Department of Computer Science
Series of Publications A
Report A-2010-2

Efficient search for statistically significant dependency rules in binary data

Wilhelmiina Hämäläinen

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in Auditorium CK112, Exactum, on October 1st, 2010, at 12 o'clock noon.

University of Helsinki
Finland

Contact information

Postal address:
Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: postmaster@cs.helsinki.fi (Internet)

URL: http://www.cs.Helsinki.FI/

Telephone: +358 9 1911

Telefax: +358 9 191 51120

ISSN 1238-8645
ISBN 978-952-10-6447-0 (paperback)
ISBN 978-952-10-6448-7 (PDF)
Computing Reviews (1998) Classification: H.2.8, G.1.6, G.2.1, G.3, I.2.6
Helsinki 2010
Helsinki University Print

Efficient search for statistically significant dependency rules in binary data

Wilhelmiina Hämäläinen

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
Wilhelmiina.Hamalainen@cs.Helsinki.FI
http://www.cs.Helsinki.FI/Wilhelmiina.Hamalainen/

PhD Thesis, Series of Publications A, Report A-2010-2
Helsinki, September 2010, 163 pages
ISSN 1238-8645
ISBN 978-952-10-6447-0 (paperback)
ISBN 978-952-10-6448-7 (PDF)

Abstract

Analyzing statistical dependencies is a fundamental problem in all empirical science. Dependencies help us understand causes and effects, create new scientific theories, and invent cures to problems. Nowadays, large amounts of data is available, but efficient computational tools for analyzing the data are missing.

In this research, we rise to the challenge, and develop efficient algorithms for a commonly occurring search problem – searching for the statistically most significant dependency rules in binary data. We consider dependency rules of the form $X \rightarrow A$ or $X \rightarrow \neg A$, where X is a set of positive-valued attributes and A is a single attribute. Such rules describe which factors either increase or decrease the probability of the consequence A. A classical example are genetic and environmental factors, which can either cause or prevent a disease.

The emphasis in this research is that the discovered dependencies should be genuine – i.e. they should also hold in future data. This is an important distinction from the traditional association rules, which – in spite of their name and a similar appearance to dependency rules – do not necessarily represent statistical dependencies at all or represent only spurious connections, which occur by chance. Therefore, the principal objective is to search for the rules with statistical significance measures. Another im-

portant objective is to search for only non-redundant rules, which express the real causes of dependence, without any occasional extra factors. The extra factors do not add any new information on the dependence, but can only blur it and make it less accurate in future data.

The problem is computationally very demanding, because the number of all possible rules increases exponentially with the number of attributes. In addition, neither the statistical dependency nor the statistical significance are monotonic properties, which means that the traditional pruning techniques do not work. As a solution, we first derive the mathematical basis for pruning the search space with any well-behaving statistical significance measures. The mathematical theory is complemented by a new algorithmic invention, which enables an efficient search without any heuristic restrictions. The resulting algorithm can be used to search for both positive and negative dependencies with any commonly used statistical measures, like Fisher's exact test, the χ^2-measure, mutual information, and z-scores.

According to our experiments, the algorithm is well-scalable, especially with Fisher's exact test. It can easily handle even the densest data sets with 10000–20000 attributes. Still, the results are globally optimal, which is a remarkable improvement over the existing solutions. In practice, this means that the user does not have to worry whether the dependencies hold in future data or if the data still contains better, but undiscovered dependencies.

Computing Reviews (1998) Categories and Subject Descriptors:
H.2.8 Database Applications — data mining
G.1.6 Optimization — global optimization
G.2.1 Combinatorics — Combinatorial algorithms
G.3 Probability and Statistics — contingency table analysis, nonparametric statistics
I.2.6 Learning — Knowledge acquisition

General Terms:
algorithms, theory, experimentation

Additional Key Words and Phrases:
dependency rule, statistical significance, rule discovery

Acknowledgements

I am very grateful to Professor Matti Nykänen for taking the invidious position as my supervisor. The task was not easy, because I had selected a challenging mission myself and he could only watch when I walked my own unknown paths. Still, he succeeded better than any imaginable supervisor could do in the most important task of the supervisor – the psychological support. Without him, I would have given up the whole work after every setback.

During my last year in the Helsinki Graduate School in Computer Science and Engineering (Hecse), I have also been priviledged to enjoy the luxury of two extra supporters, Professor Jyrki Kivinen and Jouni Seppänen, Ph.D. Both of them have done excellent work in mentally supporting me and answering my questions.

I would also like to thank my pre-examiners, Professor Geoff Webb and Siegfried Nijssen, Ph.D. as well as Jiuyong Li, Ph.D. and Professor Hannu Toivonen, for their useful comments and fruitful discussions. Marina Kurtén, M.A. has done me a valuable favour by helping to improve the language of my thesis.

I thank my half-feline family, Kimmo and Sofia, for their love and patience (and especially doing all the housework without too much whining). They believed in me and what I was doing, unquestioningly, or at least they managed to pretend it well. My parents Terttu and Yrjö have always encouraged me to follow my own paths and they have also offered good provisions for that.

This research has been supported with personal grants from the Finnish Concordia Fund (Suomalainen Konkordia-Liitto) and Emil Aaltonen Fund (Emil Aaltosen Säätiö). Conference travels have been supported by grants from following institutions: the Department of Computer Science at the University of Helsinki, the Department of Computer Science at the University of Kuopio, the Helsinki Graduate School in Computer Science and Engineering (Hecse), and the IEEE International Conference on Data Mining (ICDM). All their support is gratefully acknowledged.

To the memory of my wise Grandmother

The stone that the builders rejected
has become the cornerstone.

Matt. 21:42

Contents

List of Figures

List of Tables

List of Algorithms

List of Notations

$\mathbb{N} = \mathbb{Z}^+ \cup \{0\}$	natural numbers
$A, B, C, \ldots$	binary attributes
$a, b, c, \ldots \in \{0, 1\}$	attribute values
X, Y, Z	attribute sets
$\mathcal{R}, \mathcal{S}, \mathcal{T}, \mathcal{U}$	rule sets
$R = \{A_1, \ldots, A_k\}$	set of all attributes
$\|R\| = k$	number of attributes
$Dom(R) = \{0, 1\}^k$	attribute space
$Dom(X) = \{0, 1\}^l$	domain of X, $\|X\| = l$
$\overline{x} = (a_1, ..., a_l)$	vector of attribute values
$A_1 = a_i, A_2 = a_2$	value assignment, where $A_1 = a_1$ and $A_2 = a_2$
$(X = \overline{x}) = (A_1 = a_1), \ldots, (A_l = a_l)$	event, $X = \{A_1, ..., A_l\}$
$X = (A = 1), \ldots, (A_l = 1)$	a short hand notation for a positive-valued event
$t = (id, (A_1 = a_1), \ldots, (A_k = a_k))$	row (transaction), $id \in \mathbb{N}$
$r = \{t_1, \ldots, t_n\}$	data set
$\|r\| = n$	number of rows in r
$m(X = \overline{x})$	absolute frequency of event, number of rows where $X = \overline{x}$
$fr(X = \overline{x}) = P(X = \overline{x}) = \frac{m(X=\overline{x})}{\|r\|}$	relative frequency of event $X = \overline{x}$ in set r
$P_r(X = \overline{x})$	real probability of event $X = \overline{x}$
$cf(X = \overline{x} \to A = a) = P(A = a \| X = \overline{x})$	confidence of rule $X = \overline{x} \to A = a$
$\gamma(X = \overline{x}, A = a) = \frac{P(X=\overline{x}, A=a)}{P(X=\overline{x})P(A=a)}$	lift of rule
$\delta(X = \overline{x}, A = a)$ $= P(X = \overline{x}, A = a) - P(X = \overline{x})P(A = a)$	leverage of rule
N_X, N_A, N_{XA}, N	random variables in $\mathbb{N}$

Chapter 1

Introduction

> *Data miners are often more interested in understandability than accuracy or predictability per se.*
>
> C. Glymour, D. Madigan et al.

The core of all empirical sciences is to infer general laws and regularities from individual observations. Often the first step is to analyze statistical dependencies among different factors and then try to understand the underlying causal relationships. Computers have made systematic search possible, but still the task is computationally very demanding. In this thesis, we develop new efficient search algorithms for certain kinds of statistical dependencies. The objective is to find statistically the most significant, non-redundant dependency rules in binary data using only statistical significance measures for pruning and ranking.

In this chapter, we introduce the main problem, summarize the previous solutions, explicate the new contribution, and introduce the organization of the thesis.

1.1 Dependency rules

The concept of statistical dependence is based on statistical independence – the idea that two factors or groups of factors do not affect the occurrence of each other. If the real probabilities P_r of two events, $A = a$ and $B = b$, were known, the definition would be simple. The absolute independence states that $P_r(A = a, B = b) = P_r(A = a)P_r(B = b)$. For example – according to current knowledge – drinking coffee does not affect the probability of getting cancer. Similarly, the events would be considered statistically dependent, if $P_r(A = a, B = b) \neq P_r(A = a)P_r(B = b)$. The larger the deviation

of $P_r(A = a, B = b)$ and $P_r(A = a)P_r(B = b)$ is, the stronger is the dependency. If $P_r(A = a, B = b) < P_r(A = a)P_r(B = b)$, the dependency is negative (i.e. $B = b$ is less likely to occur, if $A = a$ had occurred than if $A = a$ had not occurred, and vice versa), and if $P_r(A = a, B = b) > P_r(A = a)P_r(B = b)$, the dependency is positive (i.e. $B = b$ is more likely to occur, if $A = a$ had occurred than if $A = a$ had not occurred, and vice versa). For example, (according to current knowledge) smoking increases the probability of getting lung cancer, but it (as well as drinking coffee) decreases the probability of getting Alzheimer's disease.

The problem is that empirical science is always based on finite sets of observations and the real probabilities P_r are not known. Instead, we have to use (typically, maximum likelihood) estimates P based on the observed relative frequencies in the data. This leads to inaccuracies, especially in small data sets. Therefore, it is quite possible to observe *spurious dependencies*, where $P(A = a, B = b)$ deviates substantially from $P(A = a)P(B = b)$, even if $A = a$ and $B = b$ were actually independent. For example, the earliest small-scale studies indicated that pipe smoking would decrease the risk of lung cancer compared to non-smokers, but according to current research, pipe smoking has no effect or may, in fact, raise the risk of lung cancer.

To solve this problem, statisticians have developed several measures of significance to prune out spurious dependencies and find the most genuine dependencies, which are likely to hold also in future data. The general idea in all statistical significance measures is to estimate the probability that the observed or a more extreme deviation between $P(A = a, B = b)$ and $P(A = a)P(B = b)$ had occurred by chance in the given data set, if $A = a$ and $B = b$ were actually independent. Given a statistical significance measure, M, we can now search only the most significant (best) dependencies or all dependencies, whose M value is inside certain limits. For example, in medical science, a focal problem is to find statistically significant dependencies between genes or gene groups and diseases. The discovered dependencies can be expressed as *dependency rules* of the form $A_1 = a_1, \ldots, A_l = a_l \rightarrow D = d$, where A_i's are gene alleles, which either occur ($a_i = 1$) or do not occur ($a_i = 0$) in the gene sequence, and D is a disorder, which was either observed ($d = 1$) or unobserved ($d = 0$). The rule does not yet say that a certain combination of alleles causes or prevents the disease, but if the rule was statistically significant, it is likely to hold also in future data. Even if the causal mechanisms were never discovered (and they seldom are), these dependency rules help to target further research and identify potential risk patients.

In this research, we concentrate on a certain, commonly occurring type

of dependency rules among binary attributes. The rule can be of the form $X \rightarrow A = a$, where the antecedent contains only true-valued attributes $X = A_1, \ldots, A_l$ (i.e. $A_i = 1$ for all i) and the consequence is a single attribute, which can be either true ($a = 1$) or false ($a = 0$). Since a has only two values, the resulting rules can be expressed simply as $X \rightarrow A$ and $X \rightarrow \neg A$. Because negative dependency between X and A is the same as positive dependency between X and $\neg A$, it is enough to search for only positive dependencies.

This type of dependency rules are sufficient in many domains. For example, in ecology, an interesting problem is to find dependencies between the occurences of plant, fungus, or animal species. Dependency rules reveal which species groups indicate the presence or absence of other species. This knowledge on ecosystems is important, when, for example, one tries to locate areas for protection, understand how to prevent erosion, or restore vegetation on already damaged sites. In addition to species information, other factors can be included by binarizing the attributes. For example, soil moistness levels can be expressed by three binary attributes corresponding to dry, moist, and wet soil; the acidity of soil can be expressed by binary attributes corresponding to certain pH levels, and so on. In gene data, one can simply create a new attribute for each allele. Environmental factors like drugs used, the subject's life style, or habits can be included with a simple binarization.

1.2 The curse of redundancy

For the correct interpretation of dependency rules it is important that the rules are *non-redundant*. This means that the condition contains only those attributes, which contribute positively to the dependency. Generally, adding new attributes to the antecedent of a dependency rule, can 1) have no effect, 2) weaken the dependency, or 3) strengthen the dependency. When the task is to search for only positive dependencies, the first two cases can only add redundancy. The third case is more difficult to judge, because now the extra attributes make the dependency stronger, but at the same time the rule usually becomes less frequent. In an extreme case, the rule is so rare that it has no statistical validity. Therefore, we need a statistical significance measure to decide whether the new attributes made the rule better or worse.

Example 1.1 A common task in medical science is to search factors which are either positively or negatively dependent with a certain disease. If all

possible dependency rules are generated, a large proportion of them is likely to be redundant. Let us now consider all three types of redundant rules, given a non-redundant, significant rule $X \rightarrow \neg A$, where X stands for a low-fat diet and physical exercise and A for liver cirrhosis. The interpretation of the rule is that a low-fat diet and physical exercise are likely to prevent liver cirrhosis. We assume that $P(\neg A|X) < 1$, which means that in principle more specific rules could express a stronger dependency.

Now it is possible to find several rules of the form $XB \rightarrow \neg A$, which express equally strong dependency to $X \rightarrow \neg A$, but where B has no effect of A. For example, B could stand for listening Schubert, reading Russian classics, or playing Mahjong. Since $P(\neg A|XB) = P(\neg A|X)$, also $P(\neg A|X\neg B) = P(\neg A|X)$, $P(A|XB) = P(A|X)$, and $(PA|X\neg B) = P(A|X)$. In the worst case, where $X \rightarrow \neg A$ was the only dependency in data, we could generate an exponential number of its redundant specializations, none of which would produce any new information. In fact, the specializations could do just the opposite, if the doctor assumed that Dostoyevsky is needed to prevent liver cirrhosis, in addition to healthy diet and exercise.

An even more serious misinterpretation could occur, if B actually weakened the dependency. For example, if B means regular alcohol consumption, there is a well-known positive dependency $B \rightarrow A$ (alcohol causes liver cirrhosis). Now B weakens rule $X \rightarrow \neg A$, but $XB \rightarrow \neg A$ can still express a positive dependency. However, it would be quite fatal to conclude that a low-fat diet, physical exercise, and regular alcohol consumption help to prevent the cirrhosis. In fact, if we analyzed all rules (including rules where the antecedent contains any number of negations), we could find that $P(\neg A|X\neg B) > P(\neg A|X)$ and $P(A|\neg XB) > P(A|B)$, i.e. that avoiding alcohol strengthens the good effect of the diet and exercises, while the diet and exercises can weaken the bad effect of alcohol. However, as we allow only restricted occurrences of negations, a misinterpretation could easily occur, unless redundant rules were pruned.

An example of the third kind of redundancy could be rule $XB \rightarrow \neg A$, where B means that the patient goes in for Fengshui. If the data set contains only a few patients who go in for Fengshui, then it can easily happen that $P(\neg A|XB) = 1$. The rule is the strongest possible and we could assume that Fengshui together with a healthy diet and exercises can totally prevent the cirrhosis. However, the dependency is so rare that it could just be due to chance. In addition, there are several patients who follow a healthy diet and go in for sports, but not for Fengshui. Therefore, the counterpart of rule $XB \rightarrow \neg A$, $\neg(XB) \rightarrow A$, is probably weaker than

$\neg X \rightarrow A$. This time, the redundancy is not discovered by comparing the confidences $P(\neg A|X)$ and $P(\neg A|XB)$, but instead we should compare the statistical significance of the rules, before drawing any conclusions on Fengshui.

In this thesis, we generalize the classical definition of redundancy (e.g. [1]), and call a rule $X \rightarrow A = a$ redundant, if there exists a more general but at least equally good rule $Y \rightarrow A = a$, where $Y \subsetneq X$. The goodness can be measured by any statistical significance function. Typically, all statistical significance measures require [64] that the more specific (and less frequent) rule $X \rightarrow A = a$ should express a stronger dependency between X and $A = a$, $\neg X$ and $A \neq a$, or both, to achieve a better value than the more general rule. However, the notion of redundancy does not yet guarantee that the improvement in the strength of the rule is statistically significant. Therefore, some authors (e.g. [75]) have suggested that instead of redundancy we should test *productivity* of the rule with respect to more general rules.

Rule $X \rightarrow A = a$ is called productive with respect to $Y \rightarrow A = a$, $Y \subsetneq X$, if $P(A = a|X) > P(A = a|Y)$. The improvement is considered significant if the probability that it had occurred by chance is less than some pre-defined threshold max_p. The problem of this approach is that it compares only $P(A = a|X)$ and $P(A = a|Y)$, but not their counterparts $P(A \neq a|\neg X)$ and $P(A \neq a|\neg Y)$. Therefore, it is possible that a rare but strong rule passes the test, even if it is less significant than its generalizations. Thus, the significant productivity alone does not guarantee non-redundancy. This problem will be further analyzed in Section 2.4.

1.3 The curse of dimensionality

The search problem for the most significant, non-redundant dependency rules can occur in two forms: In the *enumeration problem*, the task is to search for all non-redundant rules whose goodness value M does not exceed some pre-defined threshold α. In the *optimization problem*, the task is to search for the K best non-redundant dependency rules, given the desired number of rules K. Typically, the enumeration problem produces a larger number of rules, but with suitable parameter settings the results are identical. In the following, we will concentrate on the optimization problem, but – if not otherwise stated – the results are applicable to the enumeration problem, as well.

Searching for the best non-redundant dependency rules is computationally a highly demanding problem, and no polynomial time algorithms are

known. This is quite expected, because even the simpler problem – searching for the best classification rule $X \to C$ with a fixed consequence C – is known to be NP-hard with common goodness measures [57]. Given a set of attributes, R, the whole search space consists of all attribute combinations in $\mathcal{P}(R)$. For each attribute set $X \in \mathcal{P}(R)$, we can generate $|X|$ rules of the form $X \setminus \{A\} \to A$, where $|X|$ is the number of attributes in X. For each X, there is an equal number of negative rules $X \setminus \{A\} \to \neg A$, but it does not affect the number of all possible rules, because $X \to A$ and $X \to \neg A$ are mutually excluding.

Therefore, the total number of possible rules is $\sum_{i=2}^{k} i \binom{k}{i} = k2^{k-1} - k$ (Equation (A.5)), where $k = |R|$. For example, if the number of attributes is 20, there are already over 10 million possible dependency rules of the desired form. In many real-world data sets, the number of attributes can be thousands, and even if most of the attribute combinations do not occur in the data, all possible rules cannot be checked.

This so-called *curse of dimensionality* is inherent in all rule discovery problems, but statistical dependency rules have an extra difficulty – the lack of monotonicity. Statistical dependence is not a *monotonic property*, because $X \to A = a$ can express independence, even if its generalization $Y \to A = a$, $Y \subsetneq X$, expressed dependence. It is also possible that $X \to A = a$ expresses negative (positive) dependence, even if its generalization $Y \to A = a$ expressed positive (negative) dependence. On the other hand, statistical dependence is neither an *anti-monotonic property*, because $Y \to A = a$ can express independence, even if its specialization $X \to A = a$ expressed dependence. In practice this means that the lack of simple dependency rules, like $A \to B$, does not guarantee that there would not be significant dependencies in their specializations, like $ACDE \to B$.

This problem is especially problematic in genetic studies, where one would like to find statistically significant dependencies between alleles and diseases. Traditionally, scietists have tested only two-attribute rules, where a disease depends on only one allele. In this way, they have already found thousands of associations, which explain medical disorders. However, it is known that many serious and common diseases like heart disease, diabetes, and cancers, are caused by complex mechanisms, involving multiple genes and environmental factors. Analyzing these dependencies is a hot topic, but it has been admitted that first we should develop efficient computational methods. As a middle-step, the scientists have begun to analyze interactions between known or potential factors affecting a single disease. If the attribute set is relatively small (up to a few hundred alleles) and there is enough computer capacity it is possible to check dependencies in-

volving even 15 attributes. However, if all interesting genes are included, the number of attributes can be 500 000–1 000 000, and testing even the pair-wise dependencies becomes prohibitive.[22, 34, 54]

Since the brute-force approach is infeasible, a natural question is to ask whether the existing data mining methods could be applied to the problem. Unfortunately, the problem is little researched, and no efficient solutions are known. In practice, the existing solutions in data mining fall into two approaches: the first approach is to search for *association rules* [2] first, and then select the statistically significant dependency rules in the post-processing phase. The second approach is to search for statistical dependency rules directly with some statistical goodness measure.

The first approach is quite ineffective, because the objective of association rules is quite different from the dependency rules. In spite of its name, an association rule does not express an association in the statistical sense, but only a combination of frequently occurring attributes. The general idea of association rule discovery algorithms is to search for all rules of the form $X \rightarrow A$, where the relative frequency of event XA is $P(XA) \geq min_{fr}$ for some user-defined threshold min_{fr}. This does not yet state that there is any connection between X and A, and therefore it is customary to require that either the confidence of the rule, $P(A|X)$, or the *lift* of the rule, $P(A|X)/P(A)$, is sufficiently large. The minimal confidence requirement states some connection between X and A, but it is not necessarily (positive) statistical dependence. If the confidence $P(A|X)$ is large, then A is likely to occur, if X had occured, but A may still be more likely, if X had not occurred. The minimal lift requirement guarantees the latter, and it can be used to select all sufficiently strong statistical dependencies among frequent rules.

The problem of this approach is that many dependency rules are missed, because they do not fit the minimum frequency requirement, while a large number of useless rules is generated in vain. In practice, the required minimum frequency threshold has to be relatively large for feasible search. With large-dimensional data sets the threshold can be 0.60-0.80, which means that the dependency should occur on at least 60–80% of the rows. Such rules cannot have a high lift value and the strongest and most significant dependency rules are missed. In the worst case, all discovered rules can be either independence rules or express insignificant dependencies [77]. In addition, a large proportion of rules will be redundant, and pruning them afterwards among millions of rules is time consuming. We note that there exist association rule discovery algorithms, which are said to search only "non-redundant rules". However, usually the word "redundancy" is used

in another sense, referring to rules that are not necessary for representing the collection of all frequent rules, or whose frequency can be derived from other rules (e.g. [48]). In Appendix B.2.1, we analyze the most popular of such condensed representations, *closed sets* [61, 9] and *free sets* [17], and show why they do not suit for searching for non-redundant statistical dependency rules, even if the minimum frequency threshold could be set sufficiently low.

In spite of the above-mentioned statistical problems, association rule algorithms and frequency-based search in general have been popular in the past. The reason is that frequency is an anti-monotonic property, which enables an efficient pruning, namely, if a set Y is too infrequent, i.e. $P(Y) < min_{fr}$ for some threshold min_{fr}, then all of its supersets $X \supseteq Y$ are also infrequent and can be pruned without checking. Therefore, most of the algorithms which search for statistical dependencies do also use frequency-based pruning with some minimum frequency thresholds, in addition to statistical search criteria.

The second approach consists of algorithms, which search for only statistical dependency rules. None of the existing algorithms searches for the statistically most significant dependencies, but it is always possible to test the significance afterwards. Still, the search algorithm may miss some of the most significant dependencies, if they were pruned during the search. Due to the problem complexity, a common solution is to restrict the search space of possible rules heuristically and require some minimum frequency, restrict the rule length, fix the consequent attribute, or use some other criteria to prune "uninteresting" rules.

The most common restriction is to search for only classification rules of the form $X \rightarrow C$ with a fixed consequent attribute C. Morishita and Sese [57] and Nijssen and Kok [60] introduced algorithms for searching for classification rules with the χ^2-measure. Both approaches utilized the convexity of the χ^2-function and determined a minimum frequency threshold, which guarantees a certain minimum χ^2-value. This technique is quite inefficient (dependening on the frequency $P(C)$), because the resulting minimum frequency thresholds are often too low for efficient pruning. A more efficient solution was developed by Nijssen et al. [59] for searching for the best classification rule with a *closed set* in the rule antecedent (i.e. rule $X \rightarrow C$ such that for all $Z \supsetneq X$, $P(X) > P(Z)$). The goodness measure was not fixed to the χ^2-measure, but other zero-diagonal convex measures like the information gain could be used, as well. The problem of this approach is that the closed sets can contain redundant attributes and the rule can be sub-optimal in the future data (see Appendix B.2.2).

The χ^2-measure was also used by Liu et al. [47], in addition to minimum frequency thresholds and some other pruning heuristics, to test the goodness of rule $X \to C$ and whether the productivity of rule $X \to C$ with respect to its immediate generalizations ($Y \to C$, $X = YB$ for some attribute B) was significant. It is customary to test the productivity only against the immediate generalizations, because checking all $2^{|X|}$ sub-rules of the form $Y \to A$, $Y \subsetneq X$ is inefficient. Unfortunately, this also means that some rules may appear as productive, even if they are weaker than some more general rules.

Li [45] introduced an algorithm for searching for only non-redundant classification rules with different goodness easures, including confidence and lift. No minimum frequency thresholds were used, but instead it was required that $P(X|A)$ was larger than some predefined threshold.

Searching for general dependency rules with any consequent attribute is a more difficult and less studied problem. Xiong et al. [82] represented an algorithm for searching for positive dependencies between just two attributes using an upper bound for Pearson's correlation coefficient as a measure function. Webb's *MagnumOpus* software [76, 78] is able to search for general dependency rules of the form $X \to A$ with different goodness measures, including *leverage*, $P(XA) - P(X)P(A)$, and lift. In addition, it is possible to test that the rules are significantly productive. In practice, MagnumOpus tests the productivity of rule $X \to A$ with respect to its immediate generalizations ($Y \to A$, $X = YB$ for some B) with Fisher's exact test. Otherwise, redundant rules are not pruned and the rules are not necessarily the most significant. The algorithm is quite well scalable, and if only the K best rules are searched for with the leverage, it is possible to perform the search without any minimum frequency requirement. However, we note that the leverage itself favours rules with a frequent consequent, and therefore some significant and productive rules can be missed.

All of the above-mentioned algorithms search only for positive dependencies, but there are a couple of approaches which have also searched for negative dependency rules. Antonie and Zaïane [6] introduced an algorithm for searching for negative rules using a minimum frequency threshold, minimum confidence threshold, and Pearson's correlation coefficient as a goodness measure. Since Pearson's correlation coefficient for binary attributes is the same as $\sqrt{\chi^2/n}$, where n is the data size, the algorithm should be able to search dependencies with the χ^2-measure, as well.

Thiruvady and Webb [71] represented an algorithm for searching for general positive and negative dependency rules of the form $(X = x) \to (A = a)$, $x, a \in \{0, 1\}$, using the leverage as a goodness measure. No mini-

mum frequency thresholds were needed, but the rule length was restricted. Wu et al. [80] also used leverage as a goodness measure together with a minimum frequency threshold and a minimum threshold for the *certainty factor* $(P(A = a|X) - P(A = a))/P(A \neq a)$, $a \in \{0, 1\}$. Koh and Pears [40] suggested a heuristic algorithm, where a minimum frequency threshold was derived from Fisher's exact test. Unfortunately, the algorithm was based on the faulty assumption that the statistical dependence would be an anti-monotonic property. However, the algorithm should be able to find positive and negative dependency rules among two attributes correctly.

Searching for general dependency rules, where X can contain any number of negations, is an even more complex problem, and no efficient solutions are known. Meo [53] analyzed a related problem and developed a method for searching for attribute sets X, where variables corresponding to $X \setminus \{A\}$ and A are statistically dependent for any $A \in X$, and the dependence is not explained by any simpler dependencies in the immediate subsets Y, $X = YB$. Since the objective was to find dependencies between variables, and not events, all $2^{|X|}$ attribute value combinations had to be checked. The method was applied to select attribute sets containing dependencies from a set of all frequent attribute sets. Due to obvious complexity, the method is poorly scalable, depending on the data distribution and the minimum frequency threshold, and in practice it may be necessary to restrict the size of X, too.

In addition, some researchers (e.g. [18, 39]) have studied another related problem, where statistically significant "correlated sets" are searched for. The idea is that the probability of set X should deviate from its expectation, i.e. $P(X) \neq \prod_i^l P(A_i)$, where $X = A_1 \cdots A_l$. While interesting for their own sake, the correlated sets do not help to find statistically significant dependency rules [56].

1.4 Results and contribution

Figures 1.1 and 1.2 summarize the research problem. Figure 1.1 (a) shows the universe $\mathcal{U}$ of all possible rules on the given set of attributes; set $\mathcal{S}_r$ is the collection of the most significant dependency rules, and its subset $\overline{\mathcal{R}}_r$ contains only the non-redundant rules. The set of the most significant rules depends on the selected goodness measure, M. For simplicity, we assume that M is increasing by goodness, i.e. high values of M indicate a good rule. Then, $\mathcal{S}_r$ consists of those rules that would gain a measure value $M \geq min_M$ in a sample of size n, if the sample were perfect (i.e. if the estimated probabilities were fully correct, $P_r = P$). Similarly, $\overline{\mathcal{R}}_r$

consists of rules, which would be the most significant, non-redundant rules in a perfect sample of size n. Ideally, we would like to find the rules in collection $\overline{\mathcal{R}}_r$.

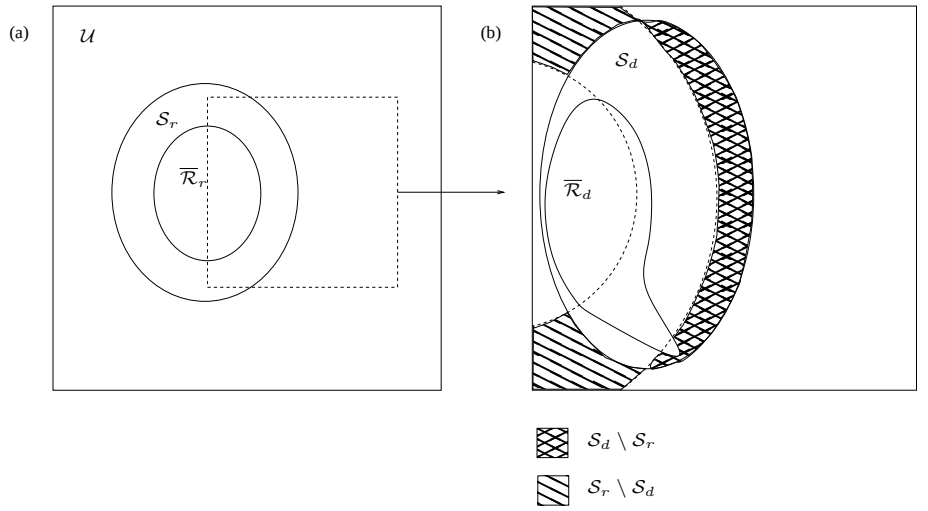


Figure 1.1: (a) Actually significant ($\mathcal{S}_r$) and non-redundant ($\overline{\mathcal{R}}_r$) dependency rules. (b) Magnification of the dashed rectagle in (a) showing rules which appear as significant ($\mathcal{S}_d$) and non-redundant ($\overline{\mathcal{R}}_d$) in the data.

The statistical problem is that we are given just a finite (and often noisy) real-world data set, where the observed probabilities P are only inaccurate estimates for the real probabilities P_r. As a result, the measure values M are also inaccurate, and it is possible that some spurious rules appear to be significant while some actually significant dependency rules appear to be insignificant. This is shown in Figure 1.1 (b), which contains all possible rules that occur in the data (have frequency $P(XA) \geq 1$). Set $\mathcal{S}_d$ is the collection of all dependency rules that appear to be significant in the data, and subset $\overline{\mathcal{R}}_d$ contains all significant rules, which appear to be non-redundant in the data. By tailoring the significance threshold min_M, it is possible to decrease either the set of spurious rules $\mathcal{S}_d \setminus \mathcal{S}_r$ or the set of missing significant rules $\mathcal{S}_r \setminus \mathcal{S}_d$, but not both. Similarly, stronger requirements for the non-redundancy can decrease the set of spuriously non-redundant rules $\overline{\mathcal{R}}_d \setminus \overline{\mathcal{R}}_r$, while weaker requirements can decrease the set of missing, actually non-redundant rules $\overline{\mathcal{R}}_r \setminus \overline{\mathcal{R}}_d$.

Figure 1.2 demonstrates the search problem and the resulting extra errors. Set $\mathcal{T}$ consists of all rules which can be found with existing search

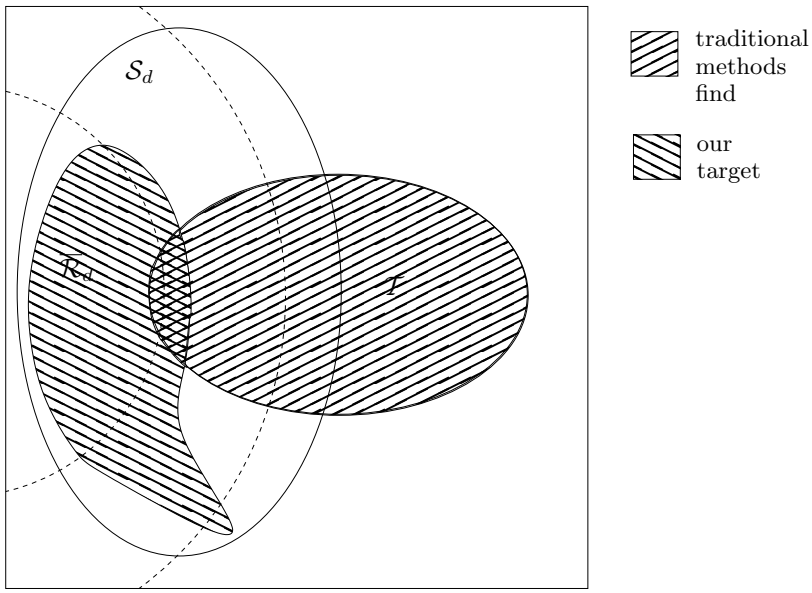


Figure 1.2: Rules discovered by traditional methods ($\mathcal{T}$) and our target rules ($\overline{\mathcal{R}}_d$). Dash lines refer to boundaries of $\mathcal{S}_r$ and $\overline{\mathcal{R}}_r$ in Figure 1.1.

methods. Due to problem complexity, all of them use either restricting pruning heuristics (especially the minimum frequency requirement) or inappropriate measure functions, which can catch only a small proportion of significant dependency rules in $\mathcal{S}_d$. This is especially true in large-dimensional, dense data sets, where it can happen that none of the most significant rules are discovered. In addition, the traditional methods produce a large number of non-significant or redundant rules. The latter can be pruned out in the post-processing phase, although it is laborious, but there is no way to restore the set of missing rules $\mathcal{S}_d \setminus \mathcal{T}$.

In this research, the objective is to develop effective search methods for the set $\overline{\mathcal{R}}_d$ – dependency rules which are the most significant and non-redundant in the data. We do not take a stand on how the related statistical problem should be solved, but clearly the set $\overline{\mathcal{R}}_d$ is a better approximation for the ideal rule set $\overline{\mathcal{R}}_r$ than $\mathcal{T}$. The new search algorithms are developed on such a general level that it is possible to apply different definitions of statistical dependency and different significance measures. We have adopted a general definition of redundancy, which leaves space for extra pruning of non-redundant rules with stricter criteria, like productivity.

The main result of this research is a new generic search algorithm, which

is able to perform the search efficiently compared to the existing solutions without any of their extra requirements for the frequency or complexity of the rule. Since no sub-optimal search heuristics are used, the resulting dependency rules are globally optimal in the given data set. In the algorithm design, the emphasis has been to minimize the time complexity, without any special attention to the space requirement. However, in practice, the implementations are capable of handling all the classical benchmark data sets (including ones with nearly 20,000 attributes) with a normal desktop computer.

In addition to the classical problem of searching for only positive dependency rules, the new algorithm can also search for negative dependencies with certain significance measures (e.g. Fisher's exact test and the χ^2-measure). This is an important step forward in some application domains, like gene–disease analysis, where the information on negative dependencies is also valuable.

The development of new efficient pruning properties has also required comprehensive theoretical work. In this thesis, we develop the theory for evaluating the satistical significance of dependency rules in different frameworks. Especially, we formalize the notion of well-behaving goodness measures, uniformly with the classical axioms for any "proper" measures of dependence. The most important new insight is that all well-behaving measures obtain their best values at the same points of the search space. In addition, they share other useful properties, which enable efficient pruning of redundant rules.

Most of the results of this thesis have been introduced in the following papers. The author has been the main contributor in all of them. The papers contain also extra material, which is not included in this thesis.

1. W. Hämäläinen and M. Nykänen. Efficient discovery of statistically significant association rules. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008)*, pages 203–212, 2008.

2. W. Hämäläinen. Lift-based search for significant dependencies in dense data sets. In *Proceedings of the Workshop on Statistical and Relational Learning in Bioinformatics (StReBio'09), in the 15th ACM SIGKDD conference on Knowledge Discovery and Data Mining*, pages 12–16. ACM, 2009.

3. W. Hämäläinen. Statapriori: an efficient algorithm for searching statistically significant association rules. *Knowledge and Informa-*

tion Systems: An International Journal (KAIS), 23(3):373–399, June 2010.

4. W. Hämäläinen. Efficient search methods for statistical dependency rules. *Fundamenta Informaticae*, 2010. Accepted pending revisions.

5. W. Hämäläinen. Efficient discovery of the top-**K** optimal dependency rules with Fisher's exact test of significance. Submitted 2010.

6. W. Hämäläinen. General upper bounds for well-behaving goodness measures on dependency rules. Submitted 2010.

7. W. Hämäläinen. New tight approximations for Fisher's exact test. Submitted 2010.

1.5 Organization

The rest of the thesis is organized as follows. In Chapter 2, we introduce the main concepts related to statistical dependence, dependency rules, statistical significance testing in different frameworks, and redundancy and improvement of dependency rules. In Chapter 3, we formalize the pruning problem and introduce well-behaving goodness measures. The theoretical basis for the search with any well-behaving goodness measures is given there. In Chapter 4, we analyze different strategies to implement the search. We introduce a new efficient pruning property, which can be combined with the basic branch-and-bound search. As a result, we intoduce a generic search algorithm, and analyze its time and space complexity. In Chapter 5, we report experiments on classical benchmark data sets. The final conclusions are drawn in Chapter 6.

At the end of the thesis are appendices, which complement the main text. Appendix A contains general mathematical results, which are used in the proofs. Appendix B contains auxiliary results (proofs for statements in the main text). Implementation details and their effect on the complexity are described in Appendix C.

Chapter 2

Statistically significant dependency rules

> *One of the most important problems in the philosophy of natural sciences is – in addition to the well-known one regarding the essence of the concept of probability itself – to make precise the premises which would make it possible to regard any given real events as independent. This question, however, is beyond the scope of this book.*
>
> A.N. Kolmogorov

In spite of their intuitive appeal, the notions of statistical dependence and – especially – statistically significant dependence are ambiguous. In this thesis, we do not try to solve which are the correct definitions or measure functions for statistically significant dependency rules, but instead we try to develop generic search methods consistent with the most common interpretations.

In this chapter, we define dependency rules on a general level, and consider two alternative interpretations – variable-based and value-based semantics – what the dependency means. We recall the general idea of the statistical significance of a dependency and give statistical measures for evaluating the significance. Finally, we discuss the notions of redundancy and improvement.

2.1 Statistical dependency rules

In its general form, a dependency rule on binary data is defined as follows:

Definition 2.1 (Dependency rule) Let R be a set of binary attributes, $X \subsetneq R$ and $Y \subseteq R \setminus X$ sets of binary attributes, and $Dom(X) = \{0, 1\}^l$, $|X| = l$, and $Dom(Y) = \{0, 1\}^m$, $|Y| = m$, their domains.

Let $Dep(\alpha, \beta)$ be a symmetric relation which defines statistical dependence between events ω_1 and ω_2; $Dep(\omega_1, \omega_2) = 1$, if ω_1 and ω are statistically dependent, and $Dep(\omega_1, \omega_2) = 0$, otherwise.

For all attribute value combinations $\overline{x} \in Dom(X)$ and $\overline{y} \in Dom(Y)$, rules $X = \overline{x} \to Y = \overline{y}$ and $Y = \overline{y} \to X = \overline{x}$ are dependency rules, if $Dep(X = \overline{x}, Y = \overline{y}) = 1$.

First, we note that the direction of the rule is customary, because the relation of statistical dependence is symmetric. Often, the rule is expressed in the order, where the confidence of rule ($cf(X = \overline{x} \to Y = \overline{y}) = P(Y = \overline{y}|X = \overline{x})$ or $cf(Y = \overline{y} \to X = \overline{x}) = P(X = \overline{x}|Y = \overline{y})$) is maximal.

In this thesis, we concentrate on a special case of dependency rules, where 1) the consequence $Y = \overline{y}$ consist of a single attribute-value combination, $A = a$, $a \in \{0, 1\}$, and 2) the antecedent $X = \overline{x}$ is a conjunction of true-valued attributes, i.e. $(X = \overline{x}) \equiv (A_1 = 1, \ldots, A_l = 1)$, where $X = \{A_1, \ldots, A_l\}$. With these restrictions, the resulting rules can be expressed in a simpler form $X \to A = a$, where $a \in \{0, 1\}$, or $X \to A$ and $X \to \neg A$. We note that with certain relations of dependency, this type of rules includes also rules of form $\neg X \to A$ and $\neg X \to \neg A$, because they are considered equivalent to $X \to \neg A$ and $X \to A$, respectively.

These two restrictions on the type of rules are commonly made also in the previous research on association rules, mainly for computational reasons. Allowing several attributes in the consequent has only a small effect on the number of all possible rules, but it complicates the definition of the redundancy and also the search for non-redundant rules. In this case, each attribute set Z can be divided into consequence and antecedent parts in $2^{|Z|} - 2$ ways (instead of $|Z|$ ways), and each rule $X \to Z \setminus X$ can be redundant with respect to $2^{|Z|} - 2^{|X|} - 2^{|Z \setminus X|}$ more general rules (see Theorems B.1 and B.2). Another reason for allowing just a single attribute in the consequence is to simplify the interpretation and application of discovered dependency rules.

Allowing any number of negations in the antecedent part increases the computational complexity of the problem remarkably. Now for each set Z there are $2^{|Z|}$ different value combinations and for each of them we can generate $|Z|$ rules with a single attribute in the consequence. The total number of all possible rules is $\mathcal{O}(k3^k)$, instead of $\mathcal{O}(k2^k)$ (see Theorem B.3).

Let us now consider what it means that there is a statistical dependency between an attribute set X and an attribute value assignment $A = a$. There are at least two different interpretations, depending on whether X and $A = a$ are interpreted as events or random variables.

Traditionally, the statistical dependence between events is defined as follows (e.g. [68, 53]):

Definition 2.2 (Independent and dependent events) Let R be as before, $X \subsetneq R$ a set of attributes, and $A \in R \setminus X$ a single attribute.

For any truth-value assignments $a, x \in \{0, 1\}$ events $X = x$ and $A = a$ are statistically independent, if $P(X = x, A = a) = P(X = x)P(A = a)$.

If the events are not independent, they are dependent. When $P(X = x, A = a) > P(X = x)P(A = a)$, the dependence is positive, and when $P(X = x, A = a) < P(X = x)P(A = a)$, the dependence is negative.

We note that actually the independence condition holds for the real but unknown probabilities $P_r(X = x, A = a)$, $P_r(X = x)$, and $P_r(A = a)$. Therefore, the condition for their frequency-based estimates in some real world data set is sometimes expressed as $P(X = x, A = a) \approx P(X = x)P(A = a)$. Because the frequency-based estimates for probabilities are likely to be inaccurate, it is customary to call events dependent only if the dependency is sufficiently strong. The strength of the statistical dependence between $X = \overline{x}$ and $A = a$ can be measured by *lift* (also called *interest* [18], *dependence* [81], or *degree of independence* [83]):

$$\gamma(X = x, A = a) = \frac{P(X = x, A = a)}{P(X = x)P(A = a)}.$$

Another commonly used measure is *leverage* (also called *dependence value* [53])

$$\delta(X = x, A = a) = P(X = x, A = a) - P(X = x)P(A = a).$$

The difference between the leverage and lift is that the first one measures the absolute deviation of the observed relative frequency $P(X = x, A = a)$ from its expectation $P(X = x)P(A = a)$, if the events were independent, while the lift measures a relative deviation from the expectation. If the dependency rules are searched by the strength of the dependence (like in [75, 45]), the results can be quite different, depending on which measure is used. However, both measures can also produce spurious rules, which are due to chance, and therefore we will use other goodness measures as search criteria.

