A Bayesian approach to forensic psychiatric data

Lydia Mennes
S0630969
l.mennes87@gmail.com

Bacholor Thesis Artificial Intelligence

24-08-2012

Internal supervisor
Dr. F. Grootjen
Faculty of Social Sciences
Radboud University, Nijmegen

External supervisor
K. von Borries
Pompestichting, Nijmegen

Additional Assessor
Dr. L.G. Vuurpijl
Faculty of Social Sciences
Radboud University, Nijmegen

Abstract

A dataset containing a large number of variables (4898) from Forensic Psychiatry is explored for this project. This dataset is provided by the forMINDs project by the Pompestichting. The method for exploration is generating a Bayesian network. The dataset has been strongly modified for this purpose. Variables have been discarded (1394 remaining), continuous variables are discretized and the large number of missing variables (30%) are imputed using distribution based imputation. For structure generation the PC-algorithm is used, with the G^2-statistic for conditional independence testing. Computation time restrictions have resulted in a further reduction of the number of variables. The resulting network of 132 variables contained cycles, indicating the existence of hidden or selection variables and making the network unusable for parameter learning and inference. Secondly the network has an average of 19 neighbors per node, making it too complex for interpretation.

Contents

Chapter 1 Introduction

The field of artificial intelligence (AI) has many definitions (Russell & Norvig, 2003), which vary along two dimensions. The first dimension consists of thought processes and reasoning versus behavior, and the second dimension consists of human performance versus the ideal concept of intelligence or rationality. Either way the field is concerned with the design of intelligent agents or systems. For this purpose numerous techniques have been developed for knowledge and reasoning, problem-solving, planning and learning. These techniques can be used for typical artificial intelligence purposes such as robotics or the generation of behavior in non-player characters in games, however these techniques can also be used in other scientific fields as illustrated by the following examples. Cognitive science for instance is an interdisciplinary field which combines computer models from AI and experimental techniques from psychology to try to construct precise and testable theories of the workings of the human mind (Russell & Norvig, 2003). In biorobotics robots provide tools for biologists studying animal behavior and testbeds for the study and evaluation of biological algorithms for potential engineering applications (Consi & Webb, 2001). In medicine there is also a wide range of possibilities for the application of AI techniques, hence the existence of the journal Artificial Intelligence in Medicine. A final example of a field using AI techniques is molecular biology. In (Levin, 1995) a genetic algorithm, which is an AI technique is used to discover the sequence of amino-acids of proteins.

As one can imagine after these examples, the possibilities are infinite. This thesis will make such a journey of applying an AI technique to a different field: the field of forensic neuropsychology. A Bayesian network technique will be used to try to give insight into a dataset containing variables related to forensic neuropsychology.

This section will first provide information on the field of forensic neuropsychology in general and the forMINDS project of the Pompestichting Forensic Psychiatric Institute in Nijmegen in particular. Then the motivation for using a Bayesian network technique is given, and the basis of Bayesian networks, Bayes rule, is explained. This is followed by an explanation of Bayesian networks and the last topic of this section consists of the research question(s) of this thesis.

1.1 Forensic neuropsychology

Forensic neuropsychology is a rather new and rapidly evolving field (Guilmette, Faust, Hart, & Arkes, 1990). In (Borries & Verkes) the field is described as following:

> *"Forensic Neuropsychology … is mainly concerned with providing information based on scientifically validated neuropsychological principles and clinical methodology relevant to the forensic question at hand. An important aspect of the field of forensic neuropsychology is the assessment of cognitive functions and informing the relation between brain and behavior. This should be grounded on scientific methods for several reasons: Ideas and hypotheses about cognitive functions in forensic populations can be systematically studied, findings can be replicated and validated leading to an ever more evidenced based theory, with the goal of finding a common standard. This process is therefore ongoing, leading to an accumulation of validated and scientifically accepted information over time. "*

Neuropsychological principles can be used in the assessment and diagnostics of forensic patients. Even though this usage is growing, there is no gold standard at this moment. Currently most forensic psychiatric clinics do not include a standard neuropsychological/ cognitive assessment procedure. The absence of both a gold standard and the presence of standard cognitive assessment can be concluded reading the care programs, "zorgprogramma's" in Dutch, which are guidelines for forensic psychiatry and can be found on (Expertisecentrum Forensische Psychiatrie). A disadvantage of using standardized tests in forensic cases is that most of the tests have been normed in a quite different (non-forensic) population. For example, the forensic population tends to be represented by those who are poor, less educated and come from minority groups (Emmerik, 2001). The traditional tests are therefore in need of new normative data to interpret these tests taking into account the different characteristics in forensic patients.
Based on the treatment programs for forensic psychiatric patients in the Netherlands, there are two characteristics which are commonly used to divide forensic patients into more homogeneous groups to which norms could be applied. First they can be (roughly) characterized by their disorders; personality disorders, psychotic disorders and substance use disorders, although most patients suffer from multiple mental diseases (Emmerik, 2001). Secondly they can be divided into two groups by the characteristics of their offence; violent offences and sexual offences.
In the last 20 years more and more studies have aimed at characterizing the mentioned subgroups of forensic patients based on cognitive functioning. For each subgroup a number of examples of such studies will be provided below. These examples and more examples for each subgroup and the related brain areas (not mentioned here) can be found in (Borries & Verkes).

- Personality disorders
  Forensic neuropsychological has focused mainly on antisocial personality disorder (ASPD) and psychopathy (PP), knowledge about neuropsychological deficits in other personality disorders have not been investigated in forensic context. A finding when comparing ASPD and PP to schizophrenia is that intellectual capacities are intact (Miller, 1987). Several executive function and attention related deficits have been implicated in psychopaths (Pham, Vanderstukken, Philippot, & Vanderlinden, 2003). Individuals with antisocial behavior have been found to be impaired in emotional face recognition (Blair & Marsh, 2008).
- Psychotic disorders
  Patients with schizophrenia show a wide range of cognitive deficits and overall performance can be around two standard deviations below healthy controls. Cognitive deficits found are often related to higher cognitive functions requiring controlled information processing, such as (sustained) attention, executive functions, working memory tasks, and different forms of learning (Anatova & Sharma, 2003) (Goldberg & Gold, 1995) (Antonova, T. Sharma, & V., 2004). Also inhibition problems (Perlstein, Carter, Barch, & Baird, 1998) and problems in strategy formation and planning (Morris, Rushe, Woodruffe, & Murray, 1995) have been found.
- Substance use disorders
  In general it has been found that successful recoverers do show intact functioning on cognitive measures. Relapsers perform poorly on tests of language, abstract reasoning, planning and cognitive flexibility. When under influence of cannabis performance measure of memory, executive functioning and psychomotor speed goes down (Bolla, Brown, Eldreth,

Tate, & Cadet, 2002). In chronic users the non acute affect is found that the ability to learn and remember new information goes down (Grant, Gonzalez, Carey, Natarajan, & Wolfson, 2003).

- Sex related offences
  Commonly assessed cognitive dysfunctions have been examined in pedophiles and other sexual offenders but most research has focused on interpersonal functioning such as empathic behavior. In (Kirsch & Becker, 2007) it is hypothesized that deficits in emotion recognition and emotional experience in sexual sadists may lead to deficits in empathic behavior. Sexual offenders in general show a profile of lower order executive functions (e.g. sustained attention and inhibition) and verbal deficits with intact or good capacities for higher order executive functioning (e.g. reasoning and cognitive flexibility) (Joval, Black, & Dassylva, 2007).
- Violent offences
  A number of cognitive deficits have been found in violent offenders, such as attentional shifting deficits by (Bergvall, Wessely, Fosman, & Hansen, 2001). (Hoaken, Allaby, & Earle, 2007) suggest abnormal executive functioning in violent and non-violent offenders, and difficulties in facial affect recognition in violent offenders.

However, these results concern the comparison of groups, while in clinical practice the goal is to characterize individual behavior, explain it and possibly predict future behavior based on cognitive abilities. Few studies have related cognitive functioning to risk assessment, treatment effectiveness and relapse prevention. It is necessary to understand these links in order to use cognitive assessment on an individual level. For this reason a large neurocognitive project called forMINDS has been set up in a forensic psychiatric institute.

1.2 The forMINDS project

Within the research department of the Pompestichting Forensic Psychiatric Institute in Nijmegen the forMINDS project is carried out by B.H. Bulten (coordinating investigator/project leader), A.K.L. von Borries (Principal investigator) and R.J. Verkes (Principal investigator). The dataset on which an AI technique is applied in this thesis, is provided by the forMINDS project and contains variables related to forensic neuropsychology.

1.2.1 The Pompestichting

The Pompestichting is a TBS-clinic in Nijmegen. In (Brazil, Bruijn, & Bulten, 2009) TBS is described as "a disposal to be treated, on behalf of the state, for people who have committed serious criminal offenses in connection with having a mental disorder. TBS is not a punishment but an entrustment act for mentally disordered offenders (diminished responsibility). These court orders are an alternative to either long-term imprisonment or confinement in a psychiatric hospital, with the goal to strike a balance between security, treatment, and protection."

1.2.2 Objectives of the forMINDs project

The forMINDS project is concerned with automated cognitive assessment in forensic context. The objectives of this project are described in (Borries & Verkes) as:

“

1) By implementing a cognitive test battery in a large population of forensic psychiatric patients, a prison population and healthy controls, we will be able to further develop and adjust the battery based on results and patterns found with the help of these cognitive tests. This will ultimately lead to a standard instrument.
2) By collecting a large body of data in forensic psychiatric patients and prison inmates, we will be able to
   a) develop normative data relevant for the interpretation of test results in these populations. Normative data from healthy controls is collected for the same reason.
   b) collect data for research into the neurocognitive differences between certain subgroups (type of offence, type of diagnosis, etc.) and healthy subjects. This will allow us to develop and test working models of cognitive dysfunction in subgroups of forensic psychiatric populations.
   c) collection of data necessary for the assessment procedure implemented in the Pompekliniek, which is also used for decisions around treatments options. This also includes the possibility of retesting at a later point in time, to evaluate the treatment.

“

The relevance of these objectives for forensic issues is explained in (Borries & Verkes) with several reasons. A short overview of these reasons is provided below.

- As mentioned in the section concerning forensic neuropsychology, most cognitive tests are normed in a population other then the population relevant to forensic psychiatry. By collecting data over time it is possible to develop normative data for the forensic population. Furthermore there are no norms available on how certain dysfunctions are related to criminal behavior such as aggression. These norms might also be developed using the collected data from the forMINDS project.
- The data collected using the test battery can be used to develop and test working models of cognitive dysfunctions for subgroups of the forensic population. Such a constructed model of cognitive dysfunction can be tested against the growing body of collected data and can thereby be refined and validated.
- Information on cognitive abilities of patients can be used in treatment, as well in decisions regarding treatment plans, as in interaction between clinical staff and patients. If for example a patient learns better based on reward compared to based on punishment, this might be useful information for treatment of and interaction with the patient.
- In practice assessment procedures often result in a list of systems which can be used to classify problems in terms of psychiatric disorders. It is stated in (Borries & Verkes) that: “assessment in terms of cognitive functions enables us to see the deficits of a certain patient in context of the relation between patient and environment without losing reliability and by adding validity. It can assist in reaching higher diagnostic differentiation within one disorder. Treatment decision should therefore not only be based on psychopathological issues, but also on cognitive capacities and the functionality of the underlying neural circuits.”
- Personality disorders are less stable than previously assumed in the list of criteria for personality disorders (Shea, et al., 2002). The symptoms can change over time, opposed to

abstract personality dimensions (e.g. perfectionism) which are more stable. Treatment interventions are mainly aiming at influencing specific behavior. Therefore it is important to find instruments which assess specific behavior instead of symptoms.

1.2.3 The forMINDS test battery and resulting data

As mentioned the project forMINDS has been running for a couple of years now and an extensive dataset has been collected. The dataset consists of three types of variables. The first part of the dataset is anamnestic information, which includes demographic information such as age, education and clinical information such as type of offence and diagnostics. Secondly tasks are included that intend to measure performance of cognitive functions. Finally questionnaires are used which measure for instance empathy. For a complete overview of the test battery see appendix A. The tasks and questionnaires cover four cognitive fields: The test battery has resulted in a dataset that contains 4898 variables and 243 subject. The variables that result from the cognitive tasks are mainly reaction times and error quantifications. The subjects consist, as mentioned previously, of detainees, TBS-patients and healthy controls.

1.3 Qualitative research

The dataset from the forMINDS project consists of structured data, i.e. it consists of distinct variables which are measured for each subject. Therefore the dataset would be suitable to use for quantitative research; testing hypotheses using statistics. Another possible research approach is qualitative research. The approach on the forMINDS dataset is a more qualitative research, although on structured data.

Qualitative methods for research are traditionally regarded as methods that investigate the why and how of a topic rather than what, where and when. Typically this type of research is used in for example social sciences and history. In (Guba & Lincoln, 1994) John Stuart Mill is said to be the first to urge social scientists to equal the so called 'harder' sciences, thus use more quantitative methods in research. It is also stated that: "There is a widespread conviction that only quantitative methods and quantitative data are ultimately valid, or of high quality." This is an ongoing debate about which can be read more in (Guba & Lincoln, 1994) and (Sechrest, 1992) for interested readers.

One way of using qualitative research is to regard it as a source of inspiration for quantitative research. According to (Guba & Lincoln, 1994)this has a number of advantages over purely quantitative methods. First of all there is no need for context stripping, which happens in purely quantitative methods through for example randomization. Also it is mentioned that the emphasis on verification of a priori hypotheses overlooks the origin of those hypotheses. Qualitative research can contribute to forming grounded a priori hypotheses for empirical research. These are just two of the mentioned arguments, since these are most applicable here.

