Inductive Causation based on Binary Data

student research project

Otto-von-Guericke-Universität Magdeburg
Department of Computer Science
Institute of Knowledge and Language Engineering
Neural Networks and Fuzzy Systems

MelTec GmbH

applicant
Sebastian Nusser

supervisors
Dr. Christian Borgelt, Otto-von-Guericke-Universität
Dr. Peter Karcher, MelTec GmbH

Contents

Chapter 1 – Introduction

From elementary statistics it is known that correlation does not imply causation. However, it was shown first by Reichenbach in [Reich1956] – but his publications are largely ignored – and later by Pearl and Cooper (e.g. in [Pearl1991] and [Cooper1997]) that, if more than two variables are observed, under certain conditions causal relationships can be inferred. This approach we call "inductive causation" in this report. Controversial discussions concerning the advantages and disadvantages are given in [CooGly1999] and [BorKru1999].

Among other objectives inductive causation can be used as an explorative tool to enhance the understanding of data and to generate working hypotheses. That means, rather than to infer a whole network of (causal) relationships, as for example in Bayesian networks, only the relationships among small subsets of variables are inferred and then the result is viewed as a causal rule set. Furthermore, the results will serve only as a guidance and working hypotheses for further experiments.

In the first chapter a short introduction to the topic of data analysis on relationships between variables is given. The second chapter introduces the MELK data, the notation and some further definitions. The third chapter describes the LCD-algorithm developed by Cooper [Cooper1997] and a possible enhancement given by Brin et al. [BSMU2000]. In Chapter 4 the application of Extended LCD for our purpose is discussed. Some measures for dependence and independence are introduced in Section 4.3, Section 4.4 describes our own implementation, and in Chapter 5 our results are discussed.

1.1 Different Methods for Data Analysis

There are many approaches that aim at making inferences about (causal) relationships among variables. Two commonly used approaches are association rules and Bayesian networks.

Association rule induction tries to find frequent item sets. It generates rules of the kind: "if variable A and variable B are observed, then variable C is likely to be observed, too". The most popular approach is called the Apriori-Algorithm. See [AgrSri1994] for an overview. One major application of association rules is market basket analysis, since association rule algorithms can handle large datasets. Unfortunately, association rules cannot be used to find causal relationships between variables.

Bayesian networks aim at describing relationships between variables which can be interpreted as possible causal relationships. They combine probability and graph theory. A Bayesian network, which is a special case of a graphical model, represents the joint distribution over a set of variables, where each variable is represented by a node in a graph. Edges represent the relationships between variables – missing edges correspond to independence relationships between variables. The direction of an edge can be interpreted as causal influence. See [Pearl1992] or [BorKru2002] for an overview over this topic.

Based on the same principle as used for Bayesian networks, Cooper [Cooper1997] and Brin et al. [BMSU2000] proposed algorithms that work only on three variables. The output for each triplet is either a certain causal relationship among the three variables, a set of possible relationships, or a "don't know" . Thus, the aim of these approaches is more conservative concerning its output, since there is no need to construct a complete network of causal relationship between all variables.

The reasons to use the approach of Cooper and Brin et al. are the difficulties that exists when interpreting edges as causal influences in other approaches that infer the whole (causal) network. See[CooGly1999] and [BSMU2000] for an overview.

1.2 Data Analysis for MelTec

MelTec is a bioscience company. The main research interest of MelTec is the development of biotechnological systems for the exploration of the human proteomics to enhance and facilitate the drug discovery and drug development process. It is possible to improve the accuracy and predictability in the drug discovery process by a better understanding of cellular networks and the interdependence of proteins.

The goal of this work is to assist biological and medical scientists in their analysis of MELK[1] data. The MELK data consists of experiments where biological markers stand for a certain proteins, protein classes or biological structures. However, the aim is not to find frequent groups of items, but to find certain markers (proteins) which are responsible for the occurrence of other markers.

To determine a complete causal model of the MELK data is computationally expensive. The approaches of Cooper and Brin et al. in comparison to association rules and complete Bayesian networks seem to be more useful for MelTec's purposes of an explorative tool. The scope of this work is to apply the approaches of Cooper and Brin et al. to MelTec's data and to assess their merits.

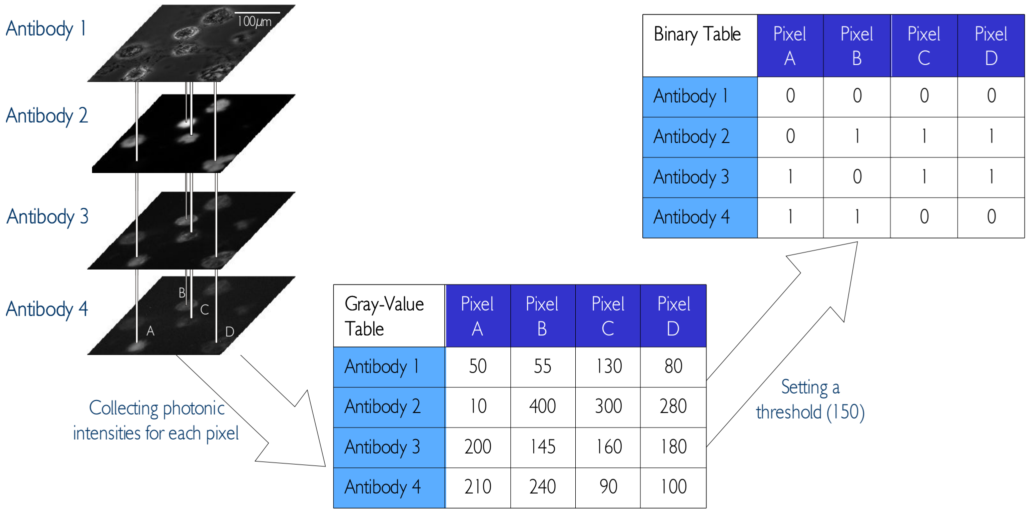
[1] MELK is an acronym, which stands for "Multi Epitop Ligand Kartierung".

Chapter 2 – MELK data and Basic Definitions

In Section 2.1 a short introduction to the MELK technology and the interpretation of its data as a multivariate probability distribution is given. Basic definitions concerning probability theory and hypothesis testing are introduced in Section 2.2 and Section 2.3. In Section 2.4 further prerequisites are given.

2.1 MELK data

Meltec's proprietary MELK robotic technology produces complex protein data in the field of bioscience. The MELK data is based on a biological sample for which a stack of n fluorescence images is produced. Each of the n images corresponds to one biological marker standing for a certain protein, a protein class or a biological structure (e.g. cell nucleus). The images within the stack are perfectly aligned – so the pixels with the same coordinates correspond to the same biological region. Then, for each image a binary image is derived, and for each stack pixel a binary vector is generated representing all markers for the same biological region. The process is shown in Figure 1.


Antibody 1
100µm
Antibody 2
Antibody 3
Antibody 4
B
C
A
D
Collecting photonic intensities for each pixel
Setting a threshold (150)

Gray-Value Table	Pixel A	Pixel B	Pixel C	Pixel D
Antibody 1	50	55	130	80
Antibody 2	10	400	300	280
Antibody 3	200	145	160	180
Antibody 4	210	240	90	100

Binary Table	Pixel A	Pixel B	Pixel C	Pixel D
Antibody 1	0	0	0	0
Antibody 2	0	1	1	1
Antibody 3	1	0	1	1
Antibody 4	1	1	0	0

Figure 1: From MELK images to binary data

The images are up to 2048 x 2048 pixel large, and there are up to 100 images in one stack. Thus, there can be up to four million 100-dimensional binary vectors. Therefore the efficiency of the algorithm is a crucial aspect.

2.2 Basic Definitions

Random variables will be denoted by capital letters (A,B,C and $A_1,\ldots,A_m$). The same notation is used for nodes in graphs. The observed values of these random variables are denoted by lowercase letters (a,b,c and $a_1,\ldots,a_m$).

P(A = a) denotes the probability of random variable A taking value a. P(A = a,B = b) stands for the joint event A = a and B = b. P(A = a|B = b) stands for the conditional probability of A = a given B = b. Whenever there is no danger of confusion, A is used as abbreviation of A = a.

Intuitively, independence means that knowing the value of one variable makes it neither more nor less probable that another random variable has a certain value. Formally, independence between random variables is defined as:

> Two random variables A and B are independent, if and only if
>
> P(A = a,B = b) = P(A = a) P(B = b)   for all a,b
>
> holds. This will be denoted by indep(A,B).

Conditional independence is defined as:

> The random variables A and B are conditionally independent given another random variable C, if and only if
>
> P(A = a,B = b|C = c) = P(A = a|C = c) P(B = b|C = c) for all a,b,c
>
> holds. indep(A,B|C) is used to denote this.

The following definition of dependence is commonly used:

> Two random variables A and B are dependent if they are not independent. This will be denoted by dep(A,B), and dep(A,B|C) respectively.

In Section 4.2 this definition will be extended.

2.3 Hypothesis Testing

The statistical procedure to make a decision between two contrary hypothesis about the process that generated a certain data set is called hypothesis testing. One hypothesis is called the null hypothesis H_0. As an example, for a population parameter θ, the null hypothesis could be that the population parameter is smaller or equal than a certain value θ_0 (H_0: $\theta \leq \theta_0$). The second hypothesis is called alternative hypothesis H_A. For the example the alternative hypothesis would be that the population parameter exceeds the value of θ_0 (H_A: $\theta > \theta_0$). The alternative hypothesis will be accepted if the observed data values are sufficiently improbable under the null hypothesis. Otherwise, the null hypothesis is not rejected. But not rejecting the null hypothesis is not equivalent to accepting the null hypothesis.