Statistical dependence between variables is defined as follows (e.g. [68, 53]):

Definition 2.3 (Independent and dependent variables) Let V and W be discrete random variables, whose domains are $Dom(V)$ and $Dom(W)$. V and W are statistically dependent, if $P(V = v, W = w) = P(V = v)P(W = w)$ for all values $v \in Dom(V)$ and $w \in Dom(W)$. Otherwise they are dependent.

Once again, the variables are called dependent only if the overall dependence is sufficiently strong. To estimate the strength of the overall dependence, we need a measure (like the χ^2-measure or mutual information MI) which takes into account all dependencies between value combinations v and w.

For dependency rule $X \rightarrow A = a$, we have now two alternative interpretations, which are sometimes called *value-based* and *variable-based* semantics [14] of the rule. In the value-based semantics, only dependence between events $X = 1$ and $A = a$ is considered, while in the variable-based semantics X and A are considered as random variables (above V and W), whose dependence is measured. In the latter case, the dependencies in all four possible event-combinations, XA, $X\neg A$, $\neg XA$, and $\neg X\neg A$, are evaluated.

Table 2.1: A contingency table with absolute frequencies of XA, $X\neg A$, $\neg XA$ and $\neg X\neg A$. Absolute leverage $\Delta = n\delta$ is the deviation from the expectation under independence, where n is the data size and δ is the leverage.

	A	$\neg A$	Σ
X	$m(XA) = m(X)P(A) + \Delta$	$m(X\neg A) = m(X)P(\neg A) - \Delta$	$m(X)$
$\neg X$	$m(\neg XA) = m(\neg X)P(A) - \Delta$	$m(\neg X\neg A) = m(\neg X)P(\neg A) + \Delta$	$m(\neg X)$
Σ	$m(A)$	$m(\neg A)$	n

The two interpretations are closely related, because for binary variables X and A, all value combinations $X = x, A = a\ x, a \in \{0, 1\}$, have the same absolute value of leverage, $|\delta|$, as shown in Table 2.1. Therefore, $|\delta|$ can now be used to measure the strength of the dependence also in the variable-based semantics. However, all four events have generally different lift values, and therefore some events may be strongly dependent, while the overall dependency can be considered weak. This can happen, if the expected frequency $P(X = x)P(A = a)$ is small and the dependency between $X = x$ and $A = a$ is strong.

Example 2.4 In the gene–disease data, many alleles or allele combinations (here X) occur quite rarely, but still they can have a strong effect on a certain disorder D. In an extreme case, the occurrence of X indicates always the occurrence of D, but D can occur also with other allele combinations. Now $P(D|X) = 1$ and the lift $\gamma(X \to D) = P(D)^{-1}$, which can be quite strong. Similarly, $\gamma(X \to \neg D) = 0$ indicates a strong negative dependency between X and $\neg D$. However, the other two events express only a weak dependency and the lift values $\gamma(\neg X \to D)$ and $\gamma(\neg X \to \neg D)$ are near 1. The leverage, $\delta(X, D) = P(X)P(\neg D)$, is also small, because $P(X)$ was small.

In the value-based semantics, dependency rule $X \to D$ would be considered strong, but in the variable-based semantics, it would be considered weak. Since the search algorithms use goodness measures, which reflect the strength of the dependency, the rule could be either discovered or undiscovered during the search, depending on the interpretation.

The two interpretations of dependency rule $X \to A = a$ lead also to different conclusions, when the dependency rule is used for prediction. Both value-based and variable-based semantics state that $A = a$ is more likely to occur, if X had occurred, than if X had not occurred. In the variable-based semantics, we can also state that $A \neq a$ is more likely to occur, if X had not occurred, than if it had occurred. However, in the value-based semantics this dependency can be so weak that no predictions can be made on $A = a$ or $A \neq a$, given that X had not occurred.

Finally, we note that there is also a third possible interpretation concerning the dependence between X and A. Attribute set X could also be thought as a variable, whose values are all possible $2^{|X|}$ value combinations $\overline{x} \in Dom(X)$. In data mining, this approach is rarely taken, due to computational complexity of the search. Another problem is that when $|X|$ is relatively large, many attribute value combinations do not occur in the data at all or so infrequently that the corresponding probabilities cannot be estimated accurately.

On the contrary, in machine learning, this interpretation is commonly used, when classifiers or Bayesian networks are learned from binary data. However, in these applications, the data sets are typically much smaller than in data mining and also the problem is easier, because the objective is not to check all sufficiently strong dependencies or even the K best dependencies, but only to find (heuristically) one sufficiently good dependency model.

2.2 Statistical significance testing

The main idea of statistical significance testing is to estimate the probability that the observed discovery has occurred by chance. If the probability is very small, we can assume that the discovery is genuine. Otherwise, it is considered spurious and discarded. The probability can be estimated either analytically or empirically. The analytical approach is used in the *traditional significance testing*, while *randomization tests* estimate the probability empirically.

Traditional significance testing can be further divided into two main classes: the frequentist and Bayesian approaches. The frequentist approach is the most commonly used and best studied (see e.g. [29, Ch. 26] or [46, Ch. 10.1]). The main idea is to estimate the probability of an observed or a rarer phenomenon under some null hypothesis. When the objective is to test the significance of the dependency between events or variables X and A, the null hypothesis H_0 is the independence assumption: $P(X, A) = P(X)P(A)$. For this purpose, we define a random variable N_{XA} which gives the absolute frequency of event XA. The task is to calculate the probability $p = P(N_{XA} \geq m(X, A) \mid H_0)$ that XA occurs at least $m(X, A)$ times in the given data set, if X and A were actually independent.

In the classical (Neyman-Pearsonian) hypothesis testing, the p-value is compared to some pre-defined threshold α. If $p \leq \alpha$, the null hypothesis is rejected and the discovery is called significant at level α. Parameter α defines the probability of committing a *type I error*, i.e. accepting a spurious rule. Another parameter, β, is used to define the probability of committing a *type II error*, i.e. rejecting a genuine dependency rule as non-significant. The problem is how to decide suitable thresholds, and often only the p-values are reported.

Deciding threshold α is even harder in the data mining where numerous patterns are tested. For example, if we use threshold $\alpha = 0.05$, then there is a 5% chance that a spurious rule passes the significance test. If we test 10 000 rules, it is likely that we will find 500 spurious rules. This so called *multiple testing problem* is inherent in the knowledge discovery, where one often performs an exhaustive search over all possible patterns.

As a solution, the more patterns we test, the stricter bounds for the significance we should use. The most famous correction method is *Bonferroni adjustment* [66], where the desired significance level p is divided by the number of tests m. In the dependency rule discovery, we can give an upper bound for the number of rules to be tested [75]. An alternative is to control the expected number of errors among selected rules [10]. In this thesis, we do not try to solve this problem, but instead our objective is to

develop search methods for the dependency rules with the best p-values – i.e. the most genuine dependencies.

The idea of Bayesian significance testing is quite similar to the frequentist approach, but now we assign some prior probabilities $P(H_0)$ and $P(H_1)$ to the null hypothesis H_0 and the research hypothesis H_1. The conditional probabilities $P(N_{XA} \geq m(X,A) \mid H_0)$ and $P(N_{XA} \geq m(X,A) \mid H_1)$ are estimated from the data, and the posterior probabilities of hypotheses $P(H_0 \mid N_{XA} \geq m(X,A))$ and $P(H_1 \mid N_{XA} \geq m(X,A))$ are calculated by the Bayes' rule. The results are asymptotically similar (under some assumptions even identical) to the traditional hypothesis testing, although the Bayesian testing is sensitive to the selected prior probabilities [5].

The main problem of the traditional significance testing is the underlying assumption that the data is a random or otherwise representative sample from the whole population [69]. Often, the tests are used even if this assumption is violated (e.g. all available data is analyzed), but in this case, the discoveries describe only the available data set and there are no guarantees that they would hold in the whole population [27]. This is especially typical for exploratory data analysis and data mining, in general. In addition, there are problems, where random sampling is not possible even in principle, because the population is infinite (e.g. the researcher is interested in the dependence between employee's smoking habits and work efficiency in general, including already dead and still unborn people) [25, 7–8].

Another restriction concerning most of the traditional significance testing are the parametric assumptions on the underlying data distribution. The problem is that the p-value cannot be estimated, unless one has assumed some form of the distribution under the null hypothesis. Typically, one selects the distribution from some parametric family like binomial or normal distributions and estimates the missing parameters from the data. In large data sets, the exact form of the distribution is less critical, because all distributions tend to normal, and also the parameters can be estimated more accurately. However, in dependence testing, the most important factor is the assumed distribution of absolute frequencies $nP(X = x)P(A = a)$, $x, a \in \{0, 1\}$. If $m(X = x)$ or $m(A = a)$ is small, there are no guarantees that the corresponding random variable would be normally distributed or that the frequency estimate would be accurate.

Both of the above mentioned problems can be solved by randomization testing (e.g. [25]). The main idea of randomization tests is to estimate the significance of the discovery empirically, by testing how often the discovery occurs in other, randomly generated data sets, which share some properties

of the original data set. The most common approach is to fix some permutation scheme, how data is permutated, and select a random subset of all possible permutations.

If only a single dependency rule $X \rightarrow A$ is tested, it is enough to generate random distributions d_j for attributes $A_i \in X$ and A. Usually, it is required that all marginal probabilities $P(A_i)$ and $P(A)$ remain the same as in the original data. For each random distribution d_j, a test statistic M, which measures the goodness or strength of the rule, is calculated. Let us now assume that the measure M is increasing by the goodness of the rule (a higher value indicates a better rule). If the original data set produced M-value M_0 and m random distributions produced M-values $M_1, \ldots, M_m$, the empirical p-value of the rule is

$$p_{em} = \frac{|\{d_j | M_j \geq M_0\}|}{m}.$$

If the data set is relatively small and X is simple, it is possible to enumerate all possible distributions, where the marginal probabilities hold. This leads to an *exact permutation test*, which gives an exact p-value. On the other hand, if the data set is large and/or X is more complex, all possibilities cannot be checked, and the empirical p-value is less accurate. In this case, the test is called a *random permutation test* or an *approximate permutation test*.

The advantage of randomization tests is that the data does not have to be a random sample from a finite population. The reason is that the permutation scheme defines a population, which is randomly sampled, when new data sets are generated. Therefore, the discoveries can be assumed to hold in the population defined by the permutation scheme. In addition, the randomization test approach does not assume any underlying parametric distribution. Still, the classical parametric test statistics can be used to estimate the p-values.[25] The only problem is that it is not always clear, how the data should be permutated. The number of random permutations plays also an important role in the testing. The more random permutations are performed, the more accurate the empirical p-values are, but in practice, extensive permutating can be too time consuming. We also note that the randomization approach does not offer any direct method for the search of significant dependency rules; it can merely be used to validate the results, when good candidates have first been found by other means.

The idea of randomization tests can be extended for estimating the overall significance of all mining results. For example, Gionis et al. [30] tested the significance of the number of all frequent sets (given a minimum frequency threshold) and the number of all pair-wise correlations among

the most frequent attributes (measured by Pearson's correlation coefficient, given a minimum correlation threshold) using randomization tests. In this case, it is necessary to generate complete data sets randomly for testing. The difficulty is to decide what properties of the original data set should be maintained. As a solution, Gionis et al. kept both column marginals ($P(A_i)$s) and row marginals (numbers of 1s on each row) fixed, and new data sets were generated by *swap randomization* [21]. A prerequisite for this method is that the attributes are semantically similar (e.g. occurrence or absence of species) and it is sensible to swap their values. In addition, there are some pathological cases, where no or only a few permutations exist with the given row and column marginals, resulting a poor p-value, even if the original data set contains a significant pattern.[30]

2.3 Measures for statistical significance

Based on the definitions of a dependency rule and statistical dependence, a rule $X \to A = a$ can be called a dependency rule, only if the dependence between events $X = 1$ and $A = a$ (in the value-based semantics) or binary variables X and A (in the variable-based semantics) is sufficiently significant. For this evaluation, we need a significance measure M. In statistics, the significance of dependence in the variable-based semantics has been well studied and several measures are available. However, the significance of dependence in the value-based semantics has been only vaguely discussed in the data mining literature.

For any situation, the appropriate measure depends on the assumptions on the origin of data. Usually, it is thought that the data is a sample from a larger (possibly infinite) population, which follows a certain distribution. Assumptions on the form of the distribution and how its parameters are estimated lead to different measures.

In the following, we will consider the most common assumptions and the related significance measures in both variable-based and value-based semantics.

2.3.1 Variable-based measures

In the variable-based semantics , the significance of an observed dependency between X and A is determined by an independence test. The task is to estimate the probability of the observed or a *more extreme* contingency table (Table 2.1), assuming that X and A were actually independent. There is no consensus how the extremeness relation should be defined, but intuitively,

contingency table T_i is more extreme than table T_j, if the dependence between X and A is stronger in T_i than in T_j. So, a necessary condition for the extremeness of T_i over T_j is that the leverage in T_i is larger than in T_j. In the following, we will notate the relation "table T_i is equally or more extreme to table T_j" by $T_i \succeq T_j$.

The probability of each contingency table T_i depends on the assumed model $\mathcal{M}$. Model $\mathcal{M}$ defines the space of all possible contingency tables $\mathcal{T}_\mathcal{M}$ (under the model assumptions) and the probability $P(T_i|\mathcal{M})$ of each table $T_i \in \mathcal{T}_\mathcal{M}$. Because the task is to test independence, the assumed model should satisfy the independence assumption $P_r(XA) = P_r(X)P_r(A)$ in some form. For the probabilities $P(T_i|\mathcal{M})$ holds

$$\sum_{T_i \in \mathcal{T}_\mathcal{M}} P(T_i|\mathcal{M}) = 1.$$

Now the probability of the observed contingency table T_0 or of any at least equally extreme contingency table T_i, $T_i \succeq T_0$, is the desired p-value

$$p = \sum_{T_i \succeq T_0} P(T_i|\mathcal{M}). \tag{2.1}$$

Classically, the models for independence testing have been divided into three main categories (sampling schemes) [8, 62], which we call *multinomial*, *double binomial*, and *hypergeometric* models. In the statistics literature (e.g. [8, 72]), the corresponding sampling schemes are called *double dichotomy*, *2×2 comparative trial*, and *2×2 independence trial*.

In the following, we describe the three models using the urn metaphor. Because there are two binary variables of interest, X and A, we cannot use the basic urn model with white and black balls. Instead, we assume a large (potentially infinite) urn of similar apples, which are either red or green, and each apple is either sweet or bitter. The problem is to decide whether there is a significant dependency between the apple colour and taste, when we have only a finite sample of apples available. For example, let us suppose that we have observed in our sample that red apples are more likely to be sweet than green ones, and we want to test if this observed dependency is just due to chance.

Multinomial model

In the multinomial model, we assume that the real probabilities of sweet red apples, bitter red apples, sweet green apples, and bitter green apples are defined by parameters p_{XA}, $p_{X\neg A}$, $p_{\neg XA}$, and $p_{\neg X\neg A}$. The probability

of red apples is p_X and of green apples $1-p_X$. Similarly, the probability of sweet apples is p_A and of bitter apples $1-p_A$. According to independence assumption, $p_{XA} = p_X p_A$, $p_{X\neg A} = p_X(1-p_A)$, $p_{\neg XA} = (1-p_X)p_A$, and $p_{\neg X\neg A} = (1-p_X)(1-p_A)$. A sample of n apples is taken *randomly* from the urn. Now the probability of obtaining N_{XA} sweet red apples, $N_{X\neg A}$ bitter red apples, $N_{\neg XA}$ sweet green apples, and $N_{\neg X\neg A}$ bitter green apples is defined by multinomial probability

$$P(N_{XA}, N_{X\neg A}, N_{\neg XA}, N_{\neg X\neg A}|n, p_X, p_A) = \binom{n}{N_{XA}, N_{X\neg A}, N_{\neg XA}, N_{\neg X\neg A}} p_X^{N_X}(1-p_X)^{n-N_X} p_A^{N_A}(1-p_A)^{n-N_A}. \quad (2.2)$$

Since data size n is given, the contingency tables can be defined by triplets $< N_{XA}, N_{X\neg A}, N_{\neg XA} >$ or, equivalently, triplets $< N_X, N_A, N_{XA} >$. Therefore, the space of all possible contingency tables is

$$\mathcal{T}_{\mathcal{M}} = \{< N_X, N_A, N_{XA} > \mid N_X = 0, \cdots, n; N_A = 0, \cdots, n; N_{XA} = 0, \cdots, \min\{N_X, N_A\}\}.$$

For estimating the p-value with Equation (2.1), we should still solve two problems. First, the parameters p_X and p_A are unknown. The most common solution is to estimate them by the observed relative frequencies, i.e. $p_X = P(X)$ and $p_A = P(A)$. Second, we should decide, when a contingency table T_i is equally or more extreme than the observed contingency table T_0. For this purpose, we need some measure, which evaluates the overall dependence in a contingency table. An example of such measures is the *odds ratio*:

$$odds(N_{XA}, N_{X\neg A}, N_{\neg XA}, N_{\neg X\neg A}) = \frac{N_{XA}N_{\neg X\neg A}}{N_{X\neg A}N_{\neg XA}}.$$

The problem of odds ratio is that it is not defined, when $N_{X\neg A}N_{\neg XA} = 0$.

In practice, the multinomial test is seldom used, but the multinomial model is an important theoretical model, from which other models can be derived as special cases.

Double binomial model

In the double binomial model, it is assumed that we have two large urns, one for red and one for green apples. Let us call these the red and the green urn. In the red urn, the probability of sweet apples is $p_{A|X}$ and of bitter apples $1-p_{A|X}$, and in the green urn the probabilities are $p_{A|\neg X}$

and $1-p_{A|\neg X}$. According to the independence assumption, the probability of sweet apples is the same in both urns: $p_A = p_{A|X} = p_{A|\neg X}$. A sample of $m(X)$ apples is taken randomly from the red urn and another random sample of $m(\neg X)$ apples is taken from the green urn. The probability of obtaining N_{XA} sweet apples among the selected $m(X)$ red apples is defined by the binomial probability

$$P(N_{XA}|m(X), p_A) = \binom{m(X)}{N_{XA}} p_A^{N_{XA}}(1-p_A)^{m(X)-N_{XA}}.$$

Similarly, the probability of obtaining $N_{\neg XA}$ sweet apples among the selected green apples is

$$P(N_{\neg XA}|m(\neg X), p_A) = \binom{m(\neg X)}{N_{\neg XA}} p_A^{N_{\neg XA}}(1-p_A)^{m(\neg X)-N_{\neg XA}}.$$

Because the two samples are independent from each other, the probability of obtaining N_{XA} sweet apples from $m(X)$ red apples and $N_{\neg XA}$ sweet apples from $m(\neg X)$ green apples is the product of the two binomials:

$$P(N_{XA}, N_{\neg XA}|n, m(X), m(\neg X), p_A) = \binom{m(X)}{N_{XA}} \binom{m(\neg X)}{N_{\neg XA}} p_A^{N_A}(1-p_A)^{n-N_A}, \quad (2.3)$$

where $N_A = N_{XA} + N_{\neg XA}$ is the total number of the obtained sweet apples. We note that the double binomial probability is not *exchangeable* with respect to the roles of X and A, i.e. generally

$$P(N_{XA}, N_{\neg XA}|n, m(X), m(\neg X), p_A) \neq P(N_{XA}, N_{X\neg A}|n, m(A), m(\neg A), p_X).$$

In practice, this means that the probability of obaining $m(XA)$ sweet red apples, $m(X\neg A)$ bitter red apples, $m(\neg XA)$ sweet green apples, and $m(\neg X\neg A)$ bitter green apples is (nearly always) different in the model of the red and green urns from the model of the sweet and bitter urns.

Since $m(X)$ and $m(\neg X)$ are given, each contingency table is defined as a pair $< N_{XA}, N_{\neg XA} >$ or, equivalently, $< N_A, N_{XA} >$. The space of all possible contingency tables is

$$\mathcal{T}_{\mathcal{M}} = \{< N_{XA}, N_{\neg XA} > \mid N_{XA} = 0, \cdots, m(X); N_{\neg XA} = 0, \cdots, m(\neg X)\}.$$

We note that N_A is not fixed, and therefore N_A is generally not equal to the observed $m(A)$.

For estimating the significance with Equation (2.1), we should once again estimate the unknown parameter p_A and decide, how the extremeness relation is defined. The most common solution is to estimate p_A from the data, and set $p_A = P(A)$. For the extremeness relation, we can use for example the odds ratio. However, if we fix also $N_A = m(A)$, it becomes easy to define, when a dependency is stronger than the observed. The positive dependence between red and sweet apples becomes stronger, when N_{XA} increases, and the negative dependence between sweet and green apples becomes stronger, when $N_{\neg XA} = m(A) - N_{XA}$ decreases. Now the p-value gets a simple expression

$$p = \sum_{i=m(XA)}^{m(X)} \binom{m(X)}{i} \binom{m(\neg X)}{m(A)-i} P(A)^{m(A)} P(\neg A)^{m(\neg A)}.$$

Another common solution is approximate the p-value with asymptotic tests, which are discussed later.

Hypergeometric model

In the hypergeometric model, there is no sampling from an urn. Instead, we can assume that we are given a finite urn of n apples, containing exactly $m(X)$ red apples and $m(\neg X)$ green apples. We test all n apples and find that $m(XA)$ of red apples are sweet and $m(\neg XA)$ of green apples are sweet. The question is how probable is our urn (or the set of all at least equally extreme urns) among all possible apple urns with $m(X)$ red apples, $m(\neg X)$ green apples, $m(A)$ sweet apples, and $m(\neg A)$ bitter apples.

Now the urns correspond to contingency tables. The number of all possible urns with the fixed totals $m(X)$, $m(\neg X)$, $m(A)$, and $m(\neg A)$ is

$$\sum_{i=0}^{m(A)} \binom{m(X)}{i} \binom{m(\neg X)}{m(A)-i} = \binom{n}{m(A)}.$$

The equality follows from the Vandermonde's identity (Equation A.1). (We recall that $\binom{m}{l} = 0$, when $l > m$.)

Assuming that all urns with these fixed totals are equally likely, the probability of an urn with N_{XA} sweet red apples is

$$P(N_{XA}|m(X), M(\neg X), m(A), m(\neg A)) = \binom{n}{m(A)}^{-1}.$$

Because all totals are fixed, the extremeness relation is also easy to define. Positive dependence is stronger than observed, when $N_{XA} > m(XA)$.

For the p-value, it is enough to sum the probabilities of urns containing at least $m(XA)$ sweet red apples. The resulting p-value is

$$p_F = \sum_{i=0}^{J_1} \frac{\binom{m(X)}{m(XA)+i}\binom{m(\neg X)}{m(\neg X\neg A)+i}}{\binom{n}{m(A)}}, \tag{2.4}$$

where $J_1 = \min\{m(X\neg A), m(\neg XA)\}$. (Instead of J_1, we could give an upper range $m(A)$, because the zero terms disappear.) This p-value is known as *Fisher's p*, because it is used in *Fisher's exact test*, an exact permutation test, where p_F is calculated. We give it a special symbol p_F, because it will be used later. For negative dependence between red and sweet apples (or positive dependence between green and sweet apples) the p-value is

$$p_F = \sum_{i=0}^{J_2} \frac{\binom{m(X)}{m(XA)-i}\binom{m(\neg X)}{m(\neg X\neg A)-i}}{\binom{n}{m(A)}}, \tag{2.5}$$

where $J_2 = \min\{m(XA), m(\neg X\neg A)\}$.

Relations between the three models

The main difference between the three models is which of the variables N, N_X, and N_A are considered fixed. In the multinomial model all variables except $N = n$ are randomized. However, if the model is conditioned with $N_X = m(X)$, it leads to the double binomial model. If the double binomial model is conditioned with $N_A = m(A)$, it leads to the hypergeometric model. For completeness, we could also consider the *Poisson model*, where all variables, including N, are unfixed Poisson variables. If the Poisson model is conditioned with the given data size, $N = n$, it leads to the multinomial model.[44, ch. 4.6-4.7]

Selecting the correct model and defining the extremeness relation is a controversial problem, which statisticians (mostly Fisherian vs. Neyman-Pearsonian schools) have argued for the last century (see e.g. [84, 4, 43, 72, 36]). One problem is that even if one or both marginal totals, N_X and/or N_A, are kept unfixed, the observed counts $m(X)$ and/or $m(A)$ are still used to estimate the unknown parameters. Therefore, Fisher and his followers have suggested that we should always assume both N_X and N_A fixed and use Fisher's exact test or – when it is heavy to compute – a suitable asymptotic test. The opponents have reminded that the results are better generalizable outside the data set, if some variables are kept unfixed,

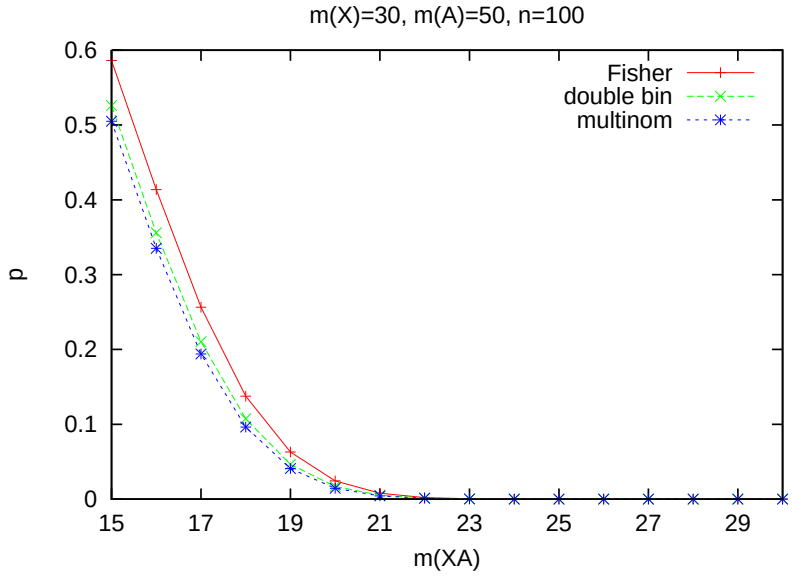


Figure 2.1: An example of p-values in different variable-based models, when $m(X)$ and $m(A)$ are fixed.

but, nevertheless, they have also suggested to use asymptotic tests, which are conditional on the observed counts.

Figure 2.1 shows an example of different p-values as functions $m(XA)$, when $m(X) = 30$, $m(A) = 50$, and $n = 100$. In all models, the extremeness relation has been defined as $D \geq \delta$, where D is a random variable for the leverage. In the beginning of the line, the leverage is $\delta = 0$, and in the end, it is maximal $\delta = 0.15$. Since the double binomial model is asymmetric with respect to X and A, the values are slightly different, when $m(X)$ and $m(A)$ are reversed. Generally, Fisher's exact test gives more cautious p-values than the multinomial and double binomial models. However, the values approach each other, when the data size is increased.

Asymptotic measures

We have seen that the p-values in the multinomial and double binomial models are quite difficult to calculate. However, the p-value can often be approximated easily using asymptotic measures. With certain assumptions, the resulting p-values converge to the correct p-values, when the data size n (or $m(X)$ and $m(\neg X)$) tend to infinity. In the following, we introduce

two commonly used asymptotic measures for independence testing: the χ^2-measure and mutual information. In statistics, the latter corresponds to *log likelihood ratio* [58].

The main idea of asymptotic tests is that instead of estimating the probability of the contingency table as such, we calculate some better behaving test statistic M. If M gets value val, we estimate the probability of $P(M \geq val)$ (assuming that large M values indicate a strong dependency).

In the case of the χ^2-test, the test statistic is the χ^2-measure:

$$\begin{aligned}\chi^2 &= \sum_{i=0}^{1}\sum_{j=0}^{1}\frac{n(P(X=i,A=j)-P(X=i)P(A=j))^2}{P(X=i)P(A=j)}\\ &= \frac{n(P(X,A)-P(X)P(A))^2}{P(X)P(\neg X)P(A)P(\neg A)}.\end{aligned} \tag{2.6}$$

So, in principle, each term tests how much the observed frequency $m(X=i,A=j)$ deviates from its expectation $nP(X=i)P(A=j)$, under the independence assumption. If the data size n is sufficiently large and none of the expected frequencies is too small, the χ^2-measure follows approximately the χ^2-distribution with one degree of freedom. This can be derived from the double binomial model as follows: N_{XA} and $N_{\neg XA}$ are binomial variables with expected values $\mu_1 = m(X)p_A$ and $\mu_2 = m(\neg X)p_A$ and standard deviations $\sigma_1 = \sqrt{m(X)p_A(1-p_A)}$ and $\sigma_2 = \sqrt{m(\neg X)p_A(1-p_A)}$. If the unknown parameter p_A is estimated by $P(A)$, we can calculate *z-scores* for N_{XA} and $N_{\neg XA}$:

$$z_1 = \frac{N_{XA} - m(X)P(A)}{\sqrt{m(X)P(A)P(\neg A)}} \quad \text{and}$$

$$z_2 = \frac{N_{\neg XA} - m(\neg X)P(A)}{\sqrt{m(\neg X)P(A)P(\neg A)}}.$$

The z-score measures how many standard deviations the observed frequency deviates from its expectation. If $m(X)$ and $m(\neg X)$ are sufficiently large, then both z_1 and z_2 follow the standard normal distribution $N(0,1)$. Therefore, their squares and also the sum of the squares $\chi^2 = z_1^2 + z_2^2$ follow the χ^2 distribution. As a classical rule of thumb [28], the χ^2-measure can be used only, if all expected frequencies $nP(X=x)P(A=a)$, $x \in \{0,1\}$, $a \in \{0,1\}$, are at least 5. However, the approximations can still be poor in some situations, when the underlying binomial distributions are skewed, e.g. if $P(A)$ is near 0 or 1, or if $m(X)$ and $m(\neg X)$ are far from each other [84, 4]. According to Carriere [20], this is quite typical for data in medical science.

One reason for the inaccuracy of the χ^2-measure is that the original binomial distributions are discrete while the χ^2-distribution is continuous. A common solution is to make a *continuity correction* and subtract 0.5 from the expected frequency $nP(X)P(A)$. According to Yates [84], the resulting continuity corrected χ^2-measure can give a good approximation to Fisher's p_F, if the underlying hypergeometric distribution is not markedly skewed. However, according to Haber [33], the resulting χ^2-value can underestimate the significance, while the uncorrected χ^2-value overestimates it.

For the χ^2-measure, we can give also alternative expressions, which are sometimes easier to analyze:

$$\chi^2 = \frac{n\delta^2}{P(X)P(\neg X)P(A)P(\neg A)} \tag{2.7}$$

$$= n(\gamma(X,A)-1)(\gamma(\neg X,\neg A)-1). \tag{2.8}$$

From these equations we see that the χ^2-measure is symmetric for positive and negative dependencies and it favours rules, where both $\gamma(X,A)$ and $\gamma(\neg X,\neg A)$ are large or – in the case of negative dependence – small.

Mutual information is another popular asymptotic measure, which has been used to test independence. It is defined as

$$\begin{aligned} MI = {} & m(XA)\log(P(XA)) + m(X\neg A)\log(P(X\neg A)) \\ & + m(\neg XA)\log(P(\neg XA)) + m(\neg X\neg A)\log(P(\neg X\neg A)) \\ & - m(X)\log(P(X)) - m(\neg X)\log(P(\neg X)) - m(A)\log(P(A)) \\ & - m(\neg A)\log(P(\neg A)). \end{aligned} \tag{2.9}$$

Mutual information is actually an information theoretic measure, but in statistics, $2MI$ is known as log likelihood ratio. The likelihood ratio can be derived from the multinomial model as follows: The likelihood of the observed counts $m(XA),\cdots,m(\neg X\neg A)$ is defined by

$$L(\overline{p}) = \binom{n}{m(XA), m(X\neg A), m(\neg XA), m(\neg X\neg A)} p_{XA}^{m(XA)} p_{X\neg A}^{m(X\neg A)} p_{\neg XA}^{m(\neg XA)} p_{\neg X\neg A}^{m(\neg X\neg A)},$$

where $\overline{p} =< p_{XA}, p_{X\neg A}, p_{\neg XA}, p_{\neg X\neg A} >$ contains the unknown parameters. The maximum likelihood estimates for the parameters are $\overline{p_1} =$ $< P(XA), P(X\neg A), P(\neg XA), P(\neg X\neg A) >$. Under the independence assumption, the parameters are $\overline{p} =< p_X p_A, p_X p_{\neg A}, p_{\neg X} p_A, p_{\neg X} p_{\neg A} >$, and the maximum likelihood estimate is

$$\overline{p_0} =< P(X)P(A), P(X)P(\neg A), P(\neg X)P(A), P(\neg X)P(\neg A) > .$$

Now the ratio of the likelihood of the data under the independence assumption and without it is

$$\lambda = \frac{L(\overline{p_0})}{L(\overline{p_1})} = \left(\frac{P(X)P(A)}{P(XA)}\right)^{m(XA)} \left(\frac{P(X)P(\neg A)}{P(X\neg A)}\right)^{m(X\neg A)} \left(\frac{P(\neg X)P(A)}{P(\neg XA)}\right)^{m(\neg XA)} \left(\frac{P(\neg X)P(\neg A)}{P(\neg X\neg A)}\right)^{m(\neg X\neg A)}.$$

The log likelihood ratio is $-2\log(\lambda) = 2MI$. It follows asymptotically the χ^2-distribution [79], and often it gives similar results to the χ^2-measure [73]. However, sometimes the two tests can give totally different results [4].

2.3.2 Value-based measures

In the value-based semantics the idea is that we would like to find events $XA = a$, which express a strong dependency, even if the general dependency between variables X and A is relatively weak. In this case, the strength of the dependency is usually measured by the lift, because the leverage has the same absolute value for all events XA, $X\neg A$, $\neg XA$, $\neg X\neg A$. However, the lift alone is an unsuitable measure, because it gets its maximal value, when $m(XA = a) = m(X) = m(A = a) = 1$ – i.e. when the rule occurs on just one row [35]. Such a rule is quite likely due to chance and hardly interesting. Therefore, we should either test the null hypothesis that the lift is $\gamma \leq 1$ (no positive dependence) [11], that the lift is less than some threshold $\gamma_0 > 1$ [41], or simply calculate the probability of observing such a large lift value, if X and A were actually independent (significance testing).

The p-value is defined like in the variable-based testing by Equation (2.1), but now the extremeness relation should not depend on the leverage but lift. A necessary condition for the extremeness of table T_i over T_j is that in T_i the lift is larger than in T_j. However, since the lift is largest, when N_X and/or N_A are smallest (and $N_{XA} = N_X$ or $N_{XA} = N_A$), it is sensible to require that also N_{XA} is larger in T_i than in T_j.

If both N_X and N_A are fixed, then the lift is larger than observed if and only if the leverage is larger than observed, and it is enough to consider tables, where $N_{XA} \geq m(XA)$. However, if either N_X, N_A, or both are unfixed, then we should always check the lift value, which is random variable

$$G = \frac{nN_{XA}}{N_X N_A},$$

and compare it to the observed lift γ.

Next, we will first survey how the value-based significance of dependence has been defined in the previous research. Then we will analyze the problem using the three classical models. Finally, we derive an alternative binomial test and the corresponding asymptotic measure for the value-based significance of dependence.

Value-based significance in the previous research

In the previous research on association rules, some authors [24, 41, 42, 19, 52] have speculated how to test the null hypothesis $P(A|X) \leq P(A)$, which is equivalent to testing $\gamma \leq 1$. For some reason, it has been taken for granted that the exact probability is defined by a binomial test *in the part of the data where X is true*. This is equivalent to adopting the double binomial model, but inspecting just one of the urns – the red apples. So, one tries to decide whether there is a dependency between red colour and sweetness by taking a sample of $m(X)$ red apples from the red urn. It is assumed that $N_{XA} \sim Bin(m(X), p_A)$, and the unknown parameter p_A is estimated from the data, as usual. For positive dependence, the p-value is defined as [24]

$$p = \sum_{i=m(XA)}^{m(X)} \binom{m(X)}{i} P(A)^i P(\neg A)^{m(X)-i} \tag{2.10}$$

and for negative dependence as [24]

$$p = \sum_{i=0}^{m(XA)} \binom{m(X)}{i} P(A)^i P(\neg A)^{m(X)-i}.$$

We see that $N_X = m(X)$ is the only variable, which has to be fixed – even N can be unfixed. In [42], it was explicitly noted that N_A is unfixed, and therefore, in the positive case, i goes from $m(XA)$ to $m(X)$, not to $\min\{m(X), m(A)\}$. The idea is that when $N_{XA} \geq m(XA)$, then $\frac{N_{XA}}{m(X)} \geq P(A|X)$, and if we already had $P(A|X) > P(A)$, then also $G \geq \gamma$. Similarly, in the negative case, $G \leq \gamma$. So, the test checks correctly all cases, where the lift is at least as large (as small) as observed.

Since the binomial p is quite difficult to calculate, it is common to estimate it asymptotically by the z-score. In the case of positive dependence, the binomial variable N_{XA} has expected value $\mu = m(X)P(A)$ and standard deviation $\sigma = \sqrt{m(X)P(A)P(\neg A)}$. The corresponding z-score is [42, 19]

$$z = \frac{m(XA) - \mu}{\sigma} = \frac{m(XA) - m(X)P(A)}{\sqrt{m(X)P(A)P(\neg A)}}$$
$$= \frac{\sqrt{m(X)}(P(A|X) - P(A))}{\sqrt{P(A)P(\neg A)}}. \tag{2.11}$$

If $m(X)$ is sufficiently large and $P(A)$ is not too near 1 or 0, the z-score follows the standard normal distribution. However, when the expected frequency $m(X)P(A)$ is low (as a rule of thumb < 5), the binomial distribution is positively skewed. This means that the z-score overestimates the significance.

The problem of this measure (and the original binomial model) is that two rules with different antecedents X are not comparable. So, all rules (with different X) are thought to be from different populations and are tested in different parts of the data. We note that this idea of checking just one urn is also underlying in the *J-measure* [70]:

$$J = m(XA) \log \frac{P(XA)}{P(X)P(A)} + m(X\neg A) \log \frac{P(X\neg A)}{P(X)P(\neg A)}, \tag{2.12}$$

which is often used to evaluate association rules. It is a reduced version of the mutual information, where all terms containing $\neg X$ have been dropped.

One solution to the comparison problem is to normalize the z-score, and divide it by the maximum value

$$\max\{z(X \to A)\} = \frac{\sqrt{m(X)}P(\neg A)}{\sqrt{P(A)P(\neg A)}}.$$

The result is

$$cfa = \frac{P(A|X) - P(A)}{P(\neg A)}, \tag{2.13}$$

which is the same as Shortliffe's *certainty factor* [67] for positive dependence. For the negative dependence, the certainty factor is

$$cfa_{neg} = \frac{P(A|X) - P(A)}{P(A)}.$$

According to [13], the certainty factor can produce quite accurate results when used for prediction purposes. However, it is very sensitive to redundant rules, because all rules with confidence 1.0 gain the maximum score, even if they occur on just one row. In addition, the certainty factor does not suit search purposes, because the upper bound is always the same.

Finally, we note that like the double binomial model, also this single binomial model is not exchangeable in the sense that generally $p(X \to A) \neq p(A \to X)$. The same holds for the corresponding z-score, J-measure, and certainty factor. This can be counter-intuitive, when the task is to search for statistical dependencies, and in the search these measures should be used with care. In addition, with this binomial model the significance of the positive dependence between X and A is generally not the same as the significance of the negative dependence between X and $\neg A$. With the corresponding z-score, the significance values are related, and

$$z_{pos}(X \to A) = -z_{neg}(X \to \neg A),$$

where z_{pos} denotes the z-score of positive dependence and z_{neg} the z-score of negative dependence. With the certainty factor and J-measure, the significance of the positive dependence between X and A and the significance of the negative dependence between X and $\neg A$ are equal.

Value-based significance under the classical models

Let us now analyze the value-based significance in the classical models. For simplicity, we consider only positive dependence. We assume that the extremeness relation is defined by comparing the lift variable G and the observed lift γ and frequency variable N_{XA} and the observed frequency $m(XA)$, i.e. $T_i \succeq T_0$, if $G_i \geq \gamma$ and $N_{XA,i} \geq m(XA)$, where G_i is the lift in table T_i and $N_{XA,i}$ is the frequency in T_i.

In the multinomial model, only the data size $N = n$ is fixed. Each contingency table, described by triplet $< N_X, N_A, N_{XA} >$, has a probability $P(N_{XA}, N_X - N_{XA}, N_A - N_{XA}, n - N_X - N_A + N_{XA}|n, p_X, p_A)$, defined by Equation (2.2). The p-value is achieved, when we sum over all possible triplets, where $G \geq \gamma$:

$$p = \sum_{N_X=0}^{n} \sum_{N_{XA}=m(XA)}^{N_X} \sum_{N_A=N_{XA}}^{Q_1} P(N_{XA}, N_X - N_{XA}, N_A - N_{XA}, n - N_X - N_A + N_{XA}|n, p_X, p_A),$$

where $Q_1 = \frac{nN_{XA}}{\gamma N_X}$. (We note that the terms are zero, if $N_X < N_{XA}$.)

In the double binomial model, $N_X = m(X)$ is also fixed. Each contingency table can be described by pair $< N_A, N_{XA} >$, and it has probability $P(N_{XA}, N_A - N_{XA}|n, m(X), m(\neg X), p_A)$ by Equation 2.3. Now we should

sum over all possible pairs, where $G \geq \gamma$:

$$p = \sum_{N_{XA}=m(XA)}^{n} \sum_{N_A=N_{XA}}^{Q_2} P(N_{XA}, N_A - N_{XA}|n, m(X), m(\neg X), p_A),$$

where $Q_2 = \frac{nN_{XA}}{\gamma m(X)}$.

In the hypergeometric model, also $N_A = m(A)$ is fixed. As noted before, the extremeness relation is now the same as in the variable-based case, and the p-value is defined by Equation (2.4). This is an important observation, because it means that *Fisher's exact test tests the significance also in the value-based semantics.* The same is not true for the first two models, where rule $X \rightarrow A$ can get a different p-value in variable-based and value-based semantics.

Alternative binomial model

When N_X and/or N_A are unfixed, the p-values are quite heavy to compute. Therefore, we will now derive a simple binomial model, where it is enough to sum over just one variable. The binomial probability can be further estimated by an equivalent z-score or the z-score can be used as an asymptotic test measure as such. Unlike in the traditional binomial model, we will test the rule in the whole data set, which means that the significance values of different rules are comparable.

Let us suppose that we have a large urn containing N apples. The probability of red apples is p_X, of sweet apples p_A, and sweet red apples p_{XA}. According to the independence assumption $p_{XA} = p_X p_A$. Therefore, the urn contains $Np_{XA} = Np_X p_A$ sweet red apples. A sample of n apples is taken randomly from the urn. The exact probability of obtaining N_{XA} sweet red apples is defined by the hypergeometric distribution:

$$P(N_{XA}|N, n, p_{XA}) = \frac{\binom{Np_{XA}}{N_{XA}}\binom{N-Np_{XA}}{n-N_{XA}}}{\binom{N}{n}}.$$

When N tends to infinity, the distribution approaches to binomial, and the probability of N_{XA} given n and p_{XA} becomes

$$P_b(N_{XA}|n, p_{XA}) = \binom{n}{N_{XA}} p_{XA}^{N_{XA}}(1 - p_{XA})^{n-N_{XA}}.$$

When the unknown parameter $p_{XA} = p_X p_A$ is estimated from the data, the probability becomes

$$p_b(N_{XA}|n, P(X)P(A)) = \binom{n}{N_{XA}} (P(X)P(A))^{N_{XA}}(1-P(X)P(A))^{n-N_{XA}}.$$

Since N_{XA} is the only variable which occurs in the probability, the extremeness relation is defined simply by $T_i \succeq T_0 \Leftrightarrow N_{XA} \geq m(XA)$. When the unknown parameter $p_X p_A$ is estimated from the data, the p-value of rule $X \rightarrow A$ is

$$p_{bin} = \sum_{i=m(XA)}^{n} \binom{n}{i} (P(X)P(A))^i (1 - P(X)P(A))^{n-i}. \tag{2.14}$$

This is in fact the classical binomial test in the whole data set, while the traditionally used binomial (Equation (2.10)) is a binomial test in the part of data, where X is true. Next we show that p_{bin} gives also an upper bound for the p-value in the multinomial model.

