

# Multi-Class Correlated Pattern Mining

Siegfried Nijssen<sup>1</sup> and Joost N. Kok<sup>2</sup>

<sup>1</sup> Albert-Ludwigs-Universität, Georges-Köhler-Allee, Gebäude 097, D-79110, Freiburg im Breisgau, Germany.

<sup>2</sup> LIACS, Leiden University, Niels Bohrweg 1, 2333 CA, Leiden, The Netherlands.  
`sniijssen@informatik.uni-freiburg.de`

**Abstract.** To mine databases in which examples are tagged with class labels, the minimum correlation constraint has been studied as an alternative to the minimum frequency constraint. We reformulate previous approaches and show that a minimum correlation constraint can be transformed into a disjunction of minimum frequency constraints. We prove that this observation extends to the multi-class  $\chi^2$  correlation measure, and thus obtain an efficient new  $O(n)$  prune test. We illustrate how the relation between correlation measures and minimum support thresholds allows for the reuse of previously discovered pattern sets, thus avoiding unnecessary database evaluations. We conclude with experimental results to assess the effectivity of algorithms based on our observations.

## 1 Introduction

One of the oldest and most popular problems in machine learning is that of classification. Classification algorithms are applicable to all databases in which examples are tagged with class labels. Surprisingly, within inductive database theory the problem of classification has received little attention. In this paper we study a problem related to classification, which was first proposed by Bay and Pazzani [2, 3] and later by Morishita and Sese [11]. These authors studied the problem of mining *contrast sets* (name proposed by Bay and Pazzani) or *correlated itemsets* (name proposed by Morishita and Sese). Both terms refer to the same straightforward problem: given a database and a function which computes a measure of correlation between a pattern and a target attribute in the database, can we find all patterns that satisfy a minimum *correlation* constraint? Clearly, from a conceptual point of view this problem is very similar to the frequent itemset mining problem, which is to find all patterns that satisfy a minimum *support* constraint. Compared to the minimum support constraint, the minimum correlation constraint is however computationally more difficult as it is neither monotonic nor anti-monotonic. Given that highly correlated patterns can be useful features for classification algorithms, it can be argued that minimum correlation is a constraint that should be supported by inductive databases. This point was observed earlier, and besides Bay, Pazzani, Morishita and Sese, also other authors have proposed algorithms to mine correlated patterns, for example Zimmermann and De Raedt [13], or the closely related *class association rules* of Liu et al. [10] and *subgroups* of Kavšek et al. [9].

In comparison with these previous approaches, this paper introduces one fundamentally new idea: that there is a relation between disjunctions of minimum frequency constraints and minimum correlation constraints. This simple, but important observation allows us to improve previous results and provide deeper insight in the use of correlation constraints in inductive databases:

- In previous research [13] it was implied that to prune branches in a search for correlated patterns, an  $O(2^d)$  test for each node in this tree would be required, where  $d$  is the number of class values. We show that an  $O(d)$  test is sufficient.
- One of the supposed key features of inductive databases is that they treat patterns as data, and that queries can also be defined on sets of patterns. We show that many searches for highly correlated patterns can be reformulated as filtering operations over sets of frequent patterns. Thus, once we have built a set of frequent patterns, our observations show which different kinds of correlation queries can be formulated over these patterns, without accessing the data from which the patterns were obtained. This allows for the reuse of pattern sets for multiple purposes.

The paper is organized as follows. In Section 2 we recall the problem of correlated pattern mining and the notion of ROC spaces. We introduce the basic idea of linking minimum frequency to minimum correlation. In Section 3 we consider the more complex  $\chi^2$  and information gain correlation measures, for the case of two classes. In Section 4 we extend this approach to multiple classes. Section 5 discusses how to compute minimum support thresholds, and illustrates how sets of patterns can be reused. Section 6 compares our approach to the work of Bay, Pazzani, Morishita and Sese. Section 7 lists several ways of using our observations in APRIORI-like algorithms, and provides experimental results. Section 8 concludes.

## 2 Plotting Frequent Patterns in ROC Space

In classification problems we consider databases  $\mathcal{D}$  of examples, where each example is labeled by one class in a domain of classes  $\mathcal{C}$  through a function  $f : \mathcal{D} \rightarrow \mathcal{C}$ ; we denote by  $\mathcal{D}_c$  the set of examples for which the class label is  $c$ . Rule learners repeatedly search for rules of the form  $x \rightarrow c$ , where  $c$  is a class label in  $\mathcal{C}$ ,  $x$  is a pattern in a pattern language  $\mathcal{X}$  and a cover relation  $\succeq$  is defined between patterns in  $\mathcal{X}$  and examples in  $\mathcal{D}$ . Rule learners search rules for which  $\rho(x \rightarrow c)$  is maximized or minimized, for a measure  $\rho$  such as accuracy, weighted accuracy, gain, or  $\chi^2$ . The measure is computed from the *contingency table*. In binary classification problems this table can be represented as follows:

$a_1(x)n_1$	$(1 - a_1(x))n_1$	$n_1$
$a_2(x)n_2$	$(1 - a_2(x))n_2$	$n_2$
$a_1(x)n_1 + a_2(x)n_1$	$n_1 + n_2 - a_1(x)n_1 - a_2(x)n_2$	$n_1 + n_2$

Here  $n_1$  is the number of examples in class 1,  $n_2$  is the number of examples in class 2 and  $a_i(x)$  is the fraction of examples of class  $i$  that is covered by the body of rule  $x \rightarrow c$ . We call  $a_i(x)$  the *frequency* of pattern  $x$  in class  $i$ . When this is clear of the context we do not denote the argument  $x$  of the  $a$  function. For convenience we furthermore denote  $N = \sum_{i=1}^d n_i$ .

When inducing a classifier from a dataset the sizes of the classes ( $n_i$ ) can be considered to be fixed. Here we furthermore assume that the head of the rule is fixed to class 1. In *Receiver Operating Characteristic curve (ROC) analysis*  $a_1(x)$  is known as the *true positive rate* (TPR) and  $a_2(x)$  is the *false positive rate* (FPR). A *ROC graph* is a graph in which rules are depicted in the FPR-TPR plane [7]. Ideally a rule has a FPR of zero and a TPR of one; the corresponding point, which is depicted in the upper left corner of the ROC graph, is known as *ROC heaven*. Heuristics of classification algorithms can be conceived as measures that determine how far from ROC heaven a classifier is.

We will start our investigation by considering the very simple *accuracy* measure, which can be formalized as  $(a_1(x)n_1 + (1 - a_2(x))n_2)/N$ , and is a function of the vector  $\mathbf{a}(x) = (a_1(x), a_2(x))$ , so we can write

$$\rho_{acc}(x \rightarrow c) = \rho_{acc}(\mathbf{a}(x)) = \rho_{acc}(a_1(x), a_2(x)) = (a_1(x)n_1 + (1 - a_2(x))n_2)/N.$$

Adapting terminology proposed by [11], we call vector  $\mathbf{a}(x)$  the stamp point of pattern  $x$ . In this paper for patterns the only property of importance is their stamp point. Usually we therefore unify a pattern with its stamp point and drop the  $x$  from our notation. If we solve the equation  $\rho_{acc}(\mathbf{a}) = \theta$  for an accuracy value  $\theta$ , we obtain the following *isometric* of possible stamp points that achieve this accuracy:

$$\begin{aligned} \frac{a_1 n_1 + (1 - a_2) n_2}{N} = \theta & \iff a_1 n_1 - a_2 n_2 = \theta N - n_2 \\ & \iff a_1 = \frac{\theta N - n_2}{n_1} + a_2 \frac{n_2}{n_1}, \quad (1) \end{aligned}$$

which is a straight line in the ROC graph. An example for this isometric with  $n_1 = 20$ ,  $n_2 = 40$  and  $\theta = \frac{44}{60}$  is given in Figure 1.

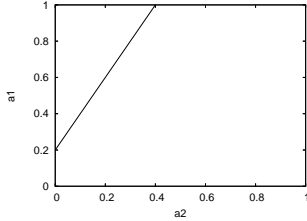
The essential observation is the following. If we consider *all* rules for which the accuracy is higher than  $\frac{44}{60}$ , then *all* these rules also have a frequency in class 1 which is higher than  $\frac{2}{10}$  (enter  $a_2 = 0$  into equation 1 to verify this). The minimum accuracy constraint can therefore be transformed into a tight *minimum frequency* constraint on one class.