The decision to reject the null hypothesis is made by observing the value of some statistic whose probability distribution is known under the assumption that θ_0 is the true value of θ. Such a statistic is called test statistic. The critical region for a test represents the values of the test statistic that lead to rejection of the null hypothesis. The probability that the observed value of the test statistic will fall into the critical region by chance if $\theta = \theta_0$ is called the level of significance of the test α. In a hypothesis testing study, α is the probability of committing a Type I error (see below).

Two possible types of errors can occur:

- Type I: the null hypothesis H_0 is rejected, even though it is correct.
- Type II: the null hypothesis H_0 is accepted, even though it is false.

2.4 Further Prerequisites

For MELK data as defined in Section 2.1 the biological markers are viewed as binary random variables. The following contingency table are examples for observed data of a binary dataset with dataset size N. $n_{i.}$ denotes the sum over a row, and $n_{.j}$ denotes the sum over a column.

	$B = b_1$	$B = b_2$	
$A = a_1$	n_{11}	n_{12}	$n_{1.}$
$A = a_2$	n_{21}	n_{22}	$n_{2.}$
	$n_{.1}$	$n_{.2}$	N

$O_{i,j}$ will denote the random variable corresponding to an entry of a contingency table and p_{ij} denotes the probability $P(A = a_i, B = b_j)$.

For MELK data this table could be:

	B = 0	B = 1	
A = 0	20	300	320
A = 1	200	400	600
	220	700	920

where (B = 1,A = 0), for example, is the number of pixels, where protein B is expressed and protein A is not expressed.

Let (A,B) be random variables. Then let h be a function and f_A be a distribution function of A. The following definitions are made:

- the expected value
  $$E[h(A)] = \int h(a)\, f_A(a)\, da$$
- the mean for A
  $$\mu_A = E[A]$$
- the variance of A
  $$\sigma_A^2 = E[(A - \mu_A)^2]$$
- the covariance of A and B
  $$\sigma_{AB} = E[(A - \mu_A)(B - \mu_B)]$$

Let { (a_i,b_i) : i = 1, ... , n } be a set of observations of the random variables (A,B), then

- μ_A is estimated by
  $$\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i$$
- the variance of A is estimated by
  $$\mathrm{Var}(A) = \frac{1}{n-1}\left(\sum_{i=1}^{n} a_i{}^2 - n\bar{a}^2\right)$$
- the covariance of A and B is estimated by
  $$\mathrm{Cov}(A,B) = \frac{1}{n-1}\left(\sum_{i=1}^{n} a_i b_i - n\bar{a}\bar{b}\right)$$

Chapter 3 – The LCD-Algorithm and Extended LCD

The LCD-Algorithm by Cooper [Cooper1997] and the Extended LCD by Brin et al. [BMSU2000] are designed for an efficient discovery of possible causal relationships from large datasets. The techniques are based on the same principles as learning Bayesian networks. The LCD-Algorithm uses tests of dependence, independence and conditional independence to restrict possible causal relationships between variables. It can find rules of the type "A causes B and B causes C" (A → B → C) or "B causes both, A and C" (A ← B → C)[2]. The Extended LCD-Algorithm by Brin et al. in [BMSU2000] uses an additional rule type to find also rules of the type "B is caused by both, A and C" (B → A ← C).

In Section 3.1 we show the connection of the LCD-Algorithm and the Extended LCD to Bayesian networks. Further assumptions and the basic rules for causal discovery are given in Section 3.2.

3.1 Motivation through Bayesian Networks

Bayesian networks [Pearl1988] and [Neapol1990] are used to represent causal relationships among random variables. They combine the probability distribution over a set of random variables with mathematical graph theory. The graph for a Bayesian network is a directed acyclic graph (DAG)[3] where each edge from a node to another node can be interpreted as a (direct) causal influence. These influences are quantified by conditional probabilities. Constraint-based [Pearl1991] and [Sprites1993] as well as Bayesian methods [CooHer1992] were proposed for learning Bayesian networks from observed data.

Figure 2 (from [Kruse2004]) shows the structure of a possible causal Bayesian network, where V = {A,B,C,D,E} is the set of nodes and each node represents a certain system variable. It shows that metastatic cancer (A) can cause a brain tumor (C) which causally influences whether the patient falls into coma (D) or experiences severe headaches (E).

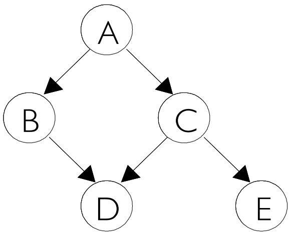


Figure 2: A - metastatic cancer, B - increased serum calcium, C - brain tumor, D - coma, E - severe headaches

[2] The actual algorithm of Cooper is slightly different from how it is presented here. However, dropping a few minor additional assumptions (e.g. it is known that a node has no causes), it is basically the same as given below.

[3] A directed acyclic graph is defined as graph G = (V,E) in which each edge in E has a direction and no node in V is its own ancestor.

Metastatic cancer can also lead to an increased serum calcium value (B) which makes it possible that the patients falls into coma, too.

Causal Markov Condition

Formally, independence relationships represented by the structure of a Bayesian network are given by the Markov condition (see [CooGly1999] Section 4.4.):

> Let Γ be a Bayesian network with node set V. Let P be a probability distribution over the nodes in V. The Markov condition is satisfied if and only if for every node A in V it holds that, according to P, node A is independent of its noneffects (nondescendants) in Γ given its direct causes (parents) in Γ.

For the Bayesian network shown in Figure 2 this means, that the chance of severe headaches (E) will be independent of metastatic cancer (A), if it is known whether the patient has a brain tumor (C) or not.

The Markov condition permits the factorization of a joint probability distribution [Pearl1988]:

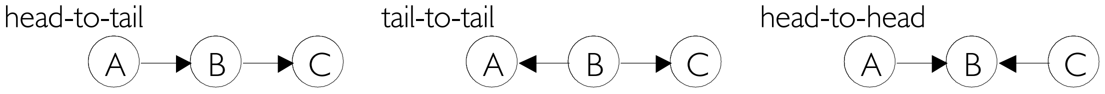
$$P(A_1, A_2, \ldots A_n) = \prod_{i=1}^{n} P(A_i \mid \text{parents}(A_i))$$

Parents(A_i) denotes the set of nodes with edges pointing to A_i. If A_i has no parents, parents(A_i) is the empty set and thus P(A_i | parents(A_i)) will be P(A_i).

d-Separation Criterion

In the theory of Bayesian networks the so-called d-separation criterion plays an important role. It serves as a method to read the independence statements from the causal structure given by the graph, that will hold in the corresponding probability distribution. It captures all conditional independence relationships that are implied by the Markov condition and it allows to determine whether two variables are conditionally independent given a set S of variables. This is possible if they are d-separated in the causal structure by S.

In Bayesian networks there are three different types for connections between A and B and B and C, where A,B,C are nodes (or sets of nodes) in a causal structure.



Formally, d-separation[4] is defined as [Kruse2004]:

> Let G = (V,E) be a directed acyclic graph, and A,B,C disjoint sets of nodes in V. A and C are d-separated by a set S ⊆ V \ {A,C}, if all paths[5] π_i between A and C are blocked by one of the following conditions:

[4] Remark: The Markov condition and the d-separation criterion are equivalent, if and only if the probability distribution is strictly positive (there are no zero probabilities).

[5] In a graph a path is a sequence of nodes such that from each of the path's nodes there is an arc to the successor node.

a) A path π is blocked by S, if at least one pair of sequenced edges in π is blocked.

b) Two head-to-tail or tail-to-tail edges with common node B are blocked if B is in S.

c) Head-to-head edges with common B are blocked if B and all descendants of B are not in S.

If S ⊆ V \ {A,C} d-separates the nodes A and C, then A and C are independent given S.

If A and C are not d-separated by S, there is no statement given by the d-separation criterion about the independence or dependence of A and C given S. It is possible that there are even more independences in the corresponding probability distribution.

Even though neither Cooper [Cooper1997] nor Brin et al. [BMSU2000] directly motivated their rule types through the d-separation criterion, the d-separation criterion seems meaningful for two reasons:

1. It shows directly the connection to Bayesian networks.
2. It turns out that the weakness of both approaches might be improved by adopting ideas from Bayesian networks.

3.2 Determining Causal Relationships

Two additional conditions are necessary to constrain possible causal relationships in a given dataset: the causal faithfulness condition and the statistical testing assumption.

Causal Faithfulness Condition

The causal faithfulness condition specifies the probabilistic dependence relationships between variables:

Variables are independent only if their independence is implied by the causal Markov condition.

According to the causal faithfulness condition, in Figure 2, examples of dependence relationships are: i) metastatic cancer and brain tumor are dependent [dep(A,C)], ii) metastatic cancer and severe headaches are dependent [dep(A,E)], and iii) severe headaches and coma are dependent [dep(D,E)]. An explanation for iii) is that having severe headaches increases the chance of having a brain tumor which increases the chance to falling into coma. So the two variables D and E become dependent because they have the common cause C (i.e. a so-called confounder[6]).

Statistical Testing Assumption

The following assumption is given in [ManCoo2001]:

[6] A confounder suggests a direct (causal) dependence relationship between variables, but there is no such relationship between this variables.