Lemma 2.5

$$p_{mul}(X \rightarrow A | n, p_X p_A) \leq \sum_{i=m(XA)}^{n} \binom{n}{i} (p_X p_A)^i (1 - p_X p_A)^{n-i}.$$

Proof In the multinomial model, the exact p-value of rule $X \rightarrow A$ is

$$p = \sum_{N_X=0}^{n} \sum_{N_{XA}=m(XA)}^{N_X} \sum_{N_A=N_{XA}}^{Q_1} \binom{n}{N_{XA}, N_X - N_{XA}, N_A - N_{XA}, n - N_X - N_A + N_{XA}} (p_X p_A)^{N_{XA}} ((1-p_X)p_A)^{N_X - N_{XA}} ((1-p_X)p_A)^{N_A - N_{XA}} ((1-p_X)(1-p_A))^{n - N_X - N_A + N_{XA}},$$

where $Q_1 = \frac{nN_{XA}}{\gamma N_X}$. Each term can be expressed in form

$$\binom{n}{N_{XA}} (p_X p_A)^{N_{XA}} B(N_{XA}, N_X, N_A),$$

where

$$B(N_{XA}, N_X, N_A) = \binom{n - N_{XA}}{N_x - N_{XA}, N_A - N_{XA}, n - N_X - N_A + N_{XA}} (p_X(1-p_A))^{N_X - N_{XA}} ((1-p_X)p_A)^{N_A - N_{XA}} ((1-p_X)(1-p_A))^{n - N_X - N_A + N_{XA}}.$$

For the exact p-value, we should sum $B(N_{XA}, N_X, N_A)$ over all values $N_X = N_{XA}, \ldots, n$ and $N_A = N_{XA}, \ldots, Q$. We get an upper bound for the

p-value, if we instead sum also N_A over the whole range (without caring about the $G \geq \gamma$ condition): $N_A = N_{XA}, \ldots, n$. Let $C(N_X, N_A|N_{XA})$ be the resulting coefficient:

$$C(N_X, N_A|N_{XA}) = \sum_{N_X=N_{XA}}^{n} \sum_{N_A=N_{XA}}^{n} B(N_{XA}, N_X, N_A).$$

By the multinomial theorem A.3, the coefficient is $C(N_X, N_A|N_{XA}) = (1 - p_X p_A)^{n-N_{XA}}$, which proves the upper bound. □

Since N_{XA} is a binomial variable with expected value $\mu = nP(X)P(A)$ and standard deviation $\sigma = \sqrt{nP(X)P(A)(1 - P(X)P(A))}$, the corresponding z-score is

$$\begin{aligned} z(X \to A) = \frac{m(X,A) - \mu}{\sigma} &= \frac{m(X,A) - nP(X)P(A)}{\sqrt{nP(X)P(A)(1 - P(X)P(A))}} \\ &= \frac{\sqrt{nP(XA)}(\gamma(X,A) - 1)}{\sqrt{\gamma(X,A) - P(X,A)}}. \end{aligned} \quad (2.15)$$

The same z-score was given also by Lallich [42] for the case, where N_X and N_A are unfixed. Because the discrete binomial distribution is approximated by the continuous normal distribution, the continuity correction can be useful, like with the χ^2-measure.

We note that this binomial model and the corresponding z-score are exchangeable, which is intuitively a desired property. However, the statistical significance of the positive dependence between X and A is generally not the same as the significance of the negative dependence between X and $\neg A$. For example, the z-score for the negative (or positive) dependence between X and $\neg A$ is

$$z(X \to \neg A) = \frac{m(X\neg A) - nP(X)P(\neg A)}{\sqrt{nP(X)P(\neg A)(1 - P(X)P(\neg A))}}.$$

Figure 2.2 shows an example of different p-values as functions $m(XA)$ (and thus lift), when $m(X) = 30$, $m(A) = 50$, and $n = 100$. In the beginning of the line, the lift is $\gamma = 1$, and in the end, it is maximal $\gamma = 2.0$. Since the double binomial and the traditional binomial ($bin2$, Equation (2.10)) models are asymmetric with respect to X and A, the values are slightly different, when $m(X)$ and $m(A)$ are reversed. Generally, the new binomial model, $bin1$, gives the most cautious p-values and the multinomial and double binomial models give the smallest p-values. However, the values approach each other, when the data size is increased.

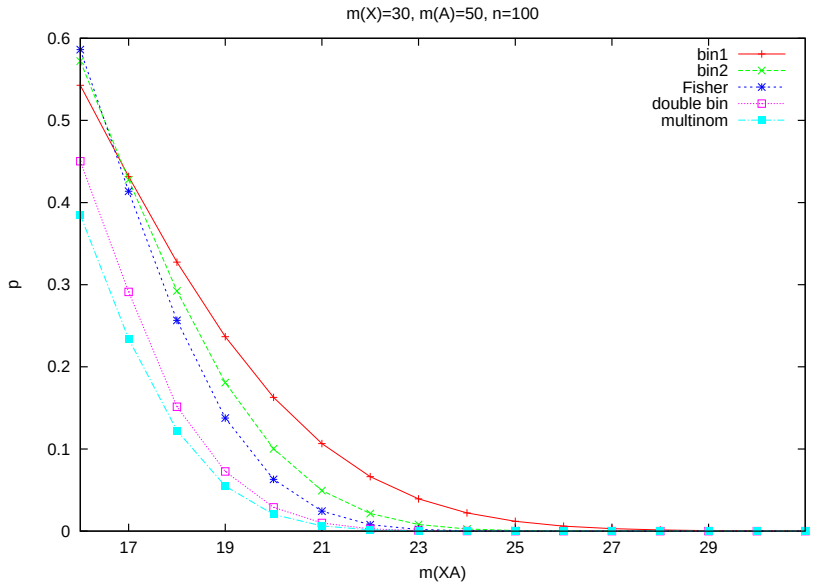


Figure 2.2: An example of p-values in different value-based models, when $m(X)$ and $m(A)$ are fixed.

In this Figure, the single binomial model ($bin2$) looks fine. However, there are pathological cases when it and the related z-score and the J-measure give opposite rankings than the other significance measures.

Example 2.6 Let us compare two rules, $X \to A$ and $Y \to A$, in the value-based semantics. The frequencies are $n = 100$, $m(A) = 50$, $m(X) = m(XA) = 30$, $m(Y) = 60$, and $m(YA) = 50$. I.e. $P(A|X) = 1$ and $P(Y|A) = 1$. The results are given in Table 2.2.

All of the traditional association rule measures (p_{bin2}, its z-score z_2, and J-measure) favour rule $X \to A$, while all the other measures (new binomial p_{bin1} and its z-score z_1, multinomial p_{mul}, double binomial p_{double}, and Fisher's p_F) rank rule $Y \to A$ better. In the classical models, the difference between the rules is quite remarkable.

2.4 Redundancy and improvement

In Section 1.2 we already discussed the problem of redundancy. Given the p-values of rules, we define the redundancy as follows. (The notion is generalized later in Definition 3.2.)

Table 2.2: Comparison of p-values and asymptotic measures for example rules $X \to A$ and $Y \to A$.

	$X \to A$	$Y \to A$
p_{bin1}	1.06e-4	2.21e-5
p_{mul}	8.86e-13	1.01e-19
p_F	1.60e-12	7.47e-19
p_{double}	2.05e-13	7.35e-20
z_1	4.20	4.36
p_{bin2}	9.31e-10	8.08e-8
z_2	5.48	5.16
J	36.12	14.56

Definition 2.7 (Redundancy (with respect to p)) Let $p(X \to A = a)$ be the p-value of rule $X \to A = a$. Rule $X \to A = a$ is redundant, if there exists rule $Y \to A = a$ such that $Y \subsetneq X$ and $p(Y \to A = a) \leq p(X \to A = a)$. Otherwise, rule $X \to A = a$ is non-redundant.

Redundancy can also be evaluated with the asymptotic measures, but it requires more care, because the asymptotic measures like the χ^2-measure and z-score tend to overestimate the significance, when the frequencies are small. Therefore, it is possible that a rule appears as non-redundant, when evaluated with an asymptotic measure, but redundant, when the exact p-values are calculated. As a solution, we suggest to use the asymptotic measures during the search to filter out rules which are definitely redundant, and check the redundancy of all remaining infrequent rules using the exact p-values. If all expected counts ($m(X)P(A)$, $m(X)P(\neg A)$, $m(\neg X)P(A)$, and $m(\neg X)P(\neg A)$ for rule $X \to A = a$) are sufficiently large, then the check for the exact p-values can be omitted.

The above definition of redundancy concerns only the observed redundancy in the given data set. It is based on the assumption that the observed probabilities P would be accurate estimates for the real, but unknown probabilities P_r. However, it is possible that a rule appears in the data as non-redundant, even if it is actually redundant (see Figure 1.1). We call such rules as *spuriously non-redundant*. To eliminate spuriously non-redundant rules, we should estimate the significance that rule $X \to A = a$ really improves a more general rule $Y \to A = a$, $Y \subsetneq X$.

Let us now denote the *improvement* of rule $X \to A = a$ over rule $Y \to A = a$ by $Imp(X \to A = a | Y \to A = a)$. In practice, the improvement can be defined by any goodness measure M. If large values of M indicate

a good rule, then

$$Imp(X \to A = a | Y \to A = a) = M(X \to A = a) - M(Y \to A = a),$$

and $Imp > 0$, if $X \to A = a$ improves $Y \to A = a$.

The task is to estimate the probability of $p = P(Imp(X \to A = a | Y \to A = a) > 0)$, assuming the null hypothesis that attributes $X \setminus Y$ do not improve rule $Y \to A = a$. If the resulting p-value is sufficiently small, we can assume that the observed improvement is not due to chance.

Unfortunately, there is no generally applicable theory for estimating the improvement of dependency rules in either variable- or value-based semantics. In the association rule research, a related problem concerning the significance of productivity has been addressed. We will now briefly review solutions to this problem and analyze why they do not suit for the general improvement testing.

Traditionally, the improvement has been defined by *productivity* (e.g. [38, 75]):

Definition 2.8 (Productivity) Rule $X \to A = a$ is productive, if $P(A = a|X) > P(A = a|Y)$ for all $Y \subsetneq X$.

So, here the goodness measure M is confidence, and it is required that a productive rule has a larger confidence than any of its generalizations. The significance of productivity is tested separately for all $Y \to A = a$, $Y \subsetneq X$, and all p-values should be below some fixed threshold max_p. In each test, the null hypothesis is that $P(A = a|X) = P(A = a|Y)$. If we denote the extra attributes in rule $X \to A = a$ by $Q = X \setminus Y$, the condition means that Q and A are independent in the set, where Y is true. The significance is estimated by calculating $p(Q \to A = a|Y)$, i.e. the p-value of rule $Q \to A = a$ in the set where Y holds. If the significance measure is Fisher's p_F, the significance of productivity of $YQ \to A = a$ over $Y \to A = a$ is [75]

$$p(Q \to A = a|Y) = \sum_{i=0}^{J} \frac{\binom{m(YQ)}{m(YQA=a)+i}\binom{m(Y\neg Q)}{m(Y\neg QA=a)-i}}{\binom{m(Y)}{m(YA=a)}}. \tag{2.16}$$

When the χ^2-measure is used to estimate the significance of productivity, the equation is [47]

$$\chi^2(Q \to A|Y) = \frac{m(Y)(P(Y)P(YQA) - P(YQ)P(YA))^2}{P(YQ)P(Y\neg Q)P(YA)P(Y\neg A)}. \tag{2.17}$$

Let us now analyze, why this approach is unsuitable for testing the improvement of dependency rules. First of all, the improvement of statistical dependency cannot be measured by the confidence or lift alone. It is true that the productivity is a necessary condition for the non-redundancy with all commonly used goodness measures (as we will see in the next chapter), but it is not a sufficient condition. A rule may have a maximal possible confidence, $cf = 1$, and still be just due to chance. In the variable-based semantics, we should also take into account the complementary rules $\neg X \to A \neq a$ and $\neg(XQ) \to A \neq a$. Typically, the confidence of the complementary rule decreases, when the confidence of rule $X \to A = a$ increases in the specializations.

The following example demonstrates that a productive rule can still be redundant, which means that the significance of the productivity does not guarantee any improvement. We calculate the significance with both Fisher's p_F and the χ^2-measure. Since p_F can be used in both variable- and value-based semantics, the example shows that the traditional productivity testing cannot identify spuriously non-redundant rules even in the value-based semantics.

Example 2.9 Let us reconsider the rules $X \to A$ ($=YQ \to A$) and $Y \to A$ in Example 2.6. Rule $X \to A$ is clearly productive with respect to $Y \to A$. Let us now calculate the significance of productivity using Fisher's p_F. Since $m(YQ) = m(YQA)$, there is just one term:

$$p(Q \to A|Y) = \frac{\binom{m(Y\neg Q)}{m(Y\neg Q\neg A)}}{\binom{m(Y)}{m(YA)}} = \frac{\binom{30}{10}}{\binom{60}{50}} = 0.00040.$$

The value is so small that we can assume that the productivity is significant. However, rule $X \to A$ is redundant ($p_F = 1.60 \cdot 10^{-12}$) with respect to $Y \to A$ ($p_F = 7.47 \cdot 10^{-19}$).

The same happens with the χ^2-measure. Now the significance of the productivity has $\chi^2(Q \to A|Y) = 12$, which is fairly good (about $p = 0.0005$). Still rule $Y \to A$ is more significant (with $\chi^2 = 66.7$) than $YQ \to A$ (with $\chi^2 = 42.9$).

A natural question arises, whether we could just replace the confidence with another goodness measure and use the same scheme for significance testing. However, it is not possible either, if we do want to keep the classical models of significance testing. The problem is that the above tests for the significance of productivity use the model, where just one urn from two are checked. Namely, the significance of $Q \to A = a$ is tested only in the Y

urn, and the $\neg Y$ urn is totally ignored. For example, let us suspect that there is a stronger dependency between big red apples and sweetness than between the red apples and sweetness. Now the traditional association rule researchers would test the dependency between the size and taste of apples in the red urn alone, and then decide, if rule *big red* $\rightarrow$ *sweet* is significantly better than rule *red* $\rightarrow$ *sweet*. In effect, this is equivalent to removing all small red apples to the green urn and testing, how much the sweetness in the red urn was improved. What one ignored is how much the bitterness in the green urn was degraded. It is clear that this model does not suit the variable-based semantics. However, it is unclear, whether it can be used in the value-based semantics.

Since we do not have any solution to this problem, we will concentrate on searching the most significant non-redundant rules. However, our intuition is that the condition for the significance of the improvement can be expressed by the *redundancy coefficient*:

$$\rho = \frac{p(XQ \rightarrow A = a)}{p(X \rightarrow A = a)},$$

where $\rho > 1$ indicates a non-redundant rule, and larger ρ values could be used to prune out spuriously non-redundant rules. If this is the fact, then the forthcoming search algorithms suit for searching genuinely non-redundant rules, as well.

Chapter 3

Pruning insignificant and redundant rules

> *We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances. Therefore, to the same natural effects we must, so far as possible, assign the same causes.*
>
> I. Newton

In our problem, the search space for dependency rules consists of all possible rules of form $X \to A_i$ or $X \to \neg A_i$, where $X \subsetneq R$ can be any attribute set and $A_i \in R \backslash X$ can be any single attribute. In the enumeration problem, the task is to find all sufficiently good, non-redundant rules, while in the optimization problem, the task is to find the K best, non-redundant rules. Both the goodness and redundancy of rules is defined with respect to some general goodness measure M. The main problem is how to prune the exponential search space effectively without losing any significant rules.

In this chapter, we introduce the theoretical basis for the search, without going to algorithmic details, yet. First, we define the general goodness measure and the related concept of redundancy and formalize the search problem. Then we introduce a class of *well-behaving* goodness measures, which covers all commonly used measures for statistical dependencies. We prove upper (or lower) bounds, which can be used to estimate the best possible M-value for an arbitrary rule $X \to A = a$, when only some information on X and A is known. We show that the upper (or lower) bounds for all well-behaving measures as well as the lower bounds for Fisher's p_F are met in the same points of the search space. This important discovery means that the same search strategy can be applied to all these measures

to prune out useless areas of the search space. Finally, we show how the concept of redundancy can be utilized in extra pruning.

3.1 Basic concepts

Let us first define a general goodness measure for dependency rules. For simplicity, we consider rules of form $X \to A = a$, where $a \in \{0, 1\}$, i.e. rules $X \to A$ and $X \to \neg A$.

Definition 3.1 (Goodness measure) Let R be a set of binary attributes and $\mathcal{U} = \{X \to A = a \mid X \subsetneq R, A \in R \setminus X, a \in \{0, 1\}\}$ the set of all possible rules, which can be constructed from attributes R.

Let $f(N_X, N_{XA}, N_A, N) : \mathbb{N}^4 \to \mathbb{R}$ be some statistical measure function, which measures the significance of positive dependency between X and $A = a$, given absolute frequencies $N_X = m(X)$, $N_{XA} = m(XA = a)$, $N_A = m(A = a)$, and data size $N = n$.

Function $M : \mathcal{U} \to \mathbb{R}$ is a goodness measure for dependency rules, if $M(X \to A = a) = f(m(X), m(XA = a), m(A = a), n)$.

Measure M is called increasing (by goodness), if large values of $M(X \to A = a)$ indicate that $X \to A = a$ is a good rule, and, respectively, decreasing, if low values indicate a good rule.

In the above definition, we have defined the statistical measure function on parameters N_X, N_{XA}, N_A, and N. First, we note that in practice, some of these parameters can be considered constant. For example, if the data size $N = n$ is given, it can be omitted. If the consequence $A = a$ is also fixed, we can use a simpler function $f_{A=a}(N_X, N_{XA})$.

Second, we note that even if we have defined the function f in whole $\mathbb{N}^4$, only some parameter value combinations can occur in any real data set. For any real frequencies hold $0 \leq n$, $0 \leq m(X) \leq n$, $0 \leq m(A = a) \leq n$, and $0 \leq m(XA = a) \leq \min\{m(X), m(A = a)\}$. In addition, for any non-trivial rule must hold $0 < n$, $0 < m(X) < n$, and $0 < m(A = a) < n$. If $n = 0$, the data set would not exist. If $m(A = a)$ or $m(X)$ were 0, the corresponding rule $X \to A = a$ would not occur in the data at all. On the other hand, if $m(X) = n$ or $m(A = a) = n$, then either X or $A = a$ would occur on all rows of data, and the rule could express only independence. Therefore, it suffices that the function is defined in the set of all legal parameter values.

Third, we note that the actual function can be defined on other parameters, if they can be derived from N_X, N_{XA}, N_A, and N. Examples of commonly occurring derived parameters are leverage δ and confidence cf.

For example, when the data size n and the consequence A are fixed, the χ^2-measure can be defined e.g. by the following two functions:

$$f_1(N_X, N_{XA}) = \frac{n(N_{XA} - N_X P(A))^2}{N_X(n - N_X)P(A)(1 - P(A))}$$

$$f_2(N_X, \delta) = \frac{n^3\delta^2}{N_X(n - N_X)P(A)(1 - P(A))}.$$

Functions f_1 and f_2 can be transformed to each other by equalities

$$f_1(N_X, N_{XA}) = f_2\left(N_X, \frac{N_{XA} - N_X P(A)}{n}\right)$$

$$f_2(N_X, \delta) = f_1(N_X, N_X P(A) + n\delta).$$

These transformations are often useful, when the behaviour of the function is analyzed.

The redundancy with respect to an arbitrary goodness measure M is defined as follows:

Definition 3.2 (Redundancy (with respect to M)) Let M be an increasing (decreasing) goodness measure. Rule $X \to A = a$ is redundant, if there exists rule $Y \to A = a$ such that $Y \subsetneq X$ and $M(Y \to A = a) \geq M(X \to A = a)$ ($M(Y \to A = a) \leq M(X \to A = a)$). Otherwise, rule $X \to A = a$ is non-redundant.

Now we can formalize the search problem, where either all sufficiently good (enumeration problem) or only the K best (optimization problem) non-redundant dependency rules are searched for.

Definition 3.3 (Search problem) Let M be an increasing (a decreasing) goodness measure and min_M (max_M) a user-defined threshold. In the optimization problem, $min_M = \min\{M(\cdot)\}$ ($max_M = \max\{M(\cdot)\}$), unless another initial value is given. In addition, the maximal number of the best rules to be searched, $K > 0$, is given in the optimization problem.

Let $\mathcal{U} = \{X \to A = a \mid X \subsetneq R, A \in R \setminus X, a \in \{0, 1\}\}$ be the set of all possible rules. The problem is to find set $\mathcal{S} \subseteq \mathcal{U}$ such that for all rules $X \to A = a \in \mathcal{S}$ holds

1. $M(X \to A = a) \geq min_M$ (or $\leq max_M$ for a decreasing M);
2. if we have an optimization problem, $M(X \to A = a) \geq M_K$ (or $\leq M_K$), where M_K is the M-value of the Kth best rule in $\mathcal{S}$; and

3. there does not exist any rule $Y \to A = a \in \mathcal{U}$ such that $Y \subsetneq X$ and $M(Y \to A = a) \geq M(X \to A = a)$ (or $M(Y \to A = a) \leq M(X \to A = a)$).

In addition, it is required in the optimization problem that $|\mathcal{S}| \leq K$.

We note that in the optimization problem, the discovered set of rules can contain less than K rules, if the data set did not contain enough non-redundant positive dependencies or the dependencies were not sufficiently good.

In practice, the searched set $\mathcal{S}$ covers only a small fraction of the whole search space $\mathcal{U}$. The problem is to find $\mathcal{S}$ from $\mathcal{U}$ such that as large area of $\mathcal{U} \setminus \mathcal{S}$ as possible can be left unchecked. For this purpose, we need effective pruning strategies. The following pruning strategies are based on the assumption that the search proceeds from more general rules to more specific rules. This is a reasonable assumption, because then the redundancy or non-redundancy of discovered rules is always known.

The basic search strategy is the following: The search space is traversed by generating sets of attributes $Z \subseteq R$ in an cumulative order, i.e. all subsets $Y \subsetneq Z$ are checked before Z. For each set Z, all possible consequences $A = a$, $A \in R$, $a \in \{0, 1\}$, are checked. If neither rule $Z \setminus \{A\} \to A = a$ nor any of its specializations $(ZQ) \setminus \{A\} \to A = a$, $Q \subseteq R \setminus Z$, belongs to the set of non-redundant, significant rules $\mathcal{S}$, then consequence $A = a$ can be pruned as impossible in the subspace defined by sets ZQ. If no consequence $A = a$, $A \in R$, $a \in \{0, 1\}$ can produce a non-redundant, significant rule with an antecedent $(ZQ) \setminus \{A\}$ (i.e. all consequences are impossible), then all sets ZQ can be pruned without any further checking. So, formally, the problem is to determine, when the value of relation

$$\mathit{Possible}(Z, A, a) = \begin{cases} 0 & \text{if } (ZQ) \setminus \{A\} \to A = a \notin \mathcal{S}, \\ 1 & \text{otherwise,} \end{cases}$$

is definitely zero.

For this purpose, we have to estimate an upper bound (or a lower bound, if M is decreasing) for the M-value of the best possible rule $(ZQ) \setminus \{A\} \to A = a$, $Q \subseteq R \setminus Z$. If the upper bound is too low compared to min_M, M_K, or $M(Y \to A = a)$, $Y \subsetneq Z \setminus \{A\}$, then we know that $\mathit{Possible}(Z, A, a) = 0$.

The available information for determining the upper bound $\mathrm{UB}(M((ZQ) \setminus \{A\} \to A = a))$ depends on the set Z and whether $A \in Z$. In practice, we need upper/lower bounds in three different cases: 1) when $Z = \{A\}$, 2) when $A \notin Z$, and 3) when $Z = X \cup \{A\}$ for some $X \subseteq R \setminus \{A\}$. The resulting upper bounds are denoted by *ub1*, *ub2*, and *ub3* and the lower bounds by *lb1*, *lb2*, *lb3*.

1. In the beginning of the search, only the absolute frequencies $m(A)$ and $m(\neg A)$ are known for all attributes $A \in R$. Therefore, we should estimate $ub1 = UB(M(X \to A = a))$ (or the corresponding lower bound *lb1*) for any possible $X \subseteq R \setminus \{A\}$ and $a \in \{0, 1\}$.

2. When $m(X)$, $A \notin X$, and $m(A = a)$ are known, we should estimate $ub2 = UB(M(XQ \to A = a))$ (or *lb2*) for any possible $Q \subseteq R \setminus (X \cup \{A\})$. As a special case, this gives also an upper/lower bound for $M(X \to A = a)$, because Q can be an empty set.

3. When $m(X)$, $m(XA = a)$, and $m(A = a)$ are known, we should estimate $ub3 = UB(M(XQ \to A = a))$ (or *lb3*) for any possible $Q \subseteq R \setminus (X \cup \{A\})$.

These upper/lower bounds can already be used for pruning out some of the forthcoming redundant rules. However, it is still possible that some forthcoming rule $X \to A = a$ is not redundant with respect to any known rules $Y \to A = a$, $Y \subsetneq X$, but it will be redundant with respect to another forthcoming rule $Z \to A = a$, $Z \subsetneq X$. Even in this case, it sometimes possible to infer the redundancy, before neither Z nor X is checked.

In the following, we will first derive upper and lower bounds for all three cases, which occur in the search. Then we will explain extra principles for pruning out redundant rules.

3.2 Upper and lower bounds for well-behaving measures

In this section, we prove general upper/lower bounds, which hold for any *well-behaving* goodness measure M. The notion of well-behaving measures is defined by the classical axioms for any "proper" goodness measures [63, 49]. In practice, the axioms hold for a large class of popular goodness measures used for evaluating the goodness of statistical dependencies, classification rules, or association rules.

3.2.1 Well-behaving measures

We recall that for any rule $X \to A$, the measure value $M(X \to A)$ can be determined as a function of four variables: absolute frequencies $N_X = m(X)$, $N_{XA} = m(XA)$, $N_A = m(A)$, and the data size $N = n$. Let us now assume that the measure M is increasing by goodness, meaning that high values of $M(X \to A)$ indicate that $X \to A$ is a good rule. According

to the classical axioms by Piatetsy-Shapiro [63] the following properties should hold for any proper measure M, measuring the goodness of a positive dependency between X and A:

Axiom (i) M is minimal, when $NN_{XA} = N_X N_A$,

Axiom (ii) M is monotonically increasing with N_{XA}, when N_X, N_A, and N remain unchanged, and

Axiom (iii) M is monotonically decreasing with N_X (or N_A), when N_{XA}, N_A (or N_X), and N remain unchanged.

The first axiom simply states that $M(X \to A)$ gets its minimum value, when X and A are independent. In addition, it is (implicitly) assumed that M gets the minimum value also for negative dependencies, i.e. when $NN_{XA} < N_X N_A$. The second axiom states that M increases, when the dependency becomes stronger (leverage δ increases) and the rule becomes more frequent. The third axiom states that M decreases, when the dependency becomes weaker (δ decreases).

Piatetsky-Shapiro [63] noticed that under these conditions M gets its maximal value for any fixed $m(X)$, when $m(XA) = m(X)$. In addition, he assumed that M would get its global maximum, when $m(XA) = m(X) = m(A)$. However, the latter does not necessarily hold, because the axioms do not tell how to compare rules $X \to A$ and $XQ \to A$, where $P(A|X) = P(A|XQ) = 1$. In this case, the more general rule, $X \to A$, has at least as large N_{XA} and N_X as $XQ \to A$ has. Major and Mangano [49] suggested a fourth axiom, which solves this problem:

Axiom (iv) M is monotonically increasing with N_X, when $cf = \frac{N_{XA}}{N_X}$ is fixed, N_A and N are fixed, and $cf > \frac{N_A}{N}$.

According to this axiom, a more general rule is better, when two rules have the same (or equally frequent) consequence and the confidence is the same. In addition, it was required that the dependency should be positive. However, based on our derivations, we assume that the same property holds also for negative dependencies, and M is non-increasing only when there is independence. In Appendix B.3, we show this for the χ^2-measure and mutual information.

In the following, we extend the axioms for negative dependencies and prove general upper bounds which hold for any measure M following these axioms. The upper bounds are the same as derived by Morishita and Sese

[57] in the case of the χ^2-measure. In their work, the upper bounds were derived by showing that the χ^2-measure is a convex function of N_X and N_{XA} for a fixed consequence A. Similar results could be achieved for other convex measures, but checking the axioms is simpler than the convexity proofs. In addition, there are non-convex goodness measures, which still follow the axioms (an example of a non-convex and non-concave well-behaving measure is the z-score, Equation (2.15), as shown in Appendix B.4).

For simplicity, assume that M is increasing by goodness; for a decreasing measure, the properties are reversed (minimum vs. maximum, increasing vs. decreasing).

Definition 3.4 (Well-behaving goodness measure) Let M be an increasing goodness measure defined by function $f(N_X, N_{XA}, N_A, N)$. Let $f_2(N_X, \delta, N_A, N)$ be another function which defines the same measure: $f_2(N_X, \delta, N_A, N) = f(N_X, \frac{N_X N_A}{N} + \delta N, N_A, N)$ and $f(N_X, N_{XA}, N_A, N) = f_2(N_X, \frac{N N_{XA} - N_X N_A}{N^2}, N_A, N)$.

Let $S \subseteq \mathbb{N}^4$ be a set of all legal parameter values (N_X, N_{XA}, N_A, N) for an arbitrary data set.

Measure M is called well-behaving, if it has the following properties in set S:

(i) f_2 gets its minimum value, when $\delta = 0$.

(ii) If N_X, N_A, and N are fixed, then

  (a) f_2 is a monotonically increasing function of δ, when $\delta > 0$ (positive dependence), and

  (b) f_2 is a monotonically decreasing function of δ, when $\delta < 0$ (negative dependence).

(iii) If $N_{XA} = m(XA = a)$, $N_A = m(A = a)$, and $N = n$ are fixed, then

  (a) f is a monotonically decreasing function of N_X, when $N_X < \frac{n \cdot m(XA=a)}{m(A=a)}$ (positive dependence), and

  (b) f is a monotonically increasing function of N_X, when $N_X > \frac{n \cdot m(XA=a)}{m(A=a)}$ (negative dependence).

(iv) If $N_A = m(A = a)$ and $N = n$ are fixed, then for all $cf_1, cf_2 \in [0, 1]$

  (a) $f(N_X, cf_1 N_X, m(A = a), n)$ is monotonically increasing with N_X, when $cf_1 > \frac{m(A=a)}{n}$ (positive dependence), and

(b) $f(N_X, m(A=a) - cf_2(n - N_X), m(A=a), n)$ is monotonically decreasing with N_X, when $cf_2 > \frac{m(A=a)}{n}$ (negative dependence).

The first two conditions are obviously equivalent to the classical axioms (i) and (ii). The only difference is that the behaviour is expressed in terms of leverage δ. This enables that the measure can be defined for negative dependencies, as well. In addition, it is often easier to check that the conditions hold for a desired function, when it is expressed as a function of δ. In practice, this can be done by differentiating $f_2(N_X, \delta, N_A, N)$ with respect to δ, where N_X, N_A, and N are considered constants. If M is an increasing function, then the derivative $f_2' = f_2'(N_X, \delta, N_A, N)$ should be $f_2' = 0$, when $\delta = 0$, $f_2' > 0$, when $\delta > 0$, and $f_2' < 0$, when $\delta < 0$. For decreasing M, the signs of the derivative are reversed. We note that it is enough that the function is defined and differentiable in the set of legal values (as noted in Section 3.1).

The third condition is also equivalent to the classical axiom (iii), when extended to both positive and negative dependencies. Before point $N_X = \frac{n \cdot m(XA=a)}{m(A=a)}$, M measures positive dependence, and after it, negative dependence. The third condition can be checked by differentiating $f(N_X, m(XA = a), m(A = a), n)$ (or – to be exact – the corresponding continuous and differentiable extension of f) with respect to N_X, where $m(XA = a)$, $m(A = a)$, and n are constants. For an increasing M, the derivative $f' = f'(N_X, m(XA = a), m(A = a), n)$ should be $f' = 0$, when $M_X = \frac{n \cdot m(XA=a)}{m(A=a)}$ (independence), $f' < 0$, when $N_X < \frac{n \cdot m(XA=a)}{m(A=a)}$ (positive dependence), and $f' > 0$, when $N_X > \frac{n \cdot m(XA=a)}{m(A=a)}$ (negative dependence). For decreasing M, the signs are reversed.

Similarly, the fourth condition is equivalent to the classical axiom (iv), when extended to both positive and negative dependencies. In the case of positive dependence, cf_1 corresponds to confidence $P(A = a|X)$, which is kept fixed. In the case of negative dependence, cf_2 corresponds to confidence $P(A = a|\neg X)$, which is kept fixed. (Now the equation is $N_{XA} = m(A = a) - cf_2(n - N_X) \Leftrightarrow m(A = a) - N_{XA} = cf_2(n - N_X) \Leftrightarrow N_{\neg XA} = cf_2 N_{\neg X}$.) We require that condition (a) holds for positive dependence ($P(A = a|X) > P(A)$) and (b) for negative dependence ($P(A = a|\neg X) > P(A = a) \Leftrightarrow P(A = a|X) < P(A = a)$). However, according to our analysis of the χ^2-measure and mutual information (Appendix B.3), both (a) and (b) hold everywhere, where $P(A = a|X) \neq P(A = a)$ and $P(A = a|\neg X) \neq P(A = a)$. It is still unproved, whether this holds for any well-behaving measure, but for the upper bound proofs the above conditions are sufficient.

In practice, the fourth condition can be checked by differentiating functions $f(N_X, cf_1 N_X, m(A=a), n)$ and $f(N_X, m(A=a) - cf_2(n - N_X), m(A=a), n)$ (the corresponding continuous and differentiable extensions) with respect to N_X. For increasing M, the derivative $f'(N_X, cf_1 N_X, m(A=a), n)$ should be $f' > 0$, when $cf_1 > \frac{m(A=a)}{n}$, and the derivative $f'(n - N_X, cf_2(n - N_X), m(A=a), n)$ should be $f' < 0$, when $cf_2 > \frac{m(A=a)}{n}$. For decreasing M, the signs are reversed.

As an example, we show that the χ^2-measure (Equation 2.6), mutual information (Equation (2.9)), two versions of the z-score (Equation (2.15) and Equation 2.11), and the J-measure (Equation (2.12)) are well-behaving. The last three measures are defined only for positive dependencies.

Theorem 3.5 *Let $S \subseteq \mathbb{N}^4$ be defined by constraints $0 < N$, $0 < N_X < N$, $0 < N_A < N$, and $0 \leq N_{XA} \leq \min\{N_X, N_A\}$. Measure M is well-behaving, if it is defined by function*

(a) $\chi^2(N_X, N_{XA}, N_A, N) = \frac{N(NN_{XA} - N_X N_A)^2}{N_X(N-N_X)N_A(N-N_A)}$;

(b) $MI(N_X, N_{XA}, N_A, N) = N_{XA} \log \frac{N \cdot N_{XA}}{N_X N_A}$
$+(N_X - N_{XA}) \log \frac{N \cdot (N_X - N_{XA})}{N_X(N-N_A)} + (N_A - N_{XA}) \log \frac{N \cdot (N_A - N_{XA})}{(N-N_X)N_A}$
$+(N - N_X - N_A + N_{XA}) \log \frac{N \cdot (N - N_X - N_A + N_{XA})}{(N-N_X)(N-N_A)}$;

(c) $z_1(N_X, N_{XA}, N_A, N) = \frac{\sqrt{N}(NN_{XA} - N_X N_A)}{\sqrt{N_X N_A (N^2 - N_X N_A)}}$, *when* $NN_{XA} > N_X N_A$, *and 0, otherwise;*

(d) $z_2(N_X, N_{XA}, N_A, N) = \frac{NN_{XA} - N_X N_A}{\sqrt{N_X N_A (N - N_A)}}$, *when* $NN_{XA} > N_X N_A$, *and 0, otherwise; and*

(e) $J(N_X, N_{XA}, N_A, N) = N_{XA} \log(\frac{N_{XA}}{N_A}) + (N_X - N_{XA}) \log(\frac{N_X - N_{XA}}{N - N_A})$
$-N_X \log(\frac{N_X}{N})$, *when* $NN_{XA} > N_X N_A$, *and 0, otherwise.*

Proof Appendix B.3. □

An example of a non-behaving measure is Shortliffe's certainty factor (Equation (2.13)), which does not satisfy the fourth condition.

3.2.2 Possible frequency values

Before we go to the theoretical results, we introduce a graphical representation, which simplifies the proofs.

Let us now consider the set of all legal frequency values, when the data size n and consequence $A = a$ (corresponding frequency $m(A = a)$) are fixed. All legal values of N_X and N_{XA} can be represented in a two-dimensional space spanned by variables N_X and N_{XA}. Figure 3.1 shows a graphical representation of the space.

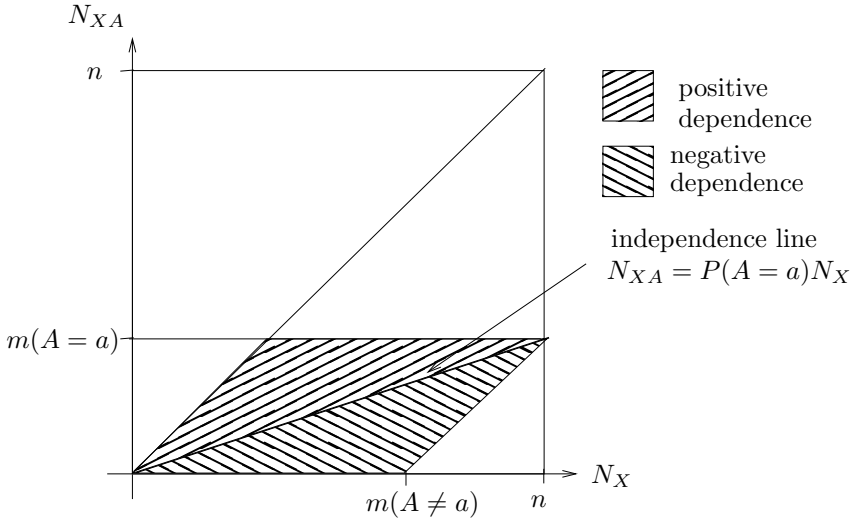


Figure 3.1: Two-dimensional space of absolute frequencies N_X and N_{XA}, when $A = a$ is fixed. In a given data set of size n, all points $(m(X), m(XA = a))$ lie in shaded areas.

In any data set of size $N = n$, all possible frequency combinations $(m(X), m(XA = a))$, where $m(XA = a) \leq m(X)$, must lie in the triangle $\{(0,0),(n,n),(n,0)\}$. If also the consequence $A = a$ is fixed, with absolute frequency $m(A = a)$, the area of possible combinations is restricted to the shaded areas in Figure 3.1. Boundary line $[(0, m(A = a)), (n, m(A = a))]$ follows from the fact that $m(XA = a) \leq m(A = a)$ and line $[(m(A \neq a), 0), (n, m(A))]$ from the fact that $m(A \neq a) \geq m(XA \neq a) \Leftrightarrow m(XA = a) \geq m(X) - m(A \neq a)$. Line $[(0,0),(n, m(A = a))]$ is called the *independence line*, because on that line $N_{XA} = P(A = a)N_X$, i.e. $m(XA = a) = P(A = a)m(X)$, and the corresponding X and $A = a$ are statistically independent. If the point lies above the independence line, the dependency is positive $(m(XA = a) > P(A = a)m(X))$, and below the line, it is negative $(m(XA = a) < P(A = a)m(X))$.

Figure 3.2 shows a point $(m(X), m(XA = a))$ corresponding to rule $X \rightarrow A = a$. In this case, the dependency is positive, because $(m(X), m(XA$

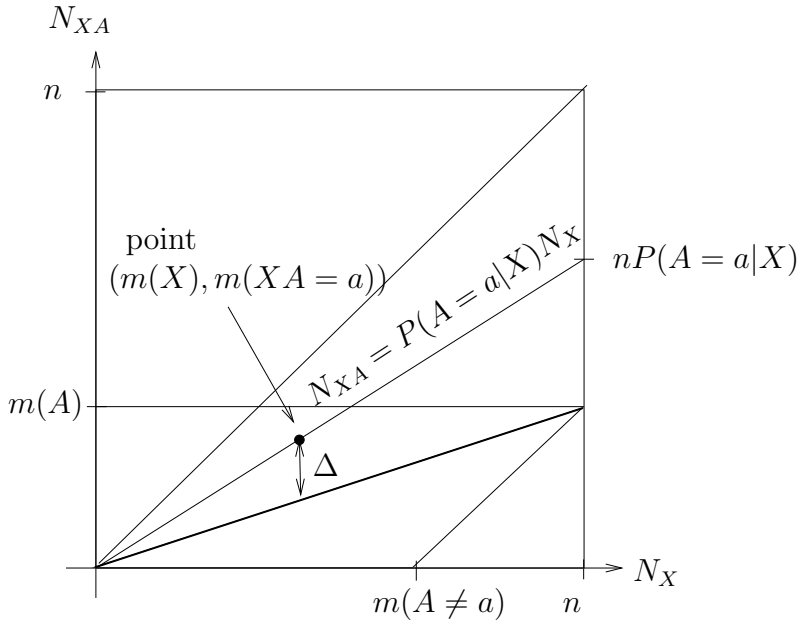


Figure 3.2: Point $(m(X), m(XA = a))$ corresponding to rule $X \rightarrow A = a$. The vertical difference Δ from the independence line measures the absolute leverage.

$= a))$ lies above the independence line. The slope of line $[(0, 0), (n, nP(A = a|X))]$ is the rule confidence $P(A = a|X) = \frac{m(XA=a)}{m(X)}$. The vertical difference between point $(m(X), m(XA = a))$ and the independence line, marked by Δ, defines the absolute leverage $\Delta = n\delta$. If the dependency is negative and the point lies below the independence line, the leverage is negative.

Figure 3.3 shows how the knowledge on a rule $X \rightarrow A = a$ can be utilized to determine the possible frequency values of more specific positive dependency rules $XQ \rightarrow A = a$. Since $m(XQA = a) \leq m(XA = a)$, all points $(m(XQ), m(XQA = a))$ must lie under the line $[(0, m(XA = a)), (n, m(XA = a))]$. Because the dependencies are positive, they also have to lie above the independence line. In the next section, we will show that for a well-behaving goodness measure M, point $(m(XA = a), m(XA = a))$ defines an upper bound (or lower bound) for the M-value of any positive dependency rule $XQ \rightarrow A = a$.

Figure 3.4 shows the area, where all possible points for negative dependency rules $XQ \rightarrow A = a$ must lie. Once again, $m(XQA = a) \leq m(XA = a)$, and because the dependence is negative, the points must lie

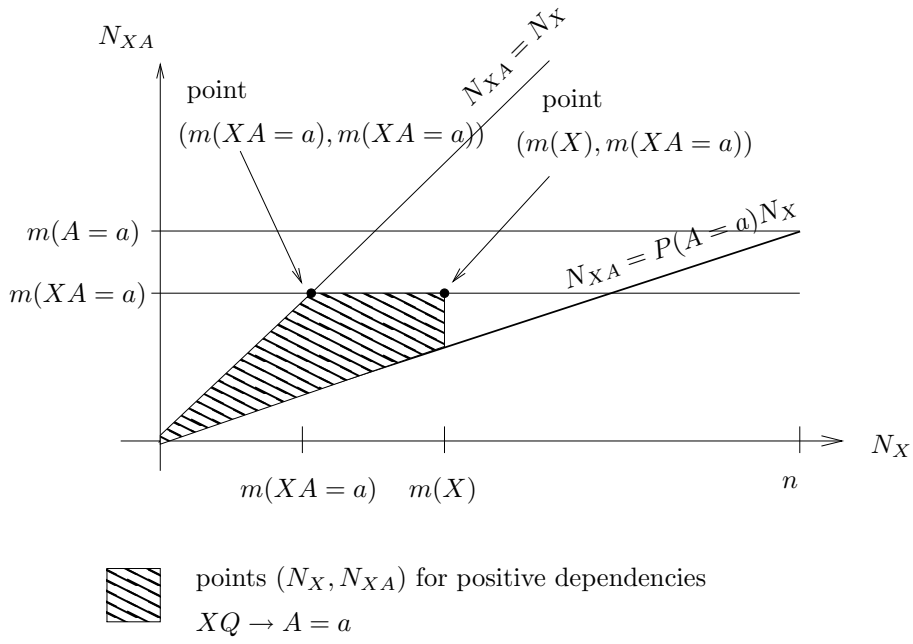
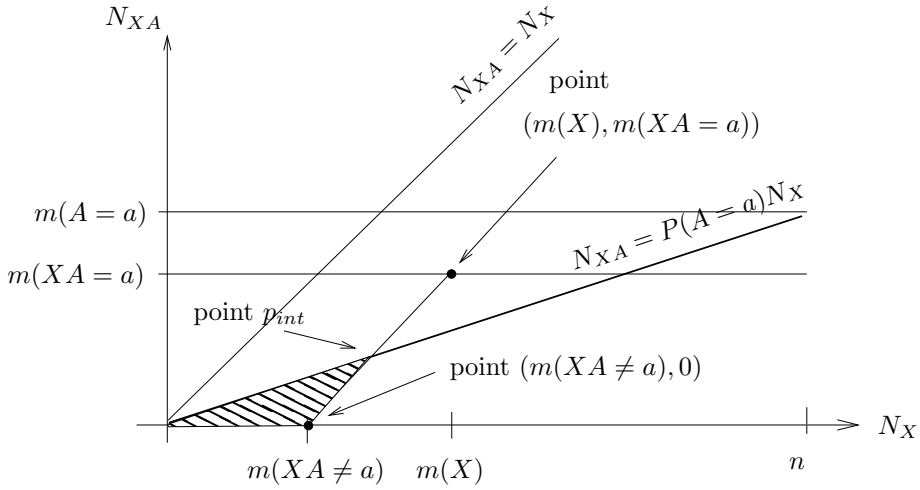


Figure 3.3: When a point $(m(X), m(XA=a))$, corresponding to rule $X \to A=a$, is given, the points associated to more specific positive dependency rules $XQ \to A=a$ lie in the shaded area.

under the independence line. In addition, the points are restricted by line $[(m(X), m(XA=a)), (m(XA \neq a), 0)]$. The reason is that $m(XQA=a) = m(XA=a) - m(X\neg QA=a)$, where $m(X\neg QA=a) \in [0, m(XA=a)]$. On the other hand, $m(XQ) = m(X) - m(X\neg QA=a) - m(X\neg QA \neq a) \leq m(X) - m(X\neg QA=a)$. So, the line contains the maximal possible values $N_X = m(XQ)$ and the corresponding $N_{XA} = m(XQA=a)$ for any $m(X\neg QA=a) \in [0, m(XA=a)]$. In point $(m(X), m(XA=a))$ holds $m(X\neg QA=a) = 0$, and in point $(m(XA \neq a), 0)$ holds $m(X\neg QA=a) = m(XA=a)$. In the following, we will show that point $(m(XA \neq a), 0)$ defines an upper bound for the M-value of any negative dependency rule $XQ \to A=a$.