More formally, let  $\mathbf{b}(i)$  denote the vector  $(b_1, \dots, b_d)$  where  $b_i = 1$  and  $b_j = 0$  for  $j \neq i$ . Then to find the threshold  $\theta_1$  on the frequency of class 1 we need to solve the equation

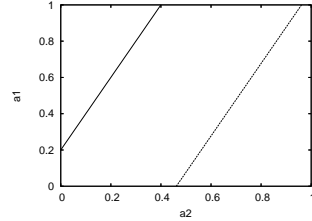
$$\rho(\theta_1 \mathbf{b}(1)) = \theta.$$

For accuracy we have that

$$\theta_1 = \frac{\theta N - n_2}{n_1}.$$



**Fig. 1.** An isometric for the accuracy measure.



**Fig. 2.** An isometric for the class neutral accuracy measure.

1. Transform minimum accuracy  $\theta$  into minimum support  $\theta_1$  for class 1;
2. Mine all frequent patterns  $\mathcal{F}$  in  $\mathcal{D}_1$  with minimum support  $\theta_1$ ;
3. Determine the support in  $\mathcal{D}_2$  of all frequent patterns in  $\mathcal{F}$ ;
4. Prune all patterns from  $\mathcal{F}$  for which accuracy is lower than  $\theta$ ;

**Fig. 3.** A simple algorithm for mining patterns with high accuracy.

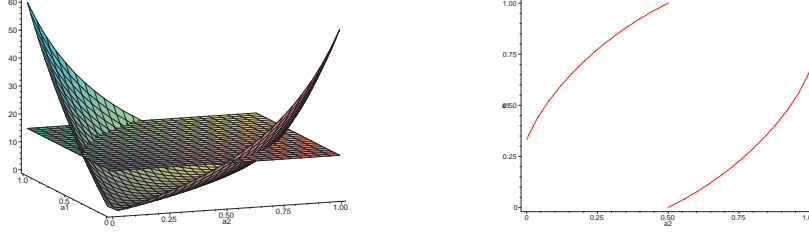
At this point we wish to give an example of the consequences of this observation. Consider the algorithm of Figure 3. Then the observation shows that this algorithm is correct: to mine patterns with high accuracy we can use any frequent pattern mining algorithm and postprocess its results.

The algorithm in the figure first determines the *entire* set of frequent patterns; their frequencies in the other part of the data are evaluated second, thus *postprocessing* results. A different approach is to evaluate each frequent pattern *immediately* in the second part of the data (thus *mixing* the evaluation). We will return later in more detail to these different approaches.

Given these observations on the 2 dimensional case of two target classes, the question is how this applies to higher numbers of classes and other correlation measures. This is what we study in the rest of the paper.

### 3 Class Neutral Measures

In the previous section we assumed that we only search for rules that have a fixed class in the head of the rule. Usually, one is interested in patterns that correlate with one of the target classes, independent of which class this is. In that case, a *class neutral* measure should be used. A simple class neutral measure is  $\max\{\rho_{acc}(x \rightarrow 1), \rho_{acc}(x \rightarrow 2)\}$ , which maximizes the correlation over all possible consequences. For some threshold value  $\theta$  the isometric is depicted in Figure 2. In comparison with the original accuracy measure there is now a ‘second ROC heaven’. To find all patterns that achieve a certain accuracy, a single minimum frequency no longer suffices. A second minimum frequency is



**Fig. 4.** The  $\chi^2$  correlation measure and the plane corresponding to a threshold value (left) and its isometric (right).

necessary, this time for the second class. Thus we have to solve two equations:

$$\rho(\theta_1 \mathbf{b}(1)) = \theta \quad \text{and} \quad \rho(\theta_2 \mathbf{b}(2)) = \theta. \quad (2)$$

Then, we have to find all patterns for which the frequency exceeds a minimum threshold value, either on the first class, or on the second class, or on both; the minimum frequency constraint on  $\mathbf{a}(x)$  is thus

$$a_1(x) \geq \theta_1 \quad \vee \quad a_2(x) \geq \theta_2. \quad (3)$$

Besides accuracy many other correlation measures are in common used. One of these is the  $\chi^2$  statistic, which is the main focus of this work. The  $\chi^2$  statistic is computed as follows. Let  $E_{i1} = (a_1 n_1 + a_2 n_2) n_i / N$ ,  $E_{i2} = ((1 - a_1) n_1 + (1 - a_2) n_2) n_i / N$ ,  $O_{i1} = a_i n_i$  and  $O_{i2} = (1 - a_i) n_i$ , then

$$\chi^2(\mathbf{a}) = \frac{(O_{11} - E_{11})^2}{E_{11}} + \frac{(O_{12} - E_{12})^2}{E_{12}} + \frac{(O_{21} - E_{21})^2}{E_{21}} + \frac{(O_{22} - E_{22})^2}{E_{22}}.$$

The  $\chi^2$  measure and an isometric are depicted in Figure 4, for  $n_1 = 20$ ,  $n_2 = 40$  and  $\theta = 15$ . Therefore also for  $\chi^2$  we can obtain thresholds by solving equation (2) and using equation (3) as pruning constraint; again the minimum frequency thresholds of the classes are determined by the points where the  $\chi^2$  statistic crosses the  $a_1$  and  $a_2$  axis, respectively.

Just like for accuracy, there is a simple expression to compute the  $\theta_i$  values. We postpone this computation however to Section 5, at which point we have introduced  $\chi^2$  for higher numbers of classes.

More-or-less similar in shape to the  $\chi^2$  measure is information gain:

$$\begin{aligned} \rho_{\text{gain}}(\mathbf{a}) = & -\frac{n_1}{N} \log \frac{n_1}{N} \log -\frac{n_2}{N} \log \frac{n_2}{N} + \frac{a_1 n_1 + a_2 n_2}{N} (P_{11} \log P_{11} + P_{21} \log P_{21}) \\ & + \frac{(1 - a_1) n_1 + (1 - a_2) n_2}{N} (P_{12} \log P_{12} + P_{22} \log P_{22}), \end{aligned}$$

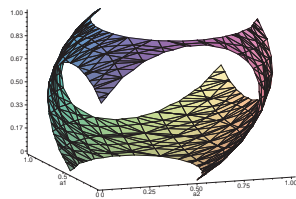
where  $P_{i1} = \frac{a_i n_i}{a_1 n_1 + a_2 n_2}$  and  $P_{i2} = \frac{(1 - a_i) n_i}{(1 - a_1) n_1 + (1 - a_2) n_2}$ . The gain measure can be treated similar as the  $\chi^2$  measure: the points where the gain isometric crosses the  $a_1$  and  $a_2$  axes, respectively, determine the minimum frequency thresholds for each of the two classes.

## 4 More than two Classes

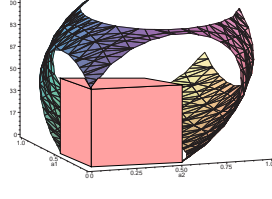
Until now only situations were considered in which there are two target classes. In general, however, there may be multiple target classes. To measure whether there is a correlation between a pattern and the target classes, we will consider the  $\chi^2$  and information gain measures here; in the next section we will consider accuracy. The contingency table is easily extended to the multi-class case:

$a_1 n_1$	$(1 - a_1) n_1$	$n_1$
$a_2 n_2$	$(1 - a_2) n_2$	$n_2$
$\vdots$	$\vdots$	$\vdots$
$a_d n_d$	$(1 - a_d) n_d$	$n_d$
$\sum_{i=1}^d a_i n_i$	$\sum_{i=1}^d (1 - a_i) n_i$	$N$

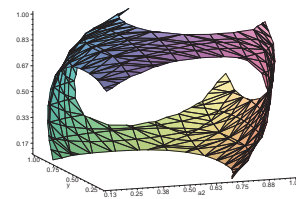
The definitions of  $E_{i1}$ ,  $E_{i2}$ ,  $O_{i1}$  and  $O_{i2}$ , are straightforwardly extended to define  $\chi^2$  as  $\chi^2(\mathbf{a}) = \sum_{i=1}^d \frac{(O_{i1} - E_{i1})^2}{E_{i1}} + \frac{(O_{i2} - E_{i2})^2}{E_{i2}}$ . Similarly, also the definition of gain ratio is extended. To give an impression of the shape of higher dimensional  $\chi^2$  and information gain measures, isometrics for three-class classification problems are given in Figure 5 and Figure 7.



**Fig. 5.** Isometric for  $\chi^2$  in a three-class classification problem.



**Fig. 6.** Isometric for  $\chi^2$  in a three-class classification problem; can a box be fitted within the isometric?



**Fig. 7.** Isometric for information gain in a three-class classification problem.

One of the main contributions of this paper is to answer this question: suppose that we want to find all itemsets for which  $\chi^2$  or information gain exceeds a predefined threshold value, is it possible to define a minimum frequency threshold on each of the classes, similar to the two dimensional case? Intuitively, this means that we want to prove that it is possible to put a ‘box’ completely inside the isometric body, such that the corners of the box are determined by the points where the isometric crosses the axes, as illustrated in Figure 6. In this section we provide an outline of our proof. Details are given in the Appendix.