> A statistical test performed to determine independence (or alternatively dependence) given a finite dataset will be correct relative to independence (dependence) in the joint probability distribution that is defined by the causal process under study.

The causal Markov condition and the causal faithfulness condition describe probabilistic relationships of dependence and independence, but in causal discovery there is no certain knowledge about probabilistic relationships among variables, since one has only a finite amount of data. Therefore, in order to make inferences, one has to assume that the dependence and independence relationships in the underlying probability distribution are correctly identified by statistical tests.

CCC Causality

Cooper [Cooper1997] proposes rules based on dependency statements and the test for conditional independence, in order to determine whether there is a causal relationship between dependent variables. This is the only rule type used in the original LCD-Algorithm and it is called CCC rule. CCC stands for three correlations among the variables.

> Let A, B and C be variables that are pairwise dependent. If A and C are independent conditioned on B, then – in the absence of hidden and confounding variables – it is possible to infer that one of the following causal relations exists between A, B and C:
>
> $$A \rightarrow B \rightarrow C \quad A \leftarrow B \rightarrow C \quad A \leftarrow B \leftarrow C$$
>
> If there is no cause for A, one can assume that only the first rule is possible – even if there are hidden or confounding variables.

If it is known that, for example, the variable A has no causes, the only possible CCC rule is A → B → C. This makes it possible to reduce the number of relationships to be tested afterwards. Additional enhancements and further performance improvements for the original LCD-Algorithm – but not for the Extended LCD – are discussed in [ManCoo2001].

CCU Causality

Brin et al. (in [BMSU2000]) extended the LCD-Algorithm by another rule; the so called CCU rule. CCU stands for two correlated and one uncorrelated variable pairs. It makes it possible to find head-to-head structures in the data.

> Let A, B and C be variables such that A and B are dependent, A and C are dependent, and B and C are independent. If B and C become dependent conditioned on A, then, in the absence of hidden and confounding variables, it is possible to infer the following causal relationship:
>
> $$B \rightarrow A \leftarrow C$$

The CCC rule and the CCU rule determine causal relationships of triplets that fulfill the d-separation criterion of a Bayesian network. Therefore, the Extended LCD goes through all triplets and tests each rule. If a CCC rule (e.g. A → B → C) is found, it may be that there is a hidden or confounding variable D mediating between B and C (e.g. A → B → D → C or A → B ← D → C). However, this result is not useless, since B and C are still causally related – even though indirectly.

Extended LCD [BMSU2000]

```
Input:      V – set of variables
Output:     list of possible causal relationships as triplets

for all A,B,C in V with $A \neq B$ and $A \neq C$ and $B \neq C$
    if dep(A,B) and dep(A,C) and dep(B,C) and indep(A,C|B) then        // (CCC-rule)
        output '$A \rightarrow B \rightarrow C$  or  $A \leftarrow B \rightarrow C$  or  $A \leftarrow B \leftarrow C$'
    endif
    if dep(A,B) and indep(A,C) and dep(B,C) and dep(A,C|B) then        // (CCU-rule)
        output '$B \rightarrow A \leftarrow C$'
    endif
endfor
```

Chapter 4 – Application of Extended LCD

In Section 4.1 a motivation for not using the standard chi-squared test for independence and why to use "strong dependence" is given. Section 4.2 introduces "strong dependence" and "weak independence". Alternative measures for dependence or independence and their suitability for our purpose are discussed in Section 4.3. The implementation of our modified LCD-Algorithm (MelCD) is given in the last section.

4.1 Motivation

The LCD-Algorithm and its extension by Brin et al. are based on independence and dependence statements. Usually, variables are defined as dependent if and only if they are not independent. This can lead to a – at the first glance – counterintuitive behavior, since a small deviation from the null hypothesis can be significant. The following tables of observed values that have the same underlying probability distribution may serve as an example.

	$B = b_1$	$B = b_2$
$A = a_1$	49	51
$A = a_2$	51	49

Table 1

	$B = b_1$	$B = b_2$
$A = a_1$	98	102
$A = a_2$	102	98

Table 2

	$B = b_1$	$B = b_2$
$A = a_1$	4900	5100
$A = a_2$	5100	4900

Table 3

Using the standard chi-squared test (see Section 4.3) to test for independence, one gets no indication of dependence (Table 1), a borderline indication of dependence (Table 2), and a strong indication of dependence (Table 3). This seems to be counterintuitive, since the underlying probability distribution of variables A and B stays the same. Only the number of observations increases. This is a general phenomena in statistical testing, and the correct interpretation is that in Table 1 "there is not enough evidence (data) to reject the independence assumption". For Table 3 there is "significant evidence for rejecting independence". This means, statistical testing can detect even the weakest deviation from the null hypothesis (in our case the assumption of independence), provided that enough data is observed. For the MELK data we usually have a large N; N = 2048 x 2048 pixel. However, detecting even the weakest deviation from the null hypothesis is unsatisfactory for our purposes, since it seems plausible for us to ignore "weak dependencies". Therefore, "strong dependence" will be defined in Section 4.2.

Our second major modification of the Extended LCD concerns the – for this purpose commonly used – chi-squared test of independence. Due to the fact that the chi-squared test has certain assumptions which can be violated – especially in an automated process that uses the chi-squared test many times – we looked at other measures for testing independence. They are introduced and investigated in Section 4.3.

4.2 Strong Dependence and Weak Independence

We want to ensure that dependence statements are strong dependencies between variables, since for CCC and CCU rules, which require dependencies between variables, it is intuitively clear that the stronger the dependence of its parts, the "stronger" the rule itself. In other words: the weakest dependence statement determines the confidence in the rule itself.

Instead of using the standard definition of dependence as given in Section 2.2, we introduce "strong dependence". Strong dependence is defined as follows:

> Let A and B be random variables. Let T(A,B) be a measure for independence that takes the value t_{indep} when A and B are independent. A and B defined to be "strongly dependent", if and only if:
>
> $$T(A,B) - t_{indep} < {}^{d}\theta_{neg.} \quad \text{or} \quad T(A,B) - t_{indep} > {}^{d}\theta_{pos.},$$
>
> where $\theta_{dep} = ({}^{d}\theta_{neg.}, {}^{d}\theta_{pos.})$. This will be denoted by $dep_{T,\theta[dep]}(A,B)$ or shortly $dep_{T,\theta}(A,B)$.

Corresponding to the strong dependence, weak independence will be defined as:

> Let A and B be random variables. Let T(A,B) be a measure for independence that takes the value t_{indep} when A and B are independent. A and B defined to be "weakly independent", if and only if:
>
> $$T(A,B) - t_{indep} > {}^{i}\theta_{neg.} \quad \text{and} \quad T(A,B) - t_{indep} < {}^{i}\theta_{pos.},$$
>
> where $\theta_{indep} = ({}^{i}\theta_{neg.}, {}^{i}\theta_{pos.})$. This will be denoted by $indep_{T,\theta[dep]}(A,B)$ or shortly $indep_{T,\theta}$ (A,B).

Note, that ${}^{d}\theta_{neg.}$ and ${}^{d}\theta_{pos.}$ for strong dependence and ${}^{i}\theta_{neg.}$ and ${}^{i}\theta_{pos.}$ for weak independence are not the same. Furthermore, it makes only sense if ${}^{d}\theta_{neg.}$ for strong dependence is smaller or equal than ${}^{i}\theta_{neg.}$ for weak independence, and ${}^{d}\theta_{pos.}$ for strong dependence is greater or equal than ${}^{i}\theta_{pos.}$ for weak independence.

Whenever the meaning is clear, dep(A,B) and indep(A,B) will be used to denote $dep_{T,\theta}(A,B)$ and $indep_{T,\theta}(A,B)$, respectively.

According to the definitions of strong dependence and weak independence θ_{indep} serves as cutoff tuple for weak independence and θ_{dep} serves as cutoff tuple for strong dependence. For values between independence and dependence cutoff, we define the relationship as "don't know". Thus, the following hypothesis test with the null hypothesis H_0 and the alternative hypothesis H_A will be used:

hypothesis test for weak independence ($\theta_{\text{indep}} = ({}^{i}\theta_{\text{neg.}}, {}^{i}\theta_{\text{pos.}})$):

H_0: $T(A,B) - t_{\text{indep}} \notin [{}^{i}\theta_{\text{neg.}}, {}^{i}\theta_{\text{pos.}}]$

H_A: $T(A,B) - t_{\text{indep}} \in ({}^{i}\theta_{\text{neg.}}, {}^{i}\theta_{\text{pos.}})$

hypothesis test for strong dependence ($\theta_{\text{dep}} = ({}^{d}\theta_{\text{neg.}}, {}^{d}\theta_{\text{pos.}})$):

H_0: $T(A,B) - t_{\text{indep}} \in [{}^{d}\theta_{\text{neg.}}, {}^{d}\theta_{\text{pos.}}]$

H_A: $T(A,B) - t_{\text{indep}} \notin ({}^{d}\theta_{\text{neg.}}, {}^{d}\theta_{\text{pos.}})$

θ_{dep} and θ_{indep} allows the user to define a “level of belief” in a dependence and independence relationship, and one can define a range of non-interesting values which are labeled as “don't know”.