Figures 3.5 and 3.6 show the four axioms graphically, when M is increasing. According to axioms (ii) and (iii), function f increases, when it departs from the independence line either horizontally or vertically (Figure 3.5). According to axiom (iv), f increases on lines $N_{XA} = cf_1 N_X$, when $cf_1 > P(A=a)$, and decreases on lines $m(A=a) - N_{XA} = cf_2(n - N_X)$, when $cf_2 > P(A=a)$ (Figure 3.6).



points (N_X, N_{XA}) for negative dependencies $XQ \to A = a$

Figure 3.4: When a point $(m(X), m(XA = a))$, corresponding to rule $X \to A = a$, is given, the points associated to more specific negative dependency rules $XQ \to A = a$ lie in the shaded area.

3.2.3 Bounds for well-behaving measures

Next we prove some useful upper bounds for well-behaving measures. For simplicity, we consider only increasing measures, but for decreasing measures, the lower bounds are met in the same points.

First we will note a couple of trivial properties, which follow from the definition of well-behaving measures.

Theorem 3.6 *Let M be a well-behaving, increasing measure. Let S be the set of legal values, as before. When $N = n$ and $N_A = m(A = a)$ are fixed, M is defined by function $f(N_X, N_{XA}, m(A = a), n)$.*

For positive dependencies hold

(i) f attains its maximal values in set S on the border defined by points $(0, 0)$, $(m(A = a), m(A = a))$, and $(n, m(A = a))$.

(ii) f attains its (global) maximum in set S in point $(m(A = a), m(A = a))$.

For negative dependencies hold

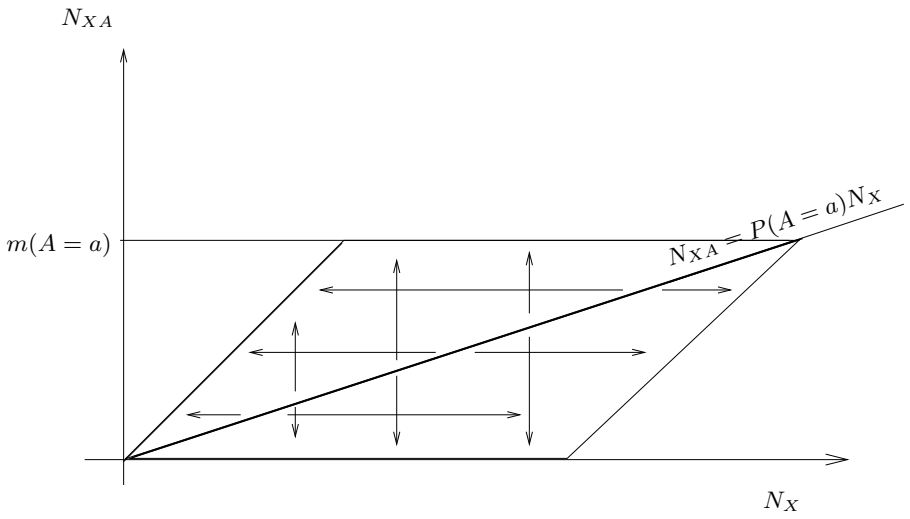


Figure 3.5: Arrows show directions where f increases (Axioms (ii) and (iii)).

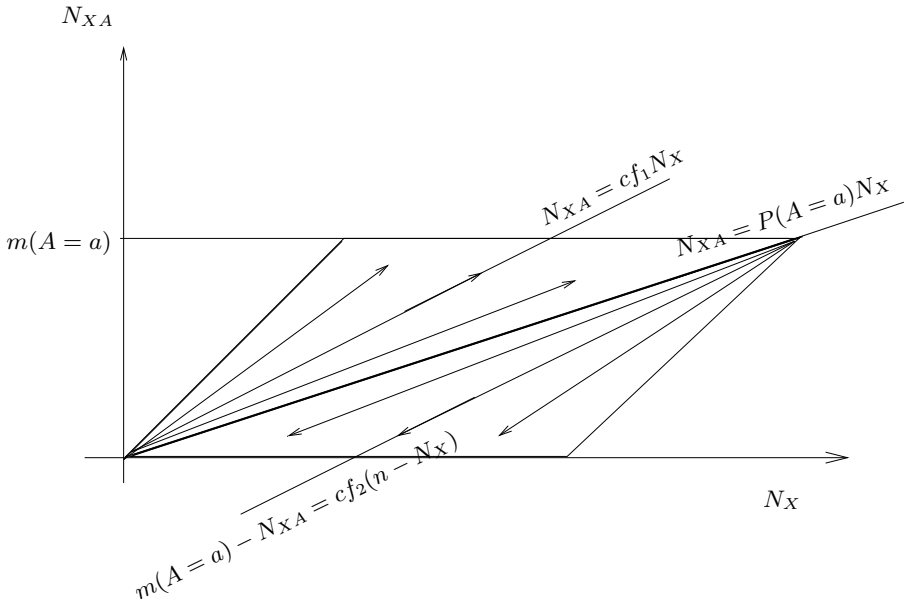


Figure 3.6: Arrows show directions where f increases (Axiom (iv)).

(i) f attains its maximal values in set S on the border defined by points $(0,0)$, $(m(A \neq a),0)$, and $(n, m(A = a))$.

(ii) f attains its maximum in set S in point $(m(A \neq a),0)$.

Proof Let us first consider positive dependencies.

(i) Since M is well-behaving, f is an increasing function of δ, when $\delta \geq 0$, in any point N_X. Therefore, it gets its maximal value on the mentioned border. (ii) When $N_{XA} = m(XA = a)$ is fixed, M being well-behaving, f is a decreasing function of N_X, when $N_X \leq \frac{n \cdot m(XA=a)}{m(A=a)}$. Therefore, f is decreasing on line $[(m(A = a), m(A = a)), (n, m(A = a))]$. On the other hand, we know that for well-behaving M, $f(N_X, cfN_X, m(A = a), n)$ is increasing with N_X, when $cf > \frac{m(A=a)}{n}$. When $cf = 1$, $f(N_X, cfN_X, m(A = a), n)$ coincides line $[(0,0), (m(A = a), m(A = a))]$. Therefore, f gets its maximum value, when $N_X = N_{XA} = m(A = a)$.

For negative dependencies, the proof is similar. The only notable exception is that now f is increasing on line $[(0,0), (m(A \neq a), 0)]$ and decreasing on line $[(m(A \neq a), 0), (n, m(A = a))]$. □

This result can already be used for pruning in two ways. In the beginning, some of the possible consequences $A = a$ may be pruned out. Given a minimum threshold min_M, $A = a$ cannot occur in the consequence of any sufficiently good positive dependency rule, if $ub1 = f(m(A = a), m(A = a), m(A = a), n) < min_M$. Similarly, $A = a$ cannot occur in the consequence of any sufficiently good negative dependency rule, if $ub1 = f(m(A \neq a), 0, m(A = a), n) < min_M$. We note that attribute A can still occur in the antecedent of good rules.

Table 3.1: Upper bound $ub1 = f(m(A = a), m(A = a), m(A = a), n)$ for common measures.

measure	*ub1*
χ^2	n
MI	$-n \log(P(A = a)^{m(A=a)} P(A \neq a)^{m(A \neq a)})$
z_1	$\frac{\sqrt{m(A \neq a)}}{\sqrt{1 + P(A=a)}}$
z_2	$\sqrt{m(A \neq a)}$
J	$-m(A = a) \log(P(A = a))$

Table 3.1 gives examples of *ub1* with common goodness measures for positive dependencies. For the χ^2-measure, the upper bound is equal to

$\max\{\chi^2(\cdot)\}$ (maximal possible value in the given data set), and no pruning is possible. For the mutual information MI, *ub1* is a function of $P(A = a)$, as desired. The function gets its best value, when $P(A = a) = 0.5$. If $P(A = a)$ deviates too much from 0.5, $ub1 < min_M$, and $A = a$ cannot occur in the consequence of any significant rule. Because MI is symmetric with respect to $A = a$ and $A \neq a$, also $A \neq a$ is an impossible consequence.

For the z-scores, z_1 and z_2, *ub1* depends on $P(A = a)$, but the best value is achieved, when $P(A = a)$ is minimal. If $P(A = a)$ is too large, $ub1 < min_M$, and $A = a$ is not a possible consequence. For the J-measure, *ub1* is also a function of $P(A = a)$, but now the best value is achieved, when $P(A = a) = e^{-1}$. If $P(A = a)$ deviates too much from e^{-1}, then $A = a$ is not a possible consequence.

The second pruning opportunity occurs, when only $m(X)$ is known, but $m(XA = a)$ is unknown. Now we can estimate an upper bound *ub2* for both $X \to A = a$ and all its specializations $XQ \to A = a$, by substituting the best possible value for N_{XA}. In the case of positive dependencies, the best possible value for N_{XA} is $\min\{m(X), m(A = a)\}$, and in the case of negative dependencies, it is $\max\{0, m(X) - m(A \neq a)\}$. In practice, this means that when $m(X) < m(A)$, the best possible M-value for positive dependence (*ub1* in point $(m(A = a), m(A = a))$ cannot be achieved any more, and $ub2 = f(m(X), m(X), m(A = a), n)$ gives a tighter upper bound than *ub1*. In the case of negative dependencies, the same happens, when $m(X)$ becomes $m(X) < m(A \neq a)$, and point $(m(A \neq a), 0)$ is no more reachable. Now $ub2 = f(m(X), 0, m(A \neq a), n) < ub1$ can be used for pruning.

Table 3.2: Upper bound $ub2 = f(m(X), m(X), m(A = a), n)$ for common measures, when $m(X) < m(A = a)$.

measure	*ub2*
χ^2	$\frac{nP(X)P(A\neq a)}{P(\neg X)P(A=a)}$
MI	$n \log\left(\frac{(P(A=a)-P(X))^{P(A=a)-P(X)}}{P(A=a)^{P(A=a)}P(\neg X)^{P(\neg X)}}\right)$
z_1	$\frac{\sqrt{m(X)}P(A\neq a)}{\sqrt{P(A=a)(1-P(X)P(A=a))}}$
z_2	$\frac{\sqrt{m(X)P(A\neq a)}}{\sqrt{P(A=a)}}$
J	$-m(X)\log(P(A = a))$

Table 3.2 gives examples of *ub2* with common goodness measures for positive dependencies, when $m(X) < m(A = a)$ and thus $ub2 < ub1$. In the

beginning, when only $P(A_i)$s, $A_i \in R$, are known, these upper bounds can be used to prune out possible consequences $A_i = a_i$ for rule $B \rightarrow A_i = a_i$ and any of its specializations, given that $P(B) < P(A_i = a_i)$.

Based on Theorem 3.6, we can prove the following observation, which enables extra pruning with some goodness measures, like the mutual information. The importance of this observation is that it can prune out attributes also from the rule antecedents, while upper bounds *ub1* and *ub2* can prune only possible consequences.

Observation 3.7 Let M be a well-behaving increasing (decreasing) goodness measure defined by function f, and $N = n$ be fixed. If

(i) f is exchangeable, i.e. $f(N_X, N_{XA}, N_A, n) = f(N_A, N_{XA}, N_X, n)$,

(ii) $f(N_A, N_A, N_A, n)$ is monotonically increasing (decreasing) in interval $[0, N_{opt}]$, $N_{opt} < n$, and

(iii) $f(m(A), m(A), m(A), n) < min_M$ (or $> max_M$) for some $m(A) \leq N_{opt}$,

then A cannot occur in the antecedent of any significant rule.

Proof Let the conditions be true for some A, $m(A) \leq N_{opt}$. Then for any rule $XA \rightarrow B = b$, $B \in R$, $B \neq A$, $b \in \{0, 1\}$, and $X \subseteq R \setminus \{A, B\}$, holds

$$
\begin{aligned}
&M(XA \rightarrow B = b) = f(m(XA), m(XAB = b), m(B = b), n) \\
&= f(m(B = b), m(XAB = b), m(XA), n) && \text{(Condition (i))} \\
&\leq f(m(B = b), m(B = b), m(B = b), n) && \text{(Theorem 3.6)} \\
&\leq f(m(A), m(A), m(A), n) \leq min_M && \text{(Conditions (ii) and (iii)).}
\end{aligned}
$$

□

When the measure is MI, the optimal value is $N_{opt} = \frac{n}{2}$. When $N_A \leq \frac{n}{2}$, function $f(N_A, N_A, N_A, n)$ is increasing, and when $N_A > \frac{n}{2}$, it is decreasing. Therefore, all consequences $C = c$, $m(C = c) \leq m(A)$, have at most as large upper bound for any $M(Y \rightarrow C = c)$ as $M(X \rightarrow A)$ has. Because MI is exchangeable, $M(XA \rightarrow B = b) = M(B = b \rightarrow XA)$ and $m(XA) \leq m(A)$. Therefore, any A which is not a possible consequence and for which $m(A) \leq \frac{n}{2}$, cannot occur in the antecedent of any rule $XA \rightarrow B = b$.

We note that it is not necessary that $f(N_A, N_A, N_A, n)$ gets its global optimum in point N_{opt} (as the measure MI does), but it is sufficient that N_{opt} is the optimum in the range $[0, N_{opt}]$. Another note is that this observation cannot be applied to the J-measure, even if $f(N_A, N_A, N_A, n)$ is

increasing in the interval $[0, \frac{n}{e}]$. The problem is that the J-measure is not exchangeable, but generally $J(X \to A) \neq J(A \to X)$.

The next theorem gives an upper bound for any positive or negative dependency rule, when a more general rule is already known. The resulting upper bound *ub3* is always at least as tight the corresponding *ub2*.

Theorem 3.8 *Let n and $m(A = a)$ be fixed and M, S and f like before. Given $m(X)$ and $m(XA = a)$ and an arbitrary attribute set $Q \subseteq R \setminus (X \cup \{A\})$*

(a) for positive dependency $XQ \to A = a$ holds $f(m(XQ), m(XQA = a), m(A = a), n) \leq f(m(XA = a), m(XA = a), m(A = a), n)$, and

(b) for negative dependency $XQ \to A = a$ holds $f(m(XQ), m(XQA = a), m(A = a), n) \leq f(m(XA \neq a), 0, m(A = a), n)$.

Proof

(a) (Positive dependence) Figure 3.3 shows the area, where possible points $(m(XQ), m(XQA = a))$ for positive dependence can lie. With any $N_X \leq m(X)$, the maximum is achieved on the border defined by points $(0, 0)$, $(m(XA = a), m(XA = a))$, and $(m(X), m(XA = a))$ (δ is maximal). On line $[(0, 0), (m(XA = a), m(XA = a))]$, f is increasing, and on line $[(m(XA = a), m(XA = a)), (m(X), m(XA = a))]$, it is decreasing. Therefore, the global maximum is achieved in point $(m(XA = a), m(XA = a))$.

(b) (Negative dependence) Figure 3.4 shows the area, where possible points $(m(XQ), m(XQA = a))$ for negative dependence can lie. Once again, f gets its maximal value for any N_X, when $-\delta$ is maximal. Therefore, the maximum must lie on the border defined by points $(0, 0)$, $(m(XA \neq a), 0)$, and the intersection point p_{int}. On line $[(0, 0), (m(XA \neq a), 0))]$, f is increasing, and the maximal value is achieved in point $(m(XA \neq a), 0)$. Therefore, it suffices to show that f gets its maximum on line $[p_{int}, (m(XA \neq a), 0)]$ in the same point, $(m(XA \neq a), 0)$.

Figure 3.7 shows the proof idea. For any point p_1 on line $[p_{int}, (m(XA \neq a), 0)]$, we can define a line of the form $m(A = a) - N_{XA} = cf_2(n - N_X)$, which goes through p_1. Because p_1 is under the independence line, $cf_2 > P(A = a)$, and line $m(A = a) - N_{XA} = cf_2(n - N_X)$ intersects N_X-axis in some point p_2 in the interval $[(0, 0), (m(XA \neq a), 0)]$. According to the definition of well-behaving measures, f is

decreasing on line $m(A = a) - N_{XA} = cf_2(n - N_X)$, and therefore f gets a better value in point p_2 than in point p_1. On the other hand, we already know that f gets a better value in $(m(XA \neq a), 0)$ than any point p_2. Therefore, f must get its upper bound in point $(m(XA \neq a), 0)$.

□

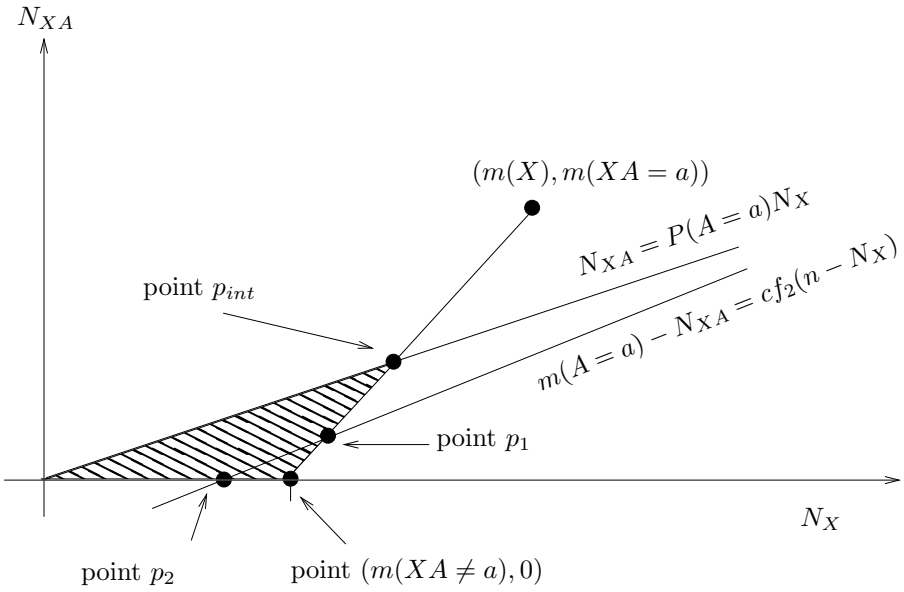


Figure 3.7: Proof idea (Theorem 3.8). Function f is better in p_2 than in p_1 and better in point $(m(XA \neq a), 0)$ than in p_2.

Theorem 3.8 (upper bound *ub3*) enables more effective pruning than Theorem 3.6 (upper bounds *ub1* and *ub2*), because now pruning is possible even if $m(X) > m(A = a)$ or $m(X) > m(A \neq a)$. The resulting upper bounds *ub3* are also tight in the sense that there can be rules $XQ \rightarrow A = a$, which reach their *ub3* values. Table 3.3 gives examples of *ub3* with common goodness measures for positive dependencies.

3.3 Lower bounds for Fisher's p_F

We assume that Fisher's p_F is also a well-behaving measure, but it is difficult to prove. Therefore, we will prove the required lower bounds *lb1*, *lb2*, and *lb3* directly.

Table 3.3: Upper bound $ub3 = f(m(XA = a), m(XA = a), m(A = a), n)$ for common measures.

measure	*ub3*
χ^2	$\frac{nP(XA=a)P(A\neq a)}{(1-P(XA=a))P(A=a)}$
MI	$n \log \left(\frac{P(\neg XA=a)^{P(\neg XA=a)}}{P(A=a)^{P(A=a)}(1-P(XA=a))^{1-P(XA=a)}} \right)$
z_1	$\frac{\sqrt{m(XA=a)}P(A\neq a)}{\sqrt{P(A=a)(1-P(XA=a)P(A=a))}}$
z_2	$\frac{\sqrt{m(XA=a)P(A\neq a)}}{\sqrt{P(A=a)}}$
J	$-m(XA = a) \log(P(A = a))$

For the first two cases (*lb1* and *lb2*) the proofs are quite simple and likely to be known also in the previous research. However, to the best of our knowledge, the proof for the third lower bound *lb3* is a new result.

Theorem 3.9 *Let us notate* $p_{abs} = \frac{m(A)!m(\neg A)!}{n!}$*, where* n *is the number of rows. For any attribute* $A \in R$ *and* $X \subseteq R \setminus \{A\}$*,* $p_F(X \to A) \geq p_{abs}$ *and* $p_F(X \to \neg A) \geq p_{abs}$*.*

Proof p_F can be expressed as

$$p_F(X \to A) = p_{abs} \sum_{i=0}^{J_1} \binom{m(X)}{m(XA)+i} \binom{m(\neg X)}{m(\neg X \neg A)+i}.$$

Since the sum is always ≥ 1, the minimum value is p_{abs}. Case $p_F(X \to \neg A)$ is similar. □

This means that $lb1 = p_{abs}$ is the absolute minimum value for any rule, where either A or $\neg A$ is the consequence. The minimum value is achieved, when $m(X) = m(A = a) = m(XA = a)$. The p_{abs}-value is lowest, when $m(A)$ is closest to $\frac{n}{2}$. On the other hand, if $m(A)$ or $m(\neg A)$ is very low, p_{abs} can be so large that no rule with consequence A or $\neg A$ could be significant. On the algorithmic level this means that A and $\neg A$ can be marked as impossible consequences and A can occur at most in the antecedent part of a significant rule.

Theorem 3.10 *Let* R *be like before. For all* $X \subsetneq R$*,* $A \in R \setminus X$*,* $a \in \{0, 1\}$*, and* $Q \subseteq R \setminus (X \cup \{A\})$ *holds*

If $m(X) \leq m(A = a)$, *then*

$$p_F(XQ \to A = a) \geq \frac{m(\neg X)!m(A = a)!}{n!(m(A = a) - m(X))!}.$$

Proof $p_F(XQ \to A = a)$ is of the form $t_0 + \ldots t_{J_1}$, where

$$t_i = p_{abs} \binom{m(XQ)}{m(XQA = a) + i} \binom{n - m(XQ)}{m(\neg(XQ)A \neq a) + i}.$$

Therefore, $p_F \geq t_{J_1}$, where $J_1 = \min\{m(XQA \neq a), m(\neg(XQ)A = a)\}$. Because $m(XQ) \leq m(X) \leq m(A = a)$, $J_1 = m(XQA \neq a)$ and p_F has a lower bound $t_{J_1} = p_{abs} \binom{n-m(XQ)}{m(A \neq a)}$. Because $\binom{m}{l}$ is an increasing function of m, $t_{J_1} \geq p_{abs} \binom{m(\neg X)}{m(A \neq a)}$, which is equal to $\frac{m(\neg X)!m(A=a)!}{n!(m(A=a)-m(X))!}$. □

This result defines the second lower bound *lb2*. However, it applies only, when $m(X) \leq m(A = a)$. If $m(X) > m(A = a)$, it is possible that there is a superset XQ such that $m(XQ) = m(A = a) = m(XQA = a)$ and p_F achieves its absolute minimum value p_{abs}. When $m(XA = a)$ is known, the following result gives a tighter (larger) lower bound *lb3*:

Theorem 3.11 *Let* R *be like before. For all* $X \subsetneq R$, $A \in R \setminus X$, $a \in \{0, 1\}$, *and* $Q \subseteq R \setminus (X \cup \{A\})$ *holds*

$$p_F(XQ \to A = a) \geq p_{abs} \binom{n - m(XA = a)}{m(A \neq a)}.$$

Proof In this proof, we use the fact that for positive dependency rule $X \to A = a$ holds $m(XA = a) > \frac{m(X)m(A=a)}{n}$ (i.e. $\gamma > 1$). We notice that it is enough to show that $p_F(X \to A = a) \geq p_{abs} \binom{n-m(XA=a)}{m(A \neq a)}$, because for any $Q \subseteq R \setminus (X \cup \{A\})$ holds

$$\binom{n - m(XQA = a)}{m(A \neq a)} \geq \binom{n - m(XA = a)}{m(A \neq a)}.$$

Let us notate $p_F = p_{abs} p_X$. Because p_{abs} is a constant, when the consequent is fixed, it can be omitted. For clarity, we present the proof for case $a = 1$. The same result is achieved for $a = 0$, when A and $\neg A$ are reversed.

If $m(X) = m(XA)$, then $p_X(X \to A)$ contains just one term, which is equal to its lower bound. Otherwise, $p_X(X \to A)$ is a sum of several terms,

but it is enough to show that for the first term t_0 holds:

$$t_0 = \binom{m(X)}{m(XA)}\binom{m(\neg X)}{m(\neg X\neg A)} \geq \binom{n-m(XA)}{m(\neg A)} \Leftrightarrow$$
$$\frac{m(X)!m(\neg A)!}{m(XA)!m(X\neg A)!(m(\neg A)-m(X\neg A))!} \geq \frac{(n-m(XA))!}{m(\neg X)} \Leftrightarrow$$
$$[m(X)\cdot\ldots\cdot(m(XA)+1)][m(\neg A)\cdot\ldots\cdot(m(\neg A)-m(X\neg A)+1)]$$
$$\geq [(n-m(XA))\cdot\ldots\cdot(m(\neg X)+1)]m(X\neg A)! \Leftrightarrow$$
$$\prod_{i=0}^{m(X\neg A)-1}(m(X)-i)(m(\neg A)-i) \geq \prod_{i=0}^{m(X\neg A)-1}(n-m(XA)-i)(m(X\neg A)-i).$$

This is true, because for all $i = 0,\ldots,m(X\neg A)-1$ holds

$$(m(X)-i)(m(\neg A)-i) \geq (n-m(XA)-i)(m(X\neg A)-i) \Leftrightarrow$$
$$m(X)m(\neg A) - im(X) - im(\neg A) + i^2 \geq m(X)(n-m(XA))$$
$$-m(XA)(n-m(XA)) - i(n-m(XA)) - im(X) + im(XA) + i^2 \Leftrightarrow$$
$$-m(X)m(A)+im(A)+m(X)m(XA)+nm(XA)-m(XA)^2-2im(XA) \geq 0.$$

Because $nm(XA) > m(X)m(A)$ and $im(A) - 2im(XA) \geq -im(XA)$, it is sufficient to show that

$$m(X)m(XA) - m(XA)^2 - im(XA) \geq 0 \Leftrightarrow$$
$$m(XA)m(X\neg A) \geq im(XA) \Leftrightarrow$$
$$m(X\neg A) \geq i,$$

which was true. □

Table 3.4: Lower bounds *lb1*, *lb2*, and *lb3* for Fisher's p_F.

$lb1 = p_{abs} = \frac{m(A)!m(\neg A)!}{n!}$
$lb2 = \frac{m(\neg X)!m(A=a)!}{n!(m(A=a)-m(X))!}$
$lb3 = p_{abs}\binom{n-m(XA=a)}{m(A\neq a)}$

The lower bounds *lb1*, *lb2*, and *lb3* for Fisher's p_F are summarized in Table 3.4. In addition, we can use Observation 3.7 to prune some attributes completely, before the search begins. Like MI, p_F is exchangeable and $f(N_A, N_A, N_A, n)$ gets its optimum value in the whole range $[0, n]$, when $N_A = \frac{n}{2}$. Therefore, attribute A cannot occur in the antecedent or consequence of any significant rule, if $p_{abs} > max_M$ and $P(A) \leq 0.5$.

3.4 Pruning redundant rules

Next, we will introduce extra pruning strategies, which can be used with both well-behaving measures and Fisher's p_F. These strategies can be applied in the special case, when a rule $X \to A = a$ with $P(A = a|X) = 1$ has been discovered. Such rules are examples of *minimal rules*. Generally, the minimality is defined as follows:

Definition 3.12 (Minimality (with respect to M)) Let M be an increasing (a decreasing) goodness measure. Rule $X \to A = a$ is minimal, if for all rules $Z \to A = a$, such that $Z \supsetneq X$, holds $M(Z \to A = a) \leq M(X \to A = a)$ ($M(Y \to A = a) \geq M(X \to A = a)$). Otherwise, rule $X \to A = a$ is non-minimal.

This means that all specializations of a minimal rule $X \to A = a$ are at most as good as $X \to A = a$ itself. We notice that a minimal rule does not have to be significant or non-redundant itself. Finding a minimal rule is always desirable, because it means that there is no reason to check any of its specializations. Based on the definition of well-behaved measures (Axiom (iv)), a rule with confidence $p(A = a|X) = 1$ is always minimal. The same holds for Fisher's p_F by Equation (3.11). In addition, there can be minimal rules with lower confidences, but they are harder to identify and we do not have any special strategies for utilizing them. In practice, they behave like any other non-redundant rules, whose specializations can be pruned out based on the upper or lower bounds of their M-values.

Let us now concentrate on minimal rules with $P(A = a|X) = 1$. First, we notice that for all specializations $XQ \to A = a$ holds $P(A = a|XQ) = 1$ and $P(A \neq a|X) = 0$. This trivial property means that we do not have to check any more special rules with consequence $A = a$ or $A \neq a$. However, we can also prune out all positive dependency rules of form $XQA \to B = b$, where $B \notin X$. The following theorem shows that all such rules will be either redundant or insignificant, if $P(A = a|X) = 1$. In the previous research, Li [45] has shown this result already for several measures (including confidence and lift), when the consequences are positive-valued. Here we extend it to cover both positive and negative consequences and all well-behaving measures as well as Fisher's p_F.

Theorem 3.13 *Let M be a well-behaving measure or Fisher's p_F. Let R be like before. If $P(A = a|X) = 1$, then all positive dependency rules $XQA \to B = b$, where $B \notin X$, $Q \subseteq R \setminus (X \cup \{A, B\})$ and $b \in \{0, 1\}$ are either redundant or insignificant.*

Proof First we notice that if $P(\neg A|X) = 1$, $P(XQAB = b) = 0$ for all Q. Therefore, rule $XQA \to B = b$ has zero frequency and cannot be significant. So it is enough to consider the case, when $P(A|X) = 1$.

When $P(A|X) = 1$, also $P(A|XZ) = 1$ for all $Z \subseteq R \setminus (X \cup \{A\})$. Therefore, $P(XQAB = b) = P(XQB = b)$ and $P(XQA) = P(XQ)$. Now rule $XQA \to B = b$ has the same M-value as a more general rule $XQ \to B = b$, and is therefore redundant:

$M(XQA \to B = b) = f(m(XQA), m(XQAB = b), m(B = b), n) = f(m(XQ), m(XQB = b), m(B = b), n) = M(XQ \to B = b)$.

The same happens, when the goodness measure is p_F. Now $p_{abs} = \binom{n}{m(B=b)}^{-1}$ and

$$p_F(XQA \to B = b) = p_{abs} \sum_{i=0}^{J_1} \binom{m(XQA)}{m(XQAB = b) + i} \binom{n - m(XQA)}{m(\neg(XQA)B \neq b) + i}.$$

This is equivalent to

$$p_F(XQ \to B = b) = p_{abs} \sum_{i=0}^{J_1} \binom{m(XQ)}{m(XQB = b) + i} \binom{n - m(XQ)}{m(\neg(XQ)B \neq b) + i},$$

because $J_1 = \min\{m(XQAB \neq b), m(\neg(XQA)B = b)\} = \min\{m(XQB \neq b), m(\neg(XQ)B = b)\}$ and $m(XQA) = m(XQ)$, $m(XQAB = b) = m(XQB = b)$, and $m(\neg(XQA)B \neq b) = m(\neg(XQ)B \neq b)$. Therefore $XQA \to B = b$ is redundant with respect to $XQ \to B = b$. □

Chapter 4

Search algorithms for non-redundant dependency rules

Hence thus and thus marvellous operations will be achieved.

On Philosopher's Stone, ancient Chinese text

In this chapter, we consider different search strategies and introduce algorithms for searching for either the K best or all sufficiently good non-redundant dependency rules with a general goodness measure M. The goodness measure can be either a well-behaving measure or Fisher's p_F, and therefore all the pruning strategies from the previous chapter can be applied. In addition, we introduce a new efficient pruning principle, called Lapis Philosophorum, which complements the basic branch-and-bound search. We give pseudocode for the Kingfisher algorithm, which implements the new pruning strategies, and analyze its time and space complexity.

4.1 Basic branch-and-bound strategy

The whole search space for dependency rules on attributes R can be represented by a complete *enumeration tree*. A complete enumeration tree lists all possible attribute sets in $\mathcal{P}(R)$. From each set $X \in \mathcal{P}(R)$, we can generate rules $X \setminus \{A_i\} \rightarrow A_i = a_i$, $A_i \in X$ and $a_i \in \{0, 1\}$, and therefore a complete enumeration tree also represents implicitly the set of all possible dependency rules. Figures 4.1 and 4.2 show two examples of complete enumeration trees, when $R = \{A, ..., E\}$. In the following, we will call the trees with the largest branches on the left (Figure 4.1) *type 1 trees*, and the trees with the largest branches on the right (Figure 4.2) *type 2 trees*. Other

kinds of enumeration trees are also possible, but these two types of trees are easiest to generate and search.

Each node v in an enumeration tree corresponds to set $X = \{A_1, \ldots, A_l\} \subseteq R$, where $A_1, \ldots, A_l$ are the labels occurring on the path from the root node to node v. In the following, we will call the nodes corresponding to sets Y_i, where $Y_i A_i = X$ for some $A_i \in X$, the *parent nodes* of node v. For example, in Figures 4.1 and 4.2, nodes corresponding to sets AC, AD and CD are the parents of the node corresponding to set ACD. The parent node, under which v is actually linked, is called the *immediate parent* of v. In Figure 4.1, the node corresponding to set AC is the immediate parent of the node corresponding to set ACD, but in Figure 4.2, the immediate parent corresponds to set CD. In the node for ACD, we can generate rules $AC \to D$, $AC \to \neg D$, $AD \to C$, $AD \to \neg C$, $CD \to A$, and $CD \to \neg A$. In addition, we should check, which consequences $A, \neg A, \ldots, E, \neg E$ are possible in the supersets of ACD (i.e. in sets $ACDE$, $ABCD$, and $ABCDE$). If a consequence was impossible in any of the parent nodes, then it is impossible also in the child node. If no consequence is possible in a node, then the node and the subtree under it can be pruned out.

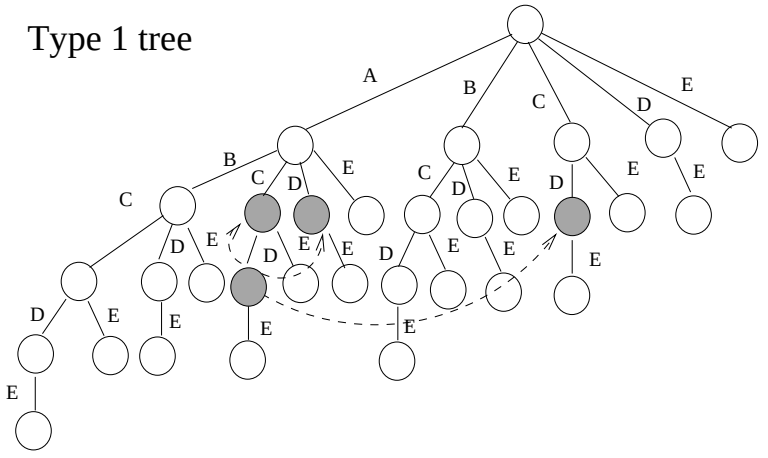


Figure 4.1: A complete enumeration tree of type 1 on attributes $A, .., E$. Arrows show the parent nodes of the node corresponding to set ACD.

In practice, the enumeration tree can be generated – either level by level or branch by branch – when the search proceeds. However, it is not necessary to generate the enumeration tree explicitly (as a data structure, where the attribute sets are stored), but still the search usually proceeds in the same manner. For example, in the classical breadth-first search (like the

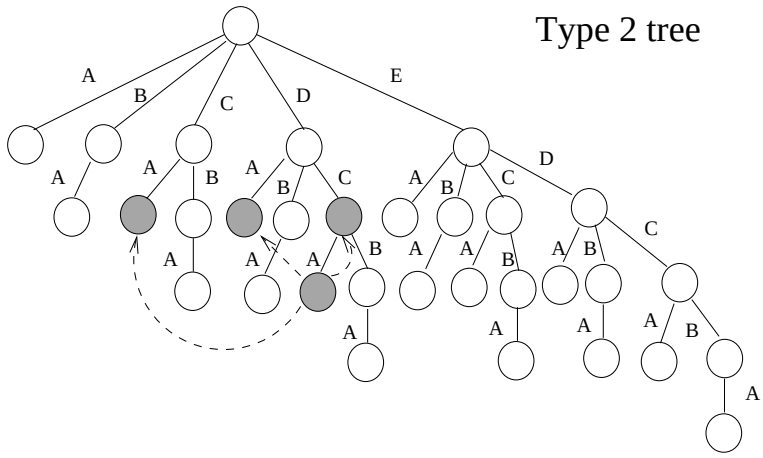


Figure 4.2: A complete enumeration tree of type 2 on attributes $A, .., E$. Arrows show the parent nodes of the node corresponding to set ACD.

Apriori algorithm for frequent sets [3, 51]) one first checks all sets containing a single attribute (1-sets), then all sets containing two attributes (2-sets), etc. This corresponds to traversing an enumeration tree level by level, from top to down. In the following algorithms, the enumeration tree is actually generated, because we need anyway a storage structure to keep record on possible consequences in the previous level sets. When the attributes are ordered, the parent nodes of any node (see an example in Figures 4.1 and 4.2) are easily located e.g. with the binary search.

Since the size of the search space (the complete enumeration tree) is in the worst case exponential, the main problem is, how to traverse as minimal an enumeration tree as possible without losing any significant dependency rules. The main strategy is an application of the branch-and-bound method, where new subtrees are generated, if they can contain sufficiently good, non-redundant rules with respect to already discovered rules. For this purpose we need the upper bounds (lower bounds) for the goodness measure M, which define the best possible M-value for any rule $(XQ) \setminus \{A_i\} \rightarrow A_i = a_i,\ A_i \in R,\ a_i \in \{0,1\},\ Q \subseteq R \setminus X$, given the information which is available in set X.

The branch-and-bound search can be implemented either in a breadth-first or depth-first manner. The main algorithm for the breadth-first search is given in Algorithm 1 and for the depth-first search in Algorithms 2 (the main program) and 3 (recursive function). For clarity, we use notation $v.set$ to refer to the set, which contains all attributes from the root of the

enumeration tree to node v. In practice, it is enough to save just the last added attribute into v.

In both search strategies, the first task is to create an empty enumeration tree t and add all single attributes as its children. For all $A_i \in R$, we create a node s_i and determine the possible consequences $A_j = a_j$, $A_j \in R$, $a_j \in \{0, 1\}$, for set $\{A_i\}$ and its supersets $X \cup \{A_i\}$ (function initialize(s_i)). In this phase, it is already possible to prune out some attributes, if they cannot occur in any significant dependency rules.

After that, the breadth-first search begins to expand the tree level by level, while the depth-first search expands it branch by branch. We do not yet go into details, how a set X can be expanded, but the idea is that all attribute sets should occur at most once in the enumeration tree. This is checked in condition `Possible`(XA_i). In addition, if we already know that all rules in X's supersets will be redundant or insignificant, then `Possible`(XA_i) returns zero for all A_i. In practice, the expansion can be coded in a more elegant way. For example, if the order of attributes is fixed, then we can add an attribute A_i to set X, only if A_i is in the canonical order after all attributes in X.

We note that in the breadth-first search condition ($num \geq l+1$) merely guarantees that we have at least $l + 1$ l-sets, before any $(l + 1)$-sets are generated. The reason is that each $(l + 1)$-set has $l + 1$ parent sets and each parent set exists in the tree, only if its supersets can produce non-redundant, significant rules. For clarity, we assume that all discovered rules are stored into a separate collection, and therefore all nodes containing only minimal rules or having too small upper bounds (too large lower bounds) for any non-redundant, significant rules can be deleted. An alternative is to keep also the discovered rules in the tree, until the search ends. In this case, we have to mark the leaf nodes, which cannot have any child nodes.

All rules which can be derived from set $X = v.set$ or its supersets are evaluated by `process`(v). Function `process` checks all v's parent nodes and combines their information on possible consequences, calculates X's frequency, estimates upper or lower bounds for $M((XQ) \setminus \{A_i\} \rightarrow A_i = a_i)$ for all possible consequences $A_i = a_i$, and decides which consequences remain as possible. If no consequence is possible, node v can be removed. Otherwise, the goodness of rules $X \setminus \{A_i\} \rightarrow A_i = a_i$ for all possible consequences $A_i = a_i$, where $A_i \in X$, is estimated by measure M, and the redundancy or minimality of rules is checked.

For efficiency, an important concern is in which order the enumeration tree should be traversed. Should we proceed in a breadth-first manner (level by level, from to top to down) or in a depth-first manner (branch

Algorithm 1 BreadthFirst

```
1   create root t
2   for all A_i ∈ R
3      add child node v_i for set {A_i}
4      initialize(v_i)
5   l ←1; num ←|R|
6   while (num ≥ l + 1)
7      for all nodes w at level l
8         X ←w.set
9         for all A_i ∈ R \ X
10           if (Possible(X A_i))
11              create child node v for set X A_i
12              if (process(v)=0)
13                 delete node v
14     l←l + 1
15     num ←number of created sets
16  output discovered rules
```

Algorithm 2 DepthFirst

```
1   create root t
2   for all A_i ∈ R
3      add child node v_i for set {A_i}
4      initialize(v_i)
5   for all children v_i
6      dfsearch(v_i)
7   output discovered rules
```

by branch, left to right)? Should we use type 1 tree or type 2 tree? And should the attributes be ordered in an ascending or descending order by their frequency?

Assuming that the search proceeds from left to right and from the top to bottom, we have four alternatives:

1. type 1 tree, when the attributes are in an ascending order, i.e. $P(A) \leq P(B) \leq ... \leq P(E)$;

2. type 1 tree, when the attributes are in a descending order, i.e. $P(A) \geq P(B) \geq ... \geq P(E)$;

Algorithm 3 dfsearch(v)

```
1    if (process(v)=0)
   // message to the call level that v can be deleted
2      return 0
3    X ←v.set
4    for all A_i ∈ R \ X
5      if (Possible(XA_i))
6        create child node w for set XA_i
7        if (dfsearch(w)=0)
8          delete node w
9    return 1
```

3. type 2 tree, when the attributes are in an ascending order, i.e. $P(A) \leq P(B) \leq ... \leq P(E)$; and

4. type 2 tree, when the attributes are in a descending order, i.e. $P(A) \geq P(B) \geq ... \geq P(E)$.

All four alternatives are possible with the breadth-first search, but type 1 tree is not sensible with the depth-first search. The problem is that in the type 1 tree, a set (e.g. ACD in Figure 4.1) would be checked before all of its parents had been checked (sets AD and CD in the Figure). This causes unnecessary extra work, if some of the parents contained better rules or would be deleted as useless. In the type 2 tree, this is avoided, but the depth-first search can still be inefficient, if we want to find only the K best non-redundant rules. The reason is that the best rules tend to occur on the first levels (with the highest frequencies), and there is no need to check the first branches in depth, if the K best rules are to be found on the first levels.

In all four alternatives, the most important concern is to restrict the largest subtree from increasing, since it can contain half of the nodes. When the largest subtree has the least frequent attribute in its root, it is also likely to stay small. Therefore, we prefer by default alternatives 1 and 4.

Alternatives 1 and 4 differ from each other only in one aspect. If we proceed a level from left to right, then the alternative 4 checks first the sets of the most common attributes, while the alternative 1 begins with the sets of the most rare attributes. The latter can be useful, if we search only the K best rules. If the measure M favours rules with a large lift but smaller frequency compared to more frequent rules with a smaller lift, then it is possible to find larger (smaller, if M is decreasing) M-values and update

min_M (max_M) in the beginning of the level. The final reason to prefer the alternative 1 is that it enables effective extra pruning together with the breadth-first search as explained in the next section. Therefore, we will consider only the alternative 1 (type 1 tree) in the following.

4.2 The Lapis Philosophorum principle

The basic branch-and-bound search prunes possible consequences only in the subtrees of a given node. However, it is also possible to prune consequences in the parent nodes, and propagate the results to other subtrees. This requires that the node is processed before any children are generated for its parents, except the immediate parent. Figure 4.1 shows an example. When we proceed level by level, from left to right, set ACD is processed before any children are created for its parent sets AD and CD. If set ACD is now removed (e.g. it has a zero frequency, which means that rule $AD \rightarrow \neg C$ was minimal), then C and $\neg C$ become impossible consequents in the node for AD and its subtrees. Similarly, if we find in the node for ACD that no rule $QCD \rightarrow A$ (for any $Q \subseteq R \setminus \{A, C, D\}$) could be significant and non-redundant, then A can be marked as an impossible consequence in the node for CD and its subtrees. This simple principle performs so effective pruning that it is called *Lapis Philosophorum*, the legendary Philosopher's stone.

Principle 4.1 (Lapis Philosophorum) Let q_{XA} be a node corresponding to set XA and q_X a node corresponding to set X as shown in Figure 4.3. If any of the following conditions holds in node q_{XA}, consequence $A = a$ can be marked as impossible in node q_X and all nodes q_{XQ} in its subtrees:

(i) Node q_{XA} does not exist (i.e. no consequence was possible in set XA or its supersets).