First, we introduce some notation. Let us denote by  $\mathcal{B}_d$  the set of all vectors  $(b_1, b_2, \dots, b_d)$  such that  $b_i \in \{0, 1\}$ . These vectors can be considered to be the corners of a higher dimensional unit rectangle. For example,  $\mathcal{B}_2 = \{(0, 0), (1, 0), (0, 1), (1, 1)\}$ . By  $\mathcal{B}_{d, \geq k}$  we denote the subset of vectors in  $\mathcal{B}_d$  for which the sum

of  $b_i$  components is higher than  $k$ . As an example,  $\mathcal{B}_{2,\geq 1} = \{(1, 0), (0, 1), (1, 1)\}$  and  $\mathcal{B}_{2,\geq 2} = \{(1, 1)\}$ .

**Definition 1.** A  $d$ -dimensional function  $\rho$  is a suitable correlation function iff it satisfies the following two properties:

- $\rho(a_1, a_2, \dots, a_d)$  is convex;
- for every  $\mathbf{b} \in \mathcal{B}_{d,\geq 2}$ , every  $0 \leq \alpha \leq 1$  and every  $1 \leq k \leq d$  it must hold that:

$$\rho(\alpha \cdot b_1, \dots, \alpha \cdot b_{k-1}, \alpha \cdot b_k, \alpha \cdot b_{k+1}, \dots, \alpha \cdot b_d) \leq \rho(\alpha \cdot b_1, \dots, \alpha \cdot b_{k-1}, 0, \alpha \cdot b_{k+1}, \dots, \alpha \cdot b_d).$$

As an example, consider the  $\chi^2$  test for two classes. Among others, in [11] it was shown that  $\chi^2$  defines a convex function. The same can be shown for our  $\chi^2$  function. The set  $\mathcal{B}_{2,\geq 2}$  consists of one single vector  $\{(1, 1)\}$ . As  $\chi^2(\alpha, \alpha) = 0$  it is clearly true that  $\chi^2(\alpha, \alpha) \leq \chi^2(\alpha, 0)$  and  $\chi^2(\alpha, \alpha) \leq \chi^2(0, \alpha)$ , for all  $0 \leq \alpha \leq 1$ . This shows that the  $\chi^2$  test for two classes defines a suitable correlation function. Note that the  $\chi^2$  function has several peculiar properties ( $\chi^2(1, 0) = \chi^2(0, 1) = n_1 + n_2$  and  $\chi^2(\alpha, \alpha) = 0$ ), but that correlation functions are not required to have these properties within our framework. We have the following theorem.

**Theorem 1.** Let  $\rho$  be a suitable correlation function. Consider a stamp point  $\mathbf{a} = (a_1, a_2, \dots, a_d)$  and let  $S_{\mathbf{a}}$  be the set of all stamp points  $(a'_1, a'_2, \dots, a'_d)$  with  $0 \leq a'_i \leq a_i$ . Then

$$\max_{\mathbf{a}' \in S_{\mathbf{a}}} \rho(\mathbf{a}') = \max\{\rho(a_1, 0, \dots, 0), \rho(0, a_2, 0, \dots, 0), \dots, \rho(0, 0, \dots, a_d)\}.$$

*Proof.* See Appendix.

From this theorem it follows that to compute an upper bound on the highest achievable correlation value for a given pattern, it suffices to compute a correlation value for each of the classes separately, or —equivalently— to consider only  $d$  thresholds in the case of  $d$  classes. To show that this theorem is also usable in practice, we also prove the following.

**Theorem 2.** The  $\chi^2$  test on a contingency table of  $d$  classes defines a suitable correlation function.

*Proof.* See Appendix.

These observations have the following consequences. Assume that we solve the following equations

$$\chi^2(\theta_1 \mathbf{b}(1)) = \theta, \quad \chi^2(\theta_2 \mathbf{b}(2)) = \theta, \quad \dots \quad \chi^2(\theta_d \mathbf{b}(d)) = \theta,$$

similar to equation (2); then we can use the following as frequency constraint:

$$a_1 \geq \theta_1 \quad \vee \quad a_2 \geq \theta_2 \quad \vee \quad \dots \quad \vee \quad a_d \geq \theta_d,$$

similar to equation (3). Thus, we have a frequency constraint which can be computed in linear time. In the next section we consider how to compute  $\theta_i$  for the  $\chi^2$  constraint.

We wish to conclude this section with an observation for another correlation measure: information gain. We can show that the nice properties of  $\chi^2$  do not apply to information gain. Consider a database with three target classes of sizes  $n_1 = 30$ ,  $n_2 = 40$  and  $n_3 = 50$ . Then  $\rho_{gain}(0.9 \times 30, 0.9 \times 40, 0) > \rho_{gain}(0.9 \times 30, 0, 0)$ . We can therefore not determine minimum frequency thresholds for each of the classes by considering the points on the  $a_1, \dots, a_d$  axes through which the iso-information gain body crosses. Still, intuitively, one should be able to determine a largest possible hyper-rectangle that fits within an iso-information gain body, and thus a set of minimum threshold values. We leave that issue as future work.

## 5 Choosing Thresholds for $\chi^2$

In this section, we first show how thresholds can be computed for  $\chi^2$ . It appears that this formula is remarkably simple and that we can draw several further conclusions; the remainder of the section is devoted to listing some of these consequences.

**Theorem 3.** *Given a stamp point  $\mathbf{a} = a_j \mathbf{b}(j)$ . Then*

$$\chi^2(\mathbf{a}) = \frac{(N - n_j)a_j N}{N - a_j n_j}.$$

*Proof.* Without loss of generality we can assume that  $j = 1$ . Then we can split the  $\chi^2$  sum into two parts: the 1th row, and the other rows. For the first row the contribution to  $\chi^2$  is:

$$N \frac{(n_1 a_1 - \frac{n_1 a_1 n_1}{N})^2}{n_1 a_1 n_1} + N \frac{((1 - a_1)n_1 - \frac{n_1(N - a_1 n_1)}{N})^2}{N(N - a_1 n_1)} = \frac{a_1(N - n_1)^2}{N - a_1 n_1}.$$

For a row  $i > 1$  the contribution to  $\chi^2$  is:

$$N \frac{(-\frac{n_i a_1 n_1}{N})^2}{n_i a_1 n_1} + N \frac{(n_i - \frac{n_i(N - a_1 n_1)}{N})^2}{n_i(N - a_1 n_1)} = \frac{a_1 n_i n_1}{N} + \frac{a_1^2 n_1^2 n_i}{N(N - a_1 n_1)} = \frac{a_1 n_1 n_i}{N - a_1 n_1}.$$

If we sum all rows  $i > 1$ , the contribution of all these rows together is  $\frac{(N - n_1)a_1 n_1}{N - a_1 n_1}$ . Summing this term and the term for the first row, we obtain  $\frac{(N - n_1)a_1 N}{N - a_1 n_1}$ .  $\square$

From this theorem follows a simple closed formula for computing the threshold minimum support for every class, starting from the  $\chi^2$  threshold.

**Theorem 4.** *Given a threshold  $\chi^2$  value  $\theta$ , the solution of  $\chi^2(\theta_i \mathbf{b}(i)) = \theta$  is*

$$\theta_i = \frac{\theta N}{N^2 - n_i N + \theta n_i}.$$

*Proof.* This follows immediately from Theorem 3.  $\square$

Until now we studied the use of a multi-dimensional  $\chi^2$  statistic to measure correlation when multiple target classes are involved. A different, perhaps more straightforward way to deal with multiple classes is not to use a more complex correlation function, but to repeatedly solve 2 dimensional search problems: assume that we have  $d$  classes, then we can also build a database in which all examples for an original class  $1 \leq i \leq d$  are put into a new class 'A' and all examples which are *not* in class  $i$  are put into class 'B'. By searching for correlated patterns in this newly labeled database, one would discover patterns that achieve the highest correlation with class  $i$ , or its complement; by repeating this procedure for each class one finds correlated patterns for each original class. A natural question is how this approach compares to the approach of the previous section.

To study this different setup we require some additional notation. Similar to the symbols  $a_i, n_i, N$ , let us introduce the symbols  $a_i^j, n_i^j$  and  $N^j$  for the two-class contingency table for class  $j$ :

$a_1^j n_1^j$	$(1 - a_1^j) n_1^j$	$n_1^j$
$a_2^j n_2^j$	$(1 - a_2^j) n_2^j$	$n_2^j$
$\sum_{i=1}^d a_i^j n_i^j$	$\sum_{i=1}^d (1 - a_i^j) n_i^j$	$N^j$

The entries are computed from the entries of the original contingency table:  $a_1^j = a_j$ ,  $a_2^j = (\sum_{k=1}^d a_k n_k - a_j n_j) / n_2^j$ ,  $n_1^j = n_j$ ,  $n_2^j = \sum_{k=1}^d n_k - n_j = N - n_j$  and  $N^j = N$ . We can also compute  $\chi^2$  for this new contingency table; let us denote this value by  $\chi_j^2(\mathbf{a}^j)$ . Then one can show that the following is not generally true:

$$\chi_j^2(\mathbf{a}^j) = \chi^2(\mathbf{a});$$

the correlation computed over the 2 class table does not equal the correlation computed over the table with multiple classes. As an example, for  $n_1 = 30$ ,  $n_2 = 40$  and  $n_5 = 50$  it does not hold that  $\chi_1^2(0.9, 0.9 \times 40/90) = \chi^2(0.9, 0.9, 0)$ .