As mentioned above a qualitative approach will be taken on the forMINDS dataset . It is meant to be an inspiration for possible quantitative research and give a more general insight or overview of the relations between the variables in the forMINDS dataset. When statistics are regarded as the only way to make truly valid conclusions (hence the debate above), the results of this thesis are not truly valid.

1.4 AI technique for exploring the forMINDS dataset

Within the field of AI there are a number of techniques which can be used to give a general overview or insight into the structure of relationships between variables in a dataset. Examples of this kind of techniques are decision trees, Byesian networks and neural networks (Russell & Norvig, 2003). It is interesting to see if applying such a technique on the forMINDS dataset has additional value for the researchers that are part of the project. This additional value might consist of inspiration for qualitative value or insight in the global structure of the relationships between variables.

The technique that will be used for exploring the forMINDS dataset is a Bayesian Network. For a while now Bayesian networks have been popular throughout science. The reason for this, and the reason why it is appealing for this thesis, is described well by (Bishop, 2006): "Graphical models are a marriage between probability theory and graph theory. They provide a natural tool for dealing with two problems that occur throughout applied mathematics and engineering - uncertainty and complexity - and in particular they are playing an increasingly important role in the design and analysis of machine learning algorithms. Fundamental to the idea of a graphical model is the notion of modularity, a complex system is built by combining simpler parts. Probability theory provides the glue whereby the parts are combined, ensuring that the system as a whole is consistent, and providing ways to interface models to data. The graph theoretic side of graphical models provides both an intuitively appealing interface by which humans can model highly-interacting sets of variables as well as a data structure that lends itself naturally to the design of efficient general-purpose algorithms. "

One advantage of a Bayesian network is the fact that the structure of a Bayesian network can be studied on itself to look at conditional (in)dependences. The nature of these conditional dependence relationships can also be studied because of the conditional probability tables. This as opposed to for example a neural network, which captures relationships implicitly. Neural networks are therefore 'black boxes' when compared to Bayesian networks. Since this thesis is supposed to be a possible source of inspiration for quantitative research the explicit capturing of relations within the network is an advantage.

1.5 Bayes' theorem

Bayes Theorem is the basis of Bayesian networks (Heckerman, A tutorial on learning with Bayesian networks, 1998). Since Bayesian Networks is the technique used to explore the forMINDS dataset, Bayes' theorem will be explained in this section. The theorem is a formula that is used for calculating conditional probabilities, see equation 1.1. It captures the relationship between the prior probabilities and the conditional probabilities of two random events. A prior probability is the probability of an event without having any further information and a conditional probability is the probability of an event in presence of other information.

Equation (1.1) $$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

To explain Bayes' theorem the example from (Kennedie, 2009) will be used which uses a Venn diagram, see figure 1.1.

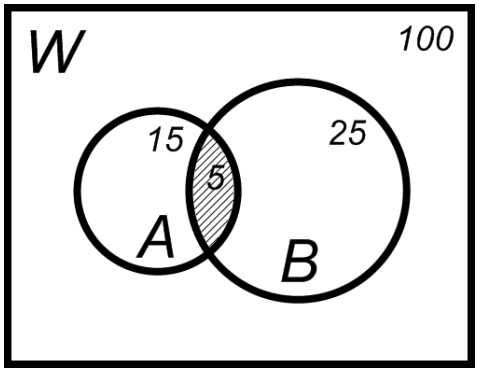


Figure 1.1. Venn diagram from (Ruskey & Waston, 1997)
for the Pen box example.

Suppose there is a box containing 100 pens, this is represented by W in the diagram. Theses pens are either a ballpoint or a fine liner and the used ink is either red or blue. All pens in B are ballpoints and all pens in A have blue ink. The shaded area then represents all ballpoints with blue ink. As can be seen in the diagram the box contains 25 ballpoints and 15 pens with blue ink from which 5 are ballpoints. The probability that a pen is a ballpoint, denoted by $P(B)$, is $\frac{|B|}{|W|} = \frac{25}{100} = 0.25$, where $|X|$means the number of elements with property X. In other words, the a priori probability of a pen being a ballpoint is 0.25. The probability that a pen is a ballpoint with blue ink, denoted by $P(A \cap B)$, is $\frac{|A \cap B|}{|W|} = \frac{5}{100} = 0.05$, where $|X \cap Y|$ means the number of elements with both property X and Y. Again, this is a prior probability. Now suppose we randomly grab a ballpoint. The probability this ballpoint contains blue ink can be calculated with equation 1.2, this is a conditional probability. The resulting probability is $\frac{0.05}{0.25} = 0.20$.

Equation (1.2) $$P(A|B) = \frac{|A \cap B|}{|B|} = \frac{P(A \cap B)}{P(B)}$$

When we grab a pen with blue ink, the probability this is a ballpoint can be calculated with equation 1.3. This probability is $\frac{0.05}{0.15} = 0.33$.

Equation (1.3) $$P(B|A) = \frac{P(B \cap A)}{P(A)}$$

The number of ballpoints with blue ink is the same as the number of pens with blue ink that are ballpoints. This means the probability of grabbing a ballpoint with blue ink is also the same as the probability of grabbing a pen with blue ink that is a ballpoint. This symmetry property is shown in equation 1.4.

Equation (1.4) $$P(A \cap B) = P(B \cap A)$$

If the symmetry property is applied to equation1.3. This results in equation 1.5.

Equation (1.5) $$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

When equation 1.5 is multiplied on both sides with $P(A)$, it is transformed into the product rule of probability.

Equation (1.6) $$P(B|A)P(A) = P(A \cap B)$$

By dividing both sides by $P(B)$ we get equation 1.7.

Equation (1.7) $$\frac{P(B|A)P(A)}{P(B)} = \frac{P(A \cap B)}{P(B)}$$

Which, using equation 2.1, can be rewritten to equation 1.8.

Equation (1.8) $$\frac{P(B|A)P(A)}{P(B)} = P(A|B)$$

When both sides of the equal sign are switched we have Bayes' theorem, which is repeated in equation 1.9.

Equation (1.9) $$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

1.6 Bayesian Network

The definition of a Bayesian network according to (Heckerman, A tutorial on learning with Bayesian networks, 1998) is:

Definition (1.1) *"A Bayesian network is a graphical model for probabilistic relationships among a set of variables".*

In line with this definition a Bayesian network consists of a network which provides the structure of the relationships among the variables and conditional probability tables, which quantify these relationships. These components and their semantics will be explained in the first three subsections. The final subsection will explain inference, which means using the network to infer any (conditional) probability of interest. More information about Bayesian networks in general can be found in (Heckerman, A tutorial on learning with Bayesian networks, 1998), (Bishop, 2006) or (Russell & Norvig, 2003).

1.6.1 Topology of a network

The "graphical model" phrase in the definition 1.1 refers to the fact that a network consists of nodes (or vertices) and edges between nodes. An example of such a model can be seen in figure 1.2. In the basic version of a Bayesian network each node in the network represents one variable in the domain of interest. The edges of the network are directed, making them arrows from one node to another. A few terms from graph theory are necessary to speak about Bayesian networks, here or in future sections. These terms will be explained now.

A path is a sequence of nodes such that from each of its nodes there is an edge to the next sequence in the sequence. In figure 1.2 *Burglary – Alarm – Dog barks* is an example of a path.
Parents of a particular node, e.g. node A, are those nodes from which an arrow goes to node A. around For example, in figure 1.2 the parents of node *Alarm* are *Burglary* and *Alarm code*.
The descendants of a node, e.g. node A, are those nodes for which there is a directed path between node A and that node. In figure 1.2 the descendants of node *Burglary* are *Neighbor calls*, *Alarm* and *Dog barks*.

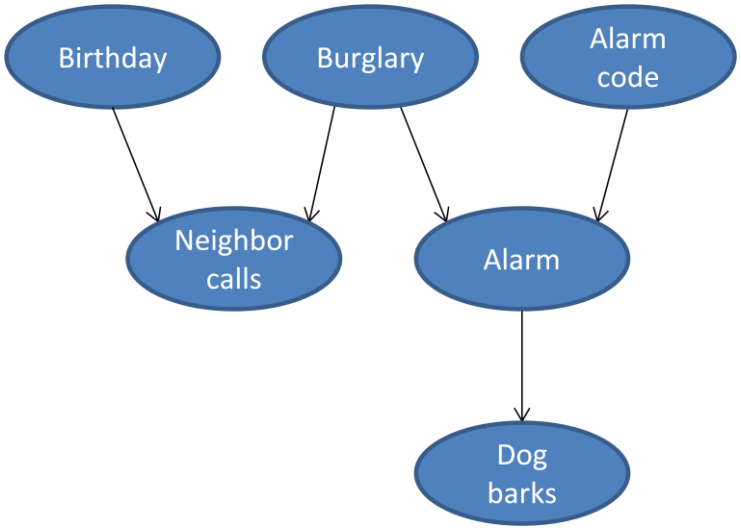


Figure 1.2. Example of the structure of a Bayesian network

These relationships can also be defined in the opposite direction. The children of a node (node A) are those nodes to which an edge goes from node A. And the predecessors of node A are all nodes from which there is a directed path to node A.
The fact that it is not allowed within a Bayesian network to have a path from a node to itself makes such a network a Directed Acyclic Graph (DAG).

1.6.2 Conditional probability tables

As mentioned in definition 1.1, Bayesian networks encode probabilistic relations among variables. Along with each node in the network there is a conditional probability table. This table encodes the probability distribution for the values of the variable encoded by this node. Table 1.1 shows such a table for variable *Alarm* from figure 1.2.

Burglary	Yes		No	
Alarm code	Correct	Incorrect	Correct	Incorrect
Yes	0.9	0.56	0.01	0.09
No	0.1	0.44	0.99	0.91

Table 1.1 Conditional probability table for variable Alarm in figure 1.2.

As can be seen in this table there is a probability distribution for each combination of values of the parents of this node. Recall that an a prior probability is the probability that a node has a certain value, without any further knowledge of the value of other nodes. Using the conditional probabilities the a priori probability distribution of any node can be calculated using Bayes rule. In equation 1.10 the formula for calculating such a distribution is shown.

Equation (1.10) $$P(A) = \sum_{i=1}^{n} P(A|b_i) * P(b_i)$$

Where A is the node of interest, b_i is an assignment of values to the parent nodes of A, and n is the number of possible assignments to A.
Suppose the variable Alarm code has the probability distribution $[0.2, 0.8]$ for respectively the alarm going off or not, and the variable Burglary has the probability distribution $[0.01, 0.99]$ for respectively a burglary taking place or not. The probability of the variable Alarm having the value yes can now be calculated:

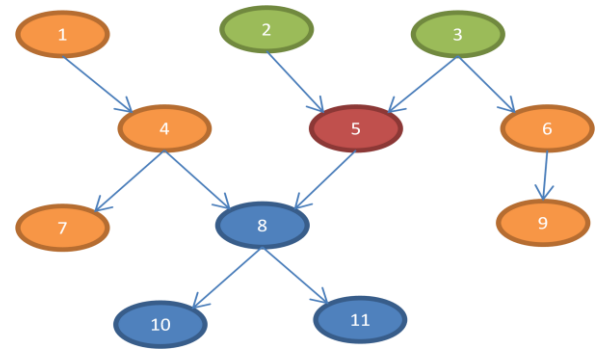
$P(Alarm = yes) =$
$P(Alarm = yes|Alarm\ code = correct, Burglary = yes) \cdot P(Alarm\ code = correct, Burglary = yes) +$
$P(Alarm = yes|Alarm\ code = correct, Burglary = no) \cdot P(Alarm\ code = correct, Burglary = no) +$
$P(Alarm = yes|Alarm\ code = wrong, Burglary = yes) \cdot P(Alarm\ code = wrong, Burglary = yes) +$
$P(Alarm = yes|Alarm\ code = wrong, Burglary = no) \cdot P(Alarm\ code = wrong, Burglary = no).$

Which results in a probability of 0.08.

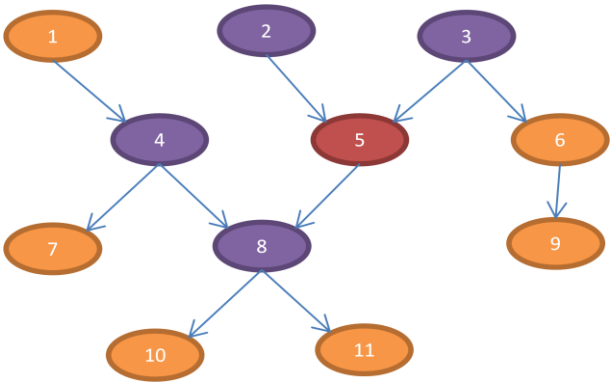
1.6.3 Semantics
The two preceding sections have explained the two key elements of a Bayesian network, its structure or topology and the probabilistic relation between a node and its parents. But what does a configuration of nodes and edges mean? The topological semantics can be explained in two ways, which are equivalent. These explanations come from (Russell & Norvig, 2003).

1. A node is conditionally independent of its non-descendents given its parents. In figure 2a the node is red, its parents are green and the non-descendents it is conditionally independent from are orange. However, it does still depend on its descendents which are blue.
2. A node is conditionally independent of all other nodes given its parents, children and children's parents. This is called the Markov blanket. In figure 2a the node is red and the Markov blanket is purple.

Both explanations come from a more general criterion called d-separation, which can decide whether a set X is independent of a set Y given a third set Z. The above explanations are more clear in this context and therefore d-separation is not explained here. For interested readers more details can be found in (Pearl, Reasoning in Intelligent Systems, 1988).



a)



b)

Figure 1.3. In a) the red node is independent of its non-descendants (orange) given its parents (green). In b) the red node is independent of all other nodes given its Markov blanket (purple).

1.6.4 Inference

Finally it is possible to use a network to infer any probability or probability distribution one would like to know. It might be interesting to know what the probability distribution of a node is, given the values of other nodes (regardless of them being in the Markov Blanket or not). These given values of other nodes are called evidence. Such a query with query variable Q and evidence e can be written as $P(Q|e)$. In general such a probability can be calculated using equation 1.11.