(ii) In node q_{XA} we find that rule $X \rightarrow A = a$ is minimal.

(iii) In node q_{XA} we find that rule $XQ \rightarrow A = a$ would be insignificant or redundant, i.e.

  (a) For an increasing goodness measure M, upper bound $ub = \mathrm{UB}(M(XQ \rightarrow A = a)) < min_M$ or $ub \leq \max\{M(Y \rightarrow A = a) | Y \subsetneq X\}$.

  (b) For a decreasing goodness measure M, lower bound $lb = \mathrm{LB}(M(XQ \rightarrow A = a)) > max_M$ or $lb \geq \min\{M(Y \rightarrow A = a) | Y \subsetneq X\}$.

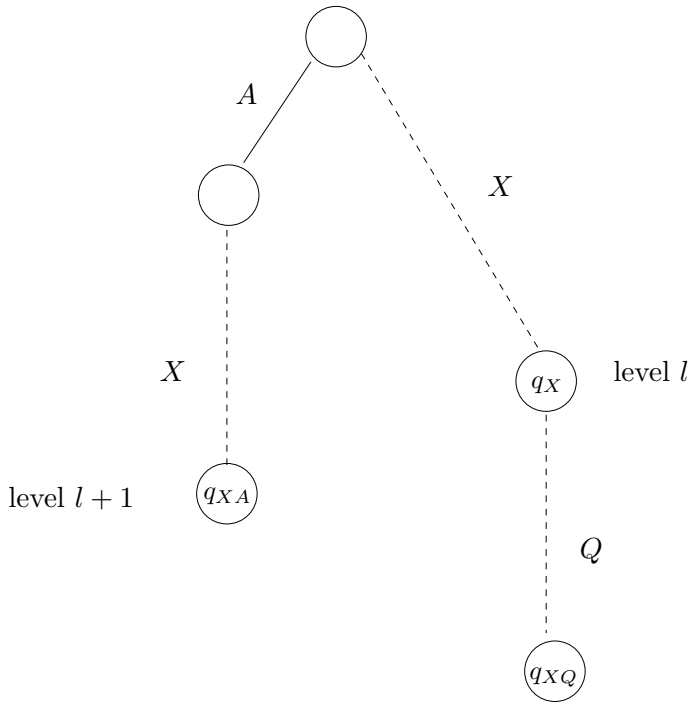


Figure 4.3: Lapis Philosophorum principle. If consequence $A = a$ is impossible in node q_{XA}, then it is impossible in nodes q_X and q_{XQ}. Here $|X| = l$. Dash lines represent paths.

The principle is based on the fact that all supersets XQA, $Q \subseteq R \setminus (X \cup \{A\})$, lie in the subtree under q_{XA}, and q_{XQ} is needed only as a parent node for rules $XQ \to A = a$.

The Lapis Philosophorum principle is especially effective, when we search positive dependency rules of the form $X \to A$. Now all rules with the least frequent consequences A_i occur in the first subtrees on the left. If set X occurs on the right-most subtrees, it is likely to have larger frequency, $P(X) > P(A)$. Therefore, the upper bound *ub2* (Table 3.2) cannot be used for pruning. On the other hand, the upper bound *ub3* (Table 3.3) cannot be used for pruning, either, because $P(XA)$ is not known in node q_X. So, in the worst case, all possible consequences in node q_X are such A_i that $P(A_i) < P(X)$. The basic breadth-first search does not offer any means to recognize, when A_i becomes an impossible consequence for X and its supersets. In the worst case, none of the consequences A_i is possible, and X is expanded in vain. With the Lapis Philosophorum principle this is

recognized immediately and node q_X can be deleted.

If we search also positive dependencies of the form $X \to \neg A$ (i.e. negative dependencies between X and A with certain measures), then infrequent consequences $A = a$ can occur also in the right subtrees. However, if A occurs in the right side of the node q_X, then A is added under q_X, and *ub3* (Table 3.3) can be used for pruning out consequence $\neg A$. If A occurs on the left side of the node q_X, then there are two alternatives: If $P(\neg A) > P(X)$, then we can use *ub2* (Table 3.2) for pruning. If $P(\neg A) \leq P(X)$, then we can use only Lapis Philosophorum.

4.3 Algorithm

Next, we give an efficient breadth-first algorithm for searching for the best non-redundant rules of the form $X \to A = a$, $a \in \{0, 1\}$, with a well-behaving goodness measure M. The algorithm is called Kingfisher, because it was originally developed for searching for dependency rules with Fisher's p_F. However, the same search method applies to any well-behaving goodness measure M.

The algorithm is designed for searching for positive dependencies, but with certain measures, like Fisher's p_F and the χ^2-measure, the positive dependence between X and $A = a$ is the same as the negative dependence between X and $A \neq a$, and therefore both positive and negative dependencies are discovered. With other measures like the z-score (Equation (2.15)), the positive dependence between X and $\neg A$ also indicates the negative dependence between X and A, but the significance is not necessarily the same, as discussed in Section 2.3.2.

The same algorithm can be used for both the optimization problem (searching for the K best rules) or enumeration problem (searching for all sufficiently good rules). For simplicity, we represent the algorithm only for the optimization problem, where both min_M (or max_M for a decreasing M) and K are given. The initial value of min_M can be set to $\min\{M(\cdot)\}$ (global minimum), in which case no rules are pruned out based on significance, until the first K rules have been found. After that the min_M-value is updated always, when a new rule is added to the collection of the best K rules. In practice, a larger initial threshold min_M is beneficial, because it can prune the first levels before any sufficiently significant rules are found.

If the algorithm is applied to the enumeration problem, then only the threshold min_M is required. The only difference to the optimization problem is that the threshold is not updated during the search. In addition, one should decide where to store all discovered rules. In the following al-

gorithm, the rules are stored into a separate rule collection, but it can be quite space consuming, if the number of discovered rules is large. One solution is to store the rules into the enumeration tree, but then the processed levels of the tree cannot be pruned as radically as with the separate rule collection.

The pseudocode for the Kingfisher algorithm is given in Algorithms 4, 5, 6, 7, and 8. The node corresponding to attribute set X is denoted by $Node(X)$. In each node $v = Node(X)$, we use the following fields:

- $v.set$ set X; in practice, it is enough to store just the last attribute in the path from the root to node v.
- $v.children$ table of pointers to v's child nodes. The size of the table is denoted by $|v.children|$.
- $v.ppossible$ and $v.npossible$ bit vectors, whose jth bits indicate, whether consequence A_j or $\neg A_j$ is a possible consequent in node v or its descendants. For simplicity, we assume that both vectors have $|R|$ bits.
- $v.pbest$ and $v.nbest$ tables for the best M-values of rules $Y \to A_j$ and $Y \to \neg A_j$, $Y \subseteq X$, $A_j \in X$. For simplicity, we assume that both tables have $|R|$ elements and index j corresponds to consequence A_j. In practice the tables can be implemented more compactly by tables of $|X|$ elements.

The main idea of the algorithm is the following:

1. Use Observation 3.7 to determine the maximal absolute frequency value min_{fr} such that even the best possible rule $X \to A = a$ with $m(A = a) = m(X) = m(XA = a) < min_{fr}$ cannot be significant. Prune out all attributes which cannot occur in significant rules. (Algorithm 4, line 1 and Algorithm 5, lines 3–4.) This step is possible only with some goodness measures like MI and p_F, satisfying the conditions of Observation 3.7.
2. Order the remaining attributes $A_i \in R$ into an ascending order by frequency and add them to the enumeration tree. Use upper bounds $ub1$ (Table 3.1) to determine consequences, which are not possible in any node $Node(A_i)$. For all A_i, use upper bounds $ub2$ (Table 3.2) to determine possible consequences $A_j = a_j$ such that $XA_i \to A_j = a_j$ can be significant. The possible consequences are marked into tables $ppossible$ and $npossible$ in node $Node(A_i)$. If A_j is possible, then $Node(A_i).ppossible[j] = 1$, and if $\neg A_j$ is possible, then $Node(A_i).npossible[j] = 1$. (Algorithm 5, lines 5–14.)

Algorithm 4 Kingfisher(R, r, min_M, K)

Input: set of attributes R, data set r, initial threshold min_M, maximal number of best rules K
Output: the best K non-redundant dependency rules for which $M \geq min_M$
Method:

```
1    determine min_fr using Observation 3.7
2    t ←check1sets(R,r,min_M,min_fr)
3    l ←2
4    while (number of (l − 1)-sets ≥ l)
     // check l-sets
5       for i = 1 to |R|
6          bfs(t.children[i], l, 0)
7       l ←l + 1
8    output the K best rules
```

Algorithm 5 check1sets(R, r, min_M, min_{fr})

Input: set of attributes R, data set r, thresholds min_M and min_{fr}
Output: root of an enumeration tree t containing the first level of nodes
Method:

```
1    for ∀A_i ∈ R
2       calculate absolute frequency m(A_i)
3       if (m(A_i) < min_fr)
4          R ←R \ {A_i}
5    order R into an ascending order by frequency
6    create root node t
7    for ∀A_i ∈ R
8       create node v = Node(A_i)
9       add v to t.children
10      for ∀A_j ∈ R
     // initialize best-tables for all consequences A_j and ¬A_j
11         v.pbest[j] ←min{M(·)}              // minimal possible M-value
12         v.nbest[j] ←min{M(·)}
     // is A_j or ¬A_j a possible consequence for set XA_i?
13         v.ppossible[j] ←possible(A_j, 1, s, A_i)
14         v.npossible[j] ←possible(A_j, 0, s, A_i)
15   return t
```

Algorithm 6 bfs(st, l, len)

Input: root of a subtree st, goal level l, path length len
Output: discovered new best rules at level l in subtree st are stored into collection *brules*
Method:

```
1    if (len = l − 2)
     // combine (l − 1)-sets to create new l-sets
2      for i = 1 to |st.children| − 1
3        Y ←st.children[i].set
4        for j = i + 1 to |st.children|
5          Z ←st.children[j].set
6          X ←Y ∪ Z
7          create node child = Node(X)
8          add child to st.children[i].children
9          initialize possible- and best-tables
           // all possible-values are set to 1 and best-values to min{M(·)}
10         if (checknode(child)=0)
11           delete child
           // use Lapis Philosophorum
12           for ∀ parent nodes v = Node(Y_m) where (X = Y_m A_m)
13             v.ppossible[m] ←0
14             v.npossible[m] ←0
15         if (Node(Y).children = ∅)
16           delete node Node(Y)
       // st's last child has never child nodes
17     delete st.children[|st.children|]
18   else for i = 1 to i = |st.children|
19     bfs(st.children[i], l, len + 1)
20   if (|st.children| = 0)                    // if all st's children were deleted
21     delete node st
```

Algorithm 7 checknode(v_X)

Input: node $v_X = Node(X)$
Output: return 0, if node v_X can be removed, and 1 otherwise; discovered new best rules in v_X are stored into collection *brules*
Method:

```
1   ismin ←0                                          // no minimal rules, yet
2   for ∀Y ⊊ X, where |Y| = |X| − 1                    // all parent sets
3     par_Y ←searchset(Y)
4     if (par_Y not found) return 0
5     updatetables(v_X, par_Y)              // update possible- and best-tables
6     if (no possible consequences left) return 0
7   calculate m(X)
8   if ((m(X) = 0) or (∃par_Y such that m(Y) = m(X)))
9     ismin ←1                                        // minimal rule found
10  for ∀A_i ∈ R                                 // update possible consequents
11    v_X.ppossible[i] ←(v_X.ppossible[i]  &  possible(A_i, 1, v_X, X))
12    v_X.npossible[i] ←(v_X.npossible[i]  &  possible(A_i, 0, v_X, X))
13    if ((A_i ∈ X) and (v_X.ppossible[i]))                // check pos. rule
14      val←M(X \ {A_i} → A_i)
15      if ((val ≥ min_M) and (val > v_X.pbest[i]))
16        add rule X → A_i to brules; v_X.pbest[i] ← val
17        update min_M                 // nothing to do, until K rules found
18    if ((A_i ∈ X) and (v_X.npossible[i]))                // check neg. rule
19      val←M(X \ {A_i} → ¬A_i)
20      if ((val ≥ min_M) and (val > v_X.nbest[i]))
21        add rule X → ¬A_i to brules; v_X.nbest[i] ← val
22        update min_M                 // nothing to do, until K rules found
23  if (ismin)                                        // pruning by minimality
24    for ∀A_i ∈ R \ X
25      v_X.ppossible[i] ←0; v_X.npossible[i] ←0
26  for ∀par_m = Node(Y_m) where (Y_m A_m = X)
27    if ((P(A_m|Y_m) = 1) or (P(¬A_m|Y_m) = 1))
28      v_X.ppossible[m] ←0; v_X.npossible[m] ←0          // by minimality
29      par_m.ppossible[m] ←0; par_m.npossible[m] ←0      // by Lapis P.
30  if (no possible consequences left) return 0
31  return 1
```

Algorithm 8 Auxiliary functions

updatetables(s, v)
// update possible- and best-tables in node s given parent node v
  for $i = 1$ **to** $i = |R|$
    $s.ppossible[i] \leftarrow (v.ppossible[i]$ **&** $s.ppossible[i])$
    $s.npossible[i] \leftarrow (v.npossible[i]$ **&** $s.npossible[i])$
    $s.pbest[i] \leftarrow \max\{v.pbest[i], s.pbest[i]\}$
    $s.nbest[i] \leftarrow \max\{n.pbest[i], s.nbest[i]\}$

possible(A_j, a_j, v, X)
// can rule $(XQ) \setminus \{A_j\} \rightarrow A_j = a_j$ *be significant and non-redundant?*
// ub2 (Table 3.2) and ub3 (Table 3.3) are implemented by UB2 and UB3
  if $(((A_j \notin X)$ **and** $(m(X) < min_{fr}))$ **or**
    $((A_j \in X)$ **and** $(m(X \setminus \{A_j\}A_j = a_j) < min_{fr})))$
      return 0 *// rule would be too infrequent*
  if $(A_j \notin X)$ $ub \leftarrow$UB2$(m(X), m(A_j = a_j))$
  else $ub \leftarrow$UB3$(m(X), m(A_j = a_j), m(XA_j = a_j))$
  if $((ub < min_M)$ **or** $((a_j = 1)$ **and** $(ub \leq v.pbest[j]))$
    or $((a_j = 0)$ **and** $(ub \leq v.nbest[j]))$
    return 0
  return 1

searchset(Y)
  return $Node(Y)$

3. Expand attribute sets as long as new non-redundant, significant rules can be found. (Algorithm 4, lines 4–7.)

   - Create l-sets from $(l-1)$-sets. (Algorithm 6, lines 2–9.)
   - For each l-set X, initialize possible consequences in $Node(X)$, given possible consequences in its parent nodes $Node(Y_m)$, where $X = Y_m A_m$ for some $A_m \in X$. Consequence $A_j = a_j$, $A_j \in R$, is possible in $Node(X)$ only if it is possible in all parent nodes $Node(Y_m)$. Initialize the best M-values for all consequences $A_j = a_j$, where $A_j \in X$, using the *best*-values in the parent nodes. (Algorithm 7, lines 2–5 and Algorithm 8, function **updatetables**.)
   - Calculate frequency $m(X)$ and check if $P(A_m = a_m|Y_m) = 1$ for any parent set Y_m. Use upper bounds *ub2* (Table 3.2) and *ub3* (Table 3.3) to decide, whether any rule $(XQ) \setminus \{A_j\} \rightarrow A_j = a_j$ can be a non-redundant, significant rule. (Upper bound *ub2* is used, when $A_j \notin X$, and *ub*3, when $A_j \in X$.) (Algorithm 7, lines 7–12 and Algorithm 8, function **possible**.) We note that in function **possible** (Algorithm 8), the frequency comparison does not prune out anything else than the upper bound comparison, if the threshold min_{fr} was determined using Observation 3.7. However, this points allows the use of other minimum frequency thresholds, if desired. For example, a common requirement is that all statistically valid rules should occur on at least five rows of data.
   - If $A_j \in X$ and $A_j = a_j$ was possible, calculate $M(X \rightarrow A_j = a_j)$. If it is sufficiently good (among the best K rules and better than the parent rules with consequence $A_j = a_j$), add it to the rule collection and update $Node(X).pbest[j]$ (if $a_j = 1$) or $Node(X).nbest[j]$ (if $a_j = 0$). Update also min_M, if K rules have been found. (Algorithm 7, lines 13–22.)
   - If minimal rules were found, mark all redundant consequences as impossible as explained in Section 3.4. (Algorithm 7, lines 23–25 and 28.)
   - Use the Lapis Philosophorum principle to propagate information on possible consequences to parents. (Algorithm 7, line 29 and Algorithm 6, lines 12–14.)

We note that when Fisher's p_F is used as a goodness measure, then the lower bounds in Table 3.4 are used instead of the upper bounds. Because

p_F is a decreasing measure, all comparisons concerning the M-value are reversed, and min_M is replaced by max_M.

Another note concerns the interchangeability of the goodness measure M. If M is interchangeable, then $M(A \to B) = M(B \to A)$ and it is sufficient to report the dependency just once. In the implementation, we report only the rule with the larger confidence. In addition, with some measures like the χ^2-measure, MI, and Fisher's p_F, also holds $M(A \to \neg B) = M(B \to \neg A)$. In this case, we also report the rule, which has the larger confidence. In both cases, the *best*-values are updated for both consequences of equivalent rules.

Technical details concerning the implementation of the enumeration tree, efficient frequency counting, and calculating Fisher's p_F (which can easily be laborious) are described in Appendix C.

4.4 Complexity

Next, we analyze the worst case time and space complexity of the Kingfisher algorithm. Both time and space complexity are exponential in the number of attributes. This is not surprising, because already a simpler problem – searching for the best classification rules of the form $X \to C = c$, $c \in \{0, 1\}$, with common goodness measures like the χ^2-measure – is NP-hard [57]. One problem in the complexity analysis is that the complexity depends on the data distribution and the effect of new pruning strategies is impossible to analyze. In Chapter 5, we will see that in practice the algorithm is quite feasible with the classical benchmark data sets.

The following theorem gives the worst case time complexity of the Kingfisher algorithm. We assume that the maximal transaction length (maximal number of 1s on any row), L, is given. Parameter L defines the maximal level (depth) and, thus, the maximal size of the generated enumeration tree. Typically, $L << k$, and the resulting complexity is more realistic than the assumption that a complete enumeration tree with 2^k nodes is generated.

Theorem 4.2 *Let n be the number of rows, k the number of attributes, K the number of best rules to be searched, and L the maximal transaction length.*

Then the worst case time complexity of searching for the K best dependency rules with the Kingfisher algorithm is

(i) $\mathcal{O}\left((k + n + \log(K))L \min\left\{2^{k-1}, \frac{L+2}{k-2(L+2)}\binom{k}{L+2}\right\}\right)$, *if* $L + 2 < \frac{k}{2}$, *and*

(ii) $\mathcal{O}((k+n+\log(K))k2^{k-1})$, *otherwise.*

Proof Processing the first level (single attribute sets) takes $\mathcal{O}(\log(n) + k\log(k)+kn+k^2) = \mathcal{O}(k(n+k))$. This is composed as follows: Determining min_{fr} (Algorithm 4 line 1) can be done in $\mathcal{O}(\log(n))$ using a binary search (iterating interval $[1, N_{opt}] \subsetneq [1, n]$). Algorithm 5 takes in the worst case $\mathcal{O}(kn + k\log(k) + k^2)$. The frequency counting for each of the k attributes takes at most n steps (see Appendix C.2), which together make kn steps. The attributes can be ordered in $\mathcal{O}(k\log(k))$ time. Possible consequences are determined for at most k nodes and for each node at most $2k$ consequences (each positive and negative consequence) is checked. These make together $2k^2$ steps. With some measures, like Fisher's p_F, this phase can be implemented more efficiently, but it does not make any difference to the overall asymptotic complexity.

Let us then analyze the complexity of processing levels $l \geq 2$. The last level is always $\leq L+1$, because after that level all sets $X \subseteq R$ have frequency 0 (on level $L+1$, $m(XA) = 0$ for all $A \notin X$, $|X| = L$, but $m(X\neg A) = m(X)$). However, $l \leq k$, and therefore the last level is $\leq \min\{L+1, k\}$. On each level $l = 2, \ldots, \min\{L+1, k\}$ the time complexity is $\mathcal{O}\left(\binom{k}{l} l(k+n+\log(K))\right)$.

Traversing all nodes on the lth level of the tree (Algorithm 6) takes at most $\mathcal{O}(lN_l)$, where N_l is the number of nodes on level l. The reason is that the tree is always pruned such that it contains only paths leading to level $l-1$. Reaching each of the new nodes on level l takes at most l steps. If the leaf nodes were linked to their successors, the traversing could be done in N_l steps. The value of N_l is difficult to evaluate, because it depends on the data distribution. However, it has always an upper bound $N_l \leq \binom{k}{l}$, which is the number of all possible l-sets.

Processing each l-set (Algorithm 6 lines 9–14 and Algorithm 7) takes $\mathcal{O}(l(k+n+\log(K)))$. This is composed as follows:

The first two parents are always known, but the rest $l-2$ parents have to be searched from the tree (Algorithm 7 line 3). Searching an $(l-1)$-set from the enumeration tree can be done in time $\mathcal{O}(l\log(k))$ (see Appendix C.1). Then, in each node, the correct child can be found in $\log_2(k)$ time using a binary search. (In practice, the number of possible children is always $< k$, except in the root.) Since the path length is $l-1$, the overall complexity is the given.

All *possible-* and *best*-values are first initialized (Algorithm 6 line 9), and then updated using the parents (Algorithm 7 line 5). This takes at most $(l+1)(2k+2l)$ steps, because there are at most k consequences in

both *possible*-tables and l possible consequences in both *best*-tables, and all tables are processed at most $l+1$ times (initialization + l times updating). Since $l \leq k$, the overall complexity is $\mathcal{O}(kl)$. (We note that in practice we also have to reserve space, when new child nodes are added, and free space, when they are deleted, but all additions and deletions can be done in $\mathcal{O}(k)$ time per node, as explained in Appendix C.1.)

Frequency counting (Algorithm 7 line 7) takes at most ln steps (see Appendix C.2). The *possible*-values are updated once again (Algorithm 7 lines 11–12), which takes at most $2k$ steps. Together, these take $\mathcal{O}(ln+k)$.

Assuming that measure M can be calculated in a constant time, the rule checking (Algorithm 7 lines 13–22) takes at most $\mathcal{O}(l\log(K))$. Each of the at most $2l$ possible rules can be checked in a constant time. Adding a rule to collection *brules* can be done in $\mathcal{O}(\log(K))$ time, if the collection is implemented as a binary heap with the worst rule on the top.

If minimal rules are found, the *possible*-tables are updated once again (lines 23–28), taking at most $2k$ steps. Lapis Philosophorum (Algorithm 7 line 29 and Algorithm 6 lines 12–14) can be implemented in $2l$ steps. Since $l \leq k$, these take together $\mathcal{O}(k)$ time.

Thus, processing each l-set takes $\mathcal{O}(l\log(k)) + \mathcal{O}(kl) + \mathcal{O}(ln+k) + \mathcal{O}(l\log(K)) + \mathcal{O}(k) = \mathcal{O}(l(k+n+\log(K)))$. The lth level takes $\mathcal{O}(N_l l(k+n+\log(K))) = \mathcal{O}\left(\binom{k}{l} l(k+n+\log(K))\right)$. Because the first level takes $\mathcal{O}\left(\binom{k}{1} 1(k+n)\right)$, the time complexity of processing levels $1,\ldots,L+1$ has an upper bound

$$\mathcal{O}\left((k+n+\log(K))\sum_{l=1}^{L+1}\binom{k}{l} l\right).$$

Because $L+1 \leq k$, we can always use an upper bound $\mathcal{O}((k+n+\log(K))k2^{k-1})$ (from Equation (A.5)), but it is often unnecessarily large. Typically, $L+2 < \frac{k}{2}$, and we can use the following upper bound (from Equation (A.6)):

$$\sum_{l=1}^{L+1}\binom{k}{l} l < (L+1)\sum_{l=1}^{L+1}\binom{k}{l} < \frac{(L+1)(L+2)}{k-2(L+2)}\binom{k}{L+2}.$$

Since $\sum_{l=0}^{\frac{k}{2}}\binom{k}{l} \leq 2^{k-1}$, the upper bound is

$$\sum_{l=1}^{L+1}\binom{k}{l} l < (L+1)\min\left\{2^{k-1}, \frac{L+2}{k-2(L+2)}\binom{k}{L+2}\right\}.$$

Therefore, the total complexity is

$$\mathcal{O}((k+n+\log(K))k2^{k-1}),$$

when $L+2 \geq \frac{k}{2}$, and

$$\mathcal{O}\left((k+n+\log(K))L\min\left\{2^{k-1}, \frac{L+2}{k-2(L+2)}\binom{k}{L+2}\right\}\right),$$

when $L+2 < \frac{k}{2}$. □

Typically $L << k$, and the complexity can be bounded by the binomial coefficient. For example, if $k \geq 3(L+2)$, the complexity reduces to $\mathcal{O}\left((k+n+\log(K))L\binom{k}{L+2}\right)$.

If the maximal transaction length L is not given, we can use an upper bound $L \leq k$. Then the time complexity becomes $\mathcal{O}((k+n+\log(K))k2^{k-1})$. In a typical case, where $k \leq n$ and $K \leq n$, the expression can be simplified to $\mathcal{O}(nk2^k)$. This is a loose upper bound, because it corresponds to checking all possible sets in $\mathcal{P}(R)$.

In the above analysis, we have assumed that the measure M can be calculated in a constant time. This is true for asymptotic measures, but some measures, like Fisher's p_F, require summing over several terms, and each term can contain binomial coefficients. In Appendix C.3, we explain how $p_F(Y \rightarrow A = a)$ can be calculated efficiently in $J = \min\{m(YA \neq a), m(\neg YA = a)\} \leq \frac{n}{4}$ steps. However, the time complexity is still quite large, $\mathcal{O}(n)$. Therefore, we give a new tight approximation for p_F, which can be calculated in a constant time.

Let us then analyze the worst-case space complexity of the Kingfisher algorithm.

Theorem 4.3 *Let k, K, and L be like before. Then the space complexity of searching for the K best dependency rules with the Kingfisher algorithm is*

(i) $\mathcal{O}\left(k\min\left\{2^{k-1}, \frac{L+2}{k-2(L+2)}\binom{k}{L+2}\right\}\right)$, *if* $L+2 < \frac{k}{2}$, *and*

(ii) $\mathcal{O}(k2^k)$, *otherwise.*

Proof First, we note that the enumeration tree contains two kinds of nodes, which we call *structure nodes* and *data nodes*. When level l is finished, nodes on levels $1, \ldots, l-1$ only code the tree structure. All data nodes (containing *best*- and *possible*-tables) are stored onto level l. On level $l+1$, new data

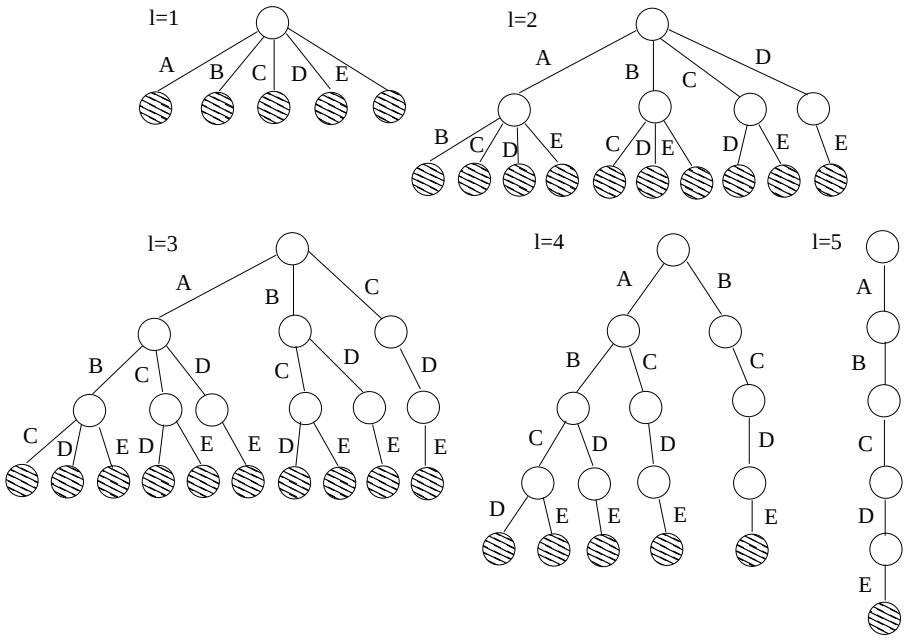


Figure 4.4: An example of the worst case tree development. Data nodes are shaded.

nodes are created onto level $l+1$ and the previous level data nodes are either removed or changed to structure nodes. All nodes which do not lead to the last level, are removed.

In the worst case, we have to construct all l-sets, $l = 1, \ldots, \min\{L+1, k\}$. After each level l, the enumeration tree contains all l-sets, and all inner nodes (structure nodes), which lead to leaf nodes. Figure 4.4 shows an example of the worst case tree development (the tree after each level l), when all sets are generated.

After each level l, the number of data nodes is $N_d = \binom{k}{l}$. For the structure nodes, we can give an upper bound

$$N_{str} \leq \sum_{j=1}^{l-1} \binom{k-1}{j}.$$

The derivation is the following: On each level, we know that at least all nodes with label A_k are leaf nodes and removed. (In addition, new nodes become leaf nodes later, when their children are removed.) If all possible sets are generated, the number of non-leaf nodes on level j is at most

the number of all j-sets, which do not contain attribute A_k, i.e. $\binom{k-1}{j}$. Summing over all levels $j = 1, \ldots, l-1$ gives N_{str}.

Because the structure node contains only the label and child pointers, all structure nodes up to level $l-1$ take together space $\mathcal{O}(N_{str} + N_d)$ (each node is pointed by one pointer). This can be bounded by $\mathcal{O}(kN_{str})$, because each structure node contains at most k child pointers.

Each data node contains two *possible*-tables, whose maximum size is k bits, and two *best*-tables, whose maximum size is l. Since $l \leq k$, the space requirement for each data node is at most $\mathcal{O}(k)$, and together, all data nodes on level l take space $\mathcal{O}(kN_d)$.

Therefore, the worst case space requirement for the whole tree after level l is

$$\mathcal{O}\left(k\sum_{j=1}^{l-1}\binom{k-1}{j} + k\binom{k}{l}\right) \leq \mathcal{O}\left(k\sum_{j=1}^{l}\binom{k}{j}\right).$$

This is largest on the last level, which is $l \leq \min\{L+1, k\}$. The sum can always be bounded by $\mathcal{O}(k2^k)$ (by Equation (A.4)). However, if $L+2 < \frac{k}{2}$, we can give a tighter upper bound (like in the proof for Theorem 4.2)

$$\mathcal{O}\left(k\min\left\{2^{k-1}, \frac{L+2}{k-2(L+2)}\binom{k}{L+2}\right\}\right).$$

In addition to the enumeration tree, we have to store the data and the best rules. Because the data can be represented by a $n \times k$ bit matrix, it takes at most $\mathcal{O}(nk)$ space. (More compact representations are possible, but this supports efficient frequency counting. See Appendix C.2.) For the K best rules, we have to store the attributes of the rule and at least the goodness measure value and information, whether the consequence is negated ("sign"). The measure value, sign, and possible other parameters (like frequencies $m(XA = a)$, $m(X)$, and $m(A = a)$) take constant space. Because the maximum rule length is $L + 1$, the best rules take at most space $\mathcal{O}(KL)$. □

In a typical case, where $n \leq 2^{k-1}$ and $K \leq n$, and L is not given, the complexity simplifies to $\mathcal{O}(k2^k)$. However, if the maximal transaction length L is known, it gives often a more realistic upper bound for the space complexity.

Chapter 5

Experiments

There remains simple experience; which, if taken as it comes, is called accident, if sought for, experiment.

F. Bacon

In this chapter, we report the experiments, which were performed to evaluate the new search algorithm. The goals were two-fold. First, we wanted to compare the quality of discovered rules, when they were searched with the Kingfisher algorithm (i.e. non-redundant rules with no or extremely small minimum frequency thresholds) and traditional methods (i.e. association rules, including redundant rules, with the required minimum frequency thresholds). With both approaches, several variable- and value-based goodness measures were used. The second goal was to evaluate the efficiency of the Kingfisher algorithm and especially the Lapis Philosophorum principle. We will first describe the test setting and then represent the results of the quality and efficiency evaluations.

5.1 Test setting

In the following, we describe the data sets, principles for the quality evaluation, tested methods, and the test environment.

5.1.1 Data sets

For testing, we used seven classical benchmark data sets from the FIMI Repository [26]. The data sets with some statistics are given in Table 5.1. Data sets Mushroom, Chess, and Pumsb are originally from the Machine Learning Repository [7], but the FIMI Repository contains binarized

Table 5.1: Description of data sets: n=number of rows, k=number of attributes, and *tlen*=average transaction length. The number of attributes having absolute frequency $m(A_i) \geq 5$ is given in the parentheses.

	n	k		*tlen*
Mushroom	8124	119	(116)	23.0
Chess	3196	75	(74)	37.0
T10I4D100K	100000	870	(869)	10.1
T40I10D100K	100000	942	(942)	39.6
Accidents	340183	468	(333)	33.8
Pumsb	49046	2113	(1734)	74.0
Retail	88162	16470	(10988)	10.3

versions of them. Unfortunately, the binarization scheme (for numerical attributes) is not known, which makes the results difficult to interpret. Sets T10I4D100K and T40I10D100K are artificially generated data sets, which are supposed to resemble typical market basket data. Retail is an example of a real market basket data. Accidents is binarized from a data set describing traffic accidents.

For each data set, Table 5.1 gives the data dimensions (numbers of rows and attributes) and average transaction length (average number of 1-valued attributes on a row). In addition, we give the number of attributes having absolute frequency $m(A_i) \geq 5$. The reason is that with all tested methods we used an absolute minimum frequency threshold five, which is commonly considered as a minimum requirement for any statistically valid conclusions.

The average transaction length describes how dense the data set is. Sets T10I4D100K and Retail are sparse data sets, which is typical for market basket data. Mushroom is also a relatively sparse set. All other data sets are quite dense. Especially, Pumsb is an exceptionally dense data set, which also contains a large number of attributes. This makes it potentially the most difficult data set for the search. Retail contains the largest number of attributes, which can make it also computationally heavy, but the set is so sparse that the search is not likely to continue deep. We note that the computationally most demanding data sets contain a large number of frequently occurring attribute combinations. This is likely, if the average transaction length is large with respect to the number of attributes. For example, Chess and T40I10D100K have nearly equal average transaction lengths, but T40I10D100K contains so many attributes that complex attribute combinations are likely to be infrequent. Accidents has the largest

number of rows, which means that frequency counting can be time consuming.

5.1.2 Evaluating results

When dependency rules are searched for, the main objective is to find strong dependencies, which hold also in future data. In the variable-based semantics, a natural measure for the strength of the dependency is the leverage. If the leverage values in the original data and future data are approximately equal, the strength of the dependency is accurately estimated. On the other hand, if the leverage in the original data has a large positive value (indicating a strong positive dependency), but in future data the leverage is near zero or negative (indicating independence or negative dependence), then the dependency can be considered spurious. It is not necessarily harmful, if the leverage is larger in future data than in the original data, but in some applications, one may want to ascertain that all dependencies are relatively weak. Therefore, we require that for an accurate dependency rule the leverage in the original data set reflects the leverage in future data accurately. When value-based semantics is used, the lift values are compared instead of the leverage.

Because future data is not known, the behaviour in future data has to be estimated by other means. The most common solution is to use cross-validation, where the best rules are learnt from one part of the data and their accuracy is tested in another part of the data. The process is repeated several times with different divisions to a learning set and a test set and the accuracies are averaged.

For good estimates, the process should be repeated sufficiently many times, but in the same time, the test sets should be sufficiently large, because otherwise the tested rules cannot be applied in them. The problem is that if a test set is too small, then many rule antecedents do not occur in the test set at all, and their accuracy cannot be evaluated. The classical 10-fold cross-validation does not suit the purpose, because the test sets would be too small (one tenth part of the data). Therefore, we have used the following scheme: each data set is divided ten times randomly into a learning set of two thirds of the data and a test set of one third of the data.

From each learning set, we have searched for the 100 best rules with the selected method, and tested them in the corresponding test set. For all rules, which can be applied to the test set, we have calculated the mean squared error of the leverage and lift. If we notate the learning set by r_l, test set by r_t, and the set of best rules by $\mathcal{R}$, then the mean squared error

of the leverage is

$$MSE_\delta(r_l, r_t, \mathcal{R}) = \frac{1}{N} \sum_{\substack{X \to A = a \in \mathcal{R}, \\ X \text{ occurs in } r_t}} (\delta_l(X \to A = a) - \delta_t(X \to A = a))^2,$$

where δ_l is the leverage in set r_l, δ_t is the leverage in set r_t, and N is the number of rules which could be applied to tr. Similarly, the mean squared error of the lift is

$$MSE_\gamma(r_l, r_t, \mathcal{R}) = \frac{1}{N} \sum_{\substack{X \to A = a \in \mathcal{R}, \\ X \text{ occurs in } r_t}} (\gamma_l(X \to A = a) - \gamma_t(X \to A = a))^2,$$

where γ_l and γ_t are the lift values in sets r_l and r_t. Because MSE_δ and MSE_γ are difficult to interpret, we report the root mean squared errors $\sqrt{MSE_\delta}$ and $\sqrt{MSE_\gamma}$, which are in the same scale as the average δ and γ.

In addition, we have calculated other statistics (average frequency, confidence, leverage, and lift in the training sets) of rules which could be applied to the test set. The number of *harmful rules*, which express a positive dependency in the learning set, but independence or negative dependence in the test set, is also recorded. For Kingfisher, we report the number of negative rules, and for Apriori, a lower bound for the number of redundant rules among the 100 best rules. All these statistics are averaged over all ten learning set and test set pairs.

5.1.3 Search methods

All experiments were performed with implementations of the Kingfisher algorithm and the traditional association rule algorithm. For the Kingfisher algorithm, we used two C-implementations. The newest implementation is simply called Kingfisher and it searches for both positive and negative rules. Available measures are Fisher's p_F and the χ^2-measure with or without a continuity correction. For the value-based measures we used an ancestor of Kingfisher, called Chitwo, which searches for only positive rules. Available measures are the χ^2-measure and the z-score (Equation (2.15)) with or without the continuity correction.

In addition, Chitwo offers a facility to traverse the search space using a minimum threshold for the z-score, min_z, but select the best rules using the binomial probability p_{bin} (Equation (2.14)). If min_z could be set sufficiently low (e.g. 2.0, corresponding to $p \approx 0.05$), the approach would guarantee the best rules by p_{bin}. However, for an efficient search the minimal z-score has

to be set quite large. If the z-scores are exaggerated (overestimate the significance), rules with a lower z-score but better p_{bin} are missed. As an alternative, one can restrict the search space with min_z, but test all rules (or all rules achieving at least a certain small z-score) in this area by p_{bin}. The problem of this approach is that the min_z threshold can still prune out areas, which would achieve a too small z-score, but still a good p_{bin}-value. Therefore, both approaches are in practice heuristic, and can only approximate the search by p_{bin}.

Both Kingfisher and Chitwo programs are available on `http://www.cs.helsinki.fi/u/whamalai/sourcecode.html`.

With Kingfisher using p_F, no minimum frequency thresholds or restrictions on the rule length were required. However, with the asymptotic measures (χ^2 and z), the search continued so deep that either minimum frequency thresholds or maximal rule lengths had to be used with some data sets. If these were needed, they were set as little restricting as possible.

For comparison, the best rules were also searched for with the traditional association rule algorithm using common goodness measures. The implementation was Borgelt's Apriori version 5.14 [15, 16]. It is an efficient trie-based implementation of the classical Apriori algorithm [51, 3]. This implementation was selected, because it is freely available, Linux-compatible program with a facility to select rules with common goodness measures. Like most association rule programs, it searches for only positive rules.

For testing Apriori, we used two variable-based measures, the χ^2-measure and mutual information (MI), and two value-based measures, certainty factor (cfa) and the J-measure (implemented by us). The best rules were searched for using as low a minimum frequency threshold as possible, but as large a threshold for the goodness measure as possible to find 100 rules. With some data sets, we also had to restrict the maximal rule length. If the 100 best rules were not unambiguous, i.e. there were several equally good rules to be selected to the last l positions, then l rules were selected randomly among them.

We note that Apriori does not prune redundant rules, and therefore the best rules are often variations of each other. We have estimated a lower bound for the number of redundant rules by checking for each rule $X \to A$, whether there is a more general rule $Y \to A$, $Y \subsetneq X$, in the list of the best rules before $X \to A$. If both rules are equally good, then $Y \to A$ can occur after $X \to A$ in the list or is not selected at all. This happened always, when Apriori found a large number of rules with the best possible M-value. For example, all data sets contained at least 100 rules with $cf = 1.0$ and

therefore obtained the maximal *cfa*-value. As a result, *cfa* seems to find only few redundant rules, even if all selected rules were redundant with respect to one simple rule.

A special case of redundant rules are rules of form $A \to B$ and $B \to A$. With the χ^2-measure and *MI* holds that $A \to B$ is equally good as $B \to A$, and therefore both occur together among the best rules.

5.1.4 Test environment

All experiments were executed in *CSC* (IT Center for Science Ltd) computing environment Hippu. Hippu consists of two servers, both of which have eight quad-core AMD Opteron 8360 SE 2.5 GHz processors. One of the servers has 512 GB and the other one 256 GB main memory. The environment has RedHat Enterprise Linux 5 operating system. The programs were compiled with Gnu gcc version 4.1.2 into 64-bit code.

We note that the memory size is very large compared to ordinary desktop computers. This is especially beneficial for Apriori, for which the memory size is often the bottle-neck. Therefore, we were able to use very low minimum frequency thresholds. However, generating a large number of frequent sets is still time consuming, and in practice some minimum frequency thresholds and/or restrictions on the maximal rule length were necessary. The asymptotic measures of Kingfisher and Chitwo do also benefit from the large memory, although it is less crucial, because the enumeration tree is pruned. On the other hand, both programs search for the best non-redundant rules, which means that the search is prone to continue deeper than with Apriori, if the same measures and thresholds were used. Therefore, we had to use some minimum frequency thresholds or maximum rule lengths with the largest data sets. All parameters were set as loose as possible such that each experiment could be executed in less than 10 minutes CPU time.

Search with Kingfisher using p_F as a goodness measure is very memory-efficient and all experiments (without any restrictions) could be executed in a normal desktop computer with just 2 GB main memory.

5.2 Results of the quality evaluation

In the following, we will first report the results of the quality evaluation for all data sets. The summary of the quality evaluation is given in Section 5.2.8.

The tested methods are Kingfisher (Kf) with p_F when all rules or only positive rules were searched for; Kingfisher with the χ^2-measure, when all

rules were searched for using the continuity correction (cc) or without it, and when only positive rules (pos) were searched for using the continuity correction; Chitwo with the z-score with or without the continuity correction; Chitwo when the search space was restricted by the z-score, but the best rules were selected by p_{bin}; and Apriori with measures χ^2, MI, cfa, and J. For Chitwo with p_{bin} we tested two alternatives: in version 1 (v1), all rules whose z-score exceeded min_z were tested by p_{bin}, while in version 2 (v2), all rules whose z-score exceeded $\frac{min_z}{4}$ were tested by p_{bin}.

5.2.1 Mushroom

Results for set Mushroom are given in Tables 5.2 and 5.3. For Kingfisher and Chitwo no minimum frequency thresholds (other than the absolute minimum frequency 5, i.e. fr=0.00092) or maximal rule lengths were used. For Apriori, we had to use min_{fr}=0.01 (absolute frequency 55) and maximal rule length 8.

Kingfisher found with both measures, p_F and χ^2, about six negative rules among the 100 best rules. The best four rules with both measures were the same in all learning sets ($A2 \rightarrow \neg A1$, $A53 \rightarrow \neg A52$, $A24 \rightarrow \neg A23$, $A39 \rightarrow \neg A38$). For all of these holds $P(X\neg A) = P(X) = P(\neg A)$, which simply states that X and A are mutually excluding attributes. Quite likely, these rules are just a side effect of the binarization. The other negative rules were more diverse, but still they seem to be related to the binarization scheme (e.g. rule pair $A36A39A59A86 \rightarrow A23$ and $A34A36A39A59 \rightarrow \neg A24$, both with cf=1.0).

With p_F, the rules were quite frequent and had large leverage, while the χ^2-measure favoured less frequent rules with larger lift. The continuity correction had a clear effect on the quality of rules by the χ^2-measure and managed to avoid the most infrequent rules (absolute frequency 25, i.e. fr=0.0046) with the highest lift values. As a result, the MSE of the lift also decreased, which means that the lift values held more accurately in the test sets.

Chitwo with the z-score found quite infrequent rules (fr=0.0048-0.0048) with high lift but small leverage. The lift values diverged quite much in the test sets, but no harmful rules were produced. The continuity correction had no effect on the results. When the best rules were selected with the binomial probability, the results approached those by p_F. The difference between the two versions of the selection strategy was marginal. With version 2, the rules were slightly simpler, more frequent, and had lower lift and leverage.