Of interest is now to study how the minimum support thresholds for the two-class search problems compare to the thresholds for the single (original) higher dimensional correlated pattern search. From Theorem 3 follows the following:

**Theorem 5.** *Given a stamp point  $\mathbf{a} = a_j \mathbf{b}(j)$  for the  $d$  dimensional search problem. Then*

$$\chi_j^2(a_j, 0) = \chi^2(\mathbf{a}).$$

*Proof.* If we consider the formula of Theorem 3, we note that the  $\chi^2$  value only depends on the total number of examples and the number of examples in the given class  $j$ . In the multi-class situation and the constructed two-class situation these are the same, and therefore also the threshold  $\chi^2$  values.  $\square$

This theorem has a practical consequence. Assume that we have determined all frequent patterns for all two-class search problems (for both classes of each problem), then it follows that we have also computed all necessary candidate patterns for the higher dimensional correlation measure. We only need to post-process the results of the two-class search problems to fill in missing support values and obtain exact  $\chi^2$  values; although therefore access to the database is required, a new frequent pattern search is not necessary.

A natural question is then whether the reverse is also true: assume that we have determined all frequent patterns for each of the classes of the multi-dimensional  $\chi^2$  statistic, have we then also determined all patterns that achieve a high correlation in each of the two-dimensional correlation problems?

Summarizing, for a class  $j$  we are interested in patterns for which  $\chi_j^2(a_1^j, a_2^j) \geq \theta$ . We assume that we have all patterns for which  $a_1 \mathbf{b}(1) \geq \theta_1 \vee \dots \vee a_d \mathbf{b}(d) \geq \theta_d$ . Then it is clear from Theorem 5 that we have also determined all patterns for which  $a_1^j = a_j \geq \theta_j = \theta_1^j$ . However, we require an additional theorem to prove that we also find all patterns for which  $a_2^j \theta_2^j$  (and thus all patterns are found for which  $a_1^j \geq \theta_1^j \vee a_2^j \geq \theta_2^j$ , which is the necessary condition to find all patterns for which  $\chi_j^2(a_1^j, a_2^j) \geq \theta$ ).

**Theorem 6.** *If for a stamp point  $\mathbf{a}$  we have  $a_2^j \geq \frac{N\theta}{N^2 - (N - n_j)(N - \theta)}$  then for at least one  $1 \leq i \leq d$ ,  $i \neq j$ :*

$$a_i \geq \frac{N\theta}{N^2 - n_i(N - \theta)}.$$

*Proof.* Without loss of generality we can assume that  $j = 1$ . Then from the assumption follows that

$$\sum_{i=2}^d a_i n_i \geq \frac{N\theta(N - n_1)}{N^2 - (N - n_1)(N - \theta)}.$$

Now let us assume that  $a_i < \frac{N\theta}{N^2 - n_i(N - \theta)}$ , for all  $1 < i < d$ . Then we have that

$$\begin{aligned} a_d n_d &\geq \frac{N\theta(N - n_1)}{N^2 - (N - n_1)(N - \theta)} - \sum_{i=2}^{d-1} a_i n_i \\ &\geq \frac{N\theta(N - n_1)}{N^2 - (N - n_1)(N - \theta)} - \sum_{i=2}^{d-1} \frac{N\theta n_i}{N^2 - n_i(N - \theta)} \\ &\geq \frac{N\theta(N - n_1)}{N^2 - (N - n_1)(N - \theta)} - \frac{N\theta(N - n_1 - n_d)}{N^2 - (N - n_1)(N - \theta)} \\ &\geq \frac{N\theta n_d}{N^2 - (N - n_1)(N - \theta)} \\ &\geq \frac{N\theta n_d}{N^2 - n_d(N - \theta)}; \end{aligned} \tag{4}$$

this shows that at least one term in the logical or must satisfy the given constraint.  $\square$

From this theorem follows that to find correlated patterns for two-class patterns, it suffices to postprocess the result from a multi-dimensional correlation search; access to the database is not even required.

The advantage of these observations is that they provide insight in the ways that sets of frequent patterns can be reused for different purposes. They show that if we search patterns that are frequent in individual classes, we can use these patterns both for multi-dimensional correlation measures as for more simplistic two-dimensional correlation measures.

At this point we can also ask ourselves how the two-way  $\chi^2$  correlation measure compares to the accuracy measure of Section 3. Assume that we were first interested in finding all patterns for which  $\chi^2(\mathbf{a}) \geq \theta_{\chi^2}$ , then we had to find all patterns for which  $a_1 \geq \theta_{\chi^2,1} = \frac{N\theta}{N^2 - n_1(N-\theta)} \vee a_2 \geq \theta_{\chi^2,2} = \frac{N\theta}{N^2 - n_2(N-\theta)}$ . Now assume that we want to find all patterns which satisfy a minimum accuracy constraint, then we can observe the following. If we solve the equation

$$\frac{\theta_{acc,i}N - (N - n_i)}{n_i} = \theta_{\chi^2,i},$$

we obtain

$$\theta_{acc,i} = \frac{N\theta_{\chi^2,i}n_i}{N(N^2 - n_i(N - \theta))} + \frac{N - n_i}{N};$$

Then for minimum accuracy thresholds  $\theta_{acc} \geq \max_i \theta_{acc,i}$  we can compute all patterns with high accuracy simply by postprocessing the results of the previous search; this follows from the comparison of class thresholds.

The same approach extends to many other situations. For example, assume that we want to contrast two classes against each other, disregarding examples of all other classes. If we already know the frequent patterns for the multi-class case, we can compute for which threshold on minimum  $\chi^2$  correlation between two classes we do not need to recompute the frequent patterns.

To conclude this section, let us sketch a possible scenario in which these observations can be exploited. Assume that we have a table with  $d > 2$  classes, and the user is first interested in finding patterns that are highly correlated according to a higher dimensional  $\chi^2$  statistic. Then we showed that we can transform this minimum correlation threshold into minimum frequency thresholds, and perform a pattern search for these thresholds; then we will find all patterns that achieve high correlation, but also some additional patterns. To answer the user's query, we postprocess the patterns. Assume that we store all patterns that achieve a high frequency in at least one of the classes, and that we also store the supports in all classes.

Then if the user changes her mind, and becomes interested in another kind of question, we showed how we can exploit the previously stored pattern set: if the user wants to find all patterns which have a high accuracy with respect to one class, we can exactly compute for which thresholds we can reuse the previously

stored pattern set, and thus, we showed how a second access to the data for this second question can be avoided.

## 6 Related work

From our point of view these results are a more simple and more efficient formulation of the methodology of Bay and Pazzani [2, 3], Morishita and Sese [11] and Zimmermann and De Raedt [13]. To show this, we will briefly review this method. By these authors the contingency table is denoted as follows:

$y$	$m - y$	$m$
$x - y$	$n - m - (x - y)$	$n - m$
$x$	$n - x$	$n$

The  $\chi^2$  statistic is defined as a function from  $(x, y)$ . If a pattern with stamp point  $(x, y)$  is refined, it is shown by Morishita and Sese that an upper bound for the  $\chi^2$  value of refined patterns is  $\max\{\chi^2(y, y), \chi^2(x - y, 0)\}$ . Clearly, this notation is a transformation of ours. The claim of Morishita and Sese can be specified equivalently in our notation. Assume that we are given a minimum  $\chi^2$  threshold. In our notation Morishita and Sese use the upper bound to stop refining if  $\max\{\chi^2(0, a_2), \chi^2(a_1, 0)\} < \theta$ . From Figure 4 we can conclude that an equivalent way to specify this test is  $a_2 < \theta_2 \wedge a_1 < \theta_1$ , where  $\theta_1$  and  $\theta_2$  are chosen such that  $\chi^2(0, \theta_2) = \theta$  and  $\chi^2(\theta_1, 0) = \theta$ , where  $\theta$  is the given threshold on  $\chi^2$ . We can thus conclude that the algorithm of Morishita and Sese which finds all correlated patterns, is a frequent itemset mining algorithm with multiple minimum support constraints.

By Zimmermann et al. [13] it was implied that an exponential number of  $\chi^2$  evaluations would be required to compute a reliable upperbound on the highest achievable  $\chi^2$  value. Extending to multiple classes the correspondence between Morishita and Sese’s approach and ours, we can prove that a linear number of thresholds is sufficient and equally strong pruning power is obtained.

