields the lowest score according to L along the line. The following equation shows that the new value $\bar{\alpha}'$ is still perpendicular to $\bar{\beta}$.

$$\begin{aligned}
\bar{\alpha}'^T \bar{\beta} &= [\bar{\alpha} + s(\bar{\lambda} - (\bar{\lambda}^T \bar{\beta})\bar{\beta})]^T \bar{\beta} \\
&= \bar{\alpha}^T \bar{\beta} + s\bar{\lambda}^T \bar{\beta} - s(\bar{\lambda}^T \bar{\beta})\bar{\beta}^T \bar{\beta} \\
&= \bar{\alpha}^T \bar{\beta} + s\bar{\lambda}^T \bar{\beta} - s\bar{\lambda}^T \bar{\beta} \qquad (24) \\
&= 0 \qquad (25)
\end{aligned}$$

(24) uses the fact that $\bar{\beta}^T \bar{\beta} = |\bar{\beta}|^2 = 1$, because $\bar{\beta}$ has unit length and (25) follows, because the starting vector $\bar{\alpha}$ is perpendicular to $\bar{\beta}$ anyway. The intercept α_0 is not subject of the perpendicularity constraint and we can simply use the derivative of L with regard to α_0 as intercept component for the line search. For the line search loop we use a simple backtracking approach with the Armijo condition as termination criterion.

As any other optimization procedure, this algorithm can get stuck in local optima with comparably high scores. For the sake of simplicity we use random restarts to obtain association rules with low score. Of course, one can utilize simulated annealing or any other global optimization strategy as well.

3.6 Postprocessing and Visualization

In section 3.4 we introduced a penalty term for non-sparse vectors. If the user provides a high weight h for this term, the vectors $\bar{\alpha}$ and $\bar{\beta}$ of the resulting association rule contain many components near zero and only a few large components. However, they are not necessarily sparse in the sense that most components are zero. Fortunately, the hyperplanes do not change too much, if one simply sets all components below a certain threshold to zero. In most cases this operation does not change

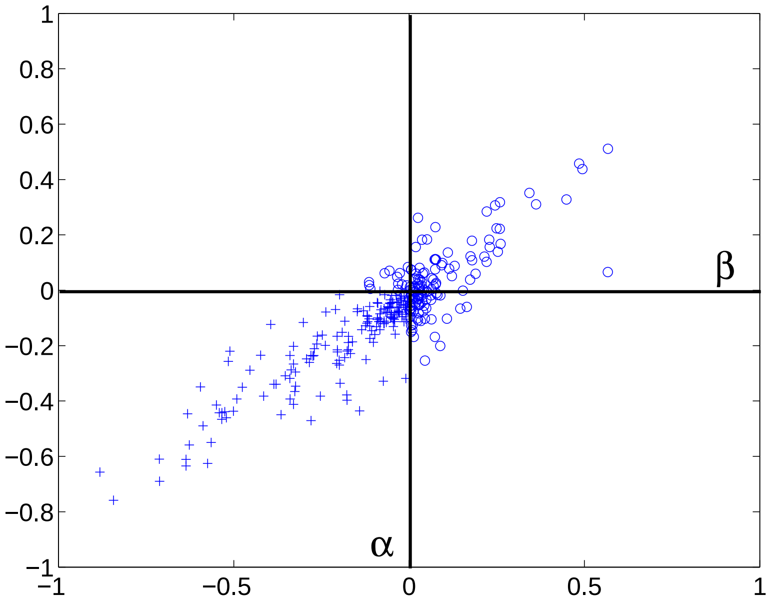


Figure 4: A scatter plot of the left-hand side term and the right-hand side term of the fifth association rule in table 1.

confidence, coverage and contrast too much. The only problem one may encounter is that the resulting $\bar{\alpha}$ and $\bar{\beta}$ are not perpendicular anymore. As outlined in algorithm 3, this can be easily fixed by determining the vector component of $\bar{\beta}$, that is perpendicular to $\bar{\alpha}$ and does not modify the zero-valued components in $\bar{\beta}$.

In practice, finding and displaying an association rule is just a small step in the knowledge discovery process. The user usually needs to validate and interpret the association rule, gather more information about the phenomenon at hand and finally assess its usefulness. A nice property of quantitative association rules is the fact that the left-hand side and right-hand side conditions are essentially projections from the instance space to $\mathbb{R}$, together with a threshold. The projections calculate the distance between an instance and the condition's hyperplane. We can therefore visualize the distances between the instances and the two hyperplanes in a scatter plot, where the x-axis specifies the the distance to the α hyperplane and the y-axis the distance to the β plane. Such a diagram can be quite useful to gather more information about the kind of dependency between the left-hand side and the right-hand side of an association rule. For example, figure 4 (visualizing the fifth rule in table 1) suggests, that there might indeed be a (noisy) linear correlation between the left-hand side and the right-hand side. Such information can be valuable for assessing the relevance and usefulness of the pattern at hand.