Equation (1.11) $$1/_{e} \sum_{y} P(Q, e, y)$$

With y being all possible combinations of values of the unobserved variables. The terms that have to be summed can be written as products of the conditional probabilities in the tables from the network.

1.7 Research question

The objective of this thesis is to investigate whether using a Bayesian network approach on the forMINDS dataset has additional value for the forMINDS project. This results in the following research question:

“Does a Bayesian network form an inspiration for possible quantitative research and does it give a more general insight of the relations between the variables in the forMINDS dataset?”

This research question is divided into two sub questions:

1. Are there unexpected configurations in the structure of the network?
2. Does the network form an inspiration for quantitative research using inference?

Chapter 2 Methods

This chapter provides information about the used methods in this thesis. The first four sections will discuss the contents of and operations on the dataset. The remaining sections discuss the software and the Bayesian techniques used for structure generation, parameter learning and inference.

2.1 The dataset

The dataset consists of tasks, questionnaires, the anamnesis and risk analysis. When each task and questionnaire and so on are regarded as a category of variables there are 25 categories. As mentioned in the introduction the tasks and questionnaires cover four cognitive fields; impulsivity and attention, moral and social behavior, emotional processing and learning. To give an idea of the contents of the dataset the categories are shown in table 2.1, together with the cognitive field it belongs to. For more detailed information about the tasks and questionnaires itself see appendix A. Multiple operations are performed on the dataset which will be discussed in the following sections. An overview of these operations can be seen in figure 2.1.

Table 2.1. The categories of variables, their corresponding field and the number of variables associated with it after removal of variables as discussed in section 2.2 and 2.3 .

Field	Category	# variables	total per field
-	Anamnesis	149	
-	Risk Analysis	48	
-	SDAS	26	223
impulsivity and attention	BisBas questionnaire	6	
	Continuous Performance Task	16	
	Perceptual Defence Task	5	
	Signal Detection Task	52	
	STOP signal task	46	
	Stroop	23	
	Trail making test	19	167
Emotional Processing	Affective Go/No go Task	93	
	Emotional Stroop Task	92	
	Faces task	90	
	Graded Facial Emotion Recognition task	221	
	Interpersonal Reactivity Inventory	5	
	state trait anger expression inventory	2	
	State trait anxiety inventory	2	505
Implicit cognition Learning	Psychopathic Personality Inventory	14	
	Casino task	235	
	ID/ED task	214	
	Kirby questionnaire	3	
	SPRQ questionnaire	2	468
Social and moral Behavior	Moral Judgement Sorting Task	8	
	Prisoners Dilemma Game	10	
	social value test	3	21

2.2 Used variables

The raw version of the forMINDS dataset contains 4898 variables. Even though Bayesian Network techniques are designed to use a lot of variables compared to regular statistics, the time needed to generate a network increases strongly when using more variables. The actual complexity varies depending on which method is used. These are discussed in section 2.6 Because this strong increase the first step in using this dataset has been to see if all variables should be used, based on their content. After this review 1384 variables remain. The reasons for removing variables are:

- The variables concerning the cognitive consist for a large part of reaction times which summarize the performance of that task. For most tasks the reaction times are summarized as a total as well as using the average. It seems redundant to use both measurements. Since not all trials are included in every reaction time variable the average is more interesting in terms of comparison. Therefore all summed variables are excluded if averaged variables are available.
- The number of variables summarizing reaction times are doubled by the fact that they are calculated both including and excluding trials that have extremely short or long reaction times. These extreme trials are regarded as mistrials. The variables that include such trials are excluded for this purpose, and the number of such trials for tasks are included.
- The errors made in tasks are often represented both by a variable containing the number of errors and the percentage of errors over trials. Both contain the information about errors which makes one of them rather redundant. The relative amount of trials seems more interesting since this already captures the information about the total numbers of trials as well. The variables containing the absolute number of errors are discarded.
- For all questionnaires the item scores are included as well as the summarizing variables used for these questionnaires. Within this research the focus lies with the relations between all tasks, questionnaires and other variable categories. The focus does not lie with whether or not the used questionnaires are of high quality. Therefore the item scores here can be assumed irrelevant, since the value of the summarizing variables is assumed to be high. All item scores have been discarded.
- A number of variables, for example the DSM-codes for the disorders or the number assigned to each subject, are (almost) unique for each subject and therefore not useful. Those variables that do carry useful information are summarized in different variables. Also there are variables that contain information which only has an administrative value, such as whether or not certain reports are available at the clinic. All these variables are excluded.
- A number of variables in the set contained dates. These dates are more probable to be interesting when considered relative. An example of this are the dates indicating when the tasks are performed. This is less probable to be interesting by itself compared to the time relative to the start of treatment. These variables have been replaced by relative variables.

2.3 Missing Values

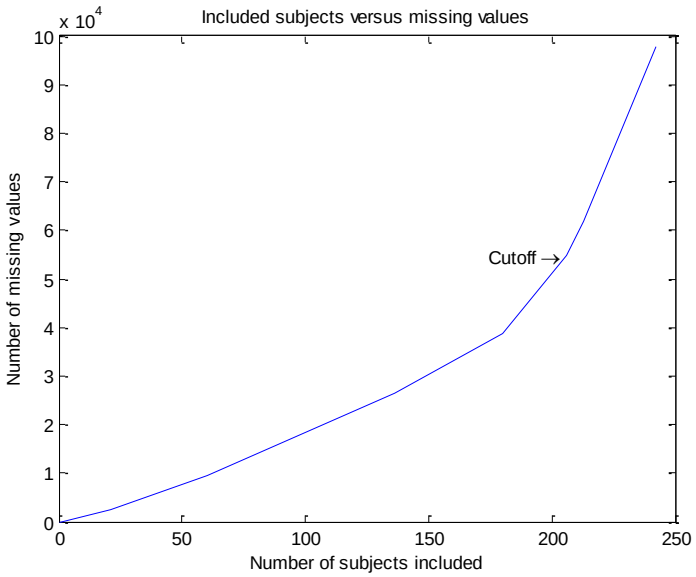
The forMINDS dataset contains many missing values. After eliminating variables as described in the previous section, 30% of all values are missing. The implementation of the algorithm for structure generation (see section 2.6.2) in the software that is used in this thesis (see section 2.5) is unable to handle missing values. There are a number of ways to cope with missing values, which will be discussed later in this section.

In order to make a good decision regarding the used method, the missing values need to be investigated regarding their type. Why are the values missing? Are the missing values randomly distributed? If the distribution is not random, is there a pattern? Might there be values missing by design? These are all relevant questions when dealing with missing values as is argued in (Cohen, Cohen, West, & Aiken, 2002), (Newman, 2003) and (Royston, 2005). The observations regarding the missing values of the forMINDS dataset are:

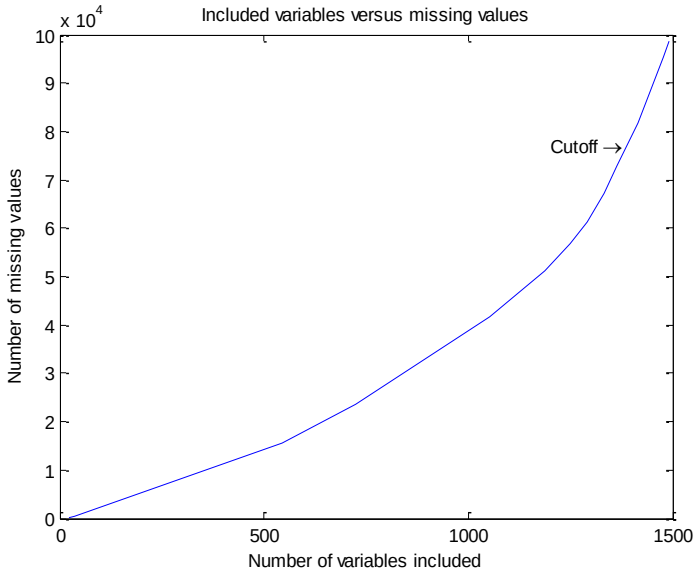
- The missing values are not randomly distributed over the variables. 25% of all missing values are found in only 9 % of the variables. The complete distribution of missing values over the variables can be seen in figure 2.1b.
- The missing values are also not randomly distributed over subjects. Here 27% of all missing values are found in 15% of the subjects. The complete distribution of missing values over the subjects can be seen in figure 2.1a.
- Part of the missing values are missing by design, due to the fact that they are not applicable given the value of another variable. For instance, if the subject has never used cocaine according to one variable, the values for the variables 'age at first time usage' and 'age at last time usage' are missing.
- The values that are not missing by design are mostly missing due to the fact that a subject did not perform one or more of the tasks, causing a chunk of values to be missing rather than a couple of values per category. In the anamnestic part of the variables however there can be single variables missing per subject.

There are a number of techniques for handling missing values. There applicability depends on the characteristics of the missing values. The techniques considered for the forMINDS dataset are:

- Dropping variables.
  One way of handling missing data is to drop those variables that include missing values (Allison, 2002). In this particular case, dropping all variables with missing values would mean dropping over 99% of all variables. Using this technique on all variables is not appealing, however the number of missing values can be greatly reduced by removing those variables with a very high percentage of missing values (recall the uneven distribution of missing values over the variables). Their effect on other variables will then not be taken into account, however so are all other possible variables that could have been included but were not. Those variables that have more than 175 missing values are eliminated from the dataset. This cutoff point has been chosen based on the balance between missing values reduction versus the elimination of variables, see figure 2.1b, resulting in the removal of 128 variables.
- Dropping subjects.
  Another way to handle missing data is to drop the subjects that include missing values, i.e. listwise deletion of missing data (Allison, 2002). This might have an effect on how representative the sample is for the population of interest. In this specific case removing all subjects that have missing values would result in a sample size of 0 subjects. A better option seems to be to remove those subjects that account for a large percentage of the missing data. Again the cutoff point has been chosen based on the balance between missing values reduction and the elimination of subjects, see figure 2.1a. This results in the removal of 36 subjects.



a)



b)

Figure 2.1 These graphs show the relation between the number of subjects (a) or variables (b) that are included versus the number of missing values.

- Add a category for categorical variables.
  For categorical variables it is an option to add another category that represents a missing value. According to (Allison, 2002) this is not a good technique because it causes biased results in regular statistics. In the case of a Bayesian network it does not seem to be a useful technique either. When generating a structure nodes might end up being connected based on the dependency of missing values. This is not desirable and therefore this technique will not be used.

- Imputation.
  Another technique for handling missing data is to substitute the missing values, which is called imputation (Cohen, Cohen, West, & Aiken, 2002), (Allison, 2002) and (Newman, 2003). Using this technique means that missing values are replaced with a plausible guess or imputation. This is a common strategy when deletion of variables and/or subjects to handle missing values is insufficient, because there would be no dataset left if only these techniques would be applied . The remaining question when using this technique is what to substitute the missing value for. Common choices are the overall mean, the mean of a subgroup or a regression estimate (Allison, 2002) and (Newman, 2003). In regular statistics mean comparison is a central theme. Substituting missing values with the mean and therefore not changing the mean of a variable (overall or of a subgroup) seems plausible, although standard deviations are altered. A Bayesian Network however does not depend on the mean of a variable. When using the mean the probability of this value would increase and therefore results in the network would become biased. This substitution therefore seems not applicable. Using regression to impute missing values seems more interesting since this would impute the missing value with a more likely estimate based on other variables. The problem lies in the 'other variables'–part of this technique. With nearly 1400 variables this is hardly applicable. First of all this would mean that a regression should be made for each variable that has even one missing value. Secondly there are missing values in all other variables as well. How should the regression be derived from those? And if one would chose a subset of variables, what would make a suitable subset? There might be an answer to this last question. In this thesis the quality of the tasks and questionnaires is assumed to be sufficient. This would mean that patterns within tasks and questionnaires should remain the same given their dependence on other variables. Regression based on other variables within the same task or questionnaire might therefore be a valid way to impute the missing values. Unfortunately most of the time all values from a questionnaire or task are missing for a particular subject, making this approach unusable. Then what would make the most suitable substitution for missing values? The equivalence of a mean for regular statistics is the probability distribution of the variables in Bayesian networks. This type of imputation is called distribution imputation. For more information see (Little & Rubin, 1987) and (Royston, 2005). From the available values of a variable the distribution is computed using 15 equally sized bins. Each imputation is now a draw from the set of bin-values belonging to that specific variable according to the accompanying distribution. For a more formal description of the used method see appendix C for pseudo-code.
- Multiple imputation.
  A way to improve single imputation is a technique called multiple imputation (Royston, Multiple Imputation of Missing Values: Update, 2005) and (Newman, 2003). This means that the data is imputed multiple times to produce a set of differently imputed complete datasets. When using a regression approach for instance, different regressions due to different parameters can be used to generated the different datasets. The resulting datasets are then combined somehow (e.g. taking the average) to give an overall estimate of the parameters. When this would be applied to the use of probability distributions in single imputation, for instance by taking the average, the imputed values would migrate towards the most common value in the variable. This would undermine the idea of the usage of the

probability distribution, since this distribution would become distorted through these operations afterwards.

When the proposed operations have been performed on the forMINDS dataset 1384 variables and 206 subjects remain. The number of variables associated with each category can be seen in table 2.1.

2.4 Discretization

The dataset contains categorical as well as continuous and discrete variables. It is possible to use a hybrid Bayesian network. Such a network needs to be able to handle two extra types of conditional distributions. The conditional distribution of a continuous variable given discrete or continuous parents and the conditional distribution for a discrete variable given continuous parents. See (Russel & Norvig, 2000) and (Murphy K. , 2001; Murphy K. , 2012) for more information. It is possible to implement Gaussian nodes with the used software, though it makes structure learning and inference more complicated. A second option is to transform continuous variables into discrete variables. This means there is a loss of information on one hand and a gain in simplicity on the other hand. Since the scope of a bachelor thesis is not infinite simplicity is chosen over a more detailed network in this case.