Our observation also provides additional insight in the work of Bay and Pazzani [2]. They propose to prune branches in a search tree using both minimum support constraints and bounds on the highest achievable correlation (similar to Morishita and Sese). We can see now that explicit pruning on class frequencies may not be required, as the correlation constraint transforms into a minimum frequency constraint. Thus, our observation makes it possible to compare the pruning power of several constraints. Additionally, our pruning strategy for multiple classes can be shown to be more tight than Bay and Pazzani’s.

Much work has been done on class association rules, which are rules with high confidence and support, and a fixed attribute in the rule head. Using ROC spaces, it can be seen that the confidence constraint transforms into a maximum support constraint, but not in a minimum support constraint. If separate supports for each class are specified, such as by Liu et al. in [10], we can see now that the amount of search tree pruning is the same as for the other algorithms.

## 7 Algorithms and Experimental Results

The observations of the previous sections can be exploited in algorithms in several ways. In this section, we provide some details of algorithms that exploit our theory, where we restrict ourselves to integrating correlated pattern mining in the well-known trie based APRIORI algorithm [1]; integration in other kinds of algorithms is left as possible future work, but is expected to deliver similar results. To test the performance of our proposed algorithms, we implemented several of them; this section also contains experimental results obtained from running these implementations.

All our experiments were run on an Intel Pentium(R) 4 CPU 2.80GHz with 2GB main memory. We used datasets that we obtained from the UCI (see Figure 8). Datasets with small and large numbers of target classes were used; furthermore the datasets vary in size and number of attributes.

To implement our algorithms, we extended the APRIORI implementation of Ferenc Bodon [4]. Although this implementation is not the fastest, it has the advantage that it is small and clean; thus, we could easily change settings and compare them to each other. Unless pointed out otherwise we use the optimisation of this algorithm which loads the dataset in main memory.

Name	$N$	$d$	Comments
Internet Advertisements	3279	2	Class sizes: 2821, 458; numeric attributes not used
Mushroom	8124	2	Class sizes: 4208, 3916
Chess (KRKPA7)	3196	2	Class sizes: 1669, 1527
Chess (KRK)	28056	18	Largest class sizes: 4553, 4194; smallest: 27
Covertypes	581012	7	Largest class sizes: 283301, 211840, 35754; smallest: 2747; 8 discretized numeric attributes

**Fig. 8.** UCI datasets that we used in our experiments [6].

*Choosing thresholds in practice* The first topic that we wish to study in practice is the choice of threshold values. In statistics there are some rules of thumb for the choice of  $\chi^2$  thresholds. The most commonly used rule is that the p-value of the test should be 5%. The p-value is the probability of obtaining a given statistic, or a better statistic, if no association between the attributes of an instance and its class is assumed. A parameter for computing the p-value is the number of degrees of freedom of the test (which is  $d - 1$  in our case). For a given number degrees of freedom, a threshold p-value can be transformed into a threshold on  $\chi^2$ . Some values are illustrated in Figure 9.

In practice it turns out to be hard to transform this rule of thumb into viable minimum support thresholds. On the chess dataset (KRKPA7) a minimum support of 4 would be required on the first class for a minimum  $\chi^2$  threshold of 3.84. On most datasets such a support value is much too low; on this particular

		Degrees of freedom		
		1	6	16
p-value	0.05	3.84	12.59	26.30
	0.01	6.64	16.81	32.00
	0.001	10.83	22.46	39.25
	$10^{-300}$	36.00	51.62	73.39

**Fig. 9.** The correspondence between  $\chi^2$  values and p-values for the degrees of freedom relevant for the databases in the experiments.

dataset if we use a  $\chi^2$  threshold of 418, which results in an (absolute) support threshold of 400 in this first class, we already obtain  $> 2.000.000$  patterns that are frequent in at least one of the two classes.

Thus, computable minimum support thresholds correspond to very low p-values, which is desirable. There are however more issues involved in the determination of good thresholds. One of the advantages of using relatively high minimum support thresholds is that it reduces the risk that expected values in the contingency table become very low. A typical rule which statisticians use to estimate the reliability of the  $\chi^2$  test is

$\chi^2$  can be used if no more than 20% of the expected frequencies are less than 5 and none is less than 1.

For example, in the KRKPA7 dataset, in which 52% of the examples are in class ‘won’, we would require a minimum support threshold of 10 on this class to avoid getting expected values which are lower than 5.

We could also transform this statistician’s rule differently into combinations of minimum frequency constraints. In this paper we will not study this possibility further.

As for a database of size  $N$  the highest achievable  $\chi^2$  value is  $N$ , we will choose  $\chi^2$  thresholds which are percentages of  $N$ .

Dataset	$d$	$\theta$	Lin.	Exp. #	Cand. #	Freq. #	Corr.
Mushroom	2	12.0%	15.1s	15.1s	158021	157243	141953
Mushroom	2	10.0%	36.1s	36.1s	284590	283699	255037
Cover type	7	1.0%	19.9s	33.7s	208246	150610	42784
Cover type	7	0.5%	33.0s	63.3s	550169	433807	151952
Chess (KRK)	18	1.0%	0.8s	108.3s	13220	8029	2637
Chess (KRK)	18	0.5%	0.9s	111.3s	23246	13760	6610

**Fig. 10.** Experiments with a linear (Lin.) and an exponential (Exp.) test for pruning the search space. Given are run times (Lin. and Exp.); number of candidates (# Cand.), number of frequent patterns (# Freq.) and number of correlated patterns exceeding the  $\theta$  threshold (# Corr.).

*Linear vs Exponential search space pruning* Our second experiment involves a comparison of pruning algorithms. The setup of the experiment is as follows: we modify the original APRIORI algorithm such that with every pattern in the trie not one support, but multiple supports are stored — for each class one. When we pass an example through the trie, like in the original APRIORI algorithm, we only increase counters of the class to which the example belongs. Thus we obtain a simple mixing approach (see Section 2). When we have to determine whether an itemset should be pruned we consider two alternatives:

- our linear disjunction of minimum frequency tests;
- a generalization of the approach of Morishita and Sese which is exponential in  $d$  [11, 13].

The exponential generalization of Morishita and Sese (as also implied in [13]) works as follows. Let  $\mathbf{a}$  be a stamp point, and consider all  $a_i$  which are not zero. Then by setting a subset of these  $a_i$ 's to zero, we obtain a new stamp point which may be an upperbound on  $\chi^2$ . By computing  $\chi^2$  for all these new stamp points, and determining the maximum, we obtain the upper bound.

Results which compare these approaches for several datasets are given in Figure 10. It is clear that only if the number of target classes grows larger, the linear pruning test becomes interesting. At first sight it may seem strange that a decrease in threshold does not always result in much longer runtimes. This can be explained by the fact that the exponential approach only generates all subsets for coordinates which are non-zero. Although the number of candidates that is evaluated is much larger for a lower threshold, many of the additional candidates have zeros in many coordinates, and require less evaluation time than the patterns which have high support values in all classes.

*Postprocessing Sets of Patterns* We showed that in stead of recomputing all frequent patterns, it is often possible to reuse the same set of frequent patterns for different kinds of correlation queries. In this section we provide a short investigation of this idea. To this purpose we use the Cover type dataset, which consists of 7 target classes. We are interested in two kind of correlated patterns: patterns that correlate with all classes according to a 7-dimensional  $\chi^2$  statistic, and patterns that correlate with the first (largest) class according to a 2-dimensional  $\chi^2$  statistic that compares the first class with the aggregation of all other classes. For both correlated statistics we wish to use the same threshold value.

We use 2 kinds of algorithms. First, we have the basic *mixing* algorithm that we used earlier this section. We can start this algorithm two times to answer both questions (see Figure 11, rows ‘Search 7 dimensional’ and ‘Search 2 dimensional’). Another possibility is to run the 7 dimensional correlation query first, and to store all frequent patterns in an additional trie during the run of the algorithm<sup>3</sup>. We answer the second query by scanning the previously constructed trie.

---

<sup>3</sup> We require an additional trie as the APRIORI implementation removes unnecessary short patterns from the trie when generating longer candidates

To obtain more insight in the run time behavior of the implementations we also include in Figure 11 the run times of an implementation which does not load the database in memory, but rescans the data from disc, like the original APRIORI algorithm.

$\chi^2$ Threshold	Memory		From disc	
	1.0%	0.5%	1.0%	0.5%
Search 7 dimensional	19.9s	33.3s	82.6s	129.9s
Search 2 dimensional	7.9s	10.6s	27.7s	40.4s
Search 7 dim., store, query once	34.9s	66.3s	99.2s	164.9s
Search 7 dim., store, query twice	35.3s	67.6s	99.8s	166.6s

**Fig. 11.** A comparison between algorithms that compute patterns from data and from pattern sets.

In the experiment we can see that the time to answer a query from a constructed trie is much shorter than to compute the same result from data. Most queries can be answered within 2 seconds.