4.3 Different Measures for Independence and Dependence

Chi-squared Test

The chi-squared test is a standard test for independence. The chi-squared test measures the degree of independence between different variables. For this purpose, it compares the observed case with the expected cases.

$$X = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\left(E_{i,j} - O_{i,j}\right)^2}{E_{i,j}} \quad \dot{\sim} \quad \mathbf{X}^2{}_1$$

where $O_{i,j}$ are the observed values as defined in Section 2.4, and $E_{i,j}$ is the expected value under the null hypothesis of independence and is estimated by:

$$E_{i,j} = \frac{n_{i.}\, n_{.j}}{N}$$

$\mathbf{X}^2{}_1$ is the chi-squared distribution with one degree of freedom.

The chi-squared test has the following disadvantages:

X is only approximately (as N → ∞) a chi-squared-distributed random variable. For small N the approximation can be poor, especially if the expected values are small. The following rule is frequently given in statistics books (e.g. in [MilArn1990] p. 581). It recommends the use of chi-squared test only if:

- all cells in the contingency table have an expected value greater than 1
- at least 80% of the cells in the contingency table have an expected value greater than 5

These conditions have to be checked each time the test is applied. In our case, the test for independence will be applied many times, and it is crucial not to ignore the validity of the test. This is particular the case when testing for conditional independence, since the cell frequencies will become smaller when we condition on variables.

Pearson Correlation

One measure used to determine whether or not two random variables are linearly dependent is the sample estimate of the Pearson correlation. The Pearson correlation is defined as:

$$\rho = \frac{\sigma_{A,B}}{\sqrt{\sigma_A^2 \sigma_B^2}} \quad \text{and estimated by} \quad R = \frac{\mathrm{Cov}(A,B)}{\sqrt{\mathrm{Var}(A)\,\mathrm{Var}(B)}}$$

The Pearson correlation assumes values between -1 and 1. Values close to 1 and -1 indicate a strong positive and negative dependence. For independent variables the parameter ρ is zero. A value of zero, however, does not imply independence between two variables in every case, since the relationship could be non-linear.

In the binary case, there is the following relationship between the Pearson correlation and the chi-squared test. A proof is given in [BMSU2000], Appendix A.

$$\mathrm{X}^2(A,B) = N \cdot \rho(A,B)^2$$

where X^2(A,B) and ρ(A,B) denotes the computation for the chi-squares test and the Pearson correlation, respectively.

A probability distribution for the Pearson correlation is given, for example, in [MilArn1990]. It is shown, assuming (A,B) are bivariant normal, that the following transformation for the Pearson correlation is approximately normal distributed.

$$\frac{1}{2}\ln\left(\frac{1+R}{1-R}\right) \stackrel{\cdot}{\sim} N(\mu,\sigma) \quad \text{, where} \quad \mu = \frac{1}{2}\ln\left(\frac{1+\rho}{1-\rho}\right) \quad \sigma = \sqrt{\frac{1}{N-3}}$$

Unfortunately, there are some problems using the Pearson correlation with binary data. This holds especially in the case of a conditional test, since the variance in a vector may be zero. In this case the Pearson correlation is undefined. In some special cases the Pearson correlation does not provide an intuitive mapping of the relationship between the variables. It becomes small, even though the vectors seem to be strongly related – see the following example.

$$\begin{array}{l} A = (1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1) \\ B = (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0) \end{array} \qquad \rho = -0.0667$$

Nevertheless, the Pearson correlation is a useful measure for dependence and independence.

Scalar Product

The scalar product is a widely used measure for assessing the similarity of vectors (especially in the field of text mining). Regard the example above – where A and B seem to be strongly related – the scalar product provides a more intuitive interpretation of the relationship between two vectors than the Pearson correlation does.

In the binary case and according to Section 2.4 the scalar product simplifies to:

$$\frac{n_{11}}{N} + \frac{n_{22}}{N} - \frac{n_{12}}{N} - \frac{n_{21}}{N}$$

The value of the scalar product is in the domain [-1,1]. 1 stands for similarity between two vectors and -1 stands stands for the case where one vector is the converse of the other vector.

Unfortunately, for certain contingency tables representing dependent and independent data the scalar product returns the same value. It is not possible to determine whether variables are dependent or independent (the following table shows an example), since $(x+\gamma) + (x-\gamma) - x - x = 0$ holds for all $\gamma \leq x$, but for a large γ one would like to infer dependence.

$x + \gamma$	x
x	$x - \gamma$

We can conclude that the scalar product is not a useful measure for dependence and independence.

Log Odds Ratio

The odds are defined as the quotient of the probability of success divided by the probability of failure – or as the quotient of number of successes divided by the number of failures.

The odds ratio is defined as the ratio for odds for variable A when $B = b_1$ and $B = b_2$. Using the notation of Section 2.4, this becomes:

$$\Theta = \frac{P(A = a_1 \mid B = b_1)}{P(A = a_2 \mid B = b_1)} \bigg/ \frac{P(A = a_1 \mid B = b_2)}{P(A = a_2 \mid B = b_2)} \quad \text{and estimated by} \quad OR = \frac{n_{11}\, n_{22}}{n_{12}\, n_{21}}$$

Using the natural logarithm of the odds ratio one gets the so called log odds ratio (logOR). The log odds ratio is zero for independent data. Large positive or negative values indicate dependent data. An advantage of taking the natural logarithm of the odds ratio is that the log odds ratio is approximately normal distributed for large values of N.

$$\ln\left(\frac{O_{11}\, O_{22}}{O_{12}\, O_{21}}\right) \;\dot{\sim}\; N(\mu,\sigma) \quad \text{, where} \quad \mu = \ln\left(\frac{p_{11}\, p_{22}}{p_{12}\, p_{21}}\right) \quad \sigma = \sqrt{\frac{1}{N}\left(\frac{1}{p_{11}} + \frac{1}{p_{12}} + \frac{1}{p_{21}} + \frac{1}{p_{22}}\right)}$$

In general, the log odds ratio provides good results. The disadvantage is that it works only on binary data.

4.4 Implementation of MelCD

Our implementation is based on the Extended LCD algorithm (see Section 3.2). Furthermore, the concepts of "strong dependence" and "weak independence" are included. The log odds ratio with and without bootstrapping, and the Pearson correlation (which is linearly related to the chi-squared test in the binary case – [BMSU2000], Appendix A) are used as measures for independence. Since the Pearson correlation has for extreme contingency tables a numerical advantage over the chi-squared test, only the Pearson correlation and the log odds ratio are implemented. From hereon, our algorithm will be called MelCD (for "MelTec Causal Discovery").

Generating rule-based Datasets

To test MelCD datasets are generated synthetically. Therefore an algorithm was implemented which allows to specify the following parameters: the underlying network structure of a data set, an activation function for each node, the variance of random noise, and the number of observations to be generated. Due to this, it is possible to test how scalable the algorithm is and it allows to verify which rules should be found – and which should not be found.

The network structure differentiates nodes of the first layer[7] and of further layers. The nodes in the first layer are initialized with 0 and 1 according to a distribution that is defined by the user. The further nodes are connected to a subset of nodes from previous layers. The connections can be chosen freely, but cycles are not permitted.

The definition of the network structure is based on the following steps:

- Initialization of the first layer:
  The initial activation of the nodes of first layer is generated according to the distribution table specified by the user.
- Propagating the initial values from the first layer through the rest of the network:
  For each node which is not in the first layer its own value is determined by its parents, its activation function act(A), and its noise parameter ε.

The activation function[8] describes the state whether a node in the network is active or inactive. In our case the activation function replaces the probability tables, which are used in Bayesian networks to describe the relationships among the nodes, and it is defined as:

$$\mathrm{act}(A) = f\left(g(B_1), g(B_2), \ldots, g(B_k)\right)$$

where $B_i \in \mathrm{parents}(A)$, k is the number of parents of A and $g(X) = \mathrm{act}(X) - \frac{1}{2} + \mathbf{E}$ with $\mathbf{E} \sim N(0,\varepsilon)$. The function f can be chosen freely, but $f: R \times R \times \ldots \times R \rightarrow \{0,1\}$ must hold. Usually a majority decision among the parent nodes is used as function f, but other functions are also possible. In the case of a binary function f, as mentioned below, the values of $g(X)$ must be converted to binary values. The resulting datasets are represented by binary matrices.

Searching for Causal Rules

Our implementation is based on the Extended LCD (see Section 3.2) and it uses the following steps to find casual rules:

- Determine positive and negative dependence, independence, and “don't known” relationships between all variable pairs. The results are stored in an extended adjacency matrix, where positive and negative dependence relationships between the variables are distinguished.
- Test for CCC and CCU rules based on triplets which comply with the unconditioned statements in the CCC and CCU rules and test for conditional independence to conclude

[7] The nodes of the first layer can be interpreted as the parentless nodes of the network.

[8] The term activation function is a commonly used concept in the theory of Artificial Neuronal Networks (see e.g. [KröSma1996] for details).

a CCC rule or test for conditional dependence to conclude a CCU rule – “don't know” relationships will be ignored.