There are different ways to make discrete variables out of continuous variables. The method used here creates uniformly sized bins, with the minimum of the variable as lower boundary of the smallest bin and the maximum of the variable as the higher boundary of the largest bin. For a more formal description of the used method see appendix C for pseudo-code. An overview of all dataset operations can be seen in figure 2.2.

2.5 Software

A lot of software has been made to apply Bayesian techniques. These vary in a lot of ways. On (Murphy K. , 2012) a large overview can be found of software for this purpose and their specifications. The required specifications for this research are that it needs to be able to handle a large amount of nodes, it needs to incorporate structure generation techniques (preferably a number of algorithms), it needs to be able to learn parameters from data, inference has to be possible (preferably multiple methods) and preferably the software is open source. The two candidates from (Murphy K. , 2012) that seem suitable are GeNIe & SMILE from (GeNIe & SMILE) and BNT (Murphy K. , 2007). Smile turned out not to be suitable because it was unable to handle the amount of nodes required in this case. This inability became apparent after experimentation with the software and personal communication with the staff from GeNIe & SMILE. The used software for this project is therefore BNT, which is an open source toolbox for Matlab which is available on (Murphy K. , 2007). In the remaining sections the Bayesian techniques are discussed that are provided by the software and applied on the forMINDS dataset. An overview of these techniques can be found in (Murphy K. , 2001).

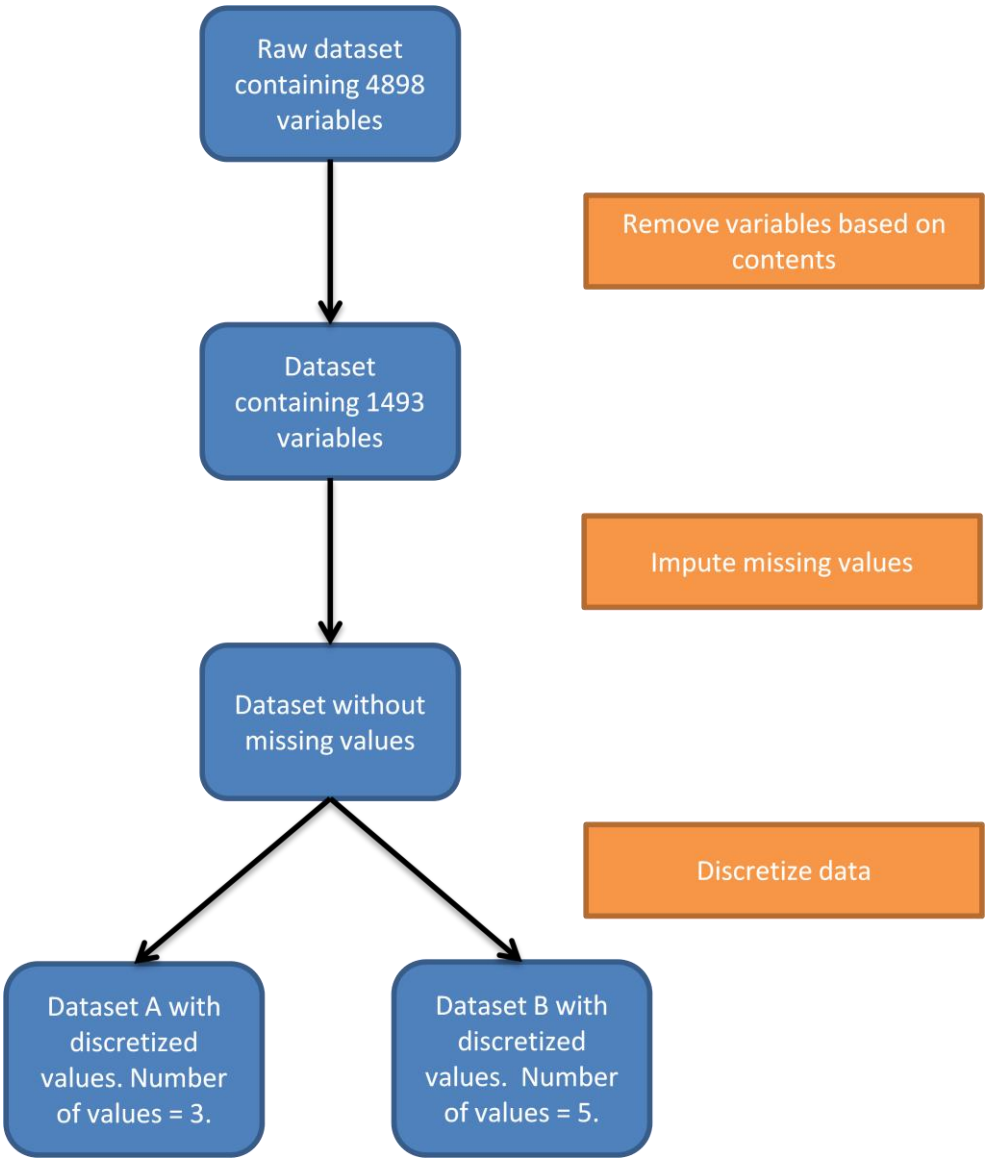


Figure 2.1 Overview of dataset operations

2.6 Structure learning

Learning a bayesian network from data is a challenging task. The number of possible DAGs is super-exponential in the number of variables (Heckerman, 1998). The methods to learn a structure from data can be divided in two types; constraint-based and search-and-score. The first type tries to form a dag using the constraints explained in section 1.4.3. The second type searches the space of possible DAGs using a score for the goodness of the model. In this section the options for structure learning given the software will be discussed and the resulting choice will be explained further.

2.6.1 Options for structure learning

There are a number of structure learning possibilities in BNT. Each will be very shortly discussed below.

- K2 algorithm.
  This is a greedy search algorithm. Initially each node has no parents, then incrementally that

parent is added which most increases the score of the resulting structure. For more detailed information see (Cooper & Herskovits, 1995).

- Hill-climbing.
  This algorithm starts at a specific point in the search space. It considers all nearest neighbor and moves to the neighbor that most increases the structure score. Neighbors are defined as adding, removing or reversing an arc in the network.
- Markov Chain Monte Carlo (MCMC) method.
  Uses the Metropolis-Hastings algorithm to search the space of all DAGs. For a more detailed explanation see (Chib & Graanberg, 1995).
- Structural EM.
  This method uses the more general expectation-maximization (EM) algorithm. This is an iterative method for finding maximum likelihood estimates of parameters. The iteration alternates between computing the expectation (E-step) and trying to maximize this expectation (M-step). For more details on the application for Bayesian networks see (Bishop, 2006).
- The PC-algorithm.
  This method starts with a fully connected network and removes edges based on conditional independence constraints. This will later be explained in further detail.
- The Fast Causal Inference (FCI) algorithm.
  This algorithm extends the PC-algorithm by being able to detect the presence of latent variables. More detailed information can be found in (Spirtes, Glymour, & Scheines, 2000).

The K2 algorithm heavily depends on the ordering of the nodes for the resulting network structure (Friedman & Koller, 2000). The forMINDS dataset contains so many variables there are so many possible orderings that this effect is not acceptable, since it is impossible to use all different orderings. This effect can be reduced by searching over the possible orderings using a MCMC method, though this increases the complexity of the resulting algorithm (Friedman & Koller, 2000). Hill Climbing can get stuck in local maxima. Starting at different points in the search space reduces this effect, however with so many variables this would have to be a large number of starting points in order to have any confidence that the resulting model is not a (small) local maximum (Russel & Norvig, 2000).
The MCMC method is not usable in this specific case because the implemented version in the software is can handle only a maximum of 10 nodes according to the user manual (Murphy K. , 2001). The remaining three methods, EM-algorithm, PC-algorithm and FCI-algorithm, could all be applied to the forMINDS dataset. One of the aims of this thesis is to be able to inspect the global network structure for possibly interesting configurations. The constraint based methods (PC and FCI) have a more insightful way of constructing a network; it has a specific meaning when a particular edge is missing. This insightfulness is missing in the EM-algorithm which therefore seems less appropriate. The downside of these algorithms is the repeated use of conditional independence tests, since this decreases its statistical power. The difference between the PC- and the FCI- algorithm is the possibility of detecting latent nodes. The forMINDS dataset includes so many variables that detecting latent variables might not be interesting at the first attempt to apply such a technique to this dataset. This might be interesting for future research. Because of its applicability, insightfulness and simplicity, the PC-algorithm seems the best algorithm to perform structure learning on the forMINDS dataset. The algorithm itself will be explained in more detail in the following section.

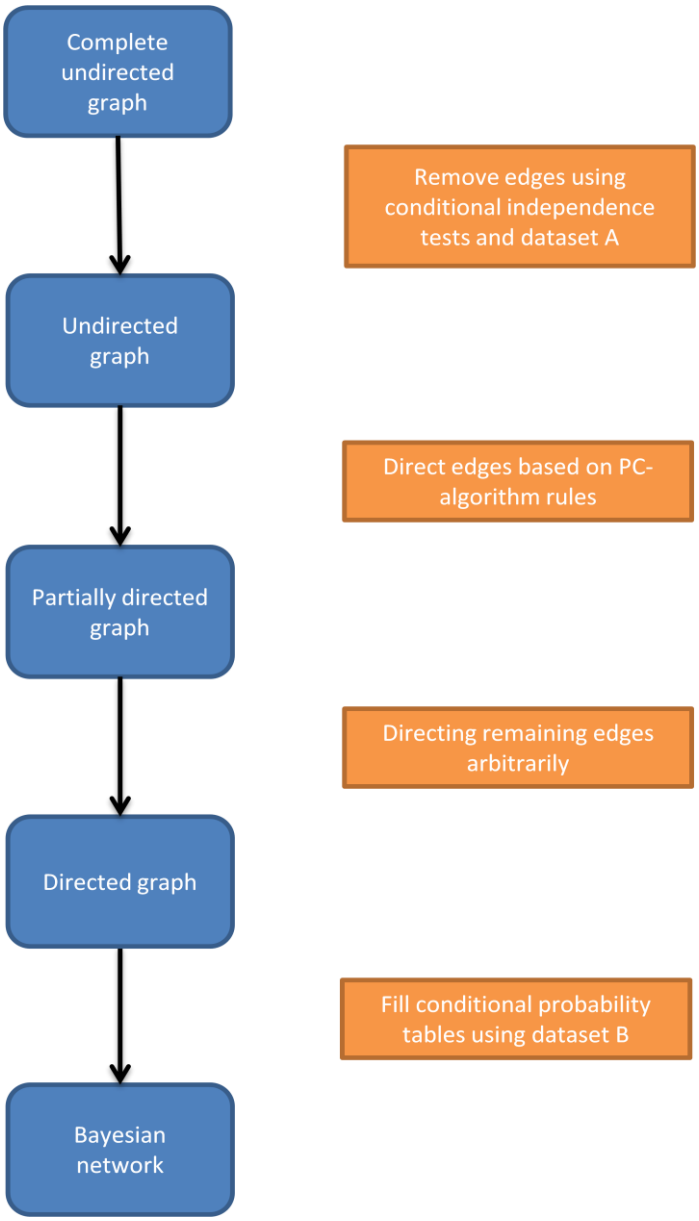


Figure 2.2Overview of generating a Bayesian network

2.6.2 The PC-algorithm

The PC-algorithm starts with a fully connected undirected graph. The algorithm then iterates over the edges to check if the nodes that an edge connects are conditionally independent. If so, the edge is removed. During the iterations the order of conditional independence is raised by one after each iteration, starting from zero. This means that at first all edges are checked for regular independence, without any conditional variables. Secondly the remaining edges are checked for conditional independence given each of their adjacent nodes. Thirdly the remaining edges are checked for conditional independence given each set of two of their adjacent nodes, and so on. For a schematic version see figure 2.3a.The output of this first stage is an undirected graph.

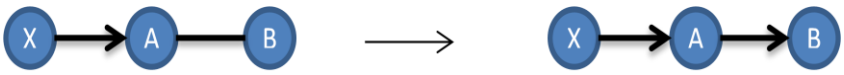
Figure 2.3Pc algorithm. In these figures the pseudo code for the structure ordering (a) and rules for directing nodes (b) are shown

V is the set of nodes
ADJ_X is the set of adjacent nodes from node X
$I(X, Y|S)$ is the test for independence of X and Y given the set of nodes S
S_{XY} is the set of nodes that separates X and Y, i.e. given this set X and Y are independent

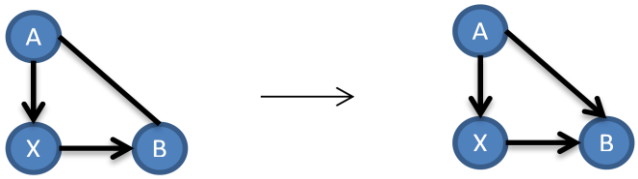
1. Start with a complete undirected graph gp
2. $order$ = 0
3. Repeat
4. For each $X \in V$
5. For each $X \in ADJ_X$
6. Determine if there is a set $S \subseteq ADJ_X - \{Y\}$ with $|S| = order$ and $I(X, Y|S)$
7. If this set exists
8. Make $S_{XY} = S$
9. Remove $X - Y$ link from gp
10. $order = order + 1$
11. Until order > maximal order or $|ADJ_X| \leq order, \forall X$

a)

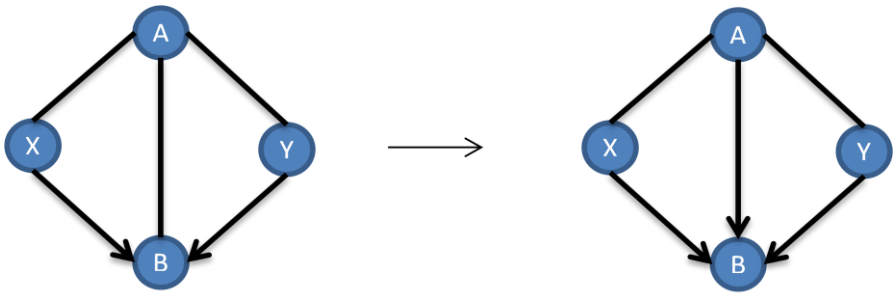
1)



2)



3)



b)

In order to have a DAG for the Bayesian network the edges need to be oriented. First the undirected graph is searched for $X - Z - Y$ connections where X and Y are not adjacent. If variable Z was not in the set based on which X and Y are concluded to be independent, $X - Z - Y$ is oriented as $X \rightarrow Z \leftarrow Y$, which is a head-to-head link. Next a set of three if-then rules is now iterated over the graph to direct edges until no more edges can be directed. The rules can be seen in figure 2.3b and more information can be found in (Meek, 1995). The resulting partially oriented graph represents a class of DAGs which are essentially equivalent. The remaining arcs are oriented on an arbitrary way, keeping the DAG conditions and not creating head-to-head links. For a more detailed description of the PC-algorithm see (Spirtes, Glymour, & Scheines, 2000). To see an overview of the different phases of the PC-algorithm and their corresponding in- and outputs see figure 2.2.