On the other hand, we also see that our implementation requires more time to construct a trie of all patterns in main memory. In some cases the additional time required for this construction is longer than the time required to perform an additional search for correlated patterns with a lower dimensional  $\chi^2$  statistic.

It can be expected that 2 dimensional  $\chi^2$  searches require less time than 7 dimensional ones, as for the 7 dimensional case some classes have rather low minimum threshold values. In the 2 dimensional case the small classes are summed together.

Some differences in run time are most likely a consequence of implementation issues and side effects of the architecture of modern computers. For example, we cannot otherwise explain that the run time for building the additional trie is larger when loading the data from disc, while in our implementation both the trie datastructures and the trie algorithms used during the construction of the second trie are exactly the same.

Dataset $\theta$	Mushroom 10%		Mushroom 12%		Internet 3.5%		Cover t. 0.5%	
	Memory	Disc	Memory	Disc	Memory	Disc	Memory	Disc
Mixing	36.1s	47.0s	15.5s	25.0s	33.1s	37.4s	10.7s	40.6s
Class 1 search	10.4s	17.5s	10.0s	16.5s	<0.1s	<0.1s	4.3s	16.1s
Class 2 search	10.4s	22.3s	1.2s	3.9s	17.9s	20.5s	5.1s	15.6s
Cl. 1 search + Cl. 2 count	18.3s	39.5s	17.5s	29.8s	<0.1s	<0.1s	16.2s	29.6s
Cl. 2 search + Cl. 1 count	18.7s	31.1s	3.5s	10.1s	35.0s	41.7s	8.6s	20.5s

**Fig. 12.** Comparison of evaluation strategies.

1. Transform minimum  $\chi^2$  into minimum supports  $\theta_1$  and  $\theta_2$ ;
2. Mine all frequent patterns  $\mathcal{F}_1$  in  $\mathcal{D}_1$  with minimum support  $\theta_1$ ;
3. Determine all supports of patterns in  $\mathcal{F}_1$  in  $\mathcal{D}_2$ ; 4. Prune all patterns from  $\mathcal{F}_1$  for which  $\chi^2$  is lower than  $\theta$ ;
5. Mine all frequent patterns  $\mathcal{F}_2$  in  $\mathcal{D}_2$  with minimum support  $\theta_2$ ;
6. Determine all supports of patterns in  $\mathcal{F}_2$  in  $\mathcal{D}_1$ ; 7. Prune all patterns from  $\mathcal{F}_1$  for which  $\chi^2$  is lower than  $\theta$ ;
8. (Optional) Merge the sets  $\mathcal{F}_1$  and  $\mathcal{F}_2$ .

**Fig. 13.** A simple algorithm for mining patterns with high  $\chi^2$  value.

*Evaluation Strategies* In our previous experiment we used a *mixing approach*, in which all patterns are evaluated in all classes. Another approach is to *postprocess* results. The simplest way to proceed is illustrated in Figure 13; first one performs a search for frequent patterns in class 1, and stores these into a new trie; then we evaluate these frequent patterns in the part of the database corresponding to the second class. Finally, we repeat the procedure with the classes reversed.

An overview of some experimental results is given in Figure 12. What is immediately remarkable in this table is the rather long additional time required to evaluate frequent patterns for the second class of examples. We investigated this phenomenon in detail, and found that the additional run time is *not* caused by scanning the examples of the second class; this scan is performed in  $< 2s$  in all cases. Furthermore, the additional run time is *not* (entirely) spent building the second trie, as the additional run time is dependent on the evaluation strategy (from memory or from disk). The main slow down seems to be caused by the mere allocation of additional main memory, and a resulting memory inefficiency of evaluating patterns in the first class. Thus, we can assume that most differences in this table are rather hardware dependent, or within margins of implementation details. We tried several further variations — including using different item orders for both classes, evaluating tries in the second class during the search in the first class, and so on, but in all cases the results do not seem to improve significantly. Thus, we can conclude that there are some differences in run time behavior of the several evaluation strategies, but that these differences are not very significant.

## 8 Conclusions

In this paper we showed that to find all patterns that correlate with a target attribute, it is sufficient to search for all patterns that satisfy a set of frequency thresholds, where these thresholds can be computed exactly by filling in a minimum correlation threshold in a correlation measure, such as information gain, accuracy, weighted accuracy or  $\chi^2$ . For the  $\chi^2$  measure we showed that this approach can even be used even if the target attribute has multiple values. We illustrated that a major consequence of this observation is that we can reuse pattern bases: if we know all patterns that satisfy a given disjunction of minimum frequency constraints, we can reuse these patterns to answer many kinds of correlation queries.

To illustrate the use of our theory, we gave several algorithms that exploit it. Although several algorithmic variations follow from our theory that are not significantly better in terms of efficiency, we showed that the main contributions of the paper do make sense:

- for large numbers of target attribute values, the reduction in run time for the  $O(d)$  prune test is significant;
- to reuse existing sets of patterns is more efficient than to recompute correlated patterns from data.

Much further research can be considered in this direction. In this paper we studied only a small amount of correlation measures, and showed only for a few of them how they relate to each other. Future inductive databases should provide a wide range of correlation measures and should be able to relate them to each other to reuse existing pattern bases efficiently. We already gave some attention to the reliability of the  $\chi^2$  test, but more work could be done in this direction. For example, for small expected values in the contingency table Fischer's exact test is considered to be more reliable than the  $\chi^2$  test. To 'automatically' switch to a more reliable test and still find all patterns, we require a further theory on the differences between the tests.

In our experiments we showed how correlated pattern mining can be performed on top of an implementation of the traditional APRIORI frequent itemset mining algorithm. There are many kinds of algorithms, such as FP GROWTH [8] or ECLAT [12], which could incorporate the same ideas. Finally, condensed representations for answering correlated pattern mining queries have not been studied yet.

**Acknowledgements** This work was partly supported by the EU FET IST project IQ ("Inductive Querying"), contract number FP6-516169.

## References

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In *Advances in knowledge discovery and data mining*, pages 307–328, 1996.
2. S. D. Bay and M. J. Pazzani. Detecting change in categorical data: Mining contrast sets. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 302–306. ACM Press, 1999.
3. S. D. Bay and M. J. Pazzani. Detecting group differences: Mining contrast sets. In *Data Mining and Knowledge Discovery*, volume 5, pages 213–246. Kluwer Academic Publishers, 2001.
4. F. Bodon. Surprising results of trie-based FIM algorithms. In *Proceedings of the Workshop on Frequent Itemset Mining Implementations (FIMI)*, volume 90 of *CEUR Workshop Proceedings*, 2004.
5. L. De Raedt, M. Jaeger, S. D. Lee, and H. Mannila. A theory of inductive query answering (extended abstract). In *Proceedings of the Second IEEE International Conference on Data Mining (ICDM)*, pages 123–130, 2002.

6. C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.
7. J. Fürnkranz and P. Flach. ROC 'n' rule learning – towards a better understanding of covering algorithms. In *Machine Learning*, volume 58, pages 39–77, 2005.
8. J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1–12, 2000.
9. B. Kavšek, N. Lavrač, and V. Jovanoski. APRIORI-SD: Adapting association rule learning to subgroup discovery. In *Proceedings of the Fifth International Symposium on Intelligent Data Analysis*, volume 2810 of *Lecture Notes in Computer Science*, pages 230–241. Springer-Verlag, 2003.
10. B. Liu, Y. Ma, and C.-K.Wong. Improving an exhaustive search based rule learner. In *Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, volume 1910 of *Lecture Notes in Artificial Intelligence*, pages 504–509, 2000.
11. S. Morishita and J. Sese. Traversing itemset lattices with statistical metric pruning. In *Proceedings of the Nineteenth ACM SIGACT-SIGMOD-SIGART Symposium on Database Systems (PODS)*, pages 226–236, 2000.
12. M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 283–286, 1997.
13. A. Zimmermann and L. De Raedt. Cluster-grouping: From subgroup discovery to clustering. In *Proceedings of the 15th European Conference on Machine Learning (ECML)*, volume 3201 of *Lecture Notes in Computer Science*, pages 575–577, 2004.

## A Proof Outlines

In this Appendix we provide some short outlines of the proofs of Theorems 1 and 2. We will illustrate our argumentation in the case of a target attribute with 3 classes. First, however, we require the following lemma.