To determine dependence and independence relationships, the concept of weak independence and strong dependence is used according to Section 4.2. As a simplification $\theta_{neg.} = - \theta_{pos.}$ is defined for our test, since the Pearson correlation and the log odds ratio are symmetric around zero ($t_{indep} = 0$). θ_{indep} and θ_{dep} will be denoted as θ_{lower} and θ_{upper}, respectively. Thus, the hypothesis tests will change to:

hypothesis test for weak independence:

H_0: $T(A,B) \notin [-\theta_{lower},\theta_{lower}]$

H_A: $T(A,B) \in (-\theta_{lower},\theta_{lower})$

hypothesis test for strong dependence:

H_0: $T(A,B) \in [-\theta_{upper},\theta_{upper}]$

H_A: $T(A,B) \notin (-\theta_{upper},\theta_{upper})$

Evaluation Measures

The results for the log odds ratio cannot be directly compared to the Pearson correlation without further conditions, since the log odds ratio takes values in $(-\infty,\infty)$ and the Pearson correlation has the domain [-1,1]. So it was necessary to find a transformation to make the cutoff values commensurable. Figure 3 shows a plot of Pearson correlation versus log odds ratio of synthetic datasets generated by networks as described above. A linear relationship with slope 6 seem to be plausible. This makes it possible to get comparable cutoff-values and thus to compare the results of both measures, the Pearson correlation and the log odds ratio.

To evaluate the quality of the results we used a scoring function for each rule. Each CCC or CCU rule is derived by testing four dependence or independence statements. For each test a score is computed by:

$$\text{score}(A,B) = \frac{|T(A,B) - \theta|}{\theta}$$

where θ is the cutoff value, which is used to decide whether there is dependence or independence, and T(A,B) is the used measure for independence between the variables A and B. This function can serve as a heuristic to find the most promising results. For the four tests for a CCC or CCU rule the minimum and the median value are given as output. This scoring function provides the facility to sort the output according to its (possible) importance.

Test Using the Pearson Correlation

The approximate distribution of the Pearson Correlation allows to test for independence, “don't know” and dependence, where r is the observed correlation in the dataset, and $\rho_{\theta[lower]}$ and $\rho_{\theta[upper]}$ the confidence interval limits depending on the freely selectable cutoff values θ_{lower} and θ_{upper} and a certain significance level α.

The following tests are performed to determine if there is independence, dependence or neither.

independence:

H_0: $|r| \geq \rho_{\theta[lower]}$

H_A: $|r| < \rho_{\theta[lower]}$

dependence:

H_0: $|r| \leq \rho_{\theta[upper]}$

H_A: $|r| > \rho_{\theta[upper]}$

“don't know”:

$\rho_{\theta[lower]} < |r| < \rho_{\theta[upper]}$

Test Using the Log Odds Ratio

The distribution of the log odds ratio is only an approximation and it holds only for large N. The approximation is poor for small N and for extreme contingency tables. We decided to use bootstrapping[9] for small N and extreme contingency tables, and to use the approximation in all other cases, since bootstrapping is time consuming. To decide whether the approximation can be used – or whether it is necessary to use bootstrapping – we investigated different measures based on the contingency table, which will be analyzed.

The quality of the approximation of the log odds ratio (see Section 4.3 – Log Odds Ratio) as a normal distributed random variable depends on two parameters: p_{min} and N, where p_{min} is the minimal cell proportion. Therefore it seems plausible to require the minimal relative cell frequency to be "large enough". An approximate distribution for a proportion is given by:

$$\hat{p} \quad \dot{\sim} \quad N\left(p, \sqrt{\frac{p(1-p)}{N}}\right)$$

where p is estimated by n/N. Thus, the lower 99% confidence interval limit is:

$$c_{lower} = \frac{n_{min}}{N} - 2.58\sqrt{\frac{(n_{min})(N-n_{min})}{N^3}} \quad , \text{where} \quad n_{min} = \min\{n_{ij} \,|\, i=1,2, j=1,2\}$$

Therefore we plotted (see Figure 4) for simulated table data c_{lower} versus $error_{logOR}$. $error_{logOR}$ is defined as:

$$error_{logOR} = \frac{\left|ci_{\alpha}^{approx} - ci_{\alpha}^{bootstrap}\right|}{logOR}$$

where ci_{α} is the confidence interval limit for the log odds ratio, which is calculated by the approximation or by the bootstrapping method.

This shows that for a value of 0.025 the approximation error is acceptable. Thus this value is used to decide whether we use the approximation or the bootstrapping method is necessary.

The following tests are performed to determine if there is independence, dependence or neither, where l is the observed log odds ratio value for the data set, and the confidence interval limits are $\lambda_{\theta[lower]}$ and $\lambda_{\theta[upper]}$ depending on the freely selectable cutoff values θ_{lower} and θ_{upper} and a certain significance level α.

independence:	dependence:	"don't know":
H_0: $\|l\| \geq \lambda_{\theta[lower]}$	H_0: $\|l\| \leq \lambda_{\theta[upper]}$	$\lambda_{\theta[lower]} < \|l\| < \lambda_{\theta[upper]}$
H_A: $\|l\| < \lambda_{\theta[lower]}$	H_A: $\|l\| > \lambda_{\theta[upper]}$	

[9] Bootstrapping ([Efron1994] and [DavHin1997]) is a resampling strategy, where the original sample is viewed as the empirical distribution from which new samples can be drawn. For this resampled data it is possible to calculate, for example, a confidence interval of arbitrary parameters. In our case, we calculate a confidence interval for the log odds ratio.

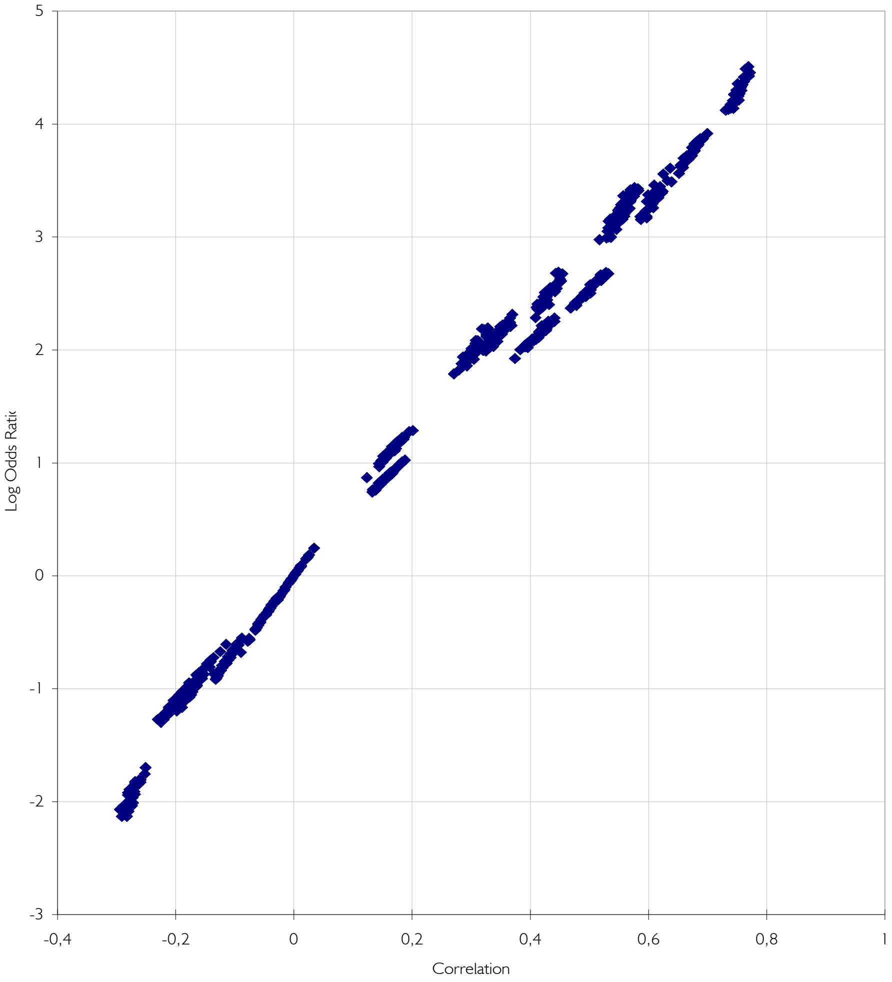


Figure 3: Plot for estimating a linear factor between Pearson correlation and log odds ratio based on synthetic datasets generated by synthetic networks

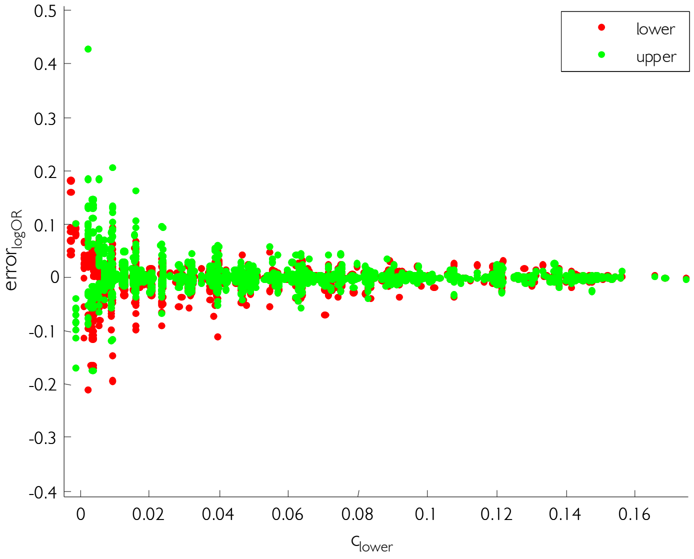


Figure 4: Estimation for decision if approximation or bootstrapping should be used (for random contingency tables and lower and upper bound)

Chapter 5 – Results

In this chapter we discuss our results and point out problems of MeICD.

As mentioned above, we used synthetic data sets to evaluate the MeICD algorithm. At least, one would expect from the algorithm that it finds rules, which are used as basic definition of the network. Furthermore, one would expect rules, where a node in a causal chain is left out, since the algorithm tests only for three variables and a causal chain can consists of even more variables.