Apriori with measures MI and J suffered from a large number of re-

Table 5.2: Results of the cross validation in set Mushroom. Method, parameters and numbers of applicable rules, negative rules, clearly redundant, and harmful rules. For harmful rules, the average numbers of negative dependence and independence rules are given in the parentheses.

method (min_{fr}, max_{len})	appl	neg	red	harmful
Kf p_F $(-,-)$	100.0	6.6	0.0	0.0
Kf p_F pos $(-,-)$	100.0	0.0	0.0	0.0
Kf χ^2 $(-,-)$	100.0	6.0	0.0	0.0
Kf χ^2 cc $(-,-)$	100.0	5.9	0.0	0.0
Kf χ^2 pos $(-,-)$	100.0	0.0	0.0	0.0
Chitwo z $(-,-)$	100.0	0.0	0.0	0.0
Chitwo z cc $(-,-)$	100.0	0.0	0.0	0.0
Chitwo p_{bin} v1 $(-,-)$	100.0	0.0	0.0	0.0
Chitwo p_{bin} v2 $(-,-)$	100.0	0.0	0.0	0.0
Apriori χ^2 $(0.01, 8)$	100.0	0.0	2.5	0.0
Apriori MI $(0.01, 8)$	100.0	0.0	33.8	0.0
Apriori cfa $(0.01, 8)$	100.0	0.0	0.0	0.0
Apriori J $(0.01, 8)$	100.0	0.0	39.6	0.0

Table 5.3: Results of the cross validation in set Mushroom. Average rule length, frequency, confidence, lift, leverage, and root mean squared errors of lift and leverage.

method	len	fr	cf	γ	δ	$\sqrt{MSE_\gamma}$	$\sqrt{MSE_\delta}$
Kf p_F	4.1	0.2364	0.97	4.3	0.17127	0.2	0.00479
Kf p_F pos	4.1	0.2204	0.97	4.5	0.16793	0.2	0.00492
Kf χ^2	3.0	0.0550	1.00	339.0	0.0232	448.0	0.00227
Kf χ^2 cc	3.7	0.1313	1.00	22.4	0.0875	3.9	0.00413
Kf χ^2 pos	3.9	0.0919	1.00	24.7	0.0778	4.1	0.00413
Chitwo z	4.9	0.0027	1.00	593.2	0.0027	652.0	0.00166
Chitwo z cc	4.0	0.0027	1.00	593.0	0.0027	652.0	0.00155
Chitwo p_{bin} v1	4.1	0.1997	0.98	5.0	0.1585	0.2	0.00516
Chitwo p_{bin} v2	3.9	0.1936	0.95	4.5	0.1457	0.2	0.00492
Apriori χ^2	7.3	0.0384	1.00	39.0	0.0349	5.6	0.00356
Apriori MI	7.1	0.2121	1.00	4.7	0.16708	0.2	0.00496
Apriori cfa	7.5	0.0230	1.00	2.6	0.0093	0.7	0.00168
Apriori J	7.0	0.2211	1.00	4.5	0.1705	0.2	0.00481

dundant rules (more than one third of the rules). All rules had the same consequence and most of them were specifications of just two rules, which were also the best positive rules by Kingfisher using p_F. Therefore, all

statistics were nearly equivalent to Kingfisher (when only positive rules were searched for), except the average rule length. When the χ^2-measure was used, Apriori found over 100 000 rules with the maximal possible χ^2-value in all learning sets. Therefore, the 100 best rules were a random sample from all equally good rules. Due to the minimum frequency threshold, Apriori managed to avoid the most infrequent rules, which were selected by Kingfisher (without the continuity correction). With *cfa*, Apriori found over 90 000 000 rules with the maximal *cfa*-value (i.e. cf=1.0). A random sample of 100 rules from these did not reveal any redundancy. The average lift value was the smallest among all tested methods. This was expected, because *cfa* does not measure the strength of the dependency in any way, if cf=1.

5.2.2 Chess

Results for set Chess are given in Tables 5.4 and 5.5. For Kingfisher and Chitwo no minimum frequency thresholds (other than the absolute minimum frequency 5, i.e. fr=0.002347) were used, but for Chitwo, the maximal rule length was set to 7. For Apriori, we had to use a quite large minimum frequency threshold (min_{fr}=0.20, i.e. absolute minimum frequency 426) along with the maximal rule length 7.

Kingfisher found with both measures, p_F and χ^2, a large number of negative dependency rules. The best rules were negative and likely a side effect of the binarization, like in Mushroom. However, the data contained also many rules of form $A \rightarrow \neg B$, where A and B were not completely mutually excluding, but with $m(A\neg B) = m(A) \approx m(\neg B)$. These rules rules were selected with both measures. In addition, p_F selected more complex negative rules, which often occurred as pairs of form $X \rightarrow A$ and $X \rightarrow \neg B$. The confidence of these rules was only modest (cf=0.60-0.80), but the frequency was quite high (fr=0.18-0.50). Instead of these, the χ^2-measure selected quite rare but strong rules (with absolute frequency 11–12, i.e. fr=0.005, and cf=0.80-0.90). When only positive rules were searched for, the best rules were more complex, especially with the χ^2-measure. This was also observed in the search, which continued deep. The continuity correction had only a marginal effect on the results.

Chitwo with the z-score found more infrequent rules than Kingfisher. The average lift was a little bit larger but leverage clearly smaller than with the χ^2-measure. The continuity correction had no effect on the results. Chitwo found also two harmful rules in one of the test sets, but both of them covered just one row of the test data. Both harmful rules had a very low absolute frequency ($m(XA) = 8$), which is often problematic for the z-

Table 5.4: Results of the cross validation in set Chess. Method, parameters and numbers of applicable rules, negative rules, clearly redundant, and harmful rules. For harmful rules, the average numbers of negative dependence and independence rules are given in the parentheses.

method (min_{fr}, max_{len})	appl	neg	red	harmful
Kf p_F $(-,-)$	100.0	61.6	0.0	0.0
Kf p_F pos $(-,-)$	100.0	0.0	0.0	0.0
Kf χ^2 $(-,-)$	100.0	36.0	0.0	0.0
Kf χ^2 cc $(-,-)$	100.0	36.0	0.0	0.0
Kf χ^2 pos $(-,-)$	100.0	0.0	0.0	0.0
Chitwo z $(-,7)$	99.1	0.0	0.0	0.2 (0.2+0.0)
Chitwo z cc $(-,7)$	99.1	0.0	0.0	0.2 (0.2+0.0)
Chitwo p_{bin} v1 $(-,7)$	100.0	0.0	0.0	0.0
Chitwo p_{bin} v2 $(-,7)$	100.0	0.0	0.0	0.0
Apriori χ^2 $(0.20,7)$	100.0	0.0	98.8	0.0
Apriori MI $(0.20,7)$	100.0	0.0	98.9	0.0
Apriori cfa $(0.20,7)$	100.0	0.0	0.0	18.0 (0.5+17.5)
Apriori J $(0.20,7)$	100.0	0.0	98.8	0.0

Table 5.5: Results of the cross validation in set Chess. Average rule length, frequency, confidence, lift, leverage, and root mean squared errors of lift and leverage.

method	len	fr	cf	γ	δ	$\sqrt{MSE_\gamma}$	$\sqrt{MSE_\delta}$
Kf p_F	3.5	0.4534	0.84	2.1	0.1672	0.1	0.0064
Kf p_F pos	4.7	0.3494	0.70	2.0	0.1513	0.2	0.0067
Kf χ^2	5.3	0.2988	0.93	123.9	0.0585	75.8	0.0046
Kf χ^2 cc	5.2	0.3060	0.90	107.5	0.0640	59.1	0.0048
Kf χ^2 pos	7.0	0.0268	0.84	167.8	0.0184	78.6	0.0033
Chitwo z	6.6	0.0044	0.86	196.8	0.0044	96.4	0.0022
Chitwo z cc	6.6	0.0044	0.86	196.8	0.0044	96.4	0.0022
Chitwo p_{bin} v1	6.3	0.0254	0.73	72.5	0.0218	37.2	0.0047
Chitwo p_{bin} v2	5.8	0.1203	0.59	5.1	0.0840	0.9	0.0089
Apriori χ^2	5.6	0.5938	1.00	1.6	0.2245	0.1	0.0058
Apriori MI	5.6	0.5938	1.00	1.6	0.2245	0.1	0.0058
Apriori cfa	6.7	0.2865	1.00	1.1	0.0120	0.0	0.0021
Apriori J	5.6	0.5938	1.00	1.6	0.2245	0.1	0.0058

score (the values become exaggerated). When the best rules were selected by the binomial probability, the harmful rules were avoided. Generally, the selection by the binomial probability managed to prune out the most

infrequent rules, and as a result the frequency and leverage were larger but the lift smaller than with the z-score alone. The difference between the two versions of the selection strategy was clear. Version 2 produced remarkably more frequent rules, with lower lift but larger leverage. The average confidence was the smallest among all tested methods.

Apriori found very similar rules with measures χ^2, MI, and J. All rules had the same consequence and nearly all of them were redundant specializations of just one rule, which was also the best positive rule by Kingfisher using p_F. Therefore, the average frequency, confidence, and leverage were all high. With *cfa*, no redundant rules were revealed, but nearly one fifth of the rules were harmful, mostly independence rules. The harmful rules were quite frequent (with fr=0.14–0.47) and had always confidence cf=1.0 (like all selected rules by *cfa*), but they expressed only a weak dependency (typically γ=1.0005) in the learning sets. Therefore, it was only expected that they would express independence also in the test sets.

All methods produced reasonable *MSE* values in proportion to the average lift and leverage values.

5.2.3 T10I4D100K

Results for set T10I4D100K are given in Tables 5.6 and 5.7. Set T10I4D100K is so sparse that no minimum frequency thresholds (other than the absolute minimum frequency 5, i.e. fr=0.000075) or maximum rule lengths were needed with any of the tested methods.

Kingfisher found only positive rules and all of them were relatively simple and infrequent. With the χ^2-measure, the lift was generally large but also unstable (the difference between the rules in the learning sets and test sets was large). The continuity correction clearly improved the accuracy of the lift. With Chitwo and the z-score, the continuity correction had a little effect. Generally, the quality of the rules with the z-score was comparable to those by the χ^2-measure without the continuity correction, and many rules were the same. Once again, the binomial probability produced more frequent rules with larger leverage but smaller and more accurate lift. The difference between the two versions was relatively small. The best rules were mostly the same as with p_F, but in a different order.

Apriori with the χ^2-measure produced more complex rules than Kingfisher, due to the large number of redundant rules. The average lift was the highest among all tested methods, but also unstable. Measures MI and J produced also a large number of redundant rules, although they were simpler than with the χ^2-measure. The best rules were – especially with measure MI – mostly the same as the best rules by p_F. In addition, the

Table 5.6: Results of the cross validation in set T10I4D100K. Method, parameters and numbers of applicable rules, negative rules, clearly redundant, and harmful rules. For harmful rules, the average numbers of negative dependence and independence rules are given in the parentheses.

method (min_{fr}, max_{len})	appl	neg	red	harmful
Kf p_F $(-,-)$	100.0	0.0	0.0	0.0
Kf χ^2 $(-,-)$	100.0	0.0	0.0	0.0
Kf χ^2 cc $(-,-)$	100.0	0.0	0.0	0.0
Chitwo z $(-,-)$	100.0	0.0	0.0	0.0
Chitwo z cc $(-,-)$	100.0	0.0	0.0	0.0
Chitwo p_{bin} v1 $(-,-)$	100.0	0.0	0.0	0.0
Chitwo p_{bin} v2 $(-,-)$	100.0	0.0	0.0	0.0
Apriori χ^2 $(-,-)$	100.0	0.0	48.2	0.0
Apriori MI $(-,-)$	100.0	0.0	39.0	0.0
Apriori cfa $(-,-)$	81.3	0.0	0.0	2.4 (2.4+0.0)
Apriori J $(-,-)$	100.0	0.0	39.9	0.0

Table 5.7: Results of the cross validation in set T10I4D100K. Average rule length, frequency, confidence, lift, leverage, and root mean squared errors of lift and leverage.

method	len	fr	cf	γ	δ	$\sqrt{MSE_\gamma}$	$\sqrt{MSE_\delta}$
Kf p_F	2.9	0.0073	0.89	68.1	0.0071	4.7	0.00053
Kf χ^2	3.9	0.0007	1.00	4221.9	0.0007	8250.0	0.00015
Kf χ^2 cc	3.9	0.0012	0.99	1468.7	0.0012	361.0	0.00023
Chitwo z	3.9	0.0006	1.00	4240.5	0.0006	8250.0	0.00015
Chitwo z cc	3.9	0.0007	1.00	4142.9	0.00067	8060.0	0.00015
Chitwo p_{bin} v1	3.0	0.0045	0.92	184.1	0.0044	21.4	0.00040
Chitwo p_{bin} v2	2.8	0.0055	0.88	126.3	0.0055	11.0	0.00044
Apriori χ^2	4.9	0.0004	1.00	5420.0	0.0004	9790.0	0.00010
Apriori MI	3.3	0.0075	0.89	74.8	0.0074	5.0	0.00056
Apriori cfa	5.3	0.0001	1.00	77.3	0.0001	23.1	0.00006
Apriori J	3.3	0.0075	0.91	72.4	0.0074	4.9	0.00056

selected rules contained their redundant specializations. Because the data set contained only short transactions, the most complex redundant rules were avoided. As a result, the quality of rules was comparable to p_F.

Measure cfa selected the most complex and infrequent rules. As a result, only 81% of them could be applied to the test sets (i.e. the rule antecedent occurred in the test set), and some of them were harmful. All harmful rules

had a quite large lift value ($\gamma = 20 - 179$) in the learning set, but zero lift in the test set. The problem was that the rules were very rare, occurring on just 6–8 rows of data, and therefore very likely due to chance.

5.2.4 T40I10D100K

Results for set T40I10D100K are given in Tables 5.8 and 5.9. For Kingfisher and Chitwo no minimum frequency thresholds (other than the absolute minimum frequency 5, i.e. fr=0.000075) were used, but for Apriori, we had to use min_{fr}=0.01 (absolute minimum frequency 667). No restrictions on the rule length were used.

The results were quite typical. Kingfisher with p_F found more frequent rules having larger leverage but much smaller lift than with the χ^2-measure. The continuity correction balanced the results by pruning out the most infrequent rules. With the z-score, the continuity correction had hardly any effect. Generally, the quality of the rules with Chitwo and the z-score was comparable to Kingfisher with the χ^2-measure (without the continuity correction). When the binomial probability was used, the results approached those by p_F. The discovered rules were also mostly the same. The difference between the two versions was marginal.

Table 5.8: Results of the cross validation in set T40I10D100K. Method, parameters and numbers of applicable rules, negative rules, clearly redundant, and harmful rules. For harmful rules, the average numbers of negative dependence and independence rules are given in the parentheses.

method (min_{fr}, max_{len})	appl	neg	red	harmful
Kf p_F $(-,-)$	100.0	0.0	0.0	0.0
Kf χ^2 $(-,-)$	99.0	0.0	0.0	0.0
Kf χ^2 cc $(-,-)$	100.0	0.0	0.0	0.0
Chitwo z $(-,-)$	99.0	0.0	0.0	0.0
Chitwo z cc $(-,-)$	99.0	0.0	0.0	0.0
Chitwo p_{bin} v1 $(-,-)$	100.0	0.0	0.0	0.0
Chitwo p_{bin} v2 $(-,-)$	100.0	0.0	0.0	0.0
Apriori χ^2 $(0.01,-)$	100.0	0.0	31.7	0.0
Apriori MI $(0.01,-)$	100.0	0.0	31.8	0.0
Apriori cfa $(0.01,-)$	100.0	0.0	3.4	0.0
Apriori J $(0.01,-)$	100.0	0.0	39.6	0.0

Apriori with the χ^2-measure produced quite many redundant rules. Due to the minimum frequency threshold, it produced more frequent rules than Kingfisher with the same measure. The average lift was also remarkably

Table 5.9: Results of the cross validation in set T40I10D100K. Average rule length, frequency, confidence, lift, leverage, and root mean squared errors of lift and leverage.

method	*len*	*fr*	*cf*	γ	δ	$\sqrt{MSE_\gamma}$	$\sqrt{MSE_\delta}$
Kf p_F	3.9	0.0150	0.96	36.0	0.0145	2.2	0.00072
Kf χ^2	4.8	0.0012	0.99	1632.6	0.0012	846.0	0.00019
Kf χ^2 cc	4.8	0.0019	0.99	703.4	0.0019	331.0	0.00023
Chitwo z	4.8	0.0011	0.99	1721.6	0.0011	860.0	0.00019
Chitwo z cc	4.8	0.0011	0.99	1679.3	0.0011	849.0	0.00019
Chitwo p_{bin} v1	3.9	0.0129	0.97	49.6	0.0127	3.1	0.00094
Chitwo p_{bin} v2	3.8	0.0130	0.96	48.9	0.0128	3.1	0.00094
Apriori χ^2	4.4	0.0104	0.98	70.3	0.0103	2.6	0.00052
Apriori *MI*	4.0	0.0151	0.97	35.1	0.0146	2.2	0.00071
Apriori *cfa*	5.7	0.0107	0.99	15.2	0.0097	0.6	0.00050
Apriori *J*	4.2	0.0156	0.97	30.8	0.0151	0.9	0.00044

smaller, but held more accurately in the test data. Both *MI* and *J* measures produced also many redundant rules. The rules were mostly (especially with *MI*) the same as with p_F or their redundant specializations. With all these measures, nearly all of the selected rules were related to just two consequences, but for some reason the *J*-measure favoured the more frequent consequence with a smaller lift value.

Certainty factor produced only a few clearly redundant rules. All rules were related to just one consequence. The rules were the most complex among all tested methods, and the average lift was also the smallest, but anyway expressed a clear dependency.

The accuracy of the lift and leverage was at least reasonable for all tested methods.

5.2.5 Accidents

Results for set Accidents are given in Tables 5.10 and 5.11. For Kingfisher with p_F no minimum frequency thresholds (except the absolute minimum frequency 5, i.e. fr=0.000022) were used. For Kingfisher with the χ^2-measure, we used min_{fr}=0.0001 (absolute minimum frequency 23), when all rules were searched for, and min_{fr}=0.0005 (absolute minimum frequency 114), when only positive rules were searched for. The latter task required a more restrictive minimum frequency threshold, because most of the best rules were negative, and therefore the search for the 100 best non-redundant positive rules continued further. For Chitwo, we used the same

min_{fr}=0.0005 along with the maximal rule length 6. The latter restriction was not likely to prune out many (if any) rules, because Kingfisher did not find any rules after level 6. For Apriori, we had to use a large minimum frequency threshold (min_{fr}=0.25, corresponding to absolute minimum frequency 56697) and also the maximum rule length 6. The reason is that Accidents is a dense data set, where the number of frequent sets is prohibitive even with modest min_{fr} values.

Kingfisher found relatively many negative rules with both measures. Once again, the χ^2-measure selected only the simplest negative rules (2-rules), while p_F produced also more complex negative rules. The reason is that the simplest negative rules described (nearly) mutually excluding events with $P(X\neg A) = P(X) \approx P(\neg A)$, thus achieving a good χ^2-value, while the more complex rules had $P(X\neg A) \leq P(X) < P(\neg A)$, and, respectively, lower χ^2-values. Generally, p_F produced low but stable lift and slightly larger leverage than the χ^2-measure, which suffered for unstable lift values.

In some learning sets, Kingfisher with the χ^2-measure found an independence rule. The rule was the same in all sets, $A17 \rightarrow \neg A293$. The problem of the rule was that it was too frequent and occurred on nearly all rows of the data. Therefore, it had minimal lift ($\gamma = 1.0001$) already in the learning set. In some of the test sets, the rule occurred on all rows (achieving $\gamma = 1.0$), and in others, it was missing only from a couple of rows.

Chitwo with the z-score produced less frequent rules than Kingfisher with the χ^2-measure (when only positive rules were searched for), but otherwise the quality of rules was similar. The continuity correction had no effect on the results. Surprisingly, the binomial probability selected in average less frequent rules, expressing weaker dependence than the z-score. The reason is that most rules by the z-score were variations of a couple of simple, relatively frequent rules, and themselves also frequent. On the other hand, the binomial probability ranked high the generalizations of these variations (2-rules), and the specializations were considered redundant and not selected.

Apriori with measures χ^2, MI, and J found very frequent rules ($fr > 0.5$), with low lift but large leverage. Most rules were redundant specializations of just two rules, which were among the best rules by p_F (when only positive rules were searched for). However, in this data set the three measures revealed a different bias. Because the minimum frequency threshold was large, the χ^2-measure did not suffer for exaggerated values related to low frequency rules. Instead, if produced very similar

Table 5.10: Results of the cross validation in set Accidents. Method, parameters and numbers of applicable rules, negative rules, clearly redundant, and harmful rules. For harmful rules, the average numbers of negative dependence and independence rules are given in the parentheses.

method (min_{fr}, max_{len})	appl	neg	red	harmful
Kf p_F $(-,-)$	100.0	42.8	0.0	0.0
Kf p_F pos $(-,-)$	100.0	0.0	0.0	0.0
Kf χ^2 $(0.0001,-)$	100.0	29.2	0.0	0.6 (0.0+0.6)
Kf χ^2 cc $(0.0001,-)$	100.0	29.2	0.0	0.6 (0.0+0.6)
Kf χ^2 pos $(0.0005,-)$	100.0	0.0	0.0	0.0
Chitwo z $(0.0005, 6)$	100.0	0.0	0.0	0.0
Chitwo z cc $(0.0005, 6)$	100.0	0.0	0.0	0.0
Chitwo p_{bin} v1 $(0.0005, 6)$	100.0	0.0	0.0	0.0
Chitwo p_{bin} v2 $(0.0005, 6)$	100.0	0.0	0.0	0.0
Apriori χ^2 $(0.25, 6)$	100.0	0.0	85.0	0.0
Apriori MI $(0.25, 6)$	100.0	0.0	84.4	0.0
Apriori cfa $(0.25, 6)$	100.0	0.0	44.5	83.8 (76.4+7.4)
Apriori J $(0.25, 6)$	100.0	0.0	90.7	0.0

Table 5.11: Results of the cross validation in set Accidents. Average rule length, frequency, confidence, lift, leverage, and root mean squared errors of lift and leverage.

method	len	fr	cf	γ	δ	$\sqrt{MSE_\gamma}$	$\sqrt{MSE_\delta}$
Kf p_F	3.7	0.3495	0.94	3.4	0.1254	0.0	0.00093
Kf p_F pos	3.9	0.2399	0.87	3.8	0.1113	0.0	0.00090
Kf χ^2	3.3	0.2210	0.92	228.5	0.0908	664.0	0.00082
Kf χ^2 cc	3.3	0.2216	0.92	221.1	0.0913	664.0	0.00083
Kf χ^2 pos	3.8	0.1568	0.88	117.0	0.0841	23.7	0.00083
Chitwo z	3.9	0.0983	0.88	121.7	0.0775	24.0	0.00086
Chitwo z cc	3.9	0.0983	0.88	121.7	0.0775	24.0	0.00086
Chitwo p_{bin} v1	3.1	0.0378	0.69	124.7	0.0298	20.1	0.00052
Chitwo p_{bin} v2	2.9	0.0689	0.59	8.4	0.0500	0.2	0.00068
Apriori χ^2	4.6	0.6226	0.92	1.3	0.1297	0.0	0.00075
Apriori MI	4.6	0.6135	0.94	1.3	0.1290	0.0	0.00075
Apriori cfa	5.6	0.2813	1.00	1.0	0.0001	0.0	0.00002
Apriori J	4.6	0.5610	0.97	1.3	0.1197	0.0	0.00073

rules with MI, specializations of rules $A8 \rightarrow A24$ (cf=0.99, $\delta = 0.13$) and $A15A25 \rightarrow A14$ (cf=0.88, $\delta = 0.14$). The J-measure favoured also specializations of $A8 \rightarrow A24$, but in addition, it produced specializations

of $A64 \rightarrow A46$ (cf=0.91, $\delta = 0.12$), which were not produced by the χ^2-measure and MI. The reason is that the J-measure tends to favour large confidence, while the χ^2-measure and MI prefer large leverage.

Certainty factor produced the least frequent but quite complex rules, because cf=1.0 occurred rarely with more frequent and simpler rules. The strength of the dependencies was extremely weak ($\gamma \approx 1.0001$), and 84% of rules were harmful. Most (76%) of the harmful rules were independence rules, but about 7% of them expressed a negative dependency in the test sets.

5.2.6 Pumsb

Results for set Pumsb are given in Tables 5.12 and 5.13. For Kingfisher with p_F no minimum frequency thresholds (except the absolute minimum frequency 5, i.e. fr=0.00015) were used, but with the χ^2-measure, we had to use min_{fr}=0.0005 (absolute minimum frequency threshold 16). For Chitwo, we used min_{fr}=0.001 (absolute minimum frequency 33) and maximum rule length 5. For Apriori, an exceptionally large minimum frequency threshold was required (min_{fr}=0.45, i.e. absolute minimum frequency 14714) along with the maximal rule length 5. The problem is that Pumsb is too dense and high-dimensional data set for frequency-based pruning.

Kingfisher with p_F found a large number of negative rules (over 60%). None of the negative rules had any recognizable type which could be a side effect of the binarization. All negative rules had large frequency (about fr=0.5) and a large confidence (cf=0.99–1.00). Kingfisher with the χ^2-measure found much less negative rules, and all of them looked like a side effect of the binarization (e.g. $A4430 \rightarrow \neg A4431$), describing mutually excluding attributes. Because most of the negative rules by p_F were relatively complex, the average rule length by χ^2 was smaller. Generally, both measures found strong and frequent rules with large leverage. With p_F, the lift and leverage held accurately in the test sets, but with the χ^2-measure, the lift was unstable. The continuity correction had a clear effect on the quality of rules.

Chitwo found much rarer but still strong rules. The lift was the largest among all tested methods and held quite well in the test sets. The continuity correction had no effect on the z-score, but the binomial probability selected more frequent rules with smaller but more accurate lift. The difference between the two selection strategies was clear, and once again, version 2 produced more frequent rules with smaller lift but larger leverage. The average confidence was remarkably low compared to other tested methods.

Apriori with measures χ^2, MI, and J found a large number of clearly

Table 5.12: Results of the cross validation in set Pumsb. Method, parameters and numbers of applicable rules, negative rules, clearly redundant, and harmful rules. For harmful rules, the average numbers of negative dependence and independence rules are given in the parentheses.

method (min_{fr}, max_{len})	appl	neg	red	harmful
Kf p_F $(-,-)$	100.0	63.3	0.0	0.0
Kf p_F pos $(-,-)$	100.0	0.0	0.0	0.0
Kf χ^2 $(0.0005,-)$	100.0	18.8	0.0	0.0
Kf χ^2 cc $(0.0005,-)$	100.0	18.0	0.0	0.0
Kf χ^2 pos $(0.0005,-)$	100.0	0.0	0.0	0.0
Chitwo z $(0.001,5)$	100.0	0.0	0.0	0.0
Chitwo z cc $(0.001,5)$	100.0	0.0	0.0	0.0
Chitwo p_{bin} v1 $(0.001,5)$	100.0	0.0	0.0	0.0
Chitwo p_{bin} v2 $(0.001,5)$	100.0	0.0	0.0	0.0
Apriori χ^2 $(0.45,5)$	100.0	0.0	55.2	0.0
Apriori MI $(0.45,5)$	100.0	0.0	73.0	0.0
Apriori cfa $(0.45,5)$	100.0	0.0	0.0	0.0
Apriori J $(0.45,5)$	100.0	0.0	95.5	0.0

Table 5.13: Results of the cross validation in set Pumsb. Average rule length, frequency, confidence, lift, leverage, and root mean squared errors of lift and leverage.

method	len	fr	cf	γ	δ	$\sqrt{MSE_\gamma}$	$\sqrt{MSE_\delta}$
Kf p_F	3.3	0.4876	1.00	2.1	0.2462	0.0	0.00038
Kf p_F pos	3.5	0.4491	0.99	2.3	0.2369	0.0	0.00074
Kf χ^2	2.8	0.3379	1.00	132.6	0.1173	132.0	0.00136
Kf χ^2 cc	2.9	0.3700	1.00	21.2	0.1384	7.4	0.00140
Kf χ^2 pos	3.8	0.2628	1.00	27.6	0.1372	8.4	0.00153
Chitwo z	3.9	0.0138	0.99	145.1	0.0136	45.6	0.00095
Chitwo z cc	3.9	0.0138	0.99	145.1	0.0136	45.6	0.00095
Chitwo p_{bin} v1	4.1	0.0163	0.99	69.6	0.0160	9.6	0.00104
Chitwo p_{bin} v2	3.6	0.0324	0.66	21.3	0.0303	1.3	0.00211
Apriori χ^2	4.1	0.7796	1.00	1.3	0.1512	0.0	0.00258
Apriori MI	3.6	0.5172	1.00	1.9	0.2484	0.0	0.00028
Apriori cfa	4.8	0.5640	1.00	1.2	0.0701	0.0	0.00125
Apriori J	4.3	0.4530	1.00	2.2	0.2470	0.0	0.00037

redundant rules. With the J-measure, nearly all rules were redundant specializations of just one rule, which was also one of the best rules by p_F. With MI, the rules were more diverse, but many were redundant

specializations of the best rules by p_F. Once again, *cfa* produced the lowest average lift, but due to the large minimum frequency threshold, the rules were very frequent and the lift values held well in the test sets.

5.2.7 Retail

Results for set Retail are given in Tables 5.14 and 5.15. For Kingfisher with p_F and Apriori no minimum frequency thresholds were needed (except the absolute minimum frequency 5, i.e. fr=0.000085). For Kingfisher with the χ^2-measure and Chitwo we had to use min_{fr}=0.0005 (absolute minimum frequency 29). No restrictions on the rule length were used.

Both Kingfisher and Chitwo found only 2-rules. The results were quite typical; Kingfisher with p_F found more frequent rules with larger leverage, while the χ^2-measure produced less frequent rules with larger lift. The continuity correction had only a marginal effect.

Chitwo with the z-score produced slightly less frequent rules with smaller leverage. The continuity correction had no effect with the z-score, but the binomial probability (especially version 2) clearly improved the quality of rules. Some of the rules were the same as with p_F, but in a different order.

Table 5.14: Results of the cross validation in set Retail. Method, parameters and numbers of applicable rules, negative rules, clearly redundant, and harmful rules. For harmful rules, the average numbers of negative dependence and independence rules are given in the parentheses.

method (min_{fr}, max_{len})	appl	neg	red	harmful
Kf p_F $(-,-)$	100.0	0.0	0.0	0.0
Kf χ^2 $(0.0005,-)$	100.0	0.0	0.0	0.0
Kf χ^2 cc $(0.0005,-)$	100.0	0.0	0.0	0.0
Chitwo z $(0.0005,-)$	100.0	0.0	0.0	0.0
Chitwo z cc $(0.0005,-)$	100.0	0.0	0.0	0.0
Chitwo p_{bin} v1 $(0.0005,-)$	100.0	0.0	0.0	0.0
Chitwo p_{bin} v2 $(0.0005,-)$	100.0	0.0	0.0	0.0
Apriori χ^2 $(-,-)$	71.6	0.0	51.7	5.1 (3.8+1.3)
Apriori MI $(-,-)$	100.0	0.0	51.9	0.0
Apriori *cfa* $(-,-)$	63.4	0.0	0.6	4.6 (4.6+0.0)
Apriori J $(-,-)$	100.0	0.0	53.7	0.0

Apriori with measures χ^2, MI, and J produced a large number (over 50%) of redundant rules. The data set was especially problematic for the χ^2-measure, which selected many rare rules. As a result, only 72% of them could be applied to the test sets, and many turned out to be harmful. The

Table 5.15: Results of the cross validation in set Retail. Average rule length, frequency, confidence, lift, leverage, and root mean squared errors of lift and leverage.

method	*len*	*fr*	*cf*	γ	δ	$\sqrt{MSE_\gamma}$	$\sqrt{MSE_\delta}$
Kf p_F	2.0	0.0096	0.61	140.1	0.0042	37.1	0.00046
Kf χ^2	2.0	0.0019	0.57	288.7	0.0017	71.1	0.00029
Kf χ^2 cc	2.0	0.0012	0.57	287.7	0.0018	71.0	0.00029
Chitwo z	2.0	0.0013	0.56	292.0	0.0013	71.5	0.00025
Chitwo z cc	2.0	0.0013	0.56	292.0	0.0013	71.5	0.00025
Chitwo p_{bin} v1	2.0	0.0015	0.50	191.8	0.0015	45.0	0.00028
Chitwo p_{bin} v2	2.0	0.0029	0.53	174.2	0.0021	42.3	0.00035
Apriori χ^2	4.0	0.0003	0.94	5553.4	0.0003	8700.0	0.00013
Apriori *MI*	2.6	0.0196	0.63	52.6	0.0096	12.2	0.00066
Apriori *cfa*	8.2	0.0001	1.00	577.4	0.0001	354.0	0.00005
Apriori *J*	2.7	0.0172	0.68	56.5	0.0089	12.6	0.00062

harmful rules were mostly independence rules, with extremely large lift values ($\gamma = 5600 - 6500$), but only small frequency (absolute frequency 6) in the learning sets. In the test sets the related consequent attributes did not occur at all. Other harmful rules had also large lift ($\gamma = 3000 - 8000$) and small frequency (absolute frequency 5–9) in the learning sets, but zero frequency in the test sets. Generally, the average lift was very large, but also unstable (held poorly in the test sets).

In this data set, Apriori with measures *MI* and *J* produced in average more frequent rules with larger leverage than p_F. The reason is that the selected rules included the best rules by p_F, but also their redundant specializations. In addition, measure *MI* listed the best 2-rules twice, in forms $A \to B$ and $B \to A$. Because all rules by p_F were 2-rules, the number of these redundant rules was large (about 25% of rules, not included into the number of clearly redundant rules).

Certainty factor favoured even more infrequent rules than the χ^2-measure. The reason is that in Retail, strong rules with cf=1.0 were complex and infrequent. Only 63% of the rules could be applied to the test sets and some of them were harmful. The harmful rules had often at least moderate lift ($\gamma = 1.7 - 1200$) in the learning set. The frequency in the learning sets was low (absolute frequency 5–9) and zero in the test sets. Generally, the rules were very complex (average length 8.2) with respect to the average transaction length (10.3).

5.2.8 Summary

The summarized results of the quality evaluation are represented in Table 5.16. The general strength of the dependencies is characterized by averages of the average lift and leverage values in all data sets. The average lift values should be interpreted with caution, because some data sets (T10I4D100K, T40I10D100K, and Retail) produced extremely large lift values with the χ^2-measure and z-score, which dominate the average values. The accuracy of the results (how well the dependencies held in the test sets) is characterized by average values of the relative errors $\frac{\sqrt{MSE_\gamma}}{\gamma}$ and $\frac{\sqrt{MSE_\delta}}{\delta}$. Due to scaling, the errors of different methods are comparable.

Table 5.16: Summarized results of the quality evaluation. γ= average lift, δ= average leverage, err_γ=average $\frac{\sqrt{MSE_\gamma}}{\gamma}$, err_δ= average $\frac{\sqrt{MSE_\delta}}{\delta}$.

Method	γ	δ	err_γ	err_δ
Kf p_F	20.0	0.105	0.070	0.044
Kf p_F pos	20.0	0.099	0.077	0.046
Kf χ^2	41.2	0.042	1.222	0.106
Kf χ^2 cc	41.1	0.055	0.720	0.088
Kf χ^2 pos	41.1	0.046	0.301	0.104
Chitwo z	41.7	0.014	0.684	0.259
Chitwo z cc	41.7	0.014	0.685	0.249
Chitwo p_{bin} v1	27.4	0.035	0.181	0.097
Chitwo p_{bin} v2	24.9	0.047	0.100	0.078
Apriori χ^2	793.3	0.079	0.517	0.126
Apriori MI	7.5	0.114	0.067	0.037
Apriori cfa	82.5	0.014	0.174	0.246
Apriori J	8.0	0.113	0.061	0.034

Let us first analyze the methods which searched for non-redundant rules. Among these methods, Kingfisher with p_F produced in average the largest leverage but smallest lift. Both the lift and leverage had the smallest relative errors among all methods. Kingfisher with the χ^2-measure produced the second best leverage (together with Chitwo p_{bin} v2), but the relative error was larger than with p_F. The continuity correction clearly improved the rules, by producing larger and more reliable leverage. The average lift was the largest among all methods together with Chitwo using the z-score. However, the relative error was extremely large (122% from the average lift), when the continuity correction was not used. The continuity correction improved the accuracy, without decreasing the average lift, but still

the error was large. Based on this, we can conclude that the continuity correction should always be used with the χ^2-measure.

When only positive rules were searched for, both p_F and χ^2 (with the continuity correction) produced smaller and somewhat less accurate leverage. The reason is that the average frequency of positive rules was smaller than the frequency of all rules. Since the leverage is bounded by the frequency, the most frequent rules can also produce larger leverage. The average lift values were not affected, but with p_F, the relative error was slightly larger, while with the χ^2-measure, it was clearly smaller. The latter was due to just one data set, Accidents, where including negative rules produced an extremely large lift error.

We conclude that Kingfisher with p_F is clearly preferable to the χ^2-measure, when the objective is to find the most significant rules with a variable-based measure. It also guarantees globally optimal results, because no minimum frequency thresholds or restrictions on the rule length are needed.

Chitwo with the z-score produced in average the smallest leverage but largest lift among all methods. The relative errors were larger than with the χ^2-measure, when only positive rules were searched for using the continuity correction. For the z-score, the continuity correction had only a marginal effect (slightly improved the accuracy of the leverage). The main reason for the inaccurate lift values were data sets Mushroom and T10I4D100K, where the z-score (like the χ^2-measure without the continuity correction) selected very rare rules (occurring on less than 30 rows of data). Therefore, we suggest that the z-score alone should not be used without a special handling of low frequency rules. The selection by binomial probabilities turned out to be a good solution to this problem. Both versions produced larger and significantly more accurate leverage, comparable to the χ^2-measure with the continuity correction. The average lift values were smaller than with the continuity corrected χ^2, but also more accurate. As expected, version 2 produced more accurate results than version 1.

If a robust value-based method is desired, then the best candidates are the continuity corrected χ^2-measure and the z-score with p_{bin} v2. If the χ^2-measure is used in the value-based semantics, then it should be checked that the goodness of rule $X \rightarrow A = a$ is due to high lift between X and $A = a$, and not between $\neg X$ and $A \neq a$. Both methods offer a heuristic solution to the problem of exaggerated values (overestimated significance), but they do not guarantee the globally optimal results measured by p-values. For this purpose, an efficient method should be invented for pruning the search space using the binomial probability alone.

Comparing the results by Apriori is more difficult, due to a large number of redundant rules. In the worst case all rules were redundant specializations of just one good rule, and the results were misleadingly good. Keeping this in mind, the results by Apriori can be compared to each other, because all Apriori experiments were executed with the same minimum frequency thresholds and maximal rule lengths, and none of them pruned out redundant rules.

Measures *MI* and *J* produced the most accurate results, with respect to both lift and leverage. The leverage was the largest among all Apriori methods, but the lift was respectively the smallest. The best rules were also mostly the same, except in T40I10D100K, Pumsbs, and – to some extent – in Accidents. All these data sets are dense and a likely explanation is that *MI* selected also rules with large $\gamma(\neg X, \neg A)$, while the *J*-measure favoured always rules with large $\gamma(X, A)$. Chess is also a dense data set, but in Chess both measures produced only redundant specializations of just one rule. Generally, the best rules by *MI* were often redundant specializations of the best rule or rules by p_F (when only positive rules were searched for), and we assume that it can approximate p_F quite well.

The χ^2-measure produced in average the largest lift among all methods (including Kingfisher and Chitwo). This was due to sets T10I4D100K and Retail, where no minimum frequency thresholds were needed, and extremely rare rules could be selected. Kingfisher with the χ^2-measure succeeded better in these data sets, even without continuity correction, because redundant (over-specific and often unreliable) rules were pruned. An interesting observation is that the χ^2-measure and *MI* produced very different rules in all data sets, except Chess and Accidents. In the previous research it has been found that both measures give often similar results, when classification rules are searched [73], although exceptions are known [4].

Certainty factor was the most problematic measure, because all data sets contained a plenty of rules with cf=1.0. Because *cfa* gets its maximal value, when ever cf=1.0, it was not able to rank the best rules. As a result, the dependencies were often extremely weak (especially in Chess, Accidents, and Pumsb). A large number of rules were harmful, expressing independence or negative dependence in the test sets. Still, the average lift was quite high, due to high lift values in Retail.

Finally, we note that in spite of the large memory size, Apriori required so large minimum frequency thresholds with most of the data sets that the globally optimal dependency rules could not be discovered, even in principle. From the 100 best non-redundant rules with Kingfisher and the

Table 5.17: Parameters of the efficiency comparison: max_M is the initial value for the threshold $\ln(max_p)$ and fr is the corresponding implicit minimum frequency, min_{fr} is the minimum frequency threshold.

	Kingfisher p_F		Kf χ^2	Chitwo z
set	max_M	fr	min_{fr}	min_{fr}
Mushroom	-2000	0.06745	–	–
Chess	-850	0.07541	–	–
T10I4D100K	-1600	0.00227	–	–
T40I10D100K	-3500	0.00569	–	–
Accidents	-73000	0.05557	0.0001	0.0005
Pumsb	-18000	0.12011	0.0005	0.001
Retail	-350	0.00047	0.0005	0.0005

χ^2-measure (when only positive rules were searched for), only 1% in Chess, 0% in T40I10D100K, 14% in Accidents, and 27% in Pumsb had so large frequency that they could have been found by Apriori.

5.3 Efficiency evaluation

In the efficiency evaluation, we compared Kingfisher with p_F and the χ^2-measure and Chitwo with the z-score, when the best 100 rules were searched for from the whole data sets. The continuity correction was used with the χ^2-measure and z-score. In addition, we tested the effect of the Lapis Philosophorum principle in Kingfisher with p_F.

The parameters for the tests are given in Table 5.17. For Kingfisher with p_F, no minimum frequency thresholds (other than the absolute minimum frequency 5) or restrictions on the rule length were needed, but thresholds $\ln(max_p)$ and the corresponding (implicitly defined) minimum frequencies are given in the Table. When the Lapis Philosophorum principle was not used, the program often got stuck, and was halted after 20 minutes CPU time. For Kingfisher with the χ^2-measure and Chitwo with the z-score, minimum frequency thresholds were used with the most demanding data sets. In addition, the maximal rule length was set to 7 for Kingfisher with the χ^2-measure in set Pumsb and 6 for Chitwo with the z-score in sets Accidents and Pumsb. The thresholds were set such that the executions could be finished in 20 minutes CPU time.

The results for Kingfisher with p_F with or without the Lapis Philosophorum principle are represented in Table 5.18. The first columns characterize the size of the traversed search space in the terms of the enumeration tree, which was generated: the last level, the widest level, and the number of

Table 5.18: Efficiency comparison of Kingfisher with p_F with and without the Lapis Philosophorum principle. l=last level, w=widest level, $wsize$= size of the widest level, and t=execution time in seconds.

	Kingfisher p_F				without LP			
set	l	w	$wsize$	t	l	w	$wsize$	t
Mushroom	6	3	454	0	7	3	959	0
Chess	15	8	2309751	169	≥10	≥10	≥36398778	≥ 1200
T10I4D100K	4	2	1399	11	8	2	3129	12
T40I10D100K	6	2	9098	17	18	7	91112	92
Accidents	13	6	142133	79	≥7	≥7	≥3448942	≥1200
Pumsb	11	6	30009	7	≥7	≥7	≥47105062	≥1200
Retail	5	2	6387	379	6	2	9208	389

sets on the widest level. We recall that the enumeration tree is pruned after each level, and thus the widest level is the bottle-neck. The last column gives the execution time in seconds.

With the original Kingfisher, the whole search space could be traversed, and therefore the discovered rules were globally optimal. The most demanding data set was Retail, where the number of attributes is extremely large. In addition, all dependencies are relatively weak, and therefore Kingfisher could not determine any effective minimum frequency thresholds from the maximal p_F-value requirement. Most of the execution time was spent on level 2, where the program had to determine lower bounds for over 17 million attribute combinations.

The implicit minimum frequency thresholds explain the efficiency of Kingfisher only on the first levels. After that the Lapis Philosophorum principle begins to play a more important role. When the principle was not used, the program got stuck with data sets Chess, Accidents, and Pumsb. The sparsest data sets could be handled without Lapis Philosophorum, but the enumeration tree was still significantly larger. In the densest data sets, the widest level was at least 15–1570 times as large as with the Lapis Philosophorum principle, and none of these experiments could have been run in an ordinary desktop computer.

The results for Kingfisher with the χ^2-measure and Chitwo with the z-score are represented in Table 5.19. Both methods were considerably slower than Kingfisher with p_F and nearly always the search continued deeper. The reason is that both asymptotic measures suffer for exaggerated values (overestimated significance), when the frequencies are low, and therefore the search space cannot be pruned efficiently with the upper bounds. Efficient pruning happens only, if good minimal rules are found on the first

Table 5.19: Efficiency comparison of Kingfisher with the χ^2-measure and Chitwo with the z-score. l=last level, w=widest level, $wsize$= size of the widest level, and t=execution time in seconds.