2.6.3 Independence testing

As described above the PC-algorithm needs a conditional independence test. The test for discrete variables described in (Spirtes, Glymour, & Scheines, 2000) is based on observed and expected frequencies. These frequencies can be used to derive conditional independence as follows.

Let N be the number of observations, $O(X = x)$ be the observed frequency of the value x of variable X. Assuming that X and Y are independent, the expected frequency of the co occurrence of value x in X and y in Y is:

$$E(X = x, Y = y) = N \cdot P(X = x, Y = y) = N \cdot P(X = x) \cdot P(Y = y)$$

Estimating $P(X = x)$ and $P(Y = y)$ by using the observed frequencies yields:

Equatio 2.1. $$E(X = x, Y = y) = N \cdot \frac{O(X=x)}{N} \cdot \frac{O(Y=y)}{N} = \frac{O(X=x) \cdot O(Y=y)}{N}$$

Now, let $E(X = x, Y = y|S = s)$ be the conditionally expected frequency of the co occurrence of X and Y under variable S.

Equation 2.2. $$E(X = x, Y = y|S = s) = N \cdot P(X = x, Y = y|S = s) = N \cdot \frac{P(X=x,Y=y,S=s)}{P(S=s)}$$

Again, assuming of X and Y are independent we find:

Equation 2.3. $$E(X = x, Y = y|S = s) = N \cdot \frac{P(X=x,S=s) \cdot P(Y=s,S=s)}{P(s=s)}$$

Estimating $P(X = x, S = s)$, $P(Y = s, S = s)$ and $P(s = s)$ using observed frequencies yields:

Equation 2.4. $$E(X = x, Y = y|S = s) = N \cdot \frac{\frac{O(X=x,S=s)}{N} \cdot \frac{O(Y=y,S=s)}{N}}{\frac{O(S=s)}{N}} = \frac{O(X=x,S=s) \cdot O(Y=y,S=s)}{O(S=s)}$$

It is easy to see that for more than one conditional variable (for example 2), the formula expands to:

Equation 2.5. $E(X = x, Y = y|S = s, T = t) = \frac{O(X=x,S=s,T=t) \cdot O(Y=y,S=s,T=t)}{O(S=s,T=t)}$

To test independence there are two options: the χ^2 test and the G^2 test. Given X, Y and a number of conditional variables $S_1, S_2 \dots S_m$ we will determine the observed values $O(X = x, Y = y|S_1 = s_1, S_2 = s_2, \dots, S_m = s_m)$ (by counting) together with the expected values $E(X = x, Y = y|S_1 = s_1, S_2 = s_2, \dots, S_m = s_m)$ (by calculation) for all possible values $i, j, s_1, \dots, s_m$. χ^2 is than calculated by

Equation2.6. $\chi^2 = \sum_{i \epsilon X} \sum_{j \epsilon Y} \sum_{s_1 \epsilon S_1} \dots \sum_{s_m \epsilon S_m} \frac{(E-O)^2}{E}$

And G^2 is calculated by

Equation 2.7. $G^2 = 2 \cdot \sum_{i \epsilon X} \sum_{j \epsilon Y} \sum_{s_1 \epsilon S_1} \dots \sum_{s_m \epsilon S_m} O \cdot ln\left(\frac{O}{E}\right)$

The χ^2 test is in fact an approximation of the log-likelihood ratio on which the G^2 test is based (Dunning, 1993). This approximation was developed by Karl Pearson because at the time it was unduly laborious to calculate log-likelihood ratios. The authors in (Spirtes, Glymour, & Scheines, Causation, Prediction, Search, 2000) have found, through simulations, that using the G^2 statistic more often leads to the correct graph than does χ^2 when dealing with discrete nodes.
The appropriate p-value indicates $P(H_0|data)$, where H_0 is the hypotheses that two variables are independent. In case of two dependent variables, this p-value will be very low. To find the appropriate p-value for G^2, the correct number of degrees of freedom, df, is needed. Let $Val(X)$ be the number of values of X. The following value for df will be used:

Equation 2.8. $df = (Val(X) - 1) \cdot (Val(Y) - 1) \cdot \prod_{i=1}^{n} Val(S_i)$

In case the distribution has a zero entry, the number of degrees of freedom is decreased by one as recommended by (Bihop, Fienberg, & Holland, 1975) and (Spirtes, Glymour, & Scheines, 2000). It is also recommended by these authors that the sample size needs to be at least five times larger than the number of cells in the independence test. The maximum number of values for a variable is 3 in this dataset, recall that this is forced through discretization. This means that, given 206 subjects, the maximal order of the independence test is three and therefore the maximum number of conditional variables is two.
The final decision in independence testing is the value of alpha. Alpha is used to decide at what p-value dependence is concluded; if the p-value is lower than alpha the two variables are concluded to be dependend. The alpha-level of the conditional independence test influences how many connections between nodes will be removed. As the order of the conditional independence test becomes larger, the number of possible combinations for conditioning rises sharply. The number of nodes in this network is large, therefore fast reduction of connections is desirable. The value of alpha is set to 0,01. The conditional independence test as described above is not implemented in BNT. The used implementation is provided in appendix C.

2.6.4 Background knowledge

It is possible in the use of Bayesian networks to supply the structure learning algorithm with background knowledge consisting of forbidden and forced arcs. This should result in a better end result with regard to network structure. In this case this pos network techniques are useful for this type of forensic psychiatric datasets and what resulting network it delivers without any assumptions. This would be interfered by background knowledge. Secondly, selecting these forced and forbidden arcs would require selecting them from $\frac{N^2-N}{2}$,where N is the number of variables, possible arcs, which is beyond the scope of this project.

2.6.5 Assumptions of the PC-algorithm

The PC-algorithm is bound in its success by a number of assumptions (Kalisch, Mächler, Colombo, Maathuis, & Bühlmann, 2012). These are:

1. The dataset must be faithful. This means that for each distributions in the dataset it is possible to find a DAG, whose list of d-separation relations (see section 1.7.3) perfectly matches the list of conditional independencies of the distribution (Kalisch, Mächler, Colombo, Maathuis, & Bühlmann, 2012).
2. No hidden or selection variables. Hidden variables are factors influencing two or more measured variables that may not themselves be measured. Selection variables are variables of which their values may influence whether a unit is included in the data sample. (Kalisch, Mächler, Colombo, Maathuis, & Bühlmann, 2012).
3. Consistent in high-dimensional settings if the underlying DAG is sparse, the data is multivariate Normal and satisfies some regularity conditions on the partial correlations and α is taken to zero appropriately (Kalisch & Bühlmann, 2007).

For all these assumptions it must be noted that they become apparent after using the PC-algorithm. Whether the data is faithful is hard to know on forehand, although it has been shown that the set of distributions that are faithful is the overwhelming majority (Meek, 1995). Whether there are hidden or selection variables and whether the underlying DAG is sparse is also hard to predict on forehand.

2.6.6 Complexity of the PC-algorithm

The maximal number of independence tests that have to be performed by the PC-algorithm for a graph G is bounded by the largest degree in G and the maximal order of the conditional independence tests which is denoted as k. Since the algorithm starts with a fully connected graph, given there is no background knowledge, the maximal degree of a vertex equals the number of vertices which is denoted as n, This results in the following upper bound (Spirtes, Glymour, & Scheines, 2000):

Equation 2.9 $$2\binom{n}{2}\sum_{i=0}^{k}\binom{n-1}{i}$$

Which is bounded by:

Equation 2.10 $$\frac{n^2(n-1)^{k-1}}{(k-1)!}$$

This means the computational requirements increase exponentially with k. However, when taking into consideration that the maximal order of the conditional independence test is 2 (see previous section) the maximal number of tests is bounded by:

Equation 2.11 $$2\binom{n}{2}\sum_{i=0}^{2}\binom{n-1}{i}$$

Which is then bounded by:

Equation 2.12 $$n^2(n-1)$$

Making the complexity of the algorithm quadratic instead of exponential. This upper bound is the worst case, which requires that there are no conditional independencies found with an order less than the maximal order. According to (Spirtes, Glymour, & Scheines, 2000) the worst case is extremely rare, and the average number of conditional independence tests is much smaller.

2.7 Parameter learning

When a network structure is defined the conditional probability tables need to be constructed for each node. The software provides a method for learning these parameters in the presence of missing values. This would be a desirable method to use since the original data could then be used instead of the data with imputed variables. However the provided code does not work properly. The remaining option for learning parameters is to use the data with the imputed values. In this case the parameters are learned by finding a point estimate of the parameters. These are maximum likelihood estimates.

2.8 Inference

Once the Bayesian network is complete, i.e. the structure and conditional probability tables have been generated, the next phase is inference. As described in the introduction this means any conditional probability can be inferred from the network. This is useful for e.g. hypothesis testing and to make predictions.

The software provides a number of methods to perform inference which will be discussed below.

- Global inference. This is the brute force method of calculating the probability distributions given evidence as described in the introduction. Since this is exponential in the number of variables this is not a useful method. For further reading see (Russel & Norvig, 2000) and (Spirtes, Glymour, & Scheines, 2000).
- Variable Elimination. This method avoids repetition of calculation and therefore increases performance. Unfortunately it is still exponential if the network structure is not a singly connected network, which means there is only one possible path from each node to every other node. This is highly unlikely to be the case in such a large network and therefore this is not a useful method. For further reading see (Russel & Norvig, 2000) and (Kschischang, B., & Loeliger, 2001).
- Quickscore. This method is mostly interesting for networks containing noisy or nodes and is therefore not suitable in this particular case. More detailled information can be found in (Heckerman, A tractable inference algorithm for diagnosing multiple diseases, 1989).

- Belief propagation. This is based on Pearl's belief propagation algorithm (Pearl, 1988), which is a technique to approximate parameters. In (Murphy, Weiss, & Jordan, 1999) it is stated that when the output of the algorithm converges the results are very good, however it might oscillate which causes very poor approximations. Whether or not oscillation will occur is hard to predict. A technique is proposed to prevent oscillation. Unfortunately this technique can make the algorithm converge to bad approximations. Because of these insecurities this is not a suitable option.
- Sampling. This type of techniques generates samples from the network. In the simplest case a large number of samples are generated and the requested query is answered through counting, however this is very inefficient. Two more efficient options provided by the software are likelihood weighting and Gibbs sampling. Likelihood weighting generates a sample through the probability distributions in the network until it reaches an evidence node. This variable is assigned the evidence value and the sample is weighted according to the probability of the value of the evidence node occurring. This way no redundant samples are generated. The Gibbs sampling method uses a MCMC method. It differs from likelihood sampling in the fact that the samples are dependent on each other as opposed to independence in likelihood sampling. Gibbs sampling is unable to handle networks that contain extreme probabilities. These extreme probabilities are very small priors. For more information on Gibbs sampling and likelihood sampling see (Geman & Geman, 1984). Since there might be small priors in the network importance sampling seems the best option in this case.

Chapter 3 Encountered problems and solutions

When generating the network structure as proposed in the method section a number of problems have been encountered.

3.1 Computation time of structure generation

Running all 1384 variables has resulted in a too long lasting calculation for building a network structure. The slowness is due to the first part of the PC-algorithm where all edges need to be validated using the conditional independence test, especially the second-order phase (independence testing conditional to two other variables). Recall that for each edge present in this phase $\binom{N}{2}$ independence tests need to be done, where N is the number of neighbors for the two nodes connected by that specific edge. Even though the reduction of the number of edges was 93.93% after the zero-order independence tests, there still were over a 100.000 edges left. The first-order phase reduced the number of edges with another 7,5%, leaving the second-order phase with too many edges to compute within anywhere near reasonable computation time. After running for 6 days less than 1000 out of 107555 edges had been handled and the calculations were ended.

In order to generate a result, the dataset needed to be reduced to be able to generate a network in a reasonable computation time, being in the magnitude of days. The test sizes that have been used consist of respectively 286, 178 and 132 variables. Included variables have been chosen in consultation with the researchers from the forMINDs project for all sets. The results will be further discussed in the next sections.

3.2 Edge reduction

The computation time heavily depends on the number of edges that are still present during the first-order and second-order phases of the first part of the PC-algorithm as discussed above. Reducing the α of the conditional independence tests therefore seems a way to reduce computation time, since it would be expected that less edges remain for each higher order phase. Secondly a resulting test network structure of 30 variables from the forMINDS dataset still contained 143 edges for an α value of 0.01, an average of 10 neighbors per node, which makes the network rather complex for interpretation. Reducing the number of resulting edges might therefore be desirable.