4 Experimental Results

To assess the applicability and feasibility of the described algorithm, we implemented a version in MATLAB. In the following paragraphs we describe experiments on two data sets. As a proof of the principle, we give some interesting results on a microarray data set and test the robustness of the induced rules. Additionally, we assess the scalability of our implementation using the larger "Cover

Id	Quantitative Association Rule	Score	Cov.	Supp.	Contr.	Conf.
(1)	0.13*PHO84 + 0.99*INO1 $<$ 0.02 → YER135C $>$ 0.04	-122.1	0.38	0.34	0.64	0.89
(2)	-0.17*SNZ1 - 0.98*GPH1 $>$ 0.04 → -0.13*INO1 + 0.11*YDR010C - 0.98*YLL059C - 0.11*PGU1 $>$ 0	-111.6	0.43	0.36	0.56	0.83
(3)	SOR1 $<$ 0.04 → ZRT1 $>$ 0.03	-100.7	0.53	0.46	0.44	0.88
(4)	YMR031W-A $<$ 0.02 → -0.98*FIG1 + 0.11*PHO84 + 0.11*YOR382W+ 0.11*YHR126C $>$ 0.06	-93.0	0.47	0.37	0.68	0.79
(5)	ZRT1 $>$ 0 → 0.10*PHO84 - 0.99*YLL059C $>$ 0	-89.8	0.41	0.32	0.63	0.78

Table 1: Some of the generated quantitative association rules.

Type” data set [3].

4.1 Yeast Gene Expression

For our first experiment, we chose the gene expression data set of Hughes *et al.* [7]. The data set was generated using microarray technology: the expression levels of 6316 genes in the yeast genome were measured for 300 diverse mutations and chemical treatments of yeast cells. The compendium is given as a table with 300 instances, where each value specifies the log base 10 of the fold change. High positive values indicate overexpressed genes, negative values denote underexpressed genes. We selected the 50 genes with the largest standard deviation for our experiments. The goal of the experiments was to show that the patterns are meaningful and potentially useful for domain experts, and that the algorithm is able to find non-random patterns.

Quantitatitive association rules based on half spaces are an interesting type of representation for the analysis of microarray data, because biochemical networks usually consist of main pathways as well as “side roads” that can be used if the other ways are blocked. This applies particularly to the Hughes dataset, where the biochemical network is exposed to all sorts of stress (chemicals, etc). Weighted sums of variables are a suitable means to model this kind of phenomenon in *one* rule: the big players obtain larger weights in the rule, while the substitutes obtain only smaller weights.

We performed experiments with the t parameter set to 0.5, g set to 1.0, and the sparseness parameter h set to 0.1, 0.3 and 0.5, respectively. Table 1 gives five of the best rules, together with their “interestingness score”, coverage, support, contrast and confidence. As can be seen, a number of rules with high confidence can be found. Contrast and coverage are centered around the target value of 0.5.

Rule (1) states that the less inositol and the less of a phosphate transporter is generated, the more of a hypothetical transmembrane protein is generated. Inositol is part of membrane lipides and thus important for cell growth. Phosphate uptake is also essential for the cell. It might be the case that the cell reacts to a stop of cell growth with the generation of the transmembrane protein. Rule (2) states that if the cell reduces SNZ1 and GPH1 (glycogen degradation) as a reaction to the depletion of nutrients, then growth is also reduced (INO1) and new energy resources are tapped (PGU1 – polygalacturonidase). Rule (3) states that if SOR1 is low, then ZRT1 is generated. SOR1 is a zinc-dependent enzyme, and ZRT1 is a high-affinity zinc transport protein. Thus, the lack of SOR1 stimulates zinc transport. Rule (4) reflects the switch from normal growth to mating. Rule (5) states that if the cell is in need of zinc, then it is also in need of phosphate, and YLL059C, supposedly an inorganic phosphate transporter, becomes superfluous.