**Lemma 1.** *Let  $\rho$  be a suitable correlation function. Given a binary vector  $\mathbf{b} \in \mathcal{B}_{d, \geq 2}$ , then for every  $k$  in this vector for which  $b_k = 1$  it holds that:*

$$\rho(\alpha \mathbf{b}) \leq \rho(\alpha \mathbf{b}'), \text{ where } \mathbf{b}' \text{ is a vector such that } b'_k = 1 \text{ and } b'_i = 0 \text{ for } i \neq k.$$

*Proof.* This follows from the second constraint on suitable correlation functions, which states that by setting one coordinate to zero the correlation value can only increase. More formally, the vector  $\mathbf{b}$  consists of ones at positions  $i_1, \dots, i_k$ , while other bits are zero. By setting first  $i_1$  to zero, then  $i_2$ , and so on, until  $i_{k-1}$  is zero, a sequence of bit vectors results, for which the correlation values increase monotonically. As we did not assume any order on the indexes in  $\mathbf{i}$ , we can conclude that we can construct a sequence which reduces every bit vector  $\mathbf{b}$  to a bit vector in which only one bit is one.  $\square$

In Figure 14 this is illustrated for the three-dimensional case. Consider the vector  $\alpha \cdot (1, 1, 1) = (\alpha, \alpha, \alpha)$ . According to the second constraint on correlation functions,  $\rho(\alpha, \alpha, \alpha) \leq \rho(0, \alpha, \alpha) \leq \rho(0, 0, \alpha)$ . Furthermore, among others,  $\rho(\alpha, 0, \alpha) \leq \rho(\alpha, 0, 0)$ . The theorem does not claim that  $\rho(\alpha, 0, \alpha) \leq \rho(0, \alpha, 0)$  holds.

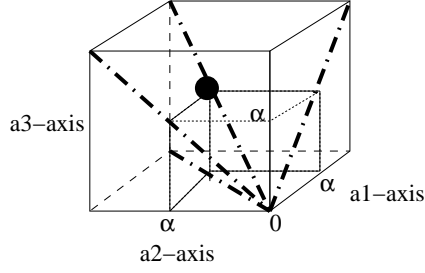


Fig. 14. Illustration of Lemma 1.

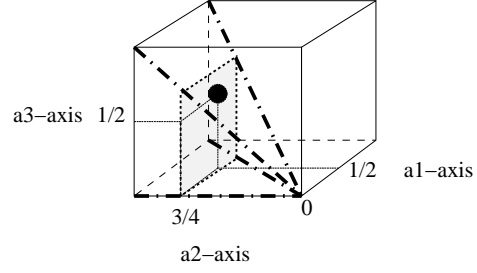


Fig. 15. An example stamp point.

*Proof. (Theorem 1)* As the function  $\rho$  is assumed to be convex the following must hold:

$$\max_{\mathbf{a}' \in \mathcal{S}_a} \rho(\mathbf{a}') = \max_{\mathbf{b} \in \mathcal{B}_d} \rho(a_1 \cdot b_1, a_2 \cdot b_2, \dots, a_d \cdot b_d).$$

This follows from the property that for convex functions any domain that can be characterized by a bounding polygon is maximized on one of the vertexes of the polygon. We now have to show that we can discard all elements of  $\mathcal{B}_{d, \geq 2}$ .

Consider the given stamp point  $\mathbf{a} = (a_1, \dots, a_d)$  and consider one of its dimensions  $k$  such that  $a_k = \max_{1 \leq j \leq d} a_j$ . Then the following points define a  $d - 1$  dimensional rectangle:

$$\{a_k \cdot \mathbf{b} \mid \mathbf{b} \in \mathcal{B}_d, b_k = 1\}$$

The stamp point  $\mathbf{a}$  is an element of this rectangle, as for all  $a_i$  it holds that  $0 \leq a_i \leq a_k$ . Please note that a rectangle in any dimension can be defined by giving two points 'opposite' from each other. The rectangle here is defined by the two points  $(0, \dots, a_k, \dots, 0)$  and  $(a_k, \dots, a_k)$ .

From the convexity of  $\rho$  it follows that for a given  $\mathbf{a}$  with  $a_k = \max_{1 \leq j \leq d} a_j$ :

$$\max_{\mathbf{b} \in \mathcal{B}_d, b_k = 1} \rho(a_k \cdot \mathbf{b}) \geq \rho(\mathbf{a}).$$

From Lemma 1 it follows that  $\max_{\mathbf{b} \in \mathcal{B}_d, b_k = 1} \rho(a_k \cdot \mathbf{b}) = \rho(a_k \cdot \mathbf{b})$ , where  $\mathbf{b}$  is the vector in which all elements are zero except  $b_k$ . For any given stamp point  $\mathbf{a}$  we may therefore conclude that  $\rho(\mathbf{a}) \leq \rho(a_k \cdot \mathbf{b})$ , where  $a_k = \max_{1 \leq i \leq d} a_i$  and  $\mathbf{b}$  is a vector that is zero in all coordinates except for the  $k$ th, which is 1.  $\square$

As an example consider the following stamp point:  $(\frac{1}{2}, \frac{3}{4}, \frac{1}{2})$ . This stamp point is illustrated in Figure 15. What we wish to show is that we do not need to consider this stamp point, as its correlation value is always lower than that of one of the points in  $\{(\frac{1}{2}, 0, 0), (0, \frac{3}{4}, 0), (0, 0, \frac{1}{2})\}$ . This would show that the only points that we need to consider are in  $\{(\frac{1}{2}, 0, 0), (0, \frac{3}{4}, 0), (0, 0, \frac{1}{2})\}$ .

As  $a_2 = \frac{3}{4} \geq \frac{1}{2} = a_1 = a_3$  the binary vectors of importance are  $\{\mathbf{b} \mid \mathbf{b} \in \mathcal{B}_d, b_2 = 1\} = \{(0, 1, 0), (0, 1, 1), (1, 1, 0), (1, 1, 1)\}$ . After multiplication with  $\frac{3}{4}$  the rectangle  $\{(0, \frac{3}{4}, 0), (0, \frac{3}{4}, \frac{3}{4}), (\frac{3}{4}, \frac{3}{4}, 0), (\frac{3}{4}, \frac{3}{4}, \frac{3}{4})\}$  is obtained. This rectangle is highlighted in the Figure. The original stamp point is part of this rectangle.

From Lemma 1 it follows that  $\max\{\rho(0, \frac{3}{4}, 0), \rho(0, \frac{3}{4}, \frac{3}{4}), \rho(\frac{3}{4}, \frac{3}{4}, 0), \rho(\frac{3}{4}, \frac{3}{4}, \frac{3}{4})\} = \rho(0, \frac{3}{4}, 0)$ . Due to convexity all points within the rectangle are lower than the highest point on the bounding polygon, therefore also  $\rho(\frac{1}{2}, \frac{3}{4}, \frac{1}{2}) \leq \rho(0, \frac{3}{4}, 0)$ . This proves that we do not need to consider the given stamp point. Similar arguments apply to the points in  $\{(\frac{1}{2}, \frac{3}{4}, 0), (0, \frac{3}{4}, \frac{1}{2}), (\frac{1}{2}, 0, \frac{1}{2})\}$ .

What remains to be shown is that suitable correlation functions indeed exist. We will show this in the proof of the following theorem.

*Proof. (Theorem 2)* It was already observed in other work that the  $\chi^2$  function for multiple classes is convex [13]. Here we concentrate on the second constraint. As one can always change the order of arguments of  $\rho$  without loss of generality we may state that we consider the following change in a contingency table:

$$\begin{array}{|c|c|c|} \hline \alpha n_1 & (1-\alpha)n_1 & n_1 \\ \alpha n_2 & (1-\alpha)n_2 & n_2 \\ \vdots & \vdots & \vdots \\ \alpha n_{k-2} & (1-\alpha)n_{k-2} & n_{k-2} \\ \alpha n_{k-1} & (1-\alpha)n_{k-1} & n_{k-1} \\ 0 & n_k & n_k \\ \vdots & \vdots & \vdots \\ 0 & n_d & n_d \\ \hline \sum_{i=1}^{k-1} \alpha n_i & \sum_{i=1}^d n_i - \sum_{i=1}^{k-1} \alpha n_i & \sum_{i=1}^d n_i \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|c|} \hline \alpha n_1 & (1-\alpha)n_1 & n_1 \\ \alpha n_2 & (1-\alpha)n_2 & n_2 \\ \vdots & \vdots & \vdots \\ \alpha n_{k-2} & (1-\alpha)n_{k-2} & n_{k-2} \\ 0 & n_{k-1} & n_{k-1} \\ 0 & n_k & n_k \\ \vdots & \vdots & \vdots \\ 0 & n_d & n_d \\ \hline \sum_{i=1}^{k-2} \alpha n_i & \sum_{i=1}^d n_i - \sum_{i=1}^{k-2} \alpha n_i & \sum_{i=1}^d n_i \\ \hline \end{array}$$

We denote the  $\chi^2$  value of the contingency table before the change as  $\chi_{\mathbf{n}}^2(\alpha, k)$ ; after the change the  $\chi^2$  value is  $\chi_{\mathbf{n}}^2(\alpha, k-1)$ . We show the following:

$$\begin{aligned} \chi_{\mathbf{n}}^2(\alpha, k) - \chi_{\mathbf{n}}^2(\alpha, k-1) = \\ \frac{\alpha(\alpha-1)n_{k-1} \left( \sum_{i=1}^d n_i \right)^2}{\left( \sum_{i=1}^{k-2} (1-\alpha)n_i + \sum_{i=k-1}^d n_i \right) \left( \sum_{i=1}^{k-1} (1-\alpha)n_i + \sum_{i=k}^d n_i \right)}. \end{aligned} \quad (5)$$