In (Kalisch & Bühlmann, 2007) the dependence of the PC-algorithm on its single tuning parameter α is compared for different numbers of observations, different levels of sparseness of the underlying DAG and its True Positivity Rate and False Positivity Rate compared to the true underlying DAG. For this comparison the authors use simulated data with 30 variables and values are averaged over 50 runs. The authors conclude that α can be used to find a good compromise between the amount of edges and their reliability. It is noted, however, that especially for larger sample sizes large changes in α result in small changes in the number of edges.
For this specific dataset a number of values for α have been compared for a set of 30 variables and the set of 132 variables. The results will be further discussed in the next sections.

It must be noted, however, that there might be another cause for the small reduction of the number of edges. Recall from section 2.6.3. that the conditional independence test heavily depends on how

the values for the subjects of the variables involved are distributed over the possibilities of combinations of values of those variables. In other words: how well each cell of the frequency table for the observed values is filled. For each empty cell the degrees of freedom is lowered by one. In case this value becomes lower or equal to zero, i.e. a large number of cells is empty, dependence is assumed. If this occurs regularly, a lot of edges appear in the network based on this assumption regardless of the level of α. The occurrence of this situation is further discussed in the next sections.

3.3 Cycles in resulting network structure

The dataset that resulted in the furthest completed network structure was the dataset containing 132 variables. The network structure generated by the software already contained a cycle in the first phase of directing the edges (using the separation sets to make head-to-head orientations). After reconsulting the literature it appeared that in case of sampling errors or hidden variables, conflicting information about edge directions might arise (Kalisch, Mächler, Colombo, Maathuis, & Bühlmann, 2012). It might be, for example, that two triples, $\mathrm{a} - \mathrm{b} - \mathrm{c}$ and $\mathrm{b} - \mathrm{c} - \mathrm{d}$, should both be oriented head-to-head in the first phase of directing edges based on the separation sets. This results in a conflict about the edge $\mathrm{b} - \mathrm{c}$, since it should be oriented as $\mathrm{b} \leftarrow \mathrm{c}$ in the first triple and as $\mathrm{b} \rightarrow \mathrm{c}$ in the second triple. Some configurations of these ambiguous edges might result in cycles and therefore an invalid network structure, others might not.

There are a number of options mentioned in the software proposed in (Kalisch, Mächler, Colombo, Maathuis, & Bühlmann, 2012) how to deal with the fact that ambiguous edges due to sampling errors or hidden variables occur.

- The default option is to simply overwrite the ambiguous edges as they occur.
- The second option is to search for a configuration of the ambiguous edges that results in a valid DAG. In the implementation in (Kalisch, Mächler, Colombo, Maathuis, & Bühlmann, 2012) a maximum of 100 such configurations are tried.
- A third option is to discard all information on directing and simply generate a random DAG on the skeleton (by which they mean the graph containing only undirected edges).
- A final option is to implement an intermediate step in the PC-algorithm, making it the conservative PC algorithm (Kalisch, Mächler, Colombo, Maathuis, & Bühlmann, 2012), which works as followed. After the skeleton is generated, all head-to-head triples are marked as either faithful or unfaithful. It is checked if a triplet $\mathrm{a} - \mathrm{b} - \mathrm{c}$ is faithful in the following way in: "We test whether variable a and c are conditionally independent given any subset of the adjacency set of a or subset of the adjacency set of c. If b is in no such conditioning set (and not in the original separation set) or in all such conditioning sets (and in the original separation set), the triple is marked as unfaithful". The faithful triples are then directed in the next step as v-structures and the unfaithful ones not. The remaining edges are then directed as before.

Overwriting ambiguous edges as they occur is the method used in the BNT software as well, resulting in cycles in this specific case. Discarding all independence information on directing and generating a random DAG on the skeleton does not seem a good option. Recall from section 1.7.3. that the interpretation of a network structure depends on its Markov-blanket, being a nodes parents, children and children's parents, which involves direction. Making a random configuration on the available skeleton would therefore result in an uninterpretable network. The conservative PC-algorithm is not

implemented in BNT. Due to time-constraints this technique is not used. The resulting option to search for a configuration of the ambiguous edges resulting in a valid DAG is used, both for the dataset containing 30 and 132 variables. This is primarily done by locating the cycle in the current graph and then turnover the first randomly chosen edge in this cycle that is ambiguous. If this does not perform well computation time wise due to the time it costs to locate cycles the method of the package for R is used and a fixed number of possible configurations of the ambiguous edges will be tried.

Chapter 4 Results

This thesis has resulted in an explorational study for the usage of Bayesian networks for this specific dataset. Different aspects of this exploration are effect of imputation , computation time, effect of α and the (partially) generated networks. The results of each of these aspects will be discussed below.

4.1 Effect of imputation

In order to try and estimate the effect of imputation using two different imputation methods a simulation has been performed. A dataset has been generated consisting of 3 variables. The first two variables are random numbers between 1 and 100 and the thirds variable contains the sum of the first two variables, where any number larger than 100 is set to 100. Applying the PC-algorithm on this dataset results in a v-structure of the form: $1 \rightarrow 3 \leftarrow 2$, where 1 is the first variable, 2 the second and 3 the third. At the start of the simulation 10% is made missing and this is increased by 10 % at each step of the simulation. These missing values are equally distributed over the entire dataset. Each dataset is imputed two times, one time with the average of the value and one time with a draw from the distribution of the values of that value as described in section 2.3. After imputation the values are discretized as described in section 2.4 and the PC-algorithm is applied on each dataset to see if the v-structure is still intact. In this simulation the v-structure was no longer present if 90% of the data had been imputed with the average and in the case of distribution based imputation the v-structure was no longer present when 50% of the data had been imputed. Let V be the set of variables and S the set of subjects, than the error of the imputation method can be calculated using

Equation 4.1. $$\sum_{i \epsilon V} \sum_{j \epsilon S} (original\ value - imputed\ value)^2$$

For imputing with the average in a dataset with 30% missing values this yields an error of 676.088 and for distribution based imputation an error of 1.224.072.

4.2 Computation time for sets of variables of different sizes

For three sets of variables of respectively 286, 178 and 132 variables the computation time of a network is compared for an α of 0.05. Only for the set of 132 variables the computations have been completed. Therefore only the phases of the PC-algorithm using zero-order and first-order conditional independence tests are compared. The results can be seen in figure 3.1. For the set of 132 variables the computation time for structure generation was 93 hours, i.e. 3 days and 21 hours.

4.3 Remaining edges for different levels of α

For two sets of variables of respectively 30 and 132 variables the number of edges remaining after independence testing are compared for different levels of α. The set of 132 variables is the smallest set of carefully chosen variables. The set of 30 variables is a test set that simply contains the first 30 variables of the set of 132 variables. The included variables are listed in appendix X. The set of 30 variables starts out with 435 connections, and the set of 132 variables with 8646. The results are visualized in figure 3.2 for each order of the independence tests done in the first part of the PC-algorithm. As described in the previous chapter these results might be influenced by the occurrence of too low values for the degrees of freedom in the conditional independence test. More information on these occurrences will be discussed in the next section.

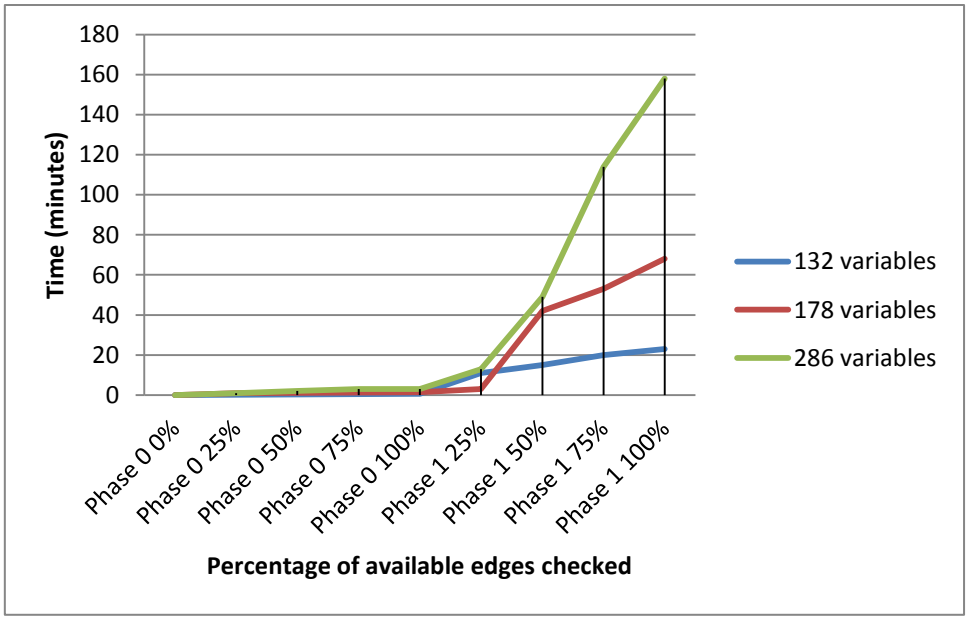


Figure 4.1 Comparison on computation time for datasets of different sizes to complete the first two phases (zero-order and first-order conditional independence tests) of computing the network structure

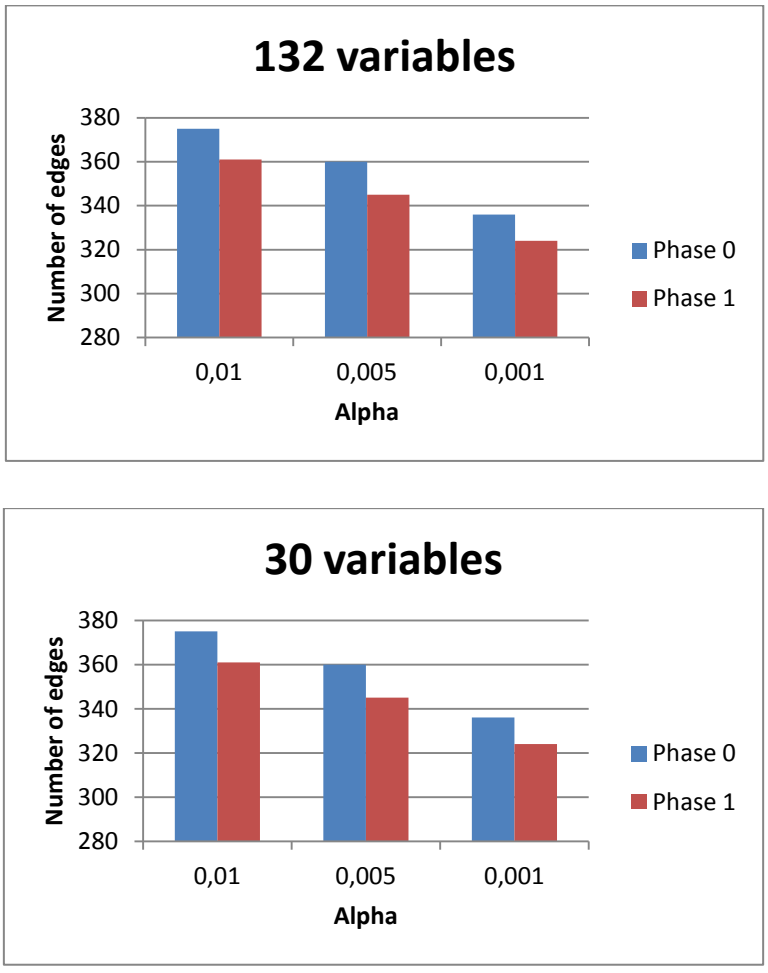


Figure 4.2. Number of variables remaining after each phase of independence testing for different values of alpha. In phase 0 the zero-order indepence tests are performed and in phase 1 the first-order independence tests. The second order independence tests are not included, since this phase has never finished completely.

4.4 Resulting networks

For the smallest network (132 variables) still containing carefully chosen variables a finished network has not been generated. The results on network structure discussed below will therefore refer to different networks in different stages of completion.

4.4.1 Network skeleton

The network skeleton is the undirected graph that is the output of the first part of the PC-algorithm. For the dataset of 132 variables the skeleton has not been completed. The results following will refer to the output at the end of the phase with first-order independence tests. At the moment of aborting the computation in the phase with second-order independence tests, no more edges had been removed and 36 of the 1231 edges had been tested.

The resulting network skeleton contains 1231 edges. Averagely each node has 19 neighbors. How these neighbors are distributed over all nodes can be seen in figure 3.2.

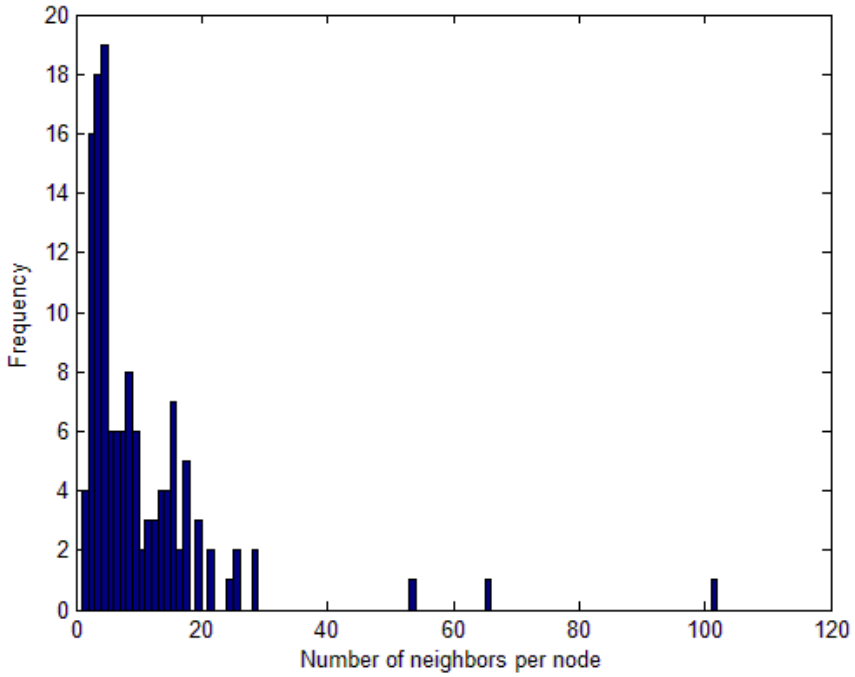


Figure 3.2. Distribution of neighbors for the network with 132 variables and α = 0.001. The network generation of this network is not complete for the second-order independence tests.