However, even though the rules do not contradict current wisdom on regulatory pathways, we

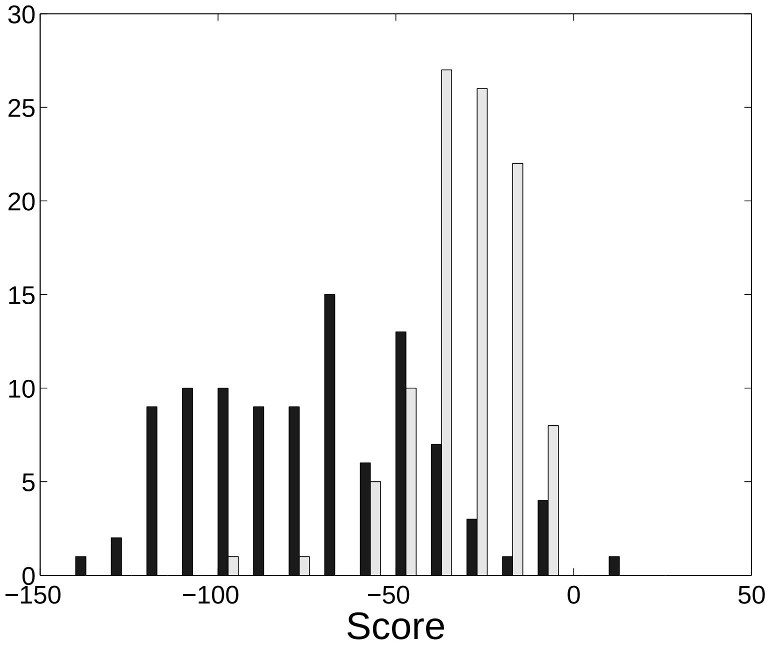


Figure 5: Distribution of scores on the original (black) and the permuted (grey) data sets.

have no statistical evidence about the significance of the rules. After all, it may well be that the generated rules only describe random fluctuations instead of an inherent structural property. We therefore performed a permutation test to assess the robustness of the rules. We permute the values in each column of the data set randomly to generate a new data set with the same distribution, but no structural relations between the columns. We then run the algorithm ten times on the permuted data set and note the best score found. This process is repeated one hundred times to get an estimate of the distribution of scores, that can be expected on random, but similar data. Figure 5 gives the resulting histograms for the original and the permuted data. The scores for the permuted data are peaked around -30, while the original data features a large number of association rules in the range between -50 and -150. Thus, we can be highly confident, that the induced rules describe indeed structural properties of the yeast data set. In practical applications, we would recommend this randomization approach to focus on significant findings.

4.2 Cover Type

The goal of the second experiment is to investigate the scalability of the optimization algorithm with regard to the size of the data set. We therefore chose the “Cover Type” data set containing 581,012 instances from the UCI repository [3]. We removed the discrete attributes, leaving ten continuous attributes describing cartographic properties of 30 x 30 meter land cells. We normalized the data set, so that each column has a mean of zero and a standard deviation of one. We then applied the optimization algorithm on subsets of different size, with the t parameter set to 0.5, g set to 1, and h set to 0.5. The experiments were performed on a Pentium IV 2.8GHz machine. As the runtime of the optimization algorithm depends on the number of line search steps and the runtime per line search, the actual runtime varies for different random restarts.

Data Set Size	Overall Runtime	Number of Line Searches	Runtime per Line Search
100	2.6 s	19	0.14 s
1,000	7.4 s	27	0.28 s
5,000	23 s	22	1.0 s
10,000	46.9 s	17	2.7 s
50,000	264 s	18	14.7 s
100,000	623 s	23	27.1 s
300,000	2004 s	23	87.2 s
500,000	3529 s	20	176.5 s

Table 2: Runtimes of the optimization algorithm as a function of data set size.

We therefore give the total runtime, the number of line searches that were performed, and the runtime per line search for the various data set sizes in table 2. The table shows that the number of line search steps remains below thirty for all data set sizes and that the runtime per search step scales favorably with the data set size.

5 Related Work

The first approach to quantitative association rules was due to Piatetsky-Shapiro [10], where the left-hand side and right-hand side of the rules were tested for equality with numeric constants.

Most of the subsequent approaches to quantitative association rules can be categorized as either interval-based [11, 9, 4, 15, 5, 13] or distribution-based [2, 12]. In the former case, the items on the left-hand side and right-hand side of the rules are defined as tests for intervals of variables. In the latter case, the numerical attributes in the right-hand side are characterized by their means and standard deviations.