Clearly, for  $0 \leq \alpha \leq 1$  it holds that  $\chi_{\mathbf{n}}^2(\alpha, k) - \chi_{\mathbf{n}}^2(\alpha, k-1) \leq 0$  and therefore that  $\chi_{\mathbf{n}}^2(\alpha, k) \leq \chi_{\mathbf{n}}^2(\alpha, k-1)$ . We show now how the first term of equation (5) can be rewritten into the second term. The right term is defined as

$$\sum_{i=1}^d \frac{(E_{i1} - O_{i1})^2}{E_{i1}} + \frac{(E_{i2} - O_{i2})^2}{E_{i2}} - \frac{(E'_{i1} - O'_{i1})^2}{E'_{i1}} - \frac{(E'_{i2} - O'_{i2})^2}{E'_{i2}}, \quad (6)$$

where

$$E_{i1} = \frac{\alpha(n_1 + \dots + n_{k-1})n_i}{N}, \quad O_{i1} = \begin{cases} \alpha n_i & \text{if } i \leq k-1; \\ 0 & \text{otherwise;} \end{cases}$$

furthermore,  $E_{i2} = n_i - E_{i1}$ ,  $O_{i2} = n_i - O_{i1}$ ,  $O'_{ij}$  is defined similar to  $O_{ij}$ , and

$$E'_{i1} = E_{i1} - \frac{\alpha n_{k-1} n_i}{N}, \quad E'_{i2} = E_{i2} + \frac{\alpha n_{k-1} n_i}{N}.$$

Equation (6) can then be rewritten as

$$\begin{aligned} & \sum_{i=1}^d (E_{i1} - 2O_{i1} + \frac{O_{i1}^2}{E_{i1}}) + (E_{i2} - 2O_{i2} + \frac{O_{i2}^2}{E_{i2}}) \\ & - (E_{i1} - \frac{\alpha n_{k-1} n_i}{N} - 2O'_{i1} + \frac{(O'_{i1})^2}{E'_{i1}}) - (E_{i2} + \frac{\alpha n_{k-1} n_i}{N} - 2O'_{i2} + \frac{(O'_{i2})^2}{E'_{i2}}), \end{aligned}$$

which reduces to

$$\sum_{i=1}^d 2(O'_{i1} - O_{i1} + O'_{i2} - O_{i2}) + \frac{O_{i1}^2}{E_{i1}} + \frac{O_{i2}^2}{E_{i2}} - \frac{(O'_{i1})^2}{E'_{i1}} - \frac{(O'_{i2})^2}{E'_{i2}}. \quad (7)$$

It is easy to see that  $\sum_{i=1}^d 2(O'_{i1} - O_{i1} + O'_{i2} - O_{i2}) = 0$ , as the  $O$  elements only sum over all observations, and this number does not change. Therefore we rewrite equation (7) to:

$$\sum_{i=1}^d \frac{O_{i1}^2}{E_{i1}} + \frac{O_{i2}^2}{E_{i2}} - \frac{(O'_{i1})^2}{E'_{i1}} - \frac{(O'_{i2})^2}{E'_{i2}},$$

which reduces to:

$$\left( \sum_{i=1}^d \frac{O_{i1}^2}{E_{i1}} + \frac{O_{i2}^2}{E_{i2}} - \frac{O_{i1}^2}{E'_{i1}} - \frac{O_{i2}^2}{E'_{i2}} \right) + \frac{(\alpha n_{k-1})^2}{E'_{(k-1)1}} - \frac{(1 - (1 - \alpha)^2) n_{k-1}^2}{E'_{(k-1)2}}.$$

or, equivalently:

$$\left( \sum_{i=1}^d \frac{O_{i1}^2}{E_{i1}} - \frac{O_{i1}^2}{E'_{i1}} \right) + \left( \sum_{i=1}^d \frac{O_{i2}^2}{E_{i2}} - \frac{O_{i2}^2}{E'_{i2}} \right) + \frac{(\alpha n_{k-1})^2}{E'_{(k-1)1}} + \frac{\alpha(\alpha - 2) n_{k-1}^2}{E'_{(k-1)2}}. \quad (8)$$

We first rewrite the first term:

$$\begin{aligned} \sum_{i=1}^d \frac{O_{i1}^2}{E_{i1}} - \frac{O_{i1}^2}{E'_{i1}} &= \sum_{i=1}^{k-1} \frac{\alpha^2 n_i^2 N}{\alpha(n_1 + \dots + n_{k-1}) n_i} - \frac{\alpha^2 n_i^2 N}{\alpha(n_1 + \dots + n_{k-2}) n_i} \\ &= \sum_{i=1}^{k-1} \frac{\alpha^2 n_i^2 (n_1 + \dots + n_{k-2}) N - \alpha^2 n_i^2 (n_1 + \dots + n_{k-1}) N}{\alpha(n_1 + \dots + n_{k-1}) (n_1 + \dots + n_{k-2}) n_i} \\ &= \sum_{i=1}^{k-1} \frac{-\alpha n_i n_{k-1} N}{(n_1 + \dots + n_{k-1}) (n_1 + \dots + n_{k-2})} \\ &= \frac{-\alpha \left( \sum_{i=1}^{k-1} n_i \right) n_{k-1} N}{(n_1 + \dots + n_{k-1}) (n_1 + \dots + n_{k-2})} = \frac{-\alpha n_{k-1} N}{n_1 + \dots + n_{k-2}} \end{aligned}$$

Furthermore, we have that:

$$\frac{(\alpha n_{k-1})^2}{E'_{(k-1)1}} = \frac{(\alpha n_{k-1})^2 N}{\alpha(n_1 + \dots + n_{k-2}) n_{k-1}} = \frac{\alpha n_{k-1} N}{n_1 + \dots + n_{k-2}},$$

therefore two of the terms in equation (8) cancel out. Next we consider:

$$\begin{aligned}
\sum_{i=1}^d \frac{O_{i2}^2}{E_{i2}} - \frac{O_{i2}^2}{E'_{i2}} &= \sum_{i=1}^{k-1} \frac{(1-\alpha)^2 n_i^2 N}{(N - \alpha(n_1 + \dots + n_{k-1}))n_i} - \frac{(1-\alpha)^2 n_i^2 N}{(N - \alpha(n_1 + \dots + n_{k-2}))n_i} + \\
&\quad \sum_{i=k}^d \frac{n_i^2 N}{(N - \alpha(n_1 + \dots + n_{k-1}))n_i} - \frac{n_i^2 N}{(N - \alpha(n_1 + \dots + n_{k-2}))n_i} \\
&= \sum_{i=1}^{k-1} \frac{\alpha(1-\alpha)^2 n_i N n_{k-1}}{(N - \alpha(n_1 + \dots + n_{k-1}))(N - \alpha(n_1 + \dots + n_{k-2}))} + \\
&\quad \sum_{i=k}^d \frac{\alpha n_i N n_{k-1}}{(N - \alpha(n_1 + \dots + n_{k-1}))(N - \alpha(n_1 + \dots + n_{k-2}))} \\
&= \frac{\alpha(\sum_{i=1}^{k-1} (1-\alpha)^2 n_i + \sum_{i=k}^d n_i) N n_{k-1}}{(N - \alpha(n_1 + \dots + n_{k-1}))(N - \alpha(n_1 + \dots + n_{k-2}))}
\end{aligned}$$

Summing the remaining terms we have that:

$$\begin{aligned}
&\left( \sum_{i=1}^d \frac{O_{i2}^2}{E_{i2}} - \frac{O_{i2}^2}{E'_{i2}} \right) + \frac{\alpha(\alpha-2)n_{k-1}^2}{E'_{(k-1)2}} = \\
&\frac{\alpha(\sum_{i=1}^{k-1} (1-\alpha)^2 n_i + \sum_{i=k}^d n_i) N n_{k-1} + \alpha(\alpha-2)n_{k-1} N (N - \alpha(n_1 + \dots + n_{k-1}))}{(N - \alpha(n_1 + \dots + n_{k-1}))(N - \alpha(n_1 + \dots + n_{k-2}))}.
\end{aligned}$$

This simplifies to

$$\frac{\alpha(\alpha-1)N^2 n_{k-1}}{(N - \alpha(n_1 + \dots + n_{k-1}))(N - \alpha(n_1 + \dots + n_{k-2}))},$$

which is the final rewritten term that we were searching. Clearly, for  $0 \leq \alpha \leq 1$  this term is negative, and  $\chi^2$  measure is therefore suitable.  $\square$