We classified the results in four groups (in parenthesis the abbreviations used in Appendix A, Table 4 and Table 5 are given):

- type 1: “rule is in rule set” (+): the rule which is found by the algorithm is one of the rules which were used to generate the dataset.
- type 2: “rule is not in rule set, but it seems to be useful” (~): a rule of this type satisfies the d-separation criterion in the graph given by the generative rule set.
- type 3: “wrong direction rule” (wd): the rule is of type 1 or type 2, but the direction of its edges is incorrect.
- type 4: “rule is not in rule set and does not seem to be useful” (-): all other rules.

In our point-of-view the “best result” would be that most of the rules given by the rule set (type 1) are found. Furthermore, there can be type 2 rules in the best result. To be more realistic, a good result for our application is that the algorithm has found type 1 and type 2 rules and it has found no or only a few type 3 and type 4 rules. As an example, we would declare a result with only three type 1 rules a “better” than a result with more type 1 rules but also with a few type 4 rules. This makes sense for us, since our goal is to generate useful working hypothesis and not to exploit the complete version space.

During the experiments the Pearson correlation and the log odds ratio were used as measures for independence and dependence. The chi-squared is not used, since a linear relationship between chi-squared test and Pearson correlation is known (see Section 4.3).

In our experiments[10] we used as standard parameters a significance level $\alpha = 0.01$ to estimate the confidence interval, a dataset size of 10 000, and a noise parameter $\varepsilon = 0.4$ which corresponds to “medium noise” in the dataset. With more noise it is clear that the number of rules that will be found is smaller, since the algorithm finds fewer dependence relationships among the variables.

[10] All experiments are performed on an Athlon system with 1250MHz running under Windows 2000 with 1024 Mbyte main memory and Matlab 6.5 R13.

With less than "medium noise" and certain activation functions we discovered the strange effect that more "not useful" rules are found. This effect results due to the fact that the child nodes are more or less "clones" of their parents. So there are dependence relationships among these child nodes which should not be there. An example is given in Figure 5: the CCU-rule $A_2 \rightarrow A_4 \leftarrow A_8$ is found, since node A_8 is a clone of node A_6 which is a combination of node A_1 and node A_3 and the nodes A_1 and A_3 are parents of node A_4.

In Figure 6 one can see that there are more dependences among the nodes found than there are given by the M9 network. This is obvious, since the dependence relationships are propagated through the network. Due to this it is also clear that rules like $A_1 \rightarrow A_9 \leftarrow A_3$ are found by the algorithm, too. However, the dependence relationships between A_1 and A_9, and A_3 and A_9 are weaker than between A_1 and A_6, and A_3 and A_6. So it is possible to infer by the scoring values that A_6 mediates between A_1 or A_3 and A_9. Fortunately the rules $A_1 \rightarrow A_6 \rightarrow A_9$ and $A_3 \rightarrow A_6 \rightarrow A_9$ are found, too. (see Appendix A – Tables 4 and 5)

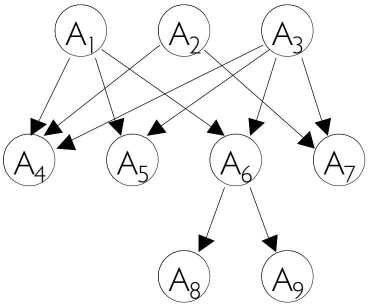


Figure 5: Network structure of M9 network (for a complete definition see Appendix B)

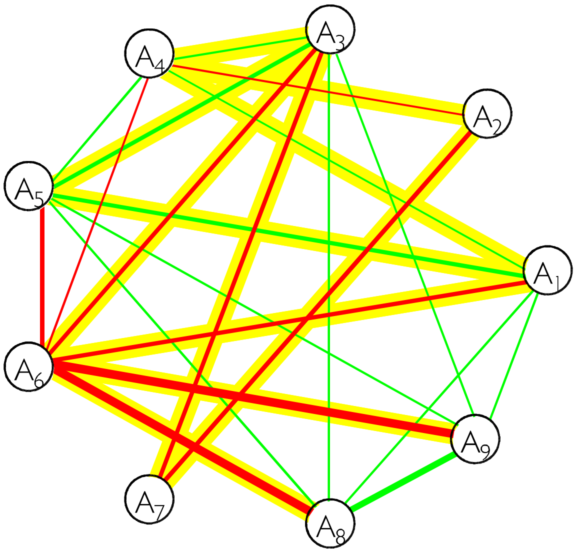


Figure 6: Dependency graph as representation of an adjacency matrix for a dataset generate by the M9 network (red edges represent negative relationships, green edges represent positive relationships and the thickness of these edges represent how strong this relationship is; yellow edges are known edges from the generative RuleSet)

In Table 4 and Table 5 the results of a test run with 10 different dataset generated by the M9 network are given for both test statistics, the Pearson correlation and the log odds ratio. The M9 network serves as an example – however, we discovered that the following parameter sets for the lower and the upper limit provide the best results for all networks we have tested:

- $\theta_{\text{lower}} = 0.2$ and $\theta_{\text{upper}} = 0.2$
- $\theta_{\text{lower}} = 0.1$ and $\theta_{\text{upper}} = 0.2$
- $\theta_{\text{lower}} = 0.1$ and $\theta_{\text{upper}} = 0.1$

Depending on the network structure one of them provides the best results. In the case of the M9 network (see Figure 5) the cutoff set (0.1,0.2) provides the best results for both measures, the Pearson correlation and the log odds ratio.

The CCU rule $A_1 \rightarrow A_6 \leftarrow A_3$ may serve as an example for Table 4: with the parameter set $\theta_{\text{lower}} = 0.2$ and $\theta_{\text{upper}} = 0.3$ the rule is not found. With $\theta_{\text{lower}} = 0.2$ and $\theta_{\text{upper}} = 0.2$ the rule is found in 100 percent of the runs, where the average minimal scores over all runs is 0.18 and the average median over all scores is 1.13. Since the minimal score is the same for the parameter set $\boldsymbol{\theta}_{\text{lower}} = 0.1$ and $\boldsymbol{\theta}_{\text{upper}} = 0.2$ and the minimal scores for rules with $A_1 \rightarrow A_6$ or $A_3 \rightarrow A_6$ are greater than 0.18, it seems plausible that the weakest dependence, in this example, is found for dep(A_1,A_3| A_6).

It seems that the number of rules of the Pearson correlation is in general larger than the number of results returned by the log odds ratio. The results of the Pearson correlation are not better than the results of the log odds ratio, since the Pearson correlation finds more non-type 1 rules. It is possible to rank the results by a scoring function.

How often a rule is found during several runs can be varying – rules with a low frequency are often rules of type 2 and type 4, this could make it possible to use the frequency over several runs also as scoring function to remove possibly bad results – but due to this it is possible to loose also type 1 rules. Unfortunately, like type 1 rules, type 3 and type 4 rules can have a high frequency, too.

In combination with the scoring function it is possible to rank the rules by using the minimum and the median distance values to the cutoff values. We discovered that most of the type 1 and type 2 rules have a higher score than rules of type 4. Unfortunately the rules of type 3 can have a high score, too. So it is necessary to have an expert to check the results.

To restrict the number of results makes sense only, if there are many more rules than the expert can handle.

Between the rules one can find in the result set, there can be differences between both measures, the Pearson correlation and the log odds ratio. It seems to be a good idea to use not only a single measure, to gain a better exploitation of the domain of results.

Known Problems

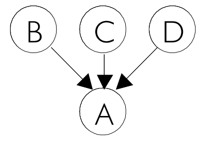


Figure 7

Our experiments show that not all rules implied by the initial network can be found. It is a problem to find certain rules in graphs with inhibiting edges, e.g. in Figure 7 the rule B → A ← C. The dependences between A and B, A and C, and the independence between B and C are found by the algorithm, but the algorithm is not able to find the conditional dependence relationship between B and C given A, since D is also a cause of A and it acts as noise concerning the relationship A,B,C. Due to this the test statistic for the conditional test dep(B,C|A) will be below the limit for "strong dependence". In this case, the formal test for conditional dependence between B and C must be dep(B,C|A,D), but such a test cannot be done by the algorithm, since it considers only triplets.

A second known problem is the fact that certain rules are found, whose edges have incorrect directions. Provided the the graph given in Figure 8, the algorithm finds one of the following CCC rules B ← A → C, B → A → C and B ← A ← C, but one would expect either the CCU rule B → A ← C or the CCC rule C → B → A. It is easy to see, that it is not possible to find the CCU rule, since B and C are dependent. Since the algorithm is a greedy approach, it incorrectly finds the CCC rule. The first match is

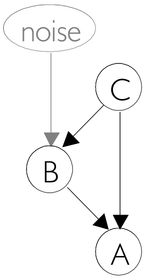


Figure 8

given as result and it is not checked whether there are better matches. In general, it is likely that the problem of edges with wrong directions arises because of the Markov-equivalence[11].

A third problem arises for special activation functions. The XOR-relationships is an example, since it is not possible to find XOR-relationships for rules $A \rightarrow B \leftarrow C$, where $B = XOR(A,C)$. This is because the unconditioned test will find no relationship between (A,B) and (B,C). The same holds for the biimplication.