The first interval-based approach discretizing the numerical attributes was proposed by Srikant and Agrawal [11]. It is interesting to note that all previous approaches based on intervals had to face similar trade-offs balancing support and confidence as our half-space association rules. Fukuda *et al.* [5] presented an efficient algorithm for quantitative association rules using computational geometry and sampling methods. While it scales up well in the size of the database, the right-hand side is restricted to exactly one categorical variable. Zhing *et al.* [15] proposed to cluster the data to improve an interval-based approach. Fukuda *et al.* [4] and Yoda *et al.* [14] introduced variants with two numerical variables on the left-hand side and one Boolean item on the right-hand side.

Ultimately, all interval-based methods have to discretize the numerical attributes in one way or the other, which inevitably leads to a loss of information. Regularities that are not axis-parallel cannot be detected or have to be approximated by several quantitative association rules. Also, cumulative effects of several numeric variables cannot easily be represented.

More recently, distribution-based approaches have been proposed: Lindell and Aumann [2] considered two types of rules. The first type contains several categorical variables on the left-hand side and a vector of the means of several numerical variables on the right-hand side. The second type consists of exactly one discretized numerical variable on the left-hand side and exactly one mean on the right-hand side. Webb [12] contributed efficient algorithms for distribution-based association rules. While this line of research is very interesting and has a high potential for applications, the expressiveness of such rules is also quite restricted, because it is not possible to relate arbitrary sets of continuous variables. Quantitative association rules based on half spaces seem to be a special case of *projection pursuit* [6], a fairly broad statistical concept developed in 1970's. It is an open question whether a reformulation of the present approach in the framework of projection pursuit would bring any algorithmic improvement. Finally, the approach presented in this paper is also related to research on subgroup discovery [8], where the goal is to search for subgroups in the population with statistically interesting properties.

6 Conclusion

We proposed a new representation for quantitative association rules based on half spaces and an optimization setting for such rules. The approach does not require a discretization step and enables the detection of regularities that are not axis-parallel. In the design of the algorithm, many trade-offs are involved. Among others, we have to balance the confidence and the coverage to find good rules. However, a relatively simple optimization algorithm is sufficient to find locally optimal half-space rules. From an algorithmic point of view, many improvements and extensions are conceivable. For instance, one could replace the scalar product with a kernel to obtain a more complex separation function or incorporate more sophisticated optimization techniques to improve the algorithm's performance. Also, the score function is obviously just one of several conceivable possibilities. Another point is that it easy to incorporate user constraints in the process to support the user interactively in exploratory data analysis. Finally, we believe that quantitative association rules based on half-spaces are an interesting new class of patterns with a high potential for applications, e.g., in transcriptomics (microarray data) and proteomics.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 12–15 1994.

[2] Y. Aumann and Y. Lindell. A statistical theory for quantitative association rules. *Journal of Intelligent Information Systems*, 20(3):255–283, 2003.

[3] C. Blake and C. Merz. UCI repository of machine learning databases, 1998.

[4] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. In *Proc. ACM SIGMOD Int. Conf. Management of Data*, pages 13–23. ACM Press, 1996.

[5] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Mining optimized association rules for numeric attributes. In *Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 182–191. ACM Press, 1996.

[6] P. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[7] T. Hughes et al. Functional discovery via a compendium of expression profiles. *Cell*, 102:109–126, July 2000.

[8] W. Kloesgen. Exploration of simulation experiments by discovery. In *Proceedings of KDD-94 Workshop*, pages 251–262, 1994.

[9] R. J. Miller and Y. Yang. Association rules over interval data. In *Proc. ACM SIGMOD Int. Conf. Management of Data*, pages 452–461. ACM Press, 1997.

[10] G. Piatetsky-Shapiro. Discovery, analysis and presentation of strong rules. In G. Piatetsky-Shapiro and W. Frawley, editors, *Knowledge Discovery in Databases*, pages 229–248. AAAI/MIT Press, 1991.

[11] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In H. V. Jagadish and I. S. Mumick, editors, *Proc. of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 1–12, 1996.

[12] G. I. Webb. Discovering associations with numeric variables. In *Proc. of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 383–388. ACM Press, 2001.

[13] J. Wijsen and R. Meersman. On the complexity of mining quantitative association rules. *Data Mining and Knowledge Discovery*, 2(3):263–281, 1998.

[14] K. Yoda, T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Computing optimized rectilinear regions for association rules. In *Proc. 3rd International Conference on Knowledge Discovery and Data Mining*, pages 96–103. AAAI Press, 1997.

[15] Z. Zhing, Y. Lu, and B. Zhang. An effective partitioning-combining algorithm for discovering quantitative association rules. In *Proc. of the First Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 1997.