This network results in a too complex picture to display here. The network is digitally available on request. To give an idea of the complexness of such a network a network with 30 variables and an average of 10 neighbors per node can be found in figure 3.3.

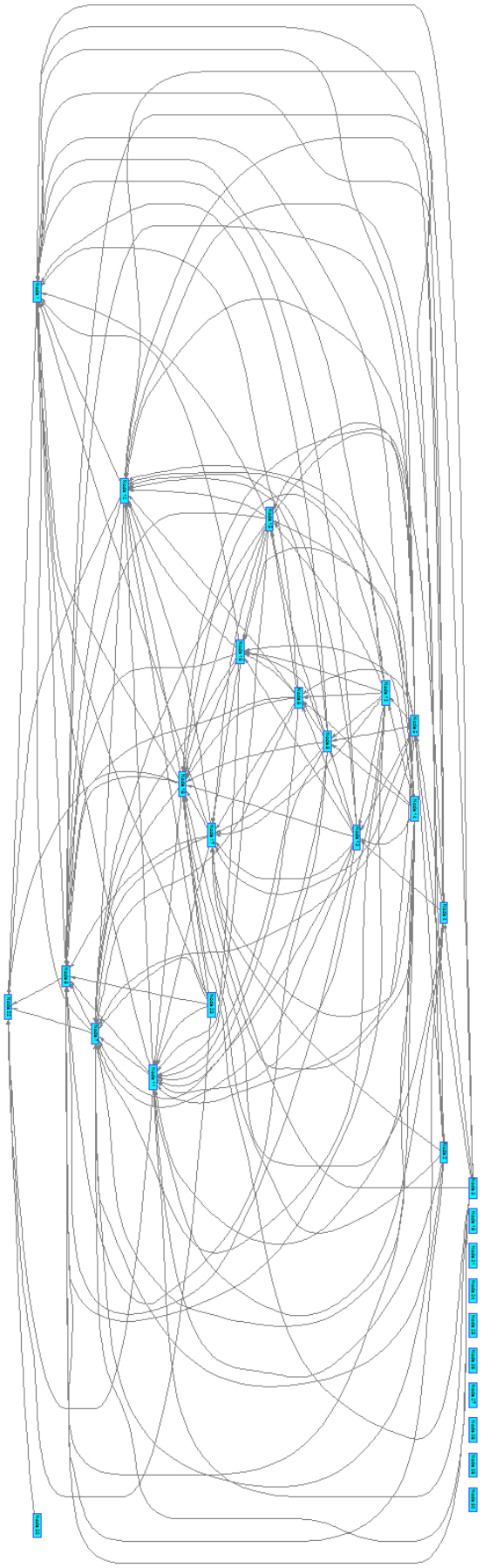


Figure 3.3 Network structure with 30 variables and α = 0.01

As discussed in the previous chapter a number of edges in this network skeleton might not result from statistical significance. Instead there are too many combination of the values of variables involved, hereafter referred to as cells, for which there are no observations to conclude independence and therefore the edges are not discarded. Within this network consisting of 123 variables there are 770 such edges encountered in the zero-order phase of constructing the network structure, which is 57% of all remaining edges. In this specific phase only one independence test per edge is performed. In the first-order phase 78.073 independence tests are unable to make a conclusion due to too many cells without observations. Part of these tests are on those same edges that did not have enough observations per cell in the zero-order phase to begin with. Recall that the number of independence tests that needs to be performed in these higher order phases depends on the amount of neighbors of the nodes connected by that particular edge.

4.4.2 Directing edges

After generating the skeleton the edges are directed. Recall that the first step in directing those edges is based on the separation sets generated in the previous phase and that ambiguous edges whose direction was uncertain cause cycles when the way to handle those ambiguous edges is to simply overwrite them as they occur. As described in the previous chapter two strategies have been tried to search for a configuration of those ambiguous edges. The first strategy, to only turnover ambiguous edges within a cycle, resulted, for a test-set of 30 variables, in a valid DAG. Unfortunately this strategy was not feasible for the set of 132 variables (only zero- and first-order independence tests) since the search for a cycle was in the network was too time consuming relative to the number of random configurations that could be tried in that time. Therefore the second strategy, to try a fixed number of possible configurations, has been applied to this network. However, after 10.000 random turnovers of edges (100 times the number of configurations tried in the R-package (Kalisch, Mächler, Colombo, Maathuis, & Bühlmann, 2012)) still no valid DAG had been found.

4.4.3 Most significant dependences

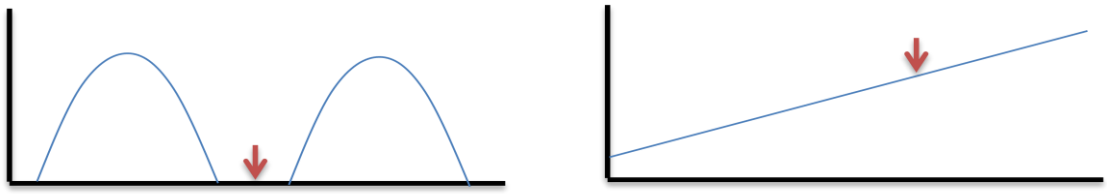
For the dataset with 132 variables there are, in the zero-order phase alone, 485 edges with a p-value which is smaller than $1{,}0 \cdot 10^{-8}$. Since any value equal or smaller than this value is registered as 0, it is not possible to sort the remaining edges based on their significance. Even though 485 edges are too much to display here, these are digitally available on request, as well as the significance of all other independence tests performed.

4.4.4 Conditional probability tables and inference

For the network containing 132 variables no valid DAG has been found and therefore it has been impossible to construct the conditional probability tables or conduct inference. For a test-network of 30 variables it has been possible to find a valid DAG. The number of edges in this network was too large, however, to construct the conditional probability tables due to memory failure when using BNT.

Chapter 5 Discussion

In the simulation regarding the comparison between imputing with the average and distribution based imputation, average imputation yields better results on the used dataset. However, this does not mean this is the case on all datasets. In this specific dataset the variables were uniformly distributed. This means that the average is always maximally half the reach of the variable besides the original value, while distribution based imputation might be further off. This explains the larger error for distribution based imputation. In case the values of a variable are normally distributed little difference between the two methods is expected, since the average has the highest probability of being imputed with distribution based imputation. However, if the values are distributed asymmetrically and the average is not the most probable value such as in the examples of figure 5.1, it is easy to imagine that distribution based imputation might yield better results when compared to imputation of the average.



a) *b)*

Figure 5.1 Examples of asymmetric distributions. The red arrows indicate the average of the distribution.

Having discussed imputation it must be noted that however clever the method for dealing with missing values is, the best way to decrease the effect of missing values is to minimize their occurrence. Designing data collection in such a way that the number of missing values is minimal can be difficult, but will in the end result in more reliable results in any research.

Because of the dependence on the distribution of the values of the different variables it is hard to say how much the forMINDs dataset is affected by the imputation of the missing variables. However, it is not likely that this effect is small, because of the large amount of missing data in the dataset.
A different method for imputation that has not been considered for this thesis, but might yield the best result of all, is using a Bayesian network . In this case the values are imputed using a simple method such as the average or distribution. A Bayesian network is generated based on this imputed data, than the missing values are imputed again with the most likely value indicated by the Bayesian network using the values of all present values of variables for that subject as evidence. This method would not have been applicable for this data, since generating a network using the imputed values resulted in cycles.

The results of how computation time increases with the number of variables are in accordance with the worst case quadratic complexity (section 2.6.4), it increases less then quadratic with the number of variables. Even though this is considered to be computationally feasible, the larger datasets still had a computation time beyond reasonable. The computation time can be partly explained by the high number of neighbors per node that are still available in the higher-order phases, which are in turn partly caused by the number of edges which are considered dependend based on too many cells without observations. Another possible cause of long computation times is the independence test

used. A lot of operations are needed to calculate all observed and expected values. Maybe using a different, less time consuming method to decide (in)dependence might reduce computation time greatly. This method could be either a different statistic or maybe even a heuristic.

The number of resulting edges in the network is too high to result in a network that is sufficiently elementary to be able to interpret a network based only on structure. This is partly due to the number of edges which are considered dependent based on too many cells without observations. A cause for the frequent appearance of zero-entries in the frequency cells might be found in the method for discretization. The boundaries of the bins for discretization are determined by using the difference between minimum and maximum of a variable and then create equally sized bins. This method is rather sensitive to extreme values, which might result in bins without any observations. However, this explanation cannot account for all variables, since there are plenty of variables which contain categorical values in the first place, or for which extreme values have already been removed (e.g. reaction times, see section 2.2). Another case in which this method of discretization has a negative effect on the number of zero-entries would be with variables with a distribution as can be seen in figure 2.3a. There are alternatives such as using equally sized bins, but instead of using the minimum and maximum of variable as a measure to determine the boundaries of the bins, use those values for which a large percentage (e.g. 80%) of the values lie in between these values. Another option is to use the ranking of values to determine the boundaries of each bin. In this case the values of a variable are ordered, and bin boundaries are chosen such that each bin contains an equal amount of values which can be easily done by counting. Which is the most suitable way of discretization depends on the nature of the variable. The best method of discretization for a dataset is probably customized discretization for each of its variables.

Chapter 6 Conclusion

The research question of this thesis as stated in the Introduction was as follows:

“Does a Bayesian network form an inspiration for possible quantitative research and does it give a more general insight of the relations between the variables in the forMINDS dataset?”

The answer to this question is: not up until now. It has appeared that, in the limited time for a bachelor thesis and with the data at hand it has not been possible to generate a valid network at all. And even if a valid network would have been generated, its reliability would be questionable. If one adds up the amount of missing data (30% of all data), leaving out 90% of the variables (132 as opposed to 1394), a large number of questionable edges because of statistic indecisiveness due to lack of observations or discretization (at least 57% of remaining edges for 132 variables) and finally the occurrence of hidden or selection variables resulting in ambiguous edges and cycles, a resulting Bayesian network would not be a reasonably reliable source to base ones inspiration for future research on.

Even though there is no resulting network suitable for use by the researchers of the forMINDs project, there is still a gain for the forMINDs project. The nature and distribution of the missing values in the dataset have been researched. The presence of hidden or selection variables with regard to the 132 variables in that set are discovered. The dataset has been converted to something more suitable to be in a program. An overview of the most significant dependences is constructed. And last but not least, researchers of a field other than Artificial Intelligence have gained knowledge of the possible use of this type of techniques.

Chapter 7 Future research

For this specific dataset it is still possible to improve performance of the Bayesian network technique. A number of such options for possible improvement are:

- Using a different imputation technique. For instance the technique using a Bayesian Network such as mentiond in section 5, or use a customary technique per variable depending on the specifications of that variable.
- Using a different method for discretization. Again there might be a lot to gain if the method for discretization is chosen per variable, depending on the specifications of that specific variable. This might reduce problems with the occurrence of zero-entries.
- Making the statistical testing more efficient somehow might reduce computation time. On itself this does not gain improvement in performance quality wise, however it might be possible to include more variables. With more included variables the resulting structure might not contain cycles since the hidden variables could be included.
- Using a different method for structure generation. Maybe the used technique is not the most suitable for this dataset and a different technique will yield better results if applied.

References

Allison, P. (2002). *Sage Monograph on Missing Data (Sage Paper #136).*

Anatova, T., & Sharma, L. (2003). Cognitive function in schizophrenia. *Psychiatric Clinics of North America* , 25-40.

Antonova, E., T. Sharma, M. R., & V., K. (2004). The relationship between brain structure and neurocognition in schizophrenia: a selective review. *Schizophrenia Research* , 117-145.

Ashkar, P., & Kenny, D. (2007). Moral reasoning of adolescent male offenders - Comparison of sexual and nonsexual offenders. *Criminal Justice and Behavior* , 108-118.

Axelrod, R. (1984). *The Evolution of Cooperation.* New York: Basic Books.

Bergvall, A. H., Wessely, H., Fosman, A., & Hansen, S. (2001). A deficit in attentional set-shifting of violent offenders. *Psychological Medicine* , 1095-1105.

Bihop, Y., Fienberg, S., & Holland, P. (1975). *Discrete Multivariate Analysis.* Cambridge: MIT Press.

Bishop, C. (2006). *Pattern Recognition and Machine Learning.* Springer.

Blair, A., & Marsh, R. (2008). Deficits in facial affect recognition among antisocial populations: A meta-analysis. *Neuroscience an Biobehavioral Reviews* , 454-465.

Blair, R. (2001). Neuro-cognitive models of aggression, the antisocial personality disorders and psychopathy. *Journal of Neurology,Neurosurgery & Psychiatry* , 727-731.

Blair, R. (2007). The amygdale and ventromedial prefrontal cortex in morality and psychopathy. *Trends in Cognitive Sciences* , 387-392.

Bolla, K., Brown, K., Eldreth, D., Tate, L., & Cadet, J. (2002). Dose-related neurocognitive effects of marijuana use. *Neurology* , 1337-1343.

Borries, K. v., & Verkes, R. *Automated cognitive assessment in forensic contexts.* Internal report Pompestichting.

Brazil, I., Bruijn, E. d., & Bulten, B. (2009). Early and late components of error monitoring in violent offenders with psychopathy. *Biological Psychiatry* , 137-143.

Chib, S., & Graanberg, G. (1995). Understanding the Metropolis-Hastings algorithm. *American Statistician* , 327-335.