		B	
		0	1
A	0	¼	¼
	1	¼	¼

		C	
		0	1
B	0	¼	¼
	1	¼	¼

There are also some strange side effects possible: given the M9 network as shown in Figure 5, with the following activation functions for the nodes $A_5 = AND(A_1,NOT(A_3))$ and $A_6 = XOR(A_1,A_3)$. As an example, with the log odds ratio a causal chain between the nodes A_3, A_5 and A_8 will be found, since the nodes A_5 and A_6 have the same inputs from the nodes A_1 and A_3, and the activation functions is similar in three of four combinations.

Summary

The Extended LCD is not capable to find all possible causal structure in a given data set, since it considers only triplets. The restriction to triplets can lead to a violation of the d-separation criterion. Nevertheless, the algorithm provides good results, which can be used as working hypothesis. It is also possible to rank the results by a scoring function. This ranking can serve as heuristic to find the most promising results.

[11] Two directed acyclic graphs are Markov-equivalent if and only if they have the same skeleton (the underlying undirected graph) and the same head-to-head structures.

Chapter 6 – Conclusions and Further Work

Inductive causation can provide good results, if it is used as explorative approach to find working hypotheses for further tests. We introduced a scoring function to judge the different rules. The scoring function proved to be helpful to restrict further experiments to the most promising rules. There are some causal relationships, which cannot be found and it turned out, that the direction of a rule can be incorrect, but nevertheless the benefit of the rules that are found can be big enough as assistance.

Pearson correlation and log odds ratio, both measures are shown to be suitable to find causal relationships in a binary dataset. The log odds ratio provides slightly better results than the Pearson correlation, since the log odds ratio is more precise for extreme contingency tables than the Pearson correlation. Unfortunately, the effort for the log odds ratios bootstrapping increases the runtime of the algorithm. Because of the distance estimation to decide when bootstrapping is needed or when the approximation is good enough, it is possible to reduce the runtime substantially – but it keeps expensive. An advantage of the Pearson Correlation is that it will work also with non-binary datasets. The scalar product is not suitable as measure for testing for independence and dependence.

It turned out that the algorithm is not capable to find all possible rules, since the d-separation criterion and the CCC and CCU rule are not based on the same conditions. d-separation describes independence relationships in a graph, the CCC and CCU try to build up a simple graph by using independence and dependence relationships between three variables in a given data set. Due to the fact that only three variables are viewed to find such a rule, the algorithm may not be capable to find more complex causal relationships.

The following topics can be interesting subjects for further work and research:

- Comparing treatment and control group results:
  The MELK data is derived for two groups: one group is treated with some substances, the other serves as a control group to determine how effective the treatment is. It is possible that the causal relationships can change between these groups.
- Combining results of several runs:
  Combing the results of several runs to enhance the profit for the experts.
- Testing conditional (in)dependence with more than one conditional variable:
  It turned out, that it is necessary to test not only for triplets, since there are some relationships where more than three variables are important. Testing with all known

neighbors as condition to determine more complex relationships might improve the results.

- Performance Tuning:
  Due to the fact that our main interest was the correctness of inductive causation – not the performance improvement of its algorithm – there might be some possibilities to enhance the performance of the MeICD algorithm. Especially the log odds ratio has a poor runtime performance. It might be possible to find a better approximation for the log odds ratio or to do further performance tuning for the bootstrapping method.
- To use other measures for dependence and independence:
  It might be useful to use, for example, the information gain as measure.

Appendix A

			theta=(0.2,0.3)			theta=(0.2,0.2)			theta=(0.1,0.2)			theta=(0.1,0.1)		
type	node indice	evaluat	perc.	min	median	perc.	min	median	perc.	min	median	perc.	min	median
CCU	1,4,2	+										100%	0,32	1,82
CCU	1,4,3	+										100%	0,28	1,81
CCU	1,5,3	+				100%	0,16	1,12	100%	0,16	1,10	100%	0,92	2,29
CCU	1,5,7	-										40%	0,03	1,01
CCU	1,6,3	+				100%	0,18	1,13	100%	0,18	1,11	100%	0,92	2,33
CCU	1,6,7	-										30%	0,05	1,02
CCC	1,6,8	+	100%	0,23	0,75	100%	0,83	1,15	100%	0,80	1,12	100%	0,86	3,18
CCC	1,6,9	+	100%	0,22	0,75	100%	0,84	1,14	100%	0,82	1,11	100%	0,88	3,17
CCU	1,8,3	~										100%	0,20	1,75
CCC	1,8,9	-												
CCU	1,9,3	~										100%	0,19	1,77
CCC	1,9,8	-	10%	0,02	0,20	10%	0,02	0,79						
CCU	2,4,3	+										100%	0,34	1,82
CCU	2,4,5	-										100%	0,21	1,74
CCU	2,4,6	-										100%	0,20	1,71
CCU	2,4,8	-										10%	0,01	1,30
CCU	2,4,9	-										10%	0,03	1,29
CCU	2,7,3	+				100%	0,19	1,12				100%	0,92	2,32
CCU	2,7,5	wd										30%	0,01	1,04
CCC	3,6,8	+	100%	0,22	0,74	100%	0,82	1,13	100%	0,81	1,10	100%	0,86	3,15
CCC	3,6,9	+	100%	0,22	0,76	100%	0,83	1,14	100%	0,83	1,12	100%	0,92	3,15
CCC	3,8,9	-												
CCC	3,9,8	-	20%	0,00	0,19	20%	0,00	0,78						
CCC	4,1,5	+												
CCC	4,1,6	+												
CCC	4,1,8	~				100%	0,10	0,60						
CCC	4,1,9	~				100%	0,11	0,60						
CCU	4,2,7	wd				20%	0,02	0,89	20%	0,02	0,85	100%	0,67	1,82
CCC	4,3,5	+												
CCC	4,3,6	+												
CCU	4,3,7	wd				20%	0,02	0,91	20%	0,02	0,88	100%	0,68	1,85
CCC	4,3,8	~				90%	0,11	0,59						
CCC	4,3,9	~				100%	0,11	0,60						
CCC	4,5,6	~												
CCC	4,5,8	-				70%	0,04	0,52						
CCC	4,5,9	-				50%	0,05	0,53						
CCC	4,6,8	~				100%	0,38	0,84	100%	0,38	0,81	100%	0,89	2,12
CCC	4,6,9	~				100%	0,37	0,85	100%	0,37	0,83	100%	0,92	2,10
CCC	4,8,9	-				60%	0,25	0,38						
CCC	4,9,8	-				40%	0,24	0,36						
CCC	5,1,8	~												
CCC	5,1,9	~	10%	0,02	0,17	10%	0,02	0,75						
CCC	5,3,7	+				30%	0,12	1,13	30%	0,12	1,11	100%	0,90	2,35
CCC	5,3,8	~												
CCC	5,3,9	~												
CCC	5,6,8	~	100%	0,14	0,69	100%	0,71	1,04	100%	0,70	1,01	100%	0,86	2,86
CCC	5,6,9	~	100%	0,12	0,69	100%	0,69	1,05	100%	0,69	1,01	100%	0,87	2,84
CCC	5,8,9	-	60%	0,06	0,14	60%	0,08	0,69						
CCC	5,9,8	-	30%	0,06	0,12	30%	0,06	0,68						
CCC	6,3,7	+				30%	0,16	1,14	30%	0,16	1,11	100%	0,86	2,35
CCC	7,3,8	~										100%	0,62	1,75
CCC	7,3,9	~										100%	0,66	1,76
CCC	7,5,8	-										10%	0,03	0,86
CCC	7,6,8	~										100%	0,62	0,99
CCC	7,6,9	~										100%	0,66	1,01
CCC	7,8,9	-										20%	0,08	0,63
CCC	7,9,8	-										60%	0,13	0,59
CCC	8,6,9	+	100%	0,94	1,35	100%	0,94	2,52	100%	0,86	2,52	100%	0,86	6,05

Table 4: Pearson correlation as test statistic for 10 datasets generated by M9 network with "medium noise"
For different parameter sets the lower and upper cutoff values are shown in the columns and when a rule was found by the algorithm, its frequency and evaluation measures are shown.