	Kingfisher χ^2				Chitwo z			
set	l	w	$wsize$	t	l	w	$wsize$	t
Mushroom	8	4	1888	0	4	3	5373	0
Chess	19	10	4638560	914	19	10	6161622	1029
T10I4D100K	4	2	604	23	4	2	43659	30
T40I10D100K	7	4	2413	28	7	2	349707	196
Accidents	25	7	1372901	1038	$\geq$6	$\geq$ 6	$\geq$3485432	$\geq$933
Pumsb	$\geq$7	$\geq$7	$\geq$10962956	$\geq$506	$\geq$6	$\geq$6	$\geq$16930559	$\geq$898
Retail	5	2	7598	631	6	2	8424	406

levels, and their specifications can be pruned as redundant.

The most difficult data sets for the χ^2-measure and z-score were Chess, Accidents, and Pumsb, which were also the densest data sets. In Pumsb, Kingfisher with χ^2 got stuck on level 8 and Chitwo with z on level 7, unless maximal rule lengths were used. Generally, the χ^2-measure managed to prune the search space better than the z-score. The reason is that the χ^2-measure gets its maximal value whenever $P(XA) = P(X) = P(A)$, while the z-score favours the least frequent rules among them.

Chapter 6

Conclusions

When you think it's all over, it's only begun.

W. Nelson

Analyzing dependencies is an never-ending problem. People have always wanted and will want to find and understand dependencies and regularities between things. In earlier times, the data sets were so small that this could be accomplished with a pen and paper or simple statistical tools. Nowadays, the amount of data is increasing all the time; in a computerized world one can easily collect and store ever larger and higher-dimensional data sets on nearly everything. All this available data is a precious source of information, which could explain the causes and cures of diseases, the behaviour of our vulnerable ecosphere and its responses to human actions, psychological laws of human nature and learning ... The applications are countless, if one could just analyze the data and find the most significant dependencies. The problem is that the tools have not developed with the needs and resources. This is exactly where this research makes its modest contribution.

In this research, we have concentrated on a commonly occurring special case of dependencies, the dependency rules between sets of positive-valued binary attributes and single binary attributes. Such rules as $X \rightarrow A$ or $X \rightarrow \neg A$ already reveal a lot of information on the predominant dependencies in natural occurrence data (like occurrences of gene alleles and diseases, market basket data, or species occurrence data) or otherwise binarized data. Still, the problem is computationally challenging, because the number of possible attribute combinations increases exponentially with the number of available attributes.

Due to the exponential size of the search space, it has often been considered impossible (or extremely inefficient) to search for all sufficiently

significant dependencies or even the best statistical dependencies in large data sets. Instead, traditional data mining has offered a heuristic solution, association rules, which are insufficient to capture statistically important dependencies. In the worst case, association rules are just frequently occurring combinations of attributes, which do not express any statistical dependencies, or spurious dependencies, which just happen to hold in the given data by chance. As such, they do not fit the needs of other sciences, where it has been demanded (e.g. [54, 55]) that computer scientists should develop efficient search methods for genuine statistical dependencies, which hold also in future data.

The problem is the same as with a gold miner, who wants to find gold in a mountain. She or he is not satisfied to get a tonne of rocks and gravel from the surface, not even if it contains the largest rocks or minerals which occur in most rocks. What the miner wants is rich ore from the pay dirt.

Quite recently, this has been admitted also among data miners, and the question has arisen how to guarantee the statistical soundness of data mining results. Still, the solutions have concentrated on evaluating the statistical significance of results in the post-processing phase, instead of searching for the statistically most significant patterns. Of course, these two approaches, searching for the most significant patterns and evaluating the significance of results afterwards, are not mutually excluding. The point is that it is useless the evaluate the significance of frequent patterns, when the most significant patterns are still undiscovered. The gold miner is not satisfied to know that the mined ore contained one percent of gold, if the mountain has veins with 10% or 50% of gold. Therefore, we have adopted the strategy of surgical strikes; we determine where the gold is to be found and mine those parts of the mountain – no deeper than needed and not in the wrong places.

The efficiency of the new mining algorithm is based on three insights. First, we realized that all proper measures for the statistical significance of dependencies behave similarly and meet their best possible values at the same points of the search space. This insight was formalized in the theory of well-behaving goodness measures. The definition of well-behaving goodness measures is consistent with the classical axioms for the proper measures of association rules. The points where the upper or lower bounds of well-behaving measures are met were also previously known for some goodness measures, but the new discovery was that the well-behaving measures do not have to be convex or concave functions, but weaker conditions are sufficient.

The second insight was that it is sufficient and even desirable to search

for only non-redundant dependency rules. The users want to know only the real causes of the dependence – not all randomly occurring extra attributes, which could be added without weakening the dependence or decreasing its significance. In fact, the redundant extra attributes can be harmful for the rule, because the rule becomes over-fitted and less likely to hold in future data. In practice, this means that the search does not have to continue any deeper, if the more complex rules would be inferior to already discovered, more general rules.

This insight does not necessarily mean that we do not have to mine as deep as when redundant rules are accepted. If for some consequent $A_i = a_i$, it is possible to find better non-redundant rules than those already found, then the search continues deeper in that area. The point is that the depth of the search depends on whether it is possible to find any better rules in the deeper levels. In the enumeration problem, where all sufficiently significant dependency rules are searched for, the requirement for non-redundancy can radically cut the depth of the search. Therefore, the new algorithms are able to enumerate all sufficiently good rules without any minimum frequency thresholds or maximal rule lengths. On the other hand, if the task is to search for the K best rules, then it is usually more laborious to find the best K non-redundant rules than merely the K best rules. It is always easier to fill a pan with one gold nugget and gravel under it than with pure gold nuggets. If only the K best (redundant) rules are searched for, then the result set can contain just one good non-redundant rule and all the other $K-1$ rules are its redundant specializations. According to our experiments, this is quite a likely scenario, if no counter-measures are used. In this case, the search for the best non-redundant rules can continue much deeper in some parts of the data, while other parts can be left unexplored. Once again, the point is the surgical strike – we search deep only when we have to.

The third insight is an algorithmic invention, discovered when the predominant ideas of the traditional frequency-based pruning algorithms were forgotten. It is the Lapis Philosophorum principle, a simple idea how the enumeration tree can be pruned, not only in the child nodes of a checked node, but also in its parents and their children. This principle turned out to be necessary for the efficient search without any minimum frequency thresholds or maximal rule lengths. Without it, the first two insights would enable a complete search (guaranteeing globally optimal results) only for moderate data sets.

The most important practical lesson of this research was that the exact tests (estimating the exact p-value of the discovery) are not necessarily

more laborious than the corresponding asymptotic tests. Traditionally, Fisher's exact test has been preferred but considered too heavy in large data sets. Therefore, by default, data miners have concentrated on the χ^2-test, which can approximate the exact test under certain circumstances. However, our experiments showed that the search with Fisher's p_F is in fact more efficient than with the χ^2-measure and, furthermore, the χ^2-measure is prone to produce unstable results. Fisher himself would have been happy with this result, because it means that the exact test can and should be used always when a variable-based goodness-measure is needed.

The value-based measures are more problematic, partially because they are lacking sound statistical foundations. Still, we must admit that sometimes the value-based semantics is relevant; a miner may be interested in that gold and mineral X are exceptionally likely to occur together, even if mineral X is rare. For the problem of searching for the best non-redundant dependency rules with a value-based measure we have only a partially satisfactory solution. The problem is that we have not been able to derive tight lower bounds for the corresponding binomial probability. Therefore, the search space has to be restricted with the corresponding asymptotic measure (the z-score), which is too inefficient in really large or dense data sets. The most straightforward solution would be to invent the required lower bounds for the binomial probability. Heuristic solutions learnt from the research of association rules (like pruning out the most frequent or infrequent attributes) could also be useful, if applied with care.

Finally, we note that in this research we have taken a very careful attitude against losing significant rules, and pruned out only clearly redundant rules. In reality (see Figure 1.1), some dependency rules may appear as non-redundant only in the given data, but in future data they would be redundant. Clearly, they could be pruned and thus the efficiency improved, but first one should develop the statistical theory for the significance of improvement, as discussed in Section 2.4. The new statistical theory can also require new algorithmic insights (similar to Lapis Philosophorum) for efficient pruning.

Another important challenge for future research is too develop an efficient algorithm for searching for general dependency rules (containing an arbitrary number of negations). The problem of the optimal binarization of multi-valued or numeric data is also complex, but possibly it could be solved simultaneously, when the most significant dependencies are searched for. As we can see, this research has just opened new promising paths, but the future still has several important challenges waiting for their solutions.

Chapter A

Useful mathematical results

The following well-known results are used in the proofs. We recall that in this thesis we assume that $0 \in \mathbb{N}$.

Vandermonde identity

For all $n, m, l \in \mathbb{N}$

$$\sum_{i=0}^{\max\{m,l\}} \binom{n}{i} \binom{n-l}{m-i} = \binom{n}{m}. \tag{A.1}$$

Pascal's rule

For all $n, m \in \mathbb{N}$

$$\binom{n}{m} + \binom{n}{m+1} = \binom{n+1}{m+1} \tag{A.2}$$

Multinomial theorem

For all $n, m_1, \ldots, m_l \in \mathbb{N}$, $l \geq 1$, and any numbers x_i holds

$$\sum_{m_1+\ldots+m_l=n} \binom{n}{m_1, \ldots, m_l} x_1^{m_1} \ldots x_l^{m_l} = (x_1 + \ldots + x_l)^n, \tag{A.3}$$

where

$$\binom{n}{m_1, \ldots, m_l} = \frac{n!}{m_1! \ldots m_l!}$$

is a multinomial coefficient.

A special case of the multinomial theorem, when $l = 2$, is the binomial theorem

$$\sum_{i=0}^{n} \binom{n}{i} x_1^i x_2^{n-i} = (x_1 + x_2)^n. \tag{A.4}$$

From the binomial theorem (and the definition of the binomial coefficients), we can derive the following equality:

$$\sum_{i=1}^{n} \binom{n}{i} i = n2^{n-1}. \tag{A.5}$$

Upper bounds for binomial coefficients

The tail of the binomial distribution $Bin(k, p)$ can be bounded by the following inequality, when $L + 1 < kp$ [23, 122-123]:

$$\begin{aligned} P(N < L + 1) &= \sum_{i=0}^{L} \binom{k}{i} p^i (1-p)^{k-i} \\ &< \frac{(L+1)(1-p)}{kp - (L+1)} \binom{k}{L+1} p^{L+1} (1-p)^{k-L-1}. \end{aligned}$$

When $p = \frac{1}{2}$, we achieve the following upper bound for the sum of binomial coefficients, when $L + 1 < \frac{k}{2}$:

$$\sum_{i=0}^{L} \binom{k}{i} < \frac{L+1}{k - 2(L+1)} \binom{k}{L+1}. \tag{A.6}$$

Chapter B

Auxiliary results

B.1 Numbers of all possible rules

In the following, we give the proofs for the numbers of all possible dependency rules, when the rule can contain 1) a set of attributes in the consequent or 2) an arbitrary number of negations in the antecedent. In case 1), we give also the number of all more general rules, which can make the rule redundant.

Theorem B.1 *Let Z be a set of binary attributes. From Z, we can produce $2^{|Z|} - 2$ different rules of form $X \rightarrow Z \setminus X$, where $X \subsetneq Z$, $X \neq \emptyset$.*

Proof Let $m = |Z|$. An antecedent X, $|X| = i$, can be selected from Z in $\binom{m}{i}$ ways. Since $i = 1, \ldots, m-1$, the number of all possible ways to select an antecedent is

$$\sum_{i=2}^{m-1} \binom{m}{i} = 2^m - 2.$$

□

We note that rules $X \rightarrow Z \setminus X$ and $Z \setminus X \rightarrow X$ express the same dependency, and therefore only one of them could be included. Then the total number of all possible rules is $2^{|Z|-1} - 1$.

When the consequent contains a set of attributes, the idea of redundancy is also enlarged. Now rule $X \rightarrow Z \setminus X$ can be redundant with respect to rules of form $Y \rightarrow Q$, where $Y \subseteq X$, $Q \subseteq Z \setminus X$, and either $|Y| < |X|$, $|Q| < |Z \setminus X|$, or both.

Theorem B.2 *Rule $X \rightarrow Z \setminus X$ can be redundant with respect to $2^{|Z|} - 2^{|X|} - 2^{|Z \setminus X|}$ more general rules.*

Proof Let $|Z| = m$ and $|X| = l$. Let $Y \to Q$ be a more general rule, where $Y \subseteq X$, $Q \subseteq Z \setminus X$, and either $|Y| < |X|$ or $|Q| < |Z \setminus X|$. If $|Y| \leq |X|$, Y can be selected in $2^l - 1$ different ways, and if $|Y| < |X|$, it can be selected in $2^l - 2$ different ways. Similarly, Q can be selected in either $2^{m-l} - 1$ or $2^{m-l} - 2$ different ways. Since Y and Q are separate, all combinations of Y and Q are possible. The number of all rules, where $|Y| < |X|$ and $|Q| \leq |Z \setminus X|$ is $(2^l - 2)(2^{m-l} - 1) = 2^m - 2^l - 2 \cdot 2^{m-l} + 2$. In addition, there are $2^{m-l} - 2$ rules, where $|Y| = |X|$ and $|Q| < |Z \setminus X|$. Together, these make $2^m - 2^l - 2^{m-l}$ different rules. □

Theorem B.3 *Let R be the set of all attributes, $|R| = k$. There are $\mathcal{O}(k3^k)$ possible rules of form $X = \overline{x} \to A = a$, where $X \subsetneq R$, $\overline{x} \in Dom(X)$, $A \in R \setminus X$, and $a \in \{0, 1\}$.*

Proof Let us mark $Z = XA$. If $|Z| = i$, Z can be selected from R in $\binom{k}{i}$ different ways. For each Z, there are 2^i different value combinations. Finally, the consequent attribute can be selected in i different ways. Since each rule must contain at least one attribute in the antecedent and exactly one attribute in the consequent, $i = 2, \ldots, k$. Therefore, the total number of all possible rules is

$$\sum_{i=2}^{k} i \binom{k}{i} 2^i < k \sum_{i=2}^{k} \binom{k}{i} 2^i = k(3^k - 2k - 1) = \mathcal{O}(k3^k).$$

□

B.2 Problems of closed and free sets

In the following, we analyze why the closed sets or free sets do not suit for searching for non-redundant statistical dependency rules. In addition, we show why closed classification rules (or redundant rules, in general) are more prone to errors than non-redundant rules.

B.2.1 Generating dependency rules from closed or free sets

The idea of *closed* attribute set is the following [61, 9]: A *closure* of an attribute set X is set

$$Cl(X) = X \cup \{A_i \mid A_i \notin X, P(A_i|X) = 1\}.$$

I.e. the closure is the most specific attribute set which covers the same rows as X. Therefore, also $P(X) = P(Cl(X))$. An attribute set X is called

closed, if $X = Cl(X)$. This means that for all $Z \supsetneq X$ holds $P(Z) < P(X)$. For each closed set X, there can be more general sets $Y_l \subsetneq Y_{l-1} \subsetneq \ldots Y_1 = X$ such that $P(Y_i) = P(X)$ and thus $Cl(Y_i) = X$ for all $i = 1, \ldots, l$. These Y_is are called *generators* of set X and the most general set, Y_l, is called the *minimal generator* of X. Minimal generators are also called *0-free sets* or simply *free sets* [17].

In the worst case, all attribute sets, which occur in the data, are both closed and free sets. This happens, for example, when the data contains each attribute combination $A_1 = a_1, \ldots, A_k = a_k$, where $R = \{A_1, \ldots, A_k\}$ and $a_i \in \{0, 1\}$ for all $i = 1, \ldots, k$, exactly once. Now the confidence of any rule $X \rightarrow A_i$, $X \subsetneq R$, $A_i \notin X$, is $P(A_i|X) = 0.5$. This means that for all sets $Z \subseteq R$, $Z = Cl(Z)$ and Z is also the only generator for itself. However, often the data contains rules with confidence $cf = 1$, and the number of all closed or all free sets is smaller than the number of all occurring attribute sets. An interesting question is, whether we could find all non-redundant statistical dependency rules by searching for only all closed sets or all free sets with some minimum frequency threshold. Here, we show that this is not possible, in spite of how small a minimum frequency threshold could be used. The problem is that neither closed sets nor free sets alone capture all non-redundant statistical dependencies, unless all occurring sets are both closed and free.

Let us first consider the case, where only free sets are searched for. Table B.1 shows an example data distribution on attributes $R = \{A, B, C\}$, where the free sets contain no statistical dependencies. In this data distribution, the only statistical dependencies occur in set ABC. Because $P(C|AB) = 1$, set ABC is not free, but all the other sets in $\mathcal{P}(R)$ are free. Therefore, only independence rules are found, if we search for only the free sets.

Table B.1: An example distribution, where free sets contain no statistical dependencies.

$P(ABC) = P(A)P(B)$
$P(AB\neg C) = 0$
$P(A\neg BC) = P(A)(P(C) - P(B))$
$P(A\neg B\neg C) = P(A)P(\neg C)$
$P(ABC) = P(B)(P(C) - P(A))$
$P(\neg AB\neg C) = P(B)P(\neg C)$
$P(\neg A\neg BC) = P(\neg A)P(C) - P(B)(P(C) - P(A))$
$P(\neg A\neg B\neg C) = P(\neg C)(1 - P(A) - P(B))$

If only closed sets are searched for, the resulting rules are often redundant. If set X is closed, then for some $A_i \in X$ holds $P(A_i|X \setminus \{A_i\}) = 1$ and set $X \setminus \{A_i\}$ is not closed. Similarly, it follows that $P(A_i|X \setminus \{A_i\}Q) = 1$ for any $Q \subseteq R \setminus X$, and all sets $X \setminus \{A_i\}Q$ are not closed. Now all rules $X \to A_j$, $A_j \notin X$, are redundant with respect to more general, but equally good rules $X \setminus \{A_i\} \to A_j$. However, the more general rules are not found, because set $X \setminus \{A_i\}A_j$ is not closed. In the next subsection we show that these redundant rules are prone to produce extra errors compared to their non-redundant counterparts.

B.2.2 The problem of closed classification rules

In the previous research [59], it has been suggested that one could search for only the best closed classification rules $X \to C$, where X is a closed set. The problem of this approach is that set X is the most specific set among all Y_i with $Cl(Y_i) = X$. Therefore, it can contain a set of redundant attributes $Q = X \setminus Y_l$, which add no information on the dependency between Y_l and C. In the worst case, attributes Q occur with Y_lC just by chance, and the rule does not hold in the future data. Generally, the rule $X \to C$ is less accurate than $Y_l \to C$ in the future data. In the following, we show this using the χ^2-measure, which was one of the goodness measures used in [59]. However, the results generalize to several other measures, as well.

Let $Y \subsetneq X$ be the most general set such that $P(Q|Y) = 1$, when $X = YQ$. Let us analyze the error in the future data (or a test set), if $X \to C$ is selected instead of a more general but equally good rule $Y \to C$.

Let us notate the real probabilities by P_r and the probabilities in the learning set by P. We show that $X \to C$ causes error, if $P_r(Q|Y) < 1$.

The most likely scenario is that Q and Y are actually dependent, but Q and C are conditionally independent given Y. Now $P_r(QC|Y) = P(Q|Y)P(C|Y) \Leftrightarrow P_r(YQC) = P_r(YC)P_r(Q|Y)$.

Let $\delta_1 = \delta_r(Y, C) = P_r(YC) - P_r(Y)P_r(C)$ be the real leverage of rule $Y \to C$. If $P(Y) = P_r(Y)$, $P(C) = P_r(C)$, and $P(YC) = P_r(YC)$, then $\delta(Y, C) = \delta_r(Y, C)$.

On the other hand, assuming the conditional independence between Q and C, $\delta_2 = \delta_r(YQ, C) = P_r(YQC) - P_r(YQ)P_r(C) = P_r(Q|Y)(P_r(YC) - P_r(Y)P_r(C)) = P_r(Q|Y)\delta_1$.

If the data size is N, the corresponding χ^2-values are

$$\chi^2_1 = \frac{N\delta_1^2}{P_r(Y)P_r(\neg Y)P_r(C)P_r(\neg C)}$$

and

$$\chi_2^2 = \frac{NP_r(Q|Y)^2\delta_1^2}{P_r(Y)P_r(Q|Y)(1-P_r(YQ))P_r(C)P_r(\neg C)} = \frac{NP_r(Q|Y)\delta_1^2}{P_r(Y)(1-P_r(YQ))P_r(C)P_r(\neg C)}.$$

The relative error in the χ^2-value is

$$\frac{\chi_1^2-\chi_2^2}{\chi_1^2} = \frac{1-P_r(Q|Y)}{1-P_r(YQ)} = \frac{1-P_r(Q|Y)}{1-P_r(Y)P_r(Q|Y)}.$$

If $P_r(Q|Y) = P(Q|Y) = 1$, the error is zero. Otherwise $X \to C$ causes some error, which is larger, the smaller $P_r(Q|Y)$ is.

We note that both χ^2-values contain some error, if $P(Y)$, $P(C)$ or $P(YC)$ differ from their correct values. However, rule $X \to C$ contains an extra source for error, which make it potentially less accurate than rule $Y \to C$.

B.3 Proofs for good behaviour

In the following, we show that the χ^2-measure, mutual information, two versions of the z-score, and the J-measure are well-behaving measures. The first two measures are defined for both positive and negative dependencies, while the last three are defined only for positive dependencies.

Theorem B.4 *Let $S \subseteq \mathbb{N}^4$ be defined by constraints $0 < N$, $0 < N_X < N$, $0 < N_A < N$, and $0 \leq N_{XA} \leq \min\{N_X, N_A\}$. Measure M is well-behaving, if it is defined by function*

(a) $\chi^2(N_X, N_{XA}, N_A, N) = \frac{N(NN_{XA}-N_XN_A)^2}{N_X(N-N_X)N_A(N-N_A)}$;

(b) $MI(N_X, N_{XA}, N_A, N) = N_{XA}\log\frac{N\cdot N_{XA}}{N_XN_A}$
$+(N_X - N_{XA})\log\frac{N\cdot(N_X-N_{XA})}{N_X(N-N_A)} + (N_A - N_{XA})\log\frac{N\cdot(N_A-N_{XA})}{(N-N_X)N_A}$
$+(N - N_X - N_A + N_{XA})\log\frac{N\cdot(N-N_X-N_A+N_{XA})}{(N-N_X)(N-N_A)}$;

(c) $z_1(N_X, N_{XA}, N_A, N) = \frac{\sqrt{N}(NN_{XA}-N_XN_A)}{\sqrt{N_XN_A(N^2-N_XN_A)}}$, *when* $NN_{XA} > N_XN_A$, *and 0, otherwise;*

(d) $z_2(N_X, N_{XA}, N_A, N) = \frac{NN_{XA} - N_X N_A}{\sqrt{N_X N_A (N - N_A)}}$, *when* $NN_{XA} > N_X N_A$, *and 0, otherwise; and*

(e) $J(N_X, N_{XA}, N_A, N) = N_{XA} \log(\frac{N_{XA}}{N_A}) + (N_X - N_{XA}) \log(\frac{N_X - N_{XA}}{N - N_A}) - N_X \log(\frac{N_X}{N})$, *when* $NN_{XA} > N_X N_A$, *and 0, otherwise.*

Proof In the proofs, we assume that $N = n$ is fixed. We will simplify the functions by substituting $P(XA) = \frac{N_{XA}}{N}$, $P(X) = \frac{N_X}{N}$, and $P(A) = \frac{N_A}{N}$.
(a) Conditions (i) and (ii): For χ^2 the alternative expression is

$$f_2(P(X), \delta, P(A), n) = \frac{n\delta^2}{P(X)(1 - P(X))P(A)(1 - P(A))}.$$

The derivative with respect to δ is

$$f_2' = \frac{2n\delta}{P(X)(1 - P(X))P(A)(1 - P(A))},$$

which satisfies the conditions (i) and (ii).

Condition (iii): When $P(XA)$, $P(A)$, and n are fixed, f can be expressed as

$$f(P(X), P(XA), P(A), n) = \frac{n}{P(A)(1 - P(A))} g(P(X)),$$

where $g(P(X)) = \frac{(P(XA) - P(X)P(A))^2}{P(X)(1 - P(X))}$. The first factor is constant, and therefore it is sufficient to differentiate $g(P(X))$ with respect to $P(X)$.

$$g'(P(X)) = \frac{-2P(XA)P(X)^2P(A) + P(X)^2P(A)^2 - P(XA)^2 + 2P(X)P(XA)^2}{P(X)^2(1 - P(X))^2}.$$

The denominator is

$$\begin{aligned}&[P(XA) - P(X)P(A)][-P(X)P(A) - P(XA) + 2P(X)P(XA)]\\ &= [P(XA) - P(X)P(A)][P(X)(P(XA) - P(A)) + P(XA)(P(X) - 1)].\end{aligned}$$

The first factor is leverage δ and the second factor is always negative. Therefore, $g' < 0$, when $\delta > 0$, and $g' > 0$, when $\delta < 0$.

Condition (iv): Let as first check the case, where $N_{XA} = cf_1 N_X$. Now f becomes

$$f(P(X), cf_1 P(X), P(A), n) = \frac{nP(X)(cf_1 - P(A))^2}{P(A)(1 - P(A))(1 - P(X))}.$$

This is clearly an increasing function of $P(X)$, when $cf_1 \neq P(A)$. Let us then check case $N_{XA} = m(A) - cf_2(n - N_X)$. Now f becomes

$$f(P(X), P(A) - cf_2(1 - P(X)), P(A), n) = \frac{n(1 - P(X))(P(A) - cf_2)^2}{P(X)P(A)(1 - P(A))}.$$

This is clearly a decreasing function of $P(X)$, when $cf_2 \neq P(A)$.

(b) In mutual information, the base of the logarithm is not defined, but usually it is assumed to be 2. However, transformation to the natural logarithm causes only a constant nominator ln(2), which disappears in differentiation. Therefore we will use the natural logarithms for simplicity. We recall that the derivative of a term of form $g(x)\ln(g(X))$ is $g'(x)\ln(g(x)) + g'(x)$.

Condition (i) and (ii): MI can be expressed as function f_2:

$$\begin{aligned} f_2(P(X), \delta, P(A), n) = n[&(P(X)P(A) + \delta)\ln(P(X)P(A) + \delta) + \\ &(P(X)(1 - P(A)) - \delta)\ln(P(X)(1 - P(A)) - \delta) + ((1 - P(X))P(A) - \delta) \\ &\ln((1 - P(X))P(A) - \delta) + ((1 - P(X))(1 - P(A)) + \delta)\ln((1 - P(X)) \\ &(1 - P(A)) + \delta) - P(A)\ln(P(A)) - (1 - P(A))\ln(1 - P(A)) - P(X)\ln(P(X)) \\ &- (1 - P(X))\ln(1 - P(X))]. \end{aligned}$$

The derivative of f_2 with respect to δ is

$$f_2' = n\ln\left(\frac{(P(X)P(A) + \delta)((1 - P(X))(1 - P(A)) + \delta)}{((P(X)(1 - P(A)) - \delta)((1 - P(X))P(A) - \delta)}\right).$$

This is the same as n times the logarithm of the odds ratio $odds$, for which holds $odds = 1$, when $\delta = 0$, $odds > 1$, when $\delta > 0$, and $odds < 1$, when $\delta < 0$. Therefore, the logarithm is zero, when $\delta = 0$, negative, when $\delta < 0$, and positive, when $\delta > 0$.

Condition (iii): When $P(XA)$, $P(A)$, and n are fixed, f can be expressed as

$$\begin{aligned} g(P(X)) = n[&P(XA)\ln(P(XA)) + (P(X) - P(XA))\ln(P(X) - P(XA)) \\ &+ (P(A) - P(XA))\ln(P(A) - P(XA)) + (1 - P(X) - P(A) + P(XA))\ln \\ &(1 - P(X) - P(A) + P(XA)) - P(A)\ln(P(A)) - (1 - P(A))\ln(1 - P(A)) \\ &- P(X)\ln(P(X)) - (1 - P(X))\ln(1 - P(X))]. \end{aligned}$$

The derivative of g with respect to $P(X)$ is

$$g' = n\ln\left(\frac{(P(X) - P(XA))(1 - P(X))}{(1 - P(X) - P(A) + P(XA))P(X)}\right).$$

Since $q = \frac{(P(X)-P(XA))(1-P(X))}{(1-P(X)-P(A)+P(XA))P(X)} = 1$, when $P(XA) = P(X)P(A)$, $q > 1$, when $P(XA) < P(X)P(A)$, and $q < 1$, when $P(XA) > P(X)P(A)$, the logarithm is zero, when X and $A = a$ are independent, negative, when X and $A = a$ are positively dependent, and positive, when X and $A = a$ are negatively dependent.

Condition (iv): When $N_{XA} = cf_1 N_X$, f becomes

$$f(P(X), cf_1 P(X), P(A), n) = n[P(X)cf_1 \ln(P(X)cf_1) + P(X)(1 - cf_1) \ln(P(X)(1 - cf_1)) + (P(A) - P(X)cf_1) \ln(P(A) - P(X)cf_1) + (1 - P(X) - P(A) + P(X)cf_1) \ln(1 - P(X) - P(A) + P(X)cf_1) - P(X) \ln P(X) - (1 - P(X)) \ln(1 - P(X)) - P(A) \ln P(A) - (1 - P(A)) \ln(1 - P(A))].$$

The derivative of f is

$$f' = n[cf_1 \ln(P(X)cf_1) + (1 - cf_1) \ln(P(X)(1 - cf_1)) - cf_1 \ln(P(A) - P(X)cf_1) - (1 - cf_1) \ln(1 - P(X) - P(A) + P(X)cf_1) - \ln(P(X)) + \ln(1 - P(X))].$$

We should show that $f' > 0$, when $cf_1 > P(A)$. To find the lowest value of f', we set $g(cf_1) = f'(P(X))$ and differentiate g with respect to cf_1.

$$\begin{aligned} g'(cf_1) &= n \left[\ln \left(\frac{P(X)cf_1(1 - P(X) - P(A) + P(X)cf_1)}{P(X)(1 - cf_1)(P(A) - P(X)cf_1)} \right) \right. \\ &\quad \left. + \frac{P(X)cf_1}{P(A) - P(X)cf_1} - \frac{P(X)(1 - cf_1)}{1 - P(X) - P(A) + P(X)cf_1} \right] \\ &= n \left[\ln \left(\frac{P(XA)P(\neg X \neg A)}{P(X\neg A)P(\neg XA)} \right) + \frac{P(XA)P(\neg X\neg A) - P(\neg XA)P(X\neg A)}{P(\neg XA)P(X\neg A)} \right] \\ &= n \left[\ln(odds) + \frac{\delta}{P(\neg XA)P(X\neg A)} \right]. \end{aligned}$$

The first term is the logarithm of the odds ratio, which is > 0, when $\delta > 0$. So $g' > 0$, when $\delta > 0$. Therefore, g is an increasing function of cf_1 and gets its minimal value, when $\delta = 0$ and $cf_1 = P(A)$. When we substitute this to f', we get $f' = \ln(1) = 0$. This is the minimal value of f', which is achieved only, when $\delta = 0$; otherwise $f' > 0$, as desired.

Let us then check case $N_{XA} = m(A) - cf_2(n - N_X)$. Now $P(XA) = P(A) - cf_2(1 - P(X))$, $P(X\neg A) = P(X)(1 - cf_2) - P(A) + cf_2$, $P(\neg XA) =$

$cf_2(1 - P(X))$, $P(\neg X \neg A) = (1 - P(X))(1 - cf_2)$, and f becomes

$$\begin{aligned}
&f(P(X), P(A) - cf_2(1 - P(X)), P(A), n) = \\
&n[(P(A) - cf_2(1 - P(X))) \ln(P(A) - cf_2(1 - P(X))) + (P(X)(1 - cf_2) \\
&- P(A) + cf_2) \ln(P(X)(1 - cf_2) - P(A) + cf_2) + cf_2(1 - P(X)) \ln(cf_2 \\
&(1 - P(X))) + (1 - P(X))(1 - cf_2) \ln((1 - P(X))(1 - cf_2)) - P(X) \ln P(X) \\
&- (1 - P(X)) \ln(1 - P(X)) - P(A) \ln(P(A)) - (1 - P(A)) \ln(1 - P(A))] \\
&= n[(P(A) - cf_2(1 - P(X))) \ln(P(A) - cf_2(1 - P(X))) + (P(X)(1 - cf_2) \\
&- P(A) + cf_2) \ln(P(X)(1 - cf_2) - P(A) + cf_2) + cf_2(1 - P(X)) \ln(cf_2) \\
&+ (1 - P(X))(1 - cf_2) \ln(1 - cf_2) - P(X) \ln P(X) - P(A) \ln(P(A)) \\
&- (1 - P(A)) \ln(1 - P(A))]
\end{aligned}$$

The derivative of f with respect to $P(X)$ is

$$f' = n \left[cf_2 \ln \frac{P(A) - cf_2(1 - P(X))}{cf_2} + (1 - cf_2) \ln \frac{P(X)(1 - cf_2) - P(A) + cf_2}{1 - cf_2} - \ln(P(X)) \right].$$

We should show that $f' < 0$, when $cf_2 > P(A)$. To find the largest value of f', we set $g(cf_2) = f'(P(X))$ and differentiate g with respect to cf_2.

$$\begin{aligned}
g'(cf_2) &= n \left[\ln \left(\frac{P(A) - cf_2(1 - P(X))}{cf_2} \right) - \ln \left(\frac{P(X) - P(A) + cf_2(1 - P(X))}{1 - cf_2} \right) \right. \\
&\quad \left. - \frac{P(A)}{P(A) - cf_2(1 - P(X))} + \frac{1 - P(A)}{P(X) - P(A) + cf_2(1 - P(X))} \right] \\
&= n \left[\ln \left(\frac{P(XA)(1 - cf_2)}{cf_2 P(X \neg A)} \right) - \frac{P(A)}{P(XA)} + \frac{1 - P(A)}{P(X \neg A)} \right].
\end{aligned}$$

Because the dependence is negative, $-P(XA) > -P(X)P(A)$ and $P(X)(1 - P(A)) < P(X \neg A)$. Therefore, the sum of the last two terms is < 0. The first term becomes $\ln(odds)$, when we substitute $cf_2 = P(A|\neg X)$. For negative dependence, also $\ln(odds) < 0$, and therefore $g' < 0$. Because g is decreasing with cf_2, f' gets its maximum value, when cf_2 is minimal, i.e. $P(A)$. When we substitute this to f', it becomes $f' = 0$. This is the maximal value of f', which is achieved only when $cf_2 = P(A)$ and $\delta = 0$. Otherwise, $f' < 0$, as desired.

(c) Now it is enough to check the conditions only for the positive dependence.

Conditions (i) and (ii): z_1 can be expressed as

$$f_2(P(X), \delta, P(A), n) = \frac{\sqrt{n}\delta}{\sqrt{P(X)P(A)(1 - P(X)P(A))}},$$

when $\delta > 0$, and $f_2 = 0$, otherwise. This is clearly an increasing function of δ and gets its minimum value 0, when $\delta \leq 0$.

Condition (iii): When $P(XA)$, $P(A)$, and n are fixed, f can be expressed as

$$f(P(X), P(XA), P(A), n) = \frac{\sqrt{n}}{\sqrt{P(A)}} g(P(X)),$$

where $g(P(X)) = \frac{P(XA)-P(X)P(A)}{\sqrt{P(X)(1-P(X)P(A))}}$. The derivative of g with respect to $P(X)$ is

$$g' = \frac{-P(X)P(A)(1-P(XA)) - P(XA)(1-P(X)P(A))}{2(P(X)(1-P(X)P(A)))^{\frac{3}{2}}}.$$

Because $P(X)P(A) < 1$ and $P(XA) \leq 1$, $g' < 0$ always.

Condition (iv): When $N_{XA} = cf_1 N_X$, f becomes

$$f(P(X), cf_1 P(X), P(A), n) = \frac{\sqrt{nP(X)}(cf_1 - P(A))}{\sqrt{P(A)(1-P(X)P(A))}},$$

which is clearly an increasing function of $P(X)$, when $cf_1 \geq P(A)$.

(d) Conditions (i) and (ii): z_2 can be expressed as

$$f_2(P(X), \delta, P(A), n) = \frac{\sqrt{n}\delta}{\sqrt{P(X)P(A)(1-P(A))}},$$

when $\delta > 0$, and $f_2 = 0$, otherwise. This is clearly an increasing function of δ and gets its minimum value 0, when $\delta \leq 0$.

Condition (iii): When $P(XA)$, $P(A)$, and n are fixed, f can be expressed as

$$f(P(X), P(XA), P(A), n) = \frac{\sqrt{n}}{\sqrt{P(A)(1-P(A))}} g(P(X)),$$

where $g(P(X)) = \frac{P(XA)-P(X)P(A)}{\sqrt{P(X)}}$. The derivative of g with respect to $P(X)$ is

$$g'(P(X)) = \frac{-P(X)P(A) - P(XA)}{2P(X)^{\frac{3}{2}}},$$

which is < 0 always.

Condition (iv): When $P(XA) = cf_1 P(X)$, f becomes

$$f(P(X), cf_1 P(X), P(A), n) = \frac{\sqrt{nP(X)}(cf_1 - P(A))}{\sqrt{P(A)(1 - P(A))}},$$

which is clearly an increasing function of $P(X)$, when $cf_1 \geq P(A)$.

(e) Like in the mutual information, we will use the natural logarithm for simplicity.

Conditions (i) and (ii): J can be expressed as

$$f_2(P(X), \delta, P(A), n) = n \left[(P(X)P(A) + \delta) \ln \left(\frac{P(X)P(A) + \delta}{P(A)} \right) \right.$$
$$\left. + (P(X)(1 - P(A)) - \delta) \ln \left(\frac{P(X)(1 - P(A)) - \delta}{(1 - P(A))} \right) - P(X) \ln(P(X)) \right].$$

The derivative of f_2 with respect to δ is

$$f_2' = n \left[\ln \left(\frac{P(X)P(A) + \delta}{P(A)} \right) - \ln \left(\frac{P(X)(1 - P(A)) - \delta}{(1 - P(A))} \right) - \ln(P(X)) \right]$$
$$= n \ln \left(\frac{(P(X)P(A) + \delta)(1 - P(A))}{(P(X)(1 - P(A)) - \delta)P(X)P(A)} \right).$$

The argument of the logarithm is 1, if $\delta = 0$, and otherwise it is > 1. Therefore, $f_2' \geq 0$, when $\delta \geq 0$.

Condition (iii): When $P(XA)$, $P(A)$, and n are fixed, f can be expressed as

$$g(P(X)) = n[P(XA) \ln \left(\frac{P(XA)}{P(A)} \right) + (P(X) - P(XA)) \ln \left(\frac{P(X) - P(XA)}{(1 - P(A))} \right)$$
$$- P(X) \ln(P(X))].$$

The derivative of g with respect to $P(X)$ is

$$g' = n[\ln \left(\frac{P(X) - P(XA)}{(1 - P(A))} \right) - \ln(P(X))] = n \ln \left(\frac{P(X) - P(XA)}{P(X)(1 - P(A))} \right).$$

The argument of the logarithm is < 1 and thus $g' < 0$, if there is a positive dependency.

Condition (iv): When $N_{XA} = cf_1 N_X$, f becomes

$$f(P(X), cf_1 P(X), P(A), n) = n \left[P(X) cf_1 \ln \left(\frac{P(X) cf_1}{P(A)} \right) + P(X) \right.$$
$$\left. (1 - cf_1) \ln \left(\frac{P(X)(1 - cf_1)}{(1 - P(A))} \right) - P(X) \ln(P(X)) \right].$$

The derivative of f with respect to $P(X)$ is

$$f' = n\left[cf_1 \ln\left(\frac{P(X)cf_1}{P(A)}\right) + (1-cf_1)\ln\left(\frac{P(X)(1-cf_1)}{(1-P(A))}\right) - \ln(P(X))\right] = n\left[cf_1 \ln\left(\frac{cf_1(1-P(A))}{(1-cf_1)P(A)}\right) + \ln\left(\frac{1-cf_1}{(1-P(A))}\right)\right].$$

We should show that $f' > 0$, when $cf_1 > P(A)$. To find the lowest value of f', we set $g(cf_1) = f'(P(X))$ and derivative g with respect to cf_1.

$$g'(cf_1) = n \ln\left(\frac{cf_1(1-P(A))}{(1-cf_1)P(A)}\right).$$

Clearly $g' = 0$, if $cf_1 = P(A)$, and $g' > 0$, if $cf_1 > P(A)$. When we substitute the minimum value $cf_1 = P(A)$ to f', we get $f' = n[P(A)\ln(1) + \ln(1)] = 0$. When $cf_1 > P(A)$, $f' > 0$, and f is an increasing function of $P(X)$.

□

B.4 Non-convexity and non-concavity of the z-score

Lemma B.5 *The z-score defined by function*

$$z_1(N_X, N_{XA}, N_A, N) = \frac{\sqrt{N}(NN_{XA} - N_X N_A)}{\sqrt{N_X N_A (N^2 - N_X N_A)}},$$

when $NN_{XA} > N_X N_A$, and 0, otherwise, is non-convex and non-concave function of N_X and N_{XA}.

Proof By the definition of convexity, function $f(x, y)$ is non-convex, if for some points (x_1, y_1) and (x_2, y_2) and some $\theta \in]0, 1[$ holds

$$f((1-\theta)x_1 + \theta x_2, (1-\theta)y_1 + \theta y_2) \geq (1-\theta)f(x_1, y_1) + \theta f(x_2, y_2).$$

On the other hand, function $f(x, y)$ is non-concave, if for some points (x_1, y_1) and (x_2, y_2) and some $\theta \in]0, 1[$ holds

$$f((1-\theta)x_1 + \theta x_2, (1-\theta)y_1 + \theta y_2) \leq (1-\theta)f(x_1, y_1) + \theta f(x_2, y_2).$$

Therefore, it is enough to give one example for both cases.

Let us now notate $x = P(X)$, $y = P(XA)$, $a = P(A)$, and $N = n$. Now the z-score can be expressed as

$$f(x, y) = \frac{\sqrt{n}}{\sqrt{a}} g(x, y),$$

where

$$g(x, y) = \frac{y - xa}{\sqrt{x(1 - xa)}}.$$

It is enough to show that $g(x, y)$ is non-convex and non-concave in some points (x_1, y_1) and (x_2, y_2).

For the non-convexity, let us consider points $(x_1, y_1) = (0, 0)$ and $(x_2, y_2) = (x_2, x_2)$ and $\theta = \frac{1}{2}$. Now both points are on line $y = x$ and $g(0, 0) = 0$. The condition becomes

$$\begin{aligned} g\left(\frac{0 + x_2}{2}, \frac{0 + x_2}{2}\right) = \frac{\sqrt{x_2}(1 - a)}{\sqrt{2 - x_2 a}} &\geq \frac{1}{2}(g(0, 0) + g(x_2, x_2)) = \frac{\sqrt{x_2}(1 - a)}{2\sqrt{1 - x_2 a}} \\ \Leftrightarrow 2\sqrt{1 - x_2 a} &\geq \sqrt{2 - x_2 a} \\ \Leftrightarrow 4(1 - x_2 a) &\geq 2 - x_2 a \\ \Leftrightarrow 2 &\geq 3 x_2 a, \end{aligned}$$

which is true at least for all $x_2 \leq \frac{2}{3}$.

For the non-concavity, let us consider points $(x_1, y_1) = (a, y_1)$ and $(x_2, y_2) = (x_2, x_2 a)$, where $x_2 > a$, and $\theta = \frac{1}{2}$. Now point $(x_2, x_2 a)$ is on the independence line and $g(x_2, x_2 a) = 0$. The condition becomes

$$\begin{aligned} g\left(\frac{a + x_2}{2}, \frac{y_1 + x_2 a}{2}\right) = \frac{a - y_1}{\sqrt{(a + x_2)(2 - a^2 - a x_2)}} & \\ \leq \frac{1}{2}(g(a, y_1) + g(x_2, x_2 a)) = \frac{a - y_1}{2\sqrt{a(1 - a^2)}} & \\ \Leftrightarrow 2\sqrt{a(1 - a^2)} \leq \sqrt{(a + x_2)(2 - a^2 - a x_2)} & \\ \Leftrightarrow 4a(1 - a^2) \leq (a + x_2)(2 - a^2 - a x_2) & \\ \Leftrightarrow 2(a - x_2)(1 - a^2) - a(a^2 - x_2^2) \leq 0 & \\ \Leftrightarrow (a - x_2)(2(1 - a^2) - a - x_2) \leq 0 & \end{aligned}$$

The last inequality is always true, because $x_2 > a$, and thus the first factor is negative, while the second factor is always positive, because now $2(1 - a^2) - a - x_2 \geq 2(1 - x_2^2 - x_2) \geq 0$ for all x_2. □

Chapter C

Implementation details

C.1 Implementing the enumeration tree

The most natural solution is to implement the enumeration tree as a *trie* (*prefix tree*). This solution is commonly used also in the search for frequent item sets (e.g. [32, 50]). In a trie implementation, each node of the tree contains a label corresponding to one attribute $A_i \in R$. If we denote the labels by $v.label$, then node v corresponds to attribute set $v_1.label, v_2.label, \ldots, v_l.label = v.label$, where the path from the root to node v consists of nodes $v_1, \ldots, v_l$.