			theta=(0.2,0.3)			theta=(0.2,0.2)			theta=(0.1,0.2)			theta=(0.1,0.1)		
type	node indice	evaluat	perc.	min	median	perc.	min	median	perc.	min	median	perc.	min	median
CCU	1,4,2	+										100%	0,80	1,27
CCU	1,4,3	+										90%	0,72	1,27
CCU	1,5,3	+				100%	0,64	0,75	100%	0,64	0,75	100%	0,94	2,33
CCU	1,5,7	-												
CCU	1,6,3	+	10%	0,10	0,22	100%	0,66	0,75	100%	0,66	0,80	100%	0,94	2,36
CCU	1,6,7	-												
CCC	1,6,8	+				100%	0,29	1,63	90%	0,28	1,63	90%	1,57	4,26
CCC	1,6,9	+				100%	0,28	1,61	100%	0,28	1,61	100%	1,57	4,23
CCU	1,8,3	~										60%	0,63	1,23
CCC	1,8,9	-				50%	0,29	0,89						
CCU	1,9,3	~										60%	0,62	1,24
CCC	1,9,8	-												
CCU	2,4,3	+										100%	0,75	1,28
CCU	2,4,5	-										60%	0,61	1,21
CCU	2,4,6	-										60%	0,61	1,20
CCU	2,4,8	-												
CCU	2,4,9	-												
CCU	2,7,3	+	10%	0,11	0,27	100%	0,64	0,78	100%	0,64	0,78	100%	0,95	2,34
CCU	2,7,5	wd												
CCC	3,6,8	+				100%	0,28	1,63	100%	0,28	1,63	100%	1,55	4,26
CCC	3,6,9	+				100%	0,28	1,61	100%	0,28	1,61	100%	1,56	4,22
CCC	3,8,9	-				10%	0,28	0,86						
CCC	3,9,8	-				30%	0,25	0,86						
CCC	4,1,5	+				70%	0,21	0,50						
CCC	4,1,6	+				80%	0,20	0,50						
CCC	4,1,8	~												
CCC	4,1,9	~												
CCU	4,2,7	wd										100%	0,94	1,52
CCC	4,3,5	+				50%	0,21	0,50						
CCC	4,3,6	+				80%	0,20	0,49						
CCU	4,3,7	wd										100%	0,94	1,54
CCC	4,3,8	~												
CCC	4,3,9	~												
CCC	4,5,6	~				20%	0,18	0,40						
CCC	4,5,8	-												
CCC	4,5,9	-												
CCC	4,6,8	~										100%	0,89	3,79
CCC	4,6,9	~										90%	0,87	3,74
CCC	4,8,9	-												
CCC	4,9,8	-												
CCC	5,1,8	~				90%	0,18	0,48						
CCC	5,1,9	~				90%	0,16	0,48						
CCC	5,3,7	+										100%	0,45	2,34
CCC	5,3,8	~				70%	0,17	0,48						
CCC	5,3,9	~				100%	0,17	0,48						
CCC	5,6,8	~				100%	0,19	1,56	100%	0,19	1,56	100%	1,37	4,12
CCC	5,6,9	~				100%	0,17	1,54	100%	0,17	1,54	100%	1,34	4,08
CCC	5,8,9	-				20%	0,17	0,85						
CCC	5,9,8	-												
CCC	6,3,7	+										100%	0,43	2,34
CCC	7,3,8	~										30%	0,26	2,05
CCC	7,3,9	~										50%	0,21	2,01
CCC	7,5,8	-												
CCC	7,6,8	~										30%	0,26	3,36
CCC	7,6,9	~										50%	0,21	3,31
CCC	7,8,9	-												
CCC	7,9,8	-												
CCC	8,6,9	+	100%	0,63	1,38	100%	1,44	2,56	50%	1,46	2,58	60%	3,91	6,14

Table 5: Log odds ratio as test statistic for 10 datasets generated by M9 network with "medium noise"
For different parameter sets the lower and upper cutoff values are shown in the columns and when a rule was found by the algorithm, its frequency and evaluation measures are shown.

Appendix B

In this Section the exact definition of the M9 network as shown in Figure 5 is given as Matlab source code. According to Section 4.4 for each node the function f is specified by a Matlab function. For the parents of a node the function g is specified by another Matlab function. A positive influence is denoted by 1 and a negative influence is denoted by -1.

The rule set format is formally defined as:

$$\text{child\_node} = \{\{f\}, \{\text{parent\_node}, g, \{\varepsilon, [1|-1]\}\}^{+}\}$$

M9 network definition

```
    std_dev = 0.4;

% M9 network rule set
    RuleSet{4} = { { @supportAggregationFunction } ,
                   { 1, @supportRule, { std_dev,  1 } },
                   { 2, @supportRule, { std_dev, -1 } },
                   { 3, @supportRule, { std_dev,  1 } }
                 };
    RuleSet{5} = { { @supportAggregationFunction } ,
                   { 1, @supportRule, { std_dev,  1 } },
                   { 3, @supportRule, { std_dev,  1 } }
                 };
    RuleSet{6} = { { @supportAggregationFunction } ,
                   { 1, @supportRule, { std_dev, -1 } },
                   { 3, @supportRule, { std_dev, -1 } }
                 };
    RuleSet{7} = { { @supportAggregationFunction } ,
                   { 2, @supportRule, { std_dev, -1 } },
                   { 3, @supportRule, { std_dev, -1 } }
                 };
    RuleSet{8} = { { @supportAggregationFunction } ,
                   { 6, @supportRule, { std_dev, -1 } }
                 };
    RuleSet{9} = { { @supportAggregationFunction } ,
                   { 6, @supportRule, { std_dev, -1 } }
                 };

% supportAggregationFunction
function [result] = supportAggregationFunction(value)
    sum_of_values = 0;

    for i = 1 : length(value)
        sum_of_values = sum_of_values + value{i};
    end

    if (sum_of_values > 0)
        result = 1;
    else
        result = 0;
    end

% supportRuleFunction
function [result] = supportRule(m_value, stddev, signum)
    result = randn * stddev + signum * (m_value - 0.5);
```

Glossary

A, B, C, A_1,..., A_m	random variables, nodes in a graph
a, b, c, a_1,..., a_m	values for random variables
P(A = a)	probability for random variable A taking value a
P(A = a,B = b)	probability for joint event A = a and B = b
P(A = a|B = b)	conditioned probability for A = a given B = b
P(A)	short form for probability for random variable A taking all values a
→, ←	(possible) causal relationship, e.g. A → B means: A causes B
dep(A,B)	dependence between A and B
indep(A,B)	independence between A and B
dep(A,B|C)	dependence between A and B given C
indep(A,B|C)	independence between A and B given C
$E_{i,j}$	expected value under null hypothesis of independence with $E_{i,j} = \frac{n_{i.}\, n_{.j}}{N}$
$\bar{a}$	sample mean value for all a_i
Var(A)	sample variance of A with $\mathrm{Var}(A) = \frac{1}{n-1}\left(\sum_{i=1}^{n} a_i^{\,2} - n\bar{a}^{2}\right)$
Cov(A,B)	sample covariance of A and B with $\mathrm{Cov}(A,B) = \frac{1}{n-1}\left(\sum_{i=1}^{n} a_i b_i - n\bar{a}\bar{b}\right)$
$\sim$ D	distributed as distribution D
$\dot{\sim}$ D	approximative distributed as distribution D
N(μ,σ)	normal distribution with the mean parameter μ and standard deviation σ
X^2_1	chi-squared distribution with one degree of freedom

References

[AgrSri1994]: R. Agrawal and R. Srikant: Fast Algorithms for Mining Association Rules, Proceedings of the 20th VLDB Conference, Santiago, 1994

[BMSU2000]: S. Brin, R. Motwani, C. Silverstein, and J. Ullman: Scalable Techniques for Mining Causal Structures, Kluwer Academic Publishers, Boston, 2000

[BorKru1999]: C. Borgelt and R. Kruse: A Critique of Inductive Causation, ECSQARU, 1999

[BorKru2002]: C. Borgelt and R. Kruse: Graphical Models - Methods for Data Analysis and Mining, J. Wiley & Sons, Chichester, 2002

[Cooper1997]: G.F. Cooper: A simple constraint-based algorithm for efficiently mining observational databases for causal relationships, Data Mining and Knowledge Discovery, 1997

[CooGly1999]: C. Glymour & G.F. Cooper (Eds.): Computation, Causation and Discovery, MIT Press, Cambridge, 1999

[CooHer1992]: G.F. Cooper and E. Herskovits: A Bayesian Method for the Induction of Probabilistic Networks from Data. Machine Learning 9 pp.309-347. Kluwer, Dordrecht, 1992

[DavHin1997]: A.C. Davison & D.V. Hinkley: Bootstrap methods and their application, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 1997

[Efron1994]: B. Efron, R.J. Tibshirani: An Introduction to the Bootstrap, Chapman & Hall/CRC, 1994

[Hecker1995]: D. Heckermann, D. Geiger & D.M. Chickering: Learning Bayesian networks – The combination of knowledge and statistical data, Machine Learning, 20, 1995

[Kruse2004]: R. Kruse: Unsicherheit und Vagheit, lecture, Otto-von-Guericke-Universität, Magdeburg, 2004

[KröSma1996]: B. Kröse & P. van der Smagt: An Introduction to Neural Networks, 8^{th} edition, 1996

[ManCoo2001]: S. Mani & G.F. Cooper: A Simulation Study of Three Related Causal Data Mining Algorithms, AISTATS, 2001

[MilArn1990]: J.S. Milton & J.C. Arnold: Introduction to probability and Statistics – Principles and Applications for Engineering and the Computing Science, Second Edition, McGraw Hill Series in Probability and Statistics, 1990

[Neapol1990]: R.E. Neapolitan: Probabilistic Reasoning in Expert Systems – Theory and Algorithms. J. Wiley & Sons, New York, 1990

[Pearl1988]: J. Pearl: Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann, San Francisco, 1988

[Pearl1991]: J. Pearl & T.S. Verma: A theory of inferred causation, Proceeding of the Second International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, pp. 441-452, San Mateo, 1991

[Pearl1992]: J. Pearl: Probabilistic Reasoning in Intelligent Systems – Networks of Plausible Inference (2nd edition). Morgan Kaufman, San Mateo, CA, 1992

[Reich1956]: H. Reichenbach (edited by M. Reichenbach): The Direction of Time, University of California Press, Berkeley and Los Angeles, 1956

[Sprites1993]: P. Sprites, C. Glymour & R. Scheines: Causation, Prediction and Search, Springer-Verlag, New York, 1993

Erklärung

Ich erkläre hiermit, dass ich die Arbeit selbstständig verfasst, keine anderen als die angegebenen Quellen benutzt und die diesen Quellen wörtlich oder sinngemäß entnommenen Ausführungen als solche kenntlich gemacht habe.

Magdeburg, 21. September 2004