In the Kingfisher algorithm, all data is stored into the leaf nodes on level l. When level $l+1$ is processed, new leaf nodes are created, and level l nodes become inner nodes. When no more children are created for a level l node, its data content can be removed, and it becomes a structure node. The structure nodes encode only the trie structure, i.e. all sets which are stored into the tree. In practice, they contain just the node label and pointers to the child nodes. The simplest solution to implement the structure nodes and data nodes is to use an extra node type for all the data content. The trie itself consists of only structure nodes, where the level l nodes have pointers to the corresponding data nodes. When the data content can be removed, it is enough to delete the corresponding data node. In this way, no extra space is wasted for data fields in the inner nodes.

Let us now consider the implementation of the trie structure. The objective is that the structure should be space efficient but implement the following operations efficiently:

- searchset(Y) returns a pointer to a node corresponding to set Y. This is needed when the parents are searched.

- deletenode(v, w) deletes node v, removes its pointer in the parent node w, and possibly compacts the child pointers in the parent node.
- addnode(v, w, A_i) adds a node v with label A_i as w's child.

In Kingfisher, the tree is always traversed in a breadth-first-manner and the nodes are deleted in the returning. Level l nodes could be accessed without traversing the upper parts of the tree (if the last level leafs were linked), but we anyway have to keep record on the corresponding attribute sets, which can be read from the root–node paths. In addition, we can delete nodes from the previous levels, if they become leafs after removals. This makes the tree structure more compact, but also saves the traversal time.

Operation searchset(Y) can be implemented in $|Y|$ steps, if the child with a given label can be found in a constant time. This is easy to implement, if each node contains a table of pointers for all possible children, whether they existed or not. In a normal trie this would mean that we have to use tables of k pointers in each node. However, in the enumeration tree the maximal number of children depends on the node. Only the root can have k children and all other nodes have less children. Because the attributes are ordered, we know that any node with label A_i can have at most $k - i$ children with labels $A_{i+1}, \ldots, A_k$. Therefore, we would need a table of size $k - i$ for a node with label A_i. This can still be too space consuming, because many attribute combinations do not occur in the data at all or they would correspond to insignificant or redundant rules. Therefore, we have implemented a more time consuming table, which contains only the pointers of the existing children.

Because the children are kept in an order (by their labels), a child with a given label can be searched with a binary search in $\mathcal{O}(\log(d))$ time, where d is the number of children. Adding a child can be done in a constant time, because the children are added in the correct order. (We have to keep record on the next free position in the children table.) We may also have to increase the children table, if there are no free positions left. (We note that the number of children to be added is not known beforehand.) For this, we use the following strategy: In the beginning, the children table has size 1, and every time the table has to be increased, its size is doubled. We can easily see that the resulting table size is at most $2d$. This means that adding d children to a node takes $\mathcal{O}(d)$ time. When all children have been added, we still remove the extra space from the end of the table. The cost is already included to the previous $\mathcal{O}(d)$ time.

When a child is removed, the table is compacted. In the worst case, this means that we have to shift all d pointers, which takes $\mathcal{O}(d)$ time. In

the worst-case asymptotic analysis, we cannot refer to the actual d, and the time requirements of the three operations become $\mathcal{O}(\log(k))$ (search child), $\mathcal{O}(k)$ (add child), and $\mathcal{O}(k)$ (delete child). We note that adding or removing all (up to k) children takes also $\mathcal{O}(k)$ total time, and the amortized worst case complexity of each addition and deletion operation is $\mathcal{O}(1)$.

The trie structure could be implemented more compactly by *succinct tree structures* [37]. In a succinct representation, each node takes only $2 + \lceil \log_2(k) \rceil$ bits, where k is the maximum number of children [12]. The overall space requirement is close to the information theoretic lower bound. In the same time, many primitive operations, like searching a child with a given label, can be implemented in a constant time, assuming a *word-RAM model* (a random access machine with $\Omega(\log(N))$ bit word, where N is the total number of nodes in the tree). However, the traditional succinct trees are static structures, which do not support efficient addition or deletion of nodes. As a solution, we could apply the techniques of [65] for dynamic trees, and implement the whole trie structure with $2N + N\lceil \log_2(k) \rceil + \mathcal{O}\left(\frac{N \log\log(N)}{\log(N)}\right)$ bits, where N is the number of nodes. In this structure, the addition and deletion of a node as well as the search for the ith child take $\mathcal{O}\left(\frac{\log(N)}{\log\log(N)}\right)$ time. Because N is very large (in the worst case $\mathcal{O}(2^k)$), this causes an overhead to the current time requirement.

In practice, the data nodes are the most space consuming part of the enumeration tree. The problem is that each node has to contain two *possible*-tables, one for positive and one for negative consequences, and in the worst case all k attributes are possible consequences. When *possible*-tables are implemented as bit vectors, they require in the worst case $2k$ bits. The further the search continues, the more consequences become impossible, but the tables are still difficult to implement more compactly.

In the current implementation of Kingfisher, we use the following heuristic to compact the tables: Because the most frequent consequences $A_i = a_i$ become impossible first, we have ordered positive consequences into an ascending order and negative consequences into a descending order by frequency $P(A_i)$. Now the zero bits are likely to occur in the end of bit vectors, and they can be cut off. When the possibility of consequence $A_i = 1$ is checked, it is enough to check that index i occurs in the table, i.e. $i \leq len$, where len is the length of the vector, and $ppossible[i] = 1$. For negative consequence $A_i = 0$, we have to check that $k - i + 1 \leq len$ and $npossible[k - i + 1] = 1$. Hence, checking and updating the ith bit can be done in a constant time. However, searching the cut point (after which all bits are zero) takes in the worst case $\mathcal{O}(\log(k))$ time, when implemented by the binary search. The bit vectors are still quite space consuming, and

in the future research, our aim is to implement them more efficiently, for example by hash tables or Bloom filters.

C.2 Efficient frequency counting

The data set consists of n rows and each row contains k binary-valued attributes $A_1, \ldots, A_k \in R$. In the frequency counting, the problem is to calculate the absolute frequency $m(X)$, $X \subseteq R$. Since we allow only positive-valued attributes in the rule antecedents, it is enough to consider the case, where $A_i = 1$ for all $A_i \in X$.

The simplest representation for the data set is a $n \times k$ bit-matrix r. If attribute A_j occurs on the ith row, then $r[i, j] = 1$, and otherwise 0. In practice, there are two basic approaches, how to construct the bit-matrix. In the *horizontal data layout*, r consists of n bit-vectors, each of them k bits long. In the *vertical data layout*, r consists of k bit-vectors, each of them n bits long. In both approaches, the asymptotic time complexity of frequency counting is the same, but in practice the difference can be remarkable. The reason is that in the vertical data layout, frequency counting can be implemented by efficient bitwise logical operations.

Let us now analyze the time requirement for frequency counting, when the data is stored into n vectors of k bits or k vectors of n bits. Each bit vector is implemented by $\frac{m}{c}$ integers, where m is the length of the bit vector (either k or n) and constant c is the number of bits in the machine word (e.g. 32 or 64, depending on the architecture). Each integer can be checked fast by bitwise logical operations. In practice, we can consider the bitwise logical operations on integers as atomic operations which take constant time.

When the horizontal layout is used, counting the frequency of each l-set requires in the worst case $n \cdot \min\{l, \frac{k}{c}\}$ time steps. If $l < k/c$, the attributes of an l-set can occur in at most l integers, and checking one row takes at most l steps. The checking is repeated on all n rows.

In the vertical layout, the frequency of $X = A_1, \ldots, A_l$ is the same as the number of 1-bits in the bitwise AND of vectors $v(A_1), \ldots, v(A_l)$. This observation was made already 1996 by Yen and Chen [85], but in that time the memory sizes were so small that the technique could not be utilized for large data sets. Counting the frequency of one l-set takes $\frac{n}{c}(l+1)$ time steps, which consist of $\frac{nl}{c}$ bitwise AND operations and counting the number of 1-bits in the result ($\frac{n}{c}$ steps) on each of l rows.

When we compare the requirements we see that the vertical layout is in practice more efficient: $\frac{n}{c}(l+1) < n \cdot \min\{\frac{k}{c}, l\}$ for all $l < k - 1$. The

asymptotic time complexity is still the same $\mathcal{O}(nl)$.

C.3 Efficient implementation of Fisher's exact test

Fisher's exact test (goodness measure p_F) is often the preferred method for significance testing, but it is not trivial to calculate in a large data set. In the following, we consider the technical problems and introduce feasible solutions.

C.3.1 Calculating Fisher's p_F

Calculating Fisher's p_F (Equation 2.4) can be a time consuming operation, because the value is a sum of several terms, each containing binomial coefficients. In addition, the evaluation can easily cause an overflow or underflow. For the latter problem, a common solution is to use logarithms in the middle steps, when each term is evaluated. For example, to evaluate binomial coefficient $c = \binom{m}{l}$, we calculate $\ln(c) = \sum_{i=1}^{m} \ln(i) - \sum_{i=1}^{l} \ln(i) - \sum_{i=1}^{m-l} \ln(i)$, and take the inverse of the result, i.e. $c = e^{\ln(c)}$. This approach is used for example in MagnumOpus [76] ([74]), when the significance of the productivity is tested by Fisher's p_F. However, the technique is unnecessarily worksome, because the same factorials are usually calculated several times during the search.

In Kingfisher, we use a more efficient strategy, where the logarithms of all factorials, $\ln(m!) = \sum_{i=1}^{m} \ln(i)$, $m = 1, \ldots, n$, are calculated just once and stored into a table. Thus, each term in p_F can be evaluated in constant time. Evaluating $p_F(Y \rightarrow A = a)$ takes $J = \min\{m(YA \neq a), m(\neg YA = a)\} \leq \frac{n}{4}$ steps. Therefore, the asymptotic time complexity is $\mathcal{O}(n)$. The initialization of the factorial table takes also $\mathcal{O}(n)$, because $(i+1)! = (i+1)i!$, i.e. $\ln((i+1)!) = \ln(i!) + \ln(i+1)$.

The asymptotic complexity of evaluating p_F is still quite large. In Kingfisher, we have two solutions to improve the efficiency. The first solution is to use exact p_F-values, but calculate them only, when a rule can be significant and non-redundant. This possibility is first checked by calculating only the first term of p_F. Since the first term contributes most to the overall p_F-value, it gives a good lower bound for the expected goodness. If the lower bound is sufficiently good, then the rest of the terms are evaluated.

The second solution is to use a tight approximation for p_F as a goodness measure, instead of the exact p_F. In the following, we introduce three approximations which can be used for this purpose. All of them are upper bounds for p_F, but they approach p_F fast, when the dependency becomes

stronger. The loosest of the upper bounds are especially suitable as goodness measures, because they can be evaluated in a constant time. However, if the dependency is really weak, these approximations may be too inaccurate. In this case, the first terms of p_F can be calculated exactly and the approximation is used to bound only the rest of the terms. In practice, it is usually sufficient to calculate just one or two terms exactly.

C.3.2 Upper bounds for p_F

The following theorem gives two useful upper bounds, which can be used to approximate Fisher's p_F. The first upper bound is more accurate, but it contains an exponent, which makes it more difficult to evaluate. The latter upper bound is always easy to evaluate and also intuitively appealing.

Theorem C.1 *Let us notate* $p_F = p_0 + p_1 + \ldots + p_J$ *and* $q_i = \frac{p_i}{p_{i-1}}$, $i \geq 1$. *For positive dependency rule* $X \rightarrow A$ *with lift* $\gamma(X, A)$ *holds*

$$\begin{aligned} p_F &\leq p_0 \left(\frac{1 - q_1^{J+1}}{1 - q_1} \right) \\ &\leq p_0 \left(1 + \frac{1 - P(A)\gamma(X,A) - P(X)\gamma(X,A) + P(X)P(A)\gamma(X,A)^2}{\gamma(X,A) - 1} \right). \end{aligned}$$

Proof Each p_i can be expressed as $p_i = p_{abs} t_i$, where $p_{abs} = \frac{m(A)!m(\neg A)!}{n!}$ is constant. Therefore, it is enough to show the result for $p_X = \frac{p_F}{p_{abs}}$.

$$p_X = t_0 + t_1 + ... + t_J = t_0 + q_1 t_0 + q_1 q_2 t_0 + ... + q_1 q_2 ... q_J t_0,$$

where

$$\begin{aligned} q_i &= \frac{t_i}{t_{i-1}} \\ &= \frac{(m(XA) + i - 1)!(m(\neg X \neg A) + i - 1)!(m(X\neg A) - i + 1)!(m(\neg XA) - i + 1)!}{(m(XA) + i)!(m(\neg X \neg A) + i)!(m(X\neg A) - i)!(m(\neg XA) - i)!} \\ &= \frac{(m(X\neg A) - i + 1)(m(\neg XA) - i + 1)}{(m(XA) + i)(m(\neg X \neg A) + i)}. \end{aligned}$$

Since q_i decreases when i increases, the largest value is q_1. We get an upper bound

$$p_X = t_0 + q_1 t_0 + q_1 q_2 t_0 + ... + q_1 q_2 ... q_J t_0 \leq t_0(1 + q_1 + q_1^2 + q_1^3 + ... + q_1^J).$$

The sum of the geometric series is $t_0 \frac{1-q_1^{J+1}}{1-q_1}$, which is the first upper bound. On the other hand,

$$t_0 \frac{1-q_1^{J+1}}{1-q_1} \leq \frac{t_0}{1-q_1} = t_0 \left(1 + \frac{q_1}{1-q_1}\right).$$

Let us insert $q_1 = \frac{m(X\neg A)m(\neg XA)}{(m(XA)+1)(m(\neg X\neg A)+1)}$, and express the frequencies using lift $\gamma = \gamma(X, A)$. For simplicity, we use notations $x = P(X)$ and $a = P(A)$. Now $m(XA) = nxa\gamma$, $m(X\neg A) = nx - nxa\gamma$, $m(\neg XA) = na - nxa\gamma$ and $m(\neg X\neg A) = n(1 - x - a + xa\gamma)$. We get

$$\begin{aligned}
\frac{q_1}{1-q_1} &= \frac{m(X\neg A)m(\neg XA)}{(m(XA)+1)(m(\neg X\neg A)+1) - m(X\neg A)m(\neg XA)} \\
&= \frac{n^2xa - n^2xa^2\gamma - n^2x^2a\gamma + n^2x^2a^2\gamma^2}{n^2xa\gamma + 2nxa\gamma + n - nx - na + 1 - n^2xa} \\
&= \frac{nxa - nxa^2\gamma - nx^2a\gamma + nx^2a^2\gamma^2}{nxa\gamma + 2xa\gamma + 1 - x - a + 1/n - nxa}.
\end{aligned}$$

The nominator is $\geq nxa\gamma - nxa$, because $2xa\gamma + 1 - x - a + 1/n \geq 0 \Leftrightarrow P(XA) + P(\neg X\neg A) + 1/n \geq 0$. Therefore

$$\frac{q_1}{1-q_1} \leq \frac{1 - a\gamma - x\gamma + xa\gamma^2}{\gamma - 1}.$$

□

In the following, we will denote the looser (simpler) upper bound by $ub1$ and the tighter upper bound (sum of the geometric series) by $ub2$. In $ub1$, the first term of p_F is always exact and the rest are approximated, while in $ub2$, the first two terms are always exact and the rest are approximated.

We note that $ub1$ can be expressed equivalently as

$$ub1 = p_0 \left(\frac{P(XA)P(\neg X\neg A)}{\delta(X, A)}\right) = p_0 \left(1 + \frac{P(X\neg A)P(\neg XA)}{\delta(X, A)}\right),$$

where $\delta(X, A)$ is the leverage. This is expression is closely related to the odds ratio

$$odds(X, A) = \frac{P(XA)P(\neg X\neg A)}{P(X\neg A)P(\neg XA)} = 1 + \frac{\delta}{P(X\neg A)P(\neg XA)},$$

which is often used to measure the strength of the dependency. The odds ratio can be expressed equivalently as

$$odds(X, A) = \frac{ub1}{ub1 - 1}.$$

We see that when the odds ratio increases (dependency becomes stronger), the upper bound decreases. In practice, it gives a tight approximation to Fisher's p_F, when the dependency is sufficiently strong. The error is difficult to bind tightly, but the following theorem gives a loose upper bound for the error, when $ub2$ is used for approximation.

Theorem C.2 *When p_F is approximated by $ub2$, the error is bounded by*

$$err \leq p_0 \left(\frac{q_1^2}{1 - q_1} \right).$$

Proof Upper bound $ub2$ can cause error only, if $J > 1$. If $J = 0$, $ub2 = p_0$ and if $J = 1$, $ub2 = p_0 \left(\frac{1-q_1^2}{1-q_1} \right) = p_0(1 + q_1) = p_0 + p_1 = p_F$. Let us now assume that $J > 1$. The error is

$$\begin{aligned} err &= ub2 - p_F \\ &= p_0(1 + q_1 + \ldots + q_1^J - 1 - q_1 - q_1q_2 - \ldots - q_1q_2 \ldots q_J) \\ &= p_0(q_1^2 + q_1^3 + \ldots q_1^J - q_1q_2 - \ldots - q_1q_2 \ldots q_J). \end{aligned}$$

It has an upper bound

$$err < p_0q_1^2(1 + q_1 + \ldots + q_1^{J-2}) = p_0q_1^2 \left(\frac{1 - q_1^{J-1}}{1 - q_1} \right) \leq p_0 \left(\frac{q_1^2}{1 - q_1} \right).$$

□

This leads to the following corollary, which gives good guarantees for the safe use of $ub2$:

Corollary C.3 *If $\gamma(X, A) \geq \frac{1+\sqrt{5}}{2} \approx 1.62$, then $err = ub2 - p_F \leq p_0$.*

Proof According to Theorem C.2, $err \leq p_0$, if $q_1^2 \leq 1 - q_1$. This is true, when $q_1 \leq \frac{\sqrt{5}-1}{2}$.

On the other hand, $q_1 < \frac{1}{\gamma}$, when $\gamma > 1$, because

$$q_1 = \frac{m(X\neg A)m(\neg XA)}{(m(XA)+1)(m(\neg X\neg A)+1)}$$
$$< \frac{P(X\neg A)P(\neg XA)}{P(XA)P(\neg X\neg A)} = \frac{xa(1-a\gamma)(1-x\gamma)}{xa\gamma(1-x-a+xa\gamma)}.$$

This is $\leq \frac{1}{\gamma}$, because

$$1-a\gamma-x\gamma+xa\gamma^2 \leq 1-x-a+xa\gamma \Leftrightarrow$$
$$0 \leq x(\gamma-1)+a(\gamma-1)-xa\gamma(\gamma-1)$$
$$= (x+a-P(XA))(\gamma-1).$$

A sufficient condition for $q_1 \leq \frac{\sqrt{5}-1}{2}$ is that

$$\frac{1}{\gamma} \leq \frac{\sqrt{5}-1}{2} \Leftrightarrow \gamma \geq \frac{2}{\sqrt{5}-1} = \frac{\sqrt{5}+1}{2}.$$

□

This result also means that $ub2 \leq 2p_F$, when the lift is as large as required.

The simpler upper bound, $ub1$, can cause a somewhat larger error than $ub2$, but it is even harder to analyze. However, we note that $ub1 = p_F$ only, when $J = 0$. When $J = 1$, there is already some error, but in practice the difference is marginal. The following theorem gives guarantees for the accuracy of $ub1$, when $\gamma \geq 2$.

Theorem C.4 *If p_F is approximated with $ub1$ and $\gamma(X, A) \geq 2$, the error is bounded by $err \leq p_0$.*

Proof The error is $err = ub1 - p_F = ub1 - ub2 + ub2 - p_F$, where $ub2 - p_F \leq p_0\left(\frac{q_1^2}{1-q_1}\right)$ by Theorem C.2.

When $\gamma \geq 2$, $ub1$ (being a decreasing function of γ) is

$$ub1 = p_0\left(\frac{\gamma(1-a\gamma-x\gamma+xa\gamma^2)}{\gamma-1}\right) \leq p_0 2(1-a-x+2xa).$$

Therefore, the error is bounded by

$$\begin{aligned} err &\leq p_0 \left(2(1-a-x+2xa) - \frac{1-q_1^{J+1}}{1-q_1} + \frac{q_1^2}{1-q} \right) \\ &= p_0 \left(2-2a-2x+4xa - \frac{(1-q_1^2-q_1^{J+1})}{(1-q_1)} \right) \\ &= p_0 \left(\frac{2(1-q_1)-2a(1-q_1)-2x(1-q_1)+4xa(1-q_1)-1+q_1^2+q_1^{J+1}}{1-q_1} \right) \\ &= p_0 \left(\frac{1-2q_1+q_1^2+q_1^{J+1}2a(1-q_1)-2x(1-q_1)+4xa(1-q_1)}{1-q_1} \right). \end{aligned}$$

When $\gamma \geq 2$, $q_1 \leq \frac{1}{2}$, and thus

$$\begin{aligned} 1-2q_1+q_1^2+q_1^{J+1} \leq 1-q_1 \Leftrightarrow \\ 0 \leq q_1 - q_1^2 - q_1^{J+1} \Leftrightarrow \\ 0 \leq q_1(1-q_1-q_1^J). \end{aligned}$$

Therefore,

$$err \leq p_0 \left(\frac{(1-q_1)(1-2a-2x+4xa)}{1-q_1} \right) = p_0(1-2a-2x+4xa).$$

$1-2a-2x+4xa \leq 1$, because $2a+2x-4xa = 2a(1-x)+2x(1-a) \geq 0$. Therefore $err \leq p_0$. □

Our experimental results support the theoretical analysis, according to which both upper bounds, $ub1$ and $ub2$, give tight approximations to Fisher's p_F, when the dependency is sufficiently strong. However, if the dependency is weak, we may need a more accurate approximation. A simple solution is to include more larger terms $p_0 + p_1 + \ldots p_{l-1}$ to the approximation and estimate an upper bound only for the smallest terms $p_l + \ldots p_J$ using the sum of the geometric series. The resulting approximation and the corresponding error bound are given in the following theorem. We omit the proofs, because they are essentially identical with the previous proofs for Theorems C.1 and C.2.

Theorem C.5 *For positive dependency rule $X \rightarrow A$ holds*

$$p_F \leq p_0 + \ldots p_{l-1} + p_l \left(\frac{1-q_{l+1}^{J-l+1}}{1-q_{l+1}} \right),$$

where

$$q_{l+1} = \frac{(m(X\neg A) - l)(m(\neg XA) - l)}{(m(XA) + l + 1)(m(\neg X\neg A) + l + 1)}$$

and $l + 1 \leq J$.

The error of the approximation is

$$err \leq p_0 \left(\frac{q_{l+1}^2}{1 - q_{l+1}} \right).$$

C.3.3 Evaluation

Figure C.1 shows the typical behaviour of the new upper bounds, when the strength of the dependency increases (i.e. $m(XA)$ increases and $m(X)$ and $m(A)$ remain unchanged). In addition to upper bounds $ub1$ and $ub2$, we consider a third upper bound, $ub3$, based on Theorem C.5, where the first three terms of p_F are evaluated exactly and the rest is approximated. All three upper bounds approach each other and the exact p_F-value, when the dependency becomes stronger.

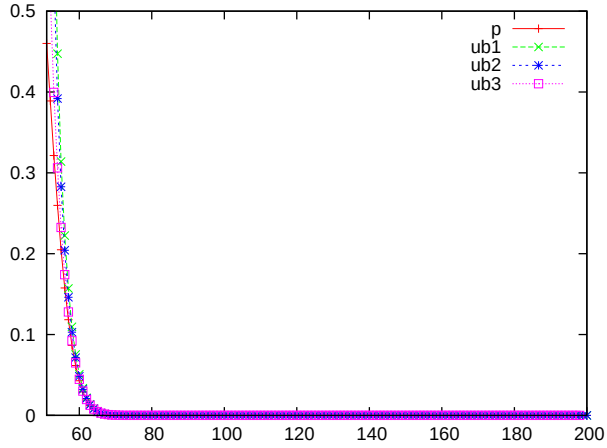


Figure C.1: Exact p_F and three upper bounds as functions of $m(XA)$, when $m(X) = 200$, $m(A) = 250$, and $n = 1000$. The strength of the dependency increases on the x-axes.

Figure C.2 shows a magnified area from Figure C.1. In this area, the dependencies are weak, and the upper bounds diverge from the exact p_F. The reason is that in this area the number of approximated terms is also the largest. For example, when $m(XA) = 55$, p_F contains 146 terms, and

when $m(XA) = 65$, it contains 136 terms. In these points the lift is 1.1 and 1.3, respectively. The difference between $ub1$ and $ub2$ is marginal, but $ub3$ clearly improves $ub1$.

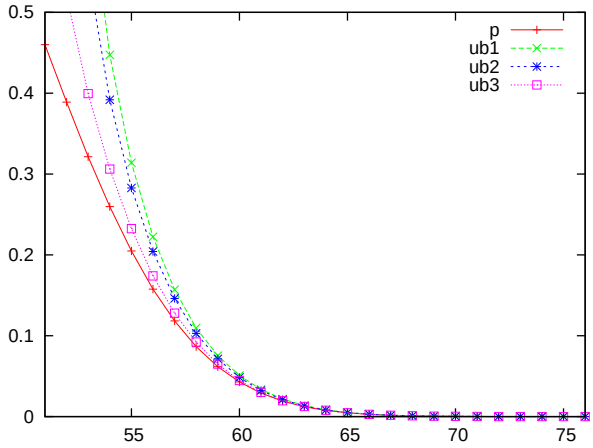


Figure C.2: A magnified area from Figure C.1 showing the differences, when the dependency is weak.

Because the new upper bound gives accurate approximations for strong dependencies, we evaluate the approximations only for the potentially problematic weak dependencies. As an example, we consider two data sets, where the data size is either $n = 1000$ or $n = 10000$. For both data sets, we have three test cases: 1) when $P(X) = P(A) = 0.5$, 2) when $P(X) = 0.2$ and $P(A) = 0.25$, and 3) when $P(X) = 0.05$ and $P(A) = 0.2$. (The second case with $n = 1000$ is shown in Figures C.1 and C.2.) For all test cases we have calculated the exact p_F, three versions of the upper bound, $ub1$, $ub2$ and $ub3$, and the p-value achieved from the one-sided χ^2-measure. The χ^2-based p-values were calculated with an online Chi-Square Calculator [31]. The values are reported for the cases, where $p_F \approx 0.05$, $p_F \approx 0.01$, and $p_F \approx 0.001$. Because the data is discrete, the exact p_F-values always deviate somewhat from the reference values.

The results for the first data set ($n = 1000$) are given in Table C.1 and for the second data set ($n = 10000$) in Table C.2. As expected, the χ^2-approximation works best, when the data size is large and the distribution is balanced (case 1). According to the classical rule of thumb, the χ^2-approximation can be used, when all expected counts are ≥ 5 [28]. This requirement is not satisfied in the third case in the smaller data set. The

resulting χ^2-based p-values are also the least accurate, but the χ^2-test produced inaccurate approximations also for the case 2, even if the smallest expected frequency was 50.

In the smaller data set, the χ^2-approximation overperformed the new upper bounds only in the first case, when $p_F \approx 0.05$. If we had calculated the first four terms exactly, the resulting $ub4$ would have already produced a better approximation.

In the larger data set, the χ^2-measure gave more accurate results for the first two cases, when $p_F \approx 0.05$ and for the case 1, when $p_F \approx 0.01$. When $p_F \approx 0.001$, the new upper bounds gave always more accurate approximations. If we had calculated the first eight terms exactly, the resulting $ub8$ would have overperformed the χ^2-approximation in case 1 with $p_F \approx 0.01$ and case 2 with $p_F \approx 0.05$. Calculating eight exact terms is quite reasonable compared to all 2442 terms, which have to be calculated for the exact p_F in case 1. With 15 exact terms, the approximation for the case 1 with $p_F \approx 0.05$ would have also been more accurate than the χ^2-based approximation. However, in so large data set (especially with an exhaustive search), a p-value of 0.05 (or even 0.01) is hardly significant. Therefore, we can conclude that for practical search purposes the new upper bounds give better approximations to the exact p_F than the χ^2-measure.

Table C.1: Comparison of the exact p_F value, three upper bounds, and the p-value based on the χ^2-measure, when $n = 1000$. The best approximations are emphasized. $E(m(XA))$ is the expected frequency of XA under the independence assumption.

1) $n = 1000$, $m(X) = 500$, $m(A) = 500$, $E(m(XA)) = 250$					
$m(XA)$	p_F	$ub1$	$ub2$	$ub3$	$p(\chi 2)$
263	0.0569	0.0696	0.0674	0.0617	**0.050**
269	0.0096	0.0107	0.0105	**0.0100**	0.0081
275	0.00096	0.00103	0.00101	**0.00100**	0.00080
2) $n = 1000$, $m(X) = 200$, $m(A) = 250$, $E(m(XA)) = 50$					
$m(XA)$	p_F	$ub1$	$ub2$	$ub3$	$p(\chi 2)$
60	0.0429	0.0508	0.0484	**0.0447**	0.0340
63	0.0123	0.0137	0.0132	**0.0125**	0.088
68	0.00089	0.00094	0.00092	**0.00089**	0.00050
3) $n = 1000$, $m(X) = 50$, $m(A) = 200$, $E(m(XA)) = 1$					
$m(XA)$	p_F	$ub1$	$ub2$	$ub3$	$p(\chi 2)$
15	0.0559	0.0655	0.0605	**0.0565**	0.0349
17	0.0123	0.0135	0.0128	**0.0124**	0.0056
19	0.00194	0.00205	0.00198	**0.00194**	0.00050

Table C.2: Comparison of the exact p_F value, three upper bounds, and the p-value based on the χ^2-measure, when $n = 10000$. The best approximations are emphasized. $E(m(XA))$ is the expected frequency of XA under the independence assumption.

1) $n = 10000$, $m(X) = 5000$, $m(A) = 5000$, $E(m(XA)) = 2500$					
$m(XA)$	p_F	$ub1$	$ub2$	$ub3$	$p(\chi 2)$
2541	0.0526	0.0655	0.0647	0.0621	**0.0505**
2559	0.0096	0.0109	0.0109	0.0106	**0.0091**
2578	0.00097	0.00105	0.00104	**0.00102**	0.00090
2) $n = 10000$, $m(X) = 2000$, $m(A) = 2500$, $E(m(XA)) = 500$					
$m(XA)$	p_F	$ub1$	$ub2$	$ub3$	$p(\chi 2)$
529	0.0504	0.0623	0.0611	0.0579	**0.047**
541	0.0100	0.0113	0.0112	**0.0108**	0.0090
554	0.00109	0.00118	0.00116	**0.00114**	0.00090
3) $n = 10000$, $m(X) = 500$, $m(A) = 2000$, $E(m(XA)) = 10$					
$m(XA)$	p_F	$ub1$	$ub2$	$ub3$	$p(\chi 2)$
115	0.0498	0.0608	0.0583	**0.0541**	0.0427
121	0.0105	0.0118	0.0115	**0.0109**	0.0080
128	0.00106	0.00114	0.00112	**0.00108**	0.00070

References

[1] C.C. Aggarwal and P.S. Yu. A new framework for itemset generation. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1998)*, pages 18–24. ACM Press, 1998.

[2] R. Agrawal, T. Imielinski, and A.N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216. ACM Press, 1993.

[3] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB'94*, pages 487–499. Morgan Kaufmann, 1994.

[4] A. Agresti. A survey of exact inference for contingency tables. *Statistical Science*, 7(1):131–153, 1992.

[5] A. Agresti and Y. Min. Frequentist performance of Bayesian confidence intervals for comparing proportions in 2×2 contingency tables. *Biometrics*, 61:515–523, 2005.

[6] M.-L. Antonie and O. R. Zaïane. Mining positive and negative association rules: an approach for confined rules. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'04)*, pages 27–38. Springer-Verlag, 2004.

[7] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[8] G.A. Barnard. Significance tests for 2×2 tables. *Biometrika*, 34(1/2):123–138, 1947.

[9] Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, and L. Lakhal. Mining minimal non-redundant association rules using frequent closed item-

sets. In *Proceedings of the First International Conference on Computational Logic (CL'00)*, volume 1861 of *Lecture Notes in Computer Science*, pages 972–986. Springer-Verlag, 2000.

[10] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, 57(1):289–300, 1995.

[11] Y. Benjamini and M. Leshno. Statistical methods for data mining. In *The Data Mining and Knowledge Discovery Handbook*, pages 565–87. Springer, 2005.

[12] D. Benoit, E.D. Demaine, J.I. Munro, R. Raman, V. Raman, and S.S. Rao. Representing trees of higher degree. *Algorithmica*, 43(4):275–292, 2005.

[13] F. Berzal, I. Blanco, D. Sánchez, and M. Amparo Vila Miranda. A new framework to assess association rules. In *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis (IDA'01)*, pages 95–104. Springer-Verlag, 2001.

[14] J. Blanchard, F. Guillet, R. Gras, and H. Briand. Using information-theoretic measures to assess association rule interestingness. In *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 66–73. IEEE Computer Society, 2005.

[15] C. Borgelt. Apriori v5.14 software, 2010. `http://www.borgelt.net/apriori.html`. Retrieved 7.6. 2010.

[16] C. Borgelt and R. Kruse. Induction of association rules: Apriori implementation. In *Proceedings of the 15th Conference on Computational Statistics (COMPSTAT 2002)*. Physica Verlag, Heidelberg, Germany, 2002.

[17] J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Approximation of frequency queris by means of free-sets. In *Proceedings of the 4th European Conference Principles of Data Mining and Knowledge Discovery (PKDD'00)*, volume 1910 of *Lecture Notes in Computer Science*, pages 75–85. Springer-Verlag, 2000.

[18] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 265–276. ACM Press, 1997.

[19] D. Bruzzese and C. Davino. Visual post-analysis of association rules. *Journal of Visual Languages & Computing*, 14:621–635, December 2003.

[20] K.C. Carriere. How good is a normal approximation for rates and proportions of low incidence events? *Communications in Statistics – Simulation and Computation*, 30:327–337, 2001.

[21] G.W. Cobb and Y.-P. Chen. An application of Markov chain Monte Carlo to community ecology. *The American Mathematical Monthly*, 110:265–288, April 2003.

[22] H.J. Cordell. Detecting gene–gene interactions that underlie human diseases. *Nature Reviews Genetics*, 10:392–404, June 2009.

[23] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachussetts, London, England, 1990.

[24] L. Dehaspe and H. Toivonen. Discovery of relational association rules. In S. Džeroski and N. Lavrač, editors, *Relational Data Mining*, pages 189–212. Springer-Verlag, Berlin, Heidelberg, 2001.

[25] E.S. Edgington. *Randomization Tests*. Marcel Dekker, Inc., New York, third edition, 1995.

[26] FIMI. Frequent itemset mining dataset repository. `http://fimi.cs.helsinki.fi/data/`. Retrieved 10.2. 2009.

[27] P.D. Finch. Description and analogy in the practice of statistics. *Biometrika*, 66:195–206, 1979.

[28] R.A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh, 1925.

[29] D. Freedman, R. Pisani, and R. Purves. *Statistics*. Norton & Company, London, 4th edition, 2007.

[30] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas. Assessing data mining results via swap randomization. *ACM Transactions on Knowledge Discovery from Data*, 1(3):14:1–14:32, 2007.

[31] Inc GraphPad Software. Quickcalcs online calculators for scientists. http://www.graphpad.com/quickcalcs/contingency1.cfm. Retrieved 18.5. 2010.

[32] L. Guner and P. Senkul. Frequent itemset minning with trie data structure and parallel execution with PVM. In *Recent Advances in Parallel Virtual Machine and Message Passing Interface, Proceedings of the 14th European PVM/MPI User's Group Meeting (PVM/MPI 2007)*, volume 4757 of *Lecture Notes in Computer Science*, pages 289–296. Springer, 2007.

[33] M. Haber. A comparison of some continuity corrections for the chi-squared test on 2× 2 tables. *Journal of the American Statistical Association*, 75(371):510–515, 1980.

[34] L.W. Hahn, M.D. Ritchie, and J.H. Moore. Multifactor dimensionality reduction software for detecting gene–gene and gene–environment interactions. *Bioinformatics*, 19:376–382, 2003.

[35] M. Hahsler, K. Hornik, and T. Reutterer. Implications of probabilistic data modeling for mining association rules. In *From Data and Information Analysis to Knowledge Engineering. Proceedings of the 29th Annual Conference of the Gesellschaft für Klassifikation*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 598–605. Springer-Verlag, 2006.

[36] J. V. Howard. The 2×2 table: A discussion from a bayesian viewpoint. *Statistical Science*, 13(4):351–367, 1998.

[37] G. Jacobson. Space-efficient static trees and graphs. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science (SFCS'89)*, pages 549–554. IEEE Computer Society, 1989.

[38] R.J. Bayardo Jr., R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense databases. *Data Mining and Knowledge Discovery*, 4(2/3):217–240, 2000.

[39] A. Kirsch, M. Mitzenmacher, A. Pietracaprina, G. Pucci, E. Upfal, and F. Vandin. An efficient rigorous approach for identifying statistically significant frequent itemsets. In *Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS'09)*, pages 117–126. ACM, 2009.

[40] Y.S. Koh and R. Pears. Efficiently finding negative association rules without support threshold. In *Advances in Artificial Intelligence, Proceedings of the 20th Australian Joint Conference on Artificial Intelligence (AI 2007)*, volume 4830 of *Lecture Notes in Computer Science*, pages 710–714. Springer, 2007.

[41] S. Lallich, O. Teytaud, and E. Prudhomme. Association rule interestingness: Measure and statistical validation. In F. Guillet and H.J. Hamilton, editors, *Quality Measures in Data Mining*, volume 43 of *Studies in Computational Intelligence*, pages 251–275. Springer, 2007.

[42] S. Lallich, B. Vaillant, and P. Lenca. Parametrised measures for the evaluation of association rule interestingness. In *Proceedings of the 11th Symposium on Applied Stochastic Models and Data Analysis (ASMDA'05)*, pages 220–229, 2005.

[43] E.L. Lehmann. The Fisher, Neyman-Pearson theories of testing hypotheses: One theory or two? *Journal of the American Statistical Association*, 88:1242–1249, December 1993.

[44] E.L. Lehmann and J.P. Romano. *Testing Statistical Hypotheses*. Texts in Statistics. Springer, New York, 3rd edition, 2005.

[45] J. Li. On optimal rule discovery. *IEEE Transactions on Knowledge and Data Engineering*, 18(4):460–471, 2006.

[46] B.W. Lindgren. *Statistical Theory*. Chapman & Hall, Boca Raton, U.S.A., 4th edition, 1993.

[47] B. Liu, W. Hsu, and Y. Ma. Pruning and summarizing the discovered associations. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99)*, pages 125–134. ACM Press, 1999.

[48] G. Liu, J. Li, and L. Wong. A new concise representation of frequent itemsets using generators and a positive border. *Knowledge and Information Systems*, 17(1):35–56, 2008.

[49] J.A. Major and J.J. Mangano. Selecting among rules induced from a hurricane database. *Journal of Intelligent Information Systems*, 4:39–52, January 1995.

[50] G.S. Manku and R. Motwani. Approximate frequency counts over data streams. In *Proceedings of 28th International Conference on Very Large Data Bases (VLDB 2002)*, pages 346–357. Morgan Kaufmann, 2002.

[51] H. Mannila, H. Toivonen, and A.I. Verkamo. Efficient algorithms for discovering association rules. In *Papers from the AAAI Workshop on Knowledge Discovery in Databases (KDD'94)*, pages 181–192. AAAI Press, 1994.

[52] N. Megiddo and R. Srikant. Discovering predictive association rules. In *Proceedings of the 4th International Conference on Knowledge Discovery in Databases and Data Mining*, pages 274–278. AAAI Press, 1998.

[53] R. Meo. Theory of dependence values. *ACM Transactions on Database Systems*, 25(3):380–406, 2000.

[54] J.H. Moore, F.W. Asselbergs, and S. M. Williams. Bioinformatics challenges for genome-wide association studies. *Bioinformatics*, 26(4):445–455, 2010.

[55] J.H. Moore and M.D. Ritchie. The challenges of whole-genome approaches to common diseases. *JAMA The Journal of the American Medical Association*, 291(13):1642–1643, 2004.

[56] S. Morishita and A. Nakaya. Parallel branch-and-bound graph search for correlated association rules. In *Revised Papers from Large-Scale Parallel Data Mining, Workshop on Large-Scale Parallel KDD Systems, SIGKDD*, volume 1759 of *Lecture Notes in Computer Science*, pages 127–144. Springer-Verlag, 2000.

[57] S. Morishita and J. Sese. Transversing itemset lattices with statistical metric pruning. In *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS'00)*, pages 226–236. ACM Press, 2000.

[58] J. Neyman and E.S. Pearson. On the use and interpretation of certain test criteria for purposes of statistical inference: Part II. *Biometrika*, 20A(3/4):263–294, 1928.

[59] S. Nijssen, T. Guns, and L. De Raed. Correlated itemset mining in ROC space: a constraint programming approach. In *Proceedings the 15th ACM SIGKDD conference on Knowledge Discovery and Data Mining (KDD'09)*, pages 647–656. ACM Press, 2009.

[60] S. Nijssen and J.N. Kok. Multi-class correlated pattern mining. In *Proceedings of the 4th International Workshop on Knowledge Discovery in Inductive Databases*, volume 3933 of *Lecture Notes in Computer Science*, pages 165–187. Springer-Verlag, 2006.

[61] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th International Conference on Database Theory (ICDT'99)*, volume 1540 of

Lecture Notes in Computer Science, pages 398–416. Springer-Verlag, 1999.

[62] E.S. Pearson. The choice of statistical tests illustrated on the interpretation of data classed in a 2×2 table. *Biometrika*, 34(1/2):139–167, 1947.

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

[64] Jr. R. J. Bayardo and R. Agrawal. Mining the most interesting rules. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99)*, pages 145–154. ACM Press, 1999.

[65] K. Sadakane and G. Navarro. Fully-functional succinct trees. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*, pages 134–149. Society for Industrial and Applied Mathematics, 2010.

[66] J.P. Shaffer. Multiple hypothesis testing. *Annual Review of Psychology*, 46:561–584, 1995.

[67] E.H. Shortliffe and B.G. Buchanan. A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23:351–379, 1975.

[68] C. Silverstein, S. Brin, and R. Motwani. Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, 2(1):39–68, 1998.

[69] T.M.F. Smith. On the validity of inferences from non-random samples. *Journal of the Royal Statistical Society. Series A (General)*, 146(4):394–403, 1983.

[70] P. Smyth and R.M. Goodman. An information theoretic approach to rule induction from databases. *IEEE Transactions on Knowledge and Data Engineering*, 4(4):301–316, 1992.

[71] D.R. Thiruvady and G.I. Webb. Mining negative rules using GRD. In *Advances in Knowledge Discovery and Data Mining, Proceedings of the 8th Pacific-Asia Conference, (PAKDD 2004)*, volume 3056 of *Lecture Notes in Computer Science*, pages 161–165. Springer, 2004.

[72] G.J.G. Upton. A comparison of alternative tests for the 2 x 2 comparative trial. *Journal of the Royal Statistical Society. Series A (General)*, 145(1):86–105, 1982.

[73] R. Vilalta and D. Oblinger. A quantification of distance bias between evaluation metrics in classification. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML'00)*, pages 1087–1094. Morgan Kaufmann Publishers Inc., 2000.

[74] G. Webb, 2009. Personal communication.

[75] G. I. Webb. Discovering significant patterns. *Machine Learning*, 68(1):1–33, 2007.

[76] G.I. Webb. MagnumOpus software. `http://www.giwebb.com/index.html`. Retrieved 10.2. 2009.

[77] G.I. Webb. Discovering significant rules. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'06)*, pages 434–443. ACM Press, 2006.

[78] G.I. Webb and S. Zhang. K-optimal rule discovery. *Data Mining and Knowledge Discovery*, 10(1):39–79, 2005.

[79] S.S. Wilks. The likelihood test of independence in contingency tables. *The Annals of Mathematical Statistics*, 6(4):190–196, 1935.

[80] X. Wu, C. Zhang, and S. Zhang. Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems*, 22(3):381–405, 2004.

[81] X. Wu, C. Zhang, and S. Zhang. Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems*, 22(3):381–405, 2004.

[82] H. Xiong, S. Shekhar, P.-N. Tan, and V. Kumar. Exploiting a support-based upper bound of pearson's correlation coefficient for efficiently identifying strongly correlated pairs. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'04)*, pages 334–343. ACM, 2004.

[83] Y.Y. Yao and N. Zhong. An analysis of quantitative measures associated with rules. In *Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining (PAKDD'99)*, pages 479–488. Springer-Verlag, 1999.

[84] F. Yates. Test of significance for 2 x 2 contingency tables. *Journal of the Royal Statistical Society. Series A (General)*, 147(3):426–463, 1984.

[85] S.-J. Yen and A.L.P. Chen. An efficient approach to discovering knowledge from large databases. In *Proceedings of the fourth international conference on on Parallel and distributed information systems (DIS'96)*, pages 8–18. IEEE Computer Society, 1996.

Index