Cohen, J., Cohen, P., West, S., & Aiken, L. (2002). *Regression/Correlation Analysis for the Behavioral Sciences.* Taylor & Francis.

Consi, B., & Webb, T. (2001). *Biorobotics: Methods and Applications.* Cambridge: The MIT Press.

Cooper, G., & Herskovits, E. (1995). A Bayesian method for the induction of probabilitstic networkks from data. *Machine Learning* , 309-347.

Dunning, T. (1993). Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics* , 61-74.

Emmerik, J. (2001). *De terbeschikkingstelling in maat en getal.* Den Haag: Ministerie van Justitie.

Enticott, P. G., Ogloff, R. P., Bradshaw, J. L., & Fitzgerald, P. B. (2008). Cognitive inhibitory control and self-reported impulsivity among violent offenders with schizophrenia. *Journal of Clinical and Experimental Neuropsychology* , 157-162.

Expertisecentrum Forensische Psychiatrie. (sd). Opgeroepen op November 12, 2011, van http://www.efp.nl/nl/project/zorgprogrammas-de-forensische-psychiatrie

Friedman, N., & Koller, D. (2000). Being Bayesian about network structure. *Proc. Sixteenth Conference on Uncertainty in Artificial Intelligence*, (pp. 201-210).

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on Pattern analysis and machine intelligence* , 721-741.

GeNIe & SMILE. (sd). Opgehaald van http://genie.sis.pitt.edu/

Goldberg, T., & Gold, J. (1995). Neurocognitive deficits in schizophrenia. In S. R. Weinberger, *Schizophrenia.* Oxford: Blackwell Science Ltd.

Grant, I., Gonzalez, R., Carey, C., Natarajan, L., & Wolfson, T. (2003). Non-acute (residual) neurocognitive effects of cannabis use: A meta-analytic study. *Journal of the international Neuropsychological Society* , 679-689.

Guba, E., & Lincoln, Y. (1994). Competing paradigms in qualitative research. In *The SAGE Handbook of Qualitative Research* (pp. 105-117). Thousand Oaks, CA: Sage.

Guilmette, T., Faust, D., Hart, K., & Arkes, H. (1990). A national survey of psychologists who offer neuropsychological services. *Archives of Clinical Neuropsychology* , 373-392.

Harmon-Jones, E., Barratt, E., & Wigg, C. (1977). Impulsiveness, Aggression, Reading and the P300 of the event related potential. *Personality and individual differences* , 439-445.

Heckerman, D. (1989). A tractable inference algorithm for diagnosing multiple diseases. *Uncertainty in Artificial Intelligence.*

Heckerman, D. (1998). *A tutorial on learning with Bayesian networks.* Cambridge: MIT Press.

Hiatt, K. S., & Newman, J. (2008). Stroop tasks reveal abnormal selective attention among psychopathic offenders. *Neuropsychology* , 50-59.

Hoaken, P. N., Allaby, D. B., & Earle, J. (2007). Executive cognitive functioning and the recognition of facial expressions of emotion in incatcerated violent offenders, non-violent offenders and controls. *Aggressive Behavior* , 412-421.

Howard, R., & Lumsden, J. (1996). A neurophysiological predictor of reoffending in special hospital patients. *Criminal behavior and Mental Health* , 147-156.

Jazbec, S., Pantelis, C., Robbins, T., Weickert, T., Weinberger, D. R., & Goldberg, T. E. (2007). Intra-dimensional/extra-dimensional set-shifting performance in schizophrenia: Impact of distractors. *Schizophrenia research* , 339-349.

Joval, C. C., Black, D. N., & Dassylva, B. (2007). The Neuropsychology and neurology of sexual deviance: a review and pilot study. *Sex Abuse* , 155-173.

Kalisch, M., & Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Reasearch* , 613-636.

Kalisch, M., Mächler, M., Colombo, D., Maathuis, M., & Bühlmann, P. (2012). Causal Inference Using Graphical Models with the R package pcalg. *Journal of Statistical software* , 1 - 26.

Kennedie, S. (2009). *Performance prediction with cognitive task load and emotional state: preliminary research for manned missions to mars.* Nijmegen: Bachelor Thesis, Radboud University.

Kirsch, L., & Becker, J. V. (2007). Emotional deficits in Psychopathy and sexual sadism: Implications for violent and sadistic behaviour. *Clinical Psychology Review* , 904-922.

Kschischang, F., B., F., & Loeliger, H. (2001). Factor graphs and the sum-product algorithm. *IEEE Trans Info. Theory* .

Levin, M. (1995). Locating putative protein signal sequences using genetic algorithms. *Applications Handbook of Genetic Algorithms* , 53-66.

Little, R., & Rubin, D. (1987). *Statistical analysis with missing data.* New York: John Wiley & Sons.

Lynch, T., Rosenthal, M., Kosson, D., Cheavens, J., Lejuez, C., & Blair, R. (2006). Heightened Sensitivity to Facial Expressions of EMotion in Borderline Personality Disorder. *Emotion* , 647-655.

Marsh, A., & Blair, R. (2008). Deficits in facial affect recognition among antisocial populations: A meta-analysis. *Neuroscience and Biobehavioral Reviews* , 454-456.

Meek, C. (1995). Causal Influence and Causal Explanation with Background Knowledge. *Uncertainty in Artificial Intelligence; Proceedings of the Eleventh Conference* (pp. 1995-200). San Mateo, California: Morgan Kaufmann.

Meek, C. (1995). Strong Completeness and Faithfulness in Bayesian Networks. *Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 411-418). Morgan Kaufmann.

Miller, L. (1987). Neuropsychology of the aggressive psychopath: an integrative review. *Aggressive behavior* , 114-119.

Morris, R., Rushe, T., Woodruffe, P., & Murray, R. (1995). Problem solving in schizophrenia: a specific deficit in planning ability. *Schizophrenia Research* , 235-246.

Mortimer, A. M., JJoyce, E., Balasubramaniam, K., Choudhary, P. C., & Saleem, P. T. (2007). Treatment with amisulpride and olanzepine improve neuropsychological function in schizophrenia. *Human Psychopharmacology* , 445-454.

Murphy, K. (2007, October 19). *Bayes Net Toolbox for Matlab*. Opgehaald van Google code: http://code.google.com/p/bnt/

Murphy, K. (2012, June 14). *Software Packages for Graphical Models/Bayesian Networks*. Opgehaald van http://www.cs.ubc.ca/~murphyk/Software/bnsoft.html

Murphy, K. (2001). *The Bayes Net Toolbox for Matlab.* Berkeley: University of California.

Murphy, K., Weiss, Y., & Jordan, M. (1999). Loopy belief propagation for approximate inference: an empirical study. *Uncertainty in Artificial Intelligence.*

Newman, D. (2003). Longitudinal Modeling with Randomly and Systematically Missing Data: A Simulation of Ad Hoc, Maximum Likelihood, and Multiple Imputation Techniques. *Organizational Research Methods* , 328-362.

Pearl, J. (1988). *Probabilistic Reasoning in intelligent systems: Network of plausible inference.* Morgan Kaufmann.

Pearl, J. (1988). *Reasoning in Intelligent Systems.* San Fransisco: Morgan Kaufmann.

Perlstein, W., Carter, C., Barch, B., & Baird, J. (1998). The Stroop task and attention deficits in schizophrenia: a critical evaluation of card and single-trial Stroop methodologies. *Neuropsychology* , 414-425.

Pham, T., Vanderstukken, O., Philippot, P., & Vanderlinden, M. (2003). Selective attention and executive functions deficits among criminal psychopaths. *Aggressive Behavior* , 393-405.

Pham, T., Vanderstukken, O., Philippot, P., & Vanderlinden, M. (2003). Selective attention and executive functions deficits among criminal psychopaths. *Aggressive Behavior* , 393-405.

Rada, E., Taracena, M., & Rodriguez, M. (2003). Antisocial personality disorder evaluation with the prisoner's dilemma. *Actas Espanolas de Psiquiatria* , 307-314.

Raine, A., Buchsbaum, M., & LaCasse, L. (1997). Brain Abnormalities in murderers indicated by positron emission tomography. *Biological psychiatry* , 495-508.

Rilling, J., Glenn, A., Jairam, M. P., Goldsmith, D., Elfenbein, H., & Lilienfels, S. (2007). Neural Correlates of Social Cooperation and Non_Cooperation as a Function of Psychopathy. *Biological Psychiatry* , 1260-1271.

Royston, P. (2005). Multiple Imputation of Missing Values: Update. *The Stata Journal* , 188-201.

Royston, P. (2005). Multiple imputation of missing values: update of ice. *Stata Journal* , 527-36.

Ruskey, F., & Waston, M. (1997). *A survey of Venn diagrams.*

Russel, S., & Norvig, P. (1995). *Artificial Intelligence: a Modern Approach.* Englewood cliffs, New Jersey: Prentice Hall.

Russell, P., & Norvig, S. (2003). *Artificial Intelligence: A Modern Approach.* New Jersey: Prentice Hall.

Sechrest, L. (1992). Roots: Back to our first generations. *Evaluation practice* , 1-8.

Shea, M. T., Stout, R., Gunderson, J., Morey, L. C., Grilo, C. M., McGlashan, T., et al. (2002). Short-term Diagnostic Stability of Schizotypal, Borderline, Avoidant and Obsessive-Compulsive Personality Disorders. *American Journal of Psychiatry* , 2036-2041.

Smith, P., & Waterman, M. (2003). Processing bias for agression words in forensic and nonforensic samples. *Cognition and Emotion* , 681-701.

Smith, P., & Waterman, M. (2004). Processing bias for secual material: the emotional stroop and sexual offenders. *Sexual Abuse: A journal of Research and Treatment* , 163-171.

Smith, S., Arnett, P., & Newman, J. (1992). Neuropsychological differentiation of psychopathic and nonpsychopathic offenders. *Personality and Individual Differences* , 1233-1243.

Spirtes, P., Glymour, C., & Scheines, R. (2000). *Causation, Prediction, Search.* Cambridge: MIT Press.

Spirtes, P., Glymour, C., & Scheines, R. (2000). *Causation, Prediction, Search.* New York: Springer - Verlag.

Walter, W., Cooper, R., McCallum, W., & Winter, A. (1964). Slow Wave Potential Waves in the Human Brain associated with expectancy,attention and decision. *Archiv für psychiatrie und Zeitschrift für die Gesamte Neurologie* , 309-322.

Appendix A

There are 25 categories of variables, which will each be discussed below. The variables resulting from the questionnaires are item scores and sum scores. The variables resulting from the different tasks are reaction times and variables related to the number of errors made by the subject. It has been mentioned in the introduction that the tasks cover five cognitive fields; emotional processing, learning, impulsivity and attention, moral and social behavior and implicit cognition. For each task the corresponding cognitive field will be mentioned below. All information about tasks and questionnaires comes from (Borries & Verkes). The abbreviations that are mentioned are the abbreviations referred to in the variable names.

A.1 Anamnesis and risk

Anamnesis (ANAM)
This category contains a few types of variables. The first type is personal information such as age, native country and education level. The second type of variables regard the diagnosis of disorders and symptoms. The next type of variables contains offence related information. Finally there are variables indicating substance abuse by the subject.

Risk (RISK)
The variables in this category are related to the risk analysis. These risks include for example how likely it is that a patient will have incidents or risks related to relapse.

A.2 Tasks

Affective Go/No go Task (AFFGO)
Information processing biases for positive and negative stimuli are assessed in this task. Positive, negative and neutral words are used to represent the different categories of stimuli. The participant is given a target category and has to press a button when a word from this category is presented. In (Walter, Cooper, McCallum, & Winter, 1964) and (Howard & Lumsden, 1996) the forensic relevance of this task is shown.

Continuous Performance Task (CPT)
The participant has to spot a specific combination of letters, 'AX' in this case, in a sequence of letters that is presented. This task measures sustained attention and concentration. The results of this task have been shown to be relevant to forensic psychiatry by (Harmon-Jones, Barratt, & Wigg, 1977) and (Raine, Buchsbaum, & LaCasse, 1997).

Graded Facial Emotion Recognition task (GERT)
A series of pictures of faces is shown representing the six basic emotions (happiness, fear, surprise, disgust, anger and sadness). In order to find more subtle differences in the ability to recognize facial emotion different intensities of each emotion are shown. The participant has to indicate which emotion they see as quick as possible. Facial Emotion recognition deficits have been found in

multiple syndroms relevant to forensic psychiatry; (Marsh & Blair, 2008) and (Lynch, Rosenthal, Kosson, Cheavens, Lejuez, & Blair, 2006).

Emotional Stroop Task (EMOS)
Words with an emotional load and neutral words are presented in color. The participants have to indicate the color of the words. When the emotional theme is relevant to the participant and captures attention an interaction effect is expected in reaction time between color and the load of the word. This task measures attentional bias with resprec to a certain emotional theme or problem. Relevance to forensic psychiatry is shown in (Smith & Waterman, 2003) and (Smith & Waterman, 2004).

Faces Task (FACES)
During the task a number of drawings of faces are very shortly shown at the same time in each trial. Each face has a emotion; neutral, happy or angry. In one trial either all faces have the same emotion, or the emotion of one face deviates. The participant has to respond whether or not there was a deviating face in the trial. This task measures perceptual sensitivity and response bias, especially for the distinction in perception of angry and happy faces.

Prisoners Dilemma Game (PDG)
The participant is has to play multiple rounds of the Prisoners Dilemma Game as described in (Axelrod, 1984) against a computer using different strategies in different conditions. Because of the repetition punishment plays a role in this task. Balanced cooperation becomes the most benefitting strategy. Deficits in cooperations have been found in (Rilling, Glenn, Jairam, Goldsmith, Elfenbein, & Lilienfels, 2007) and (Rada, Taracena, & Rodriguez, 2003).

Perceptual Defence Task (PDT)
During this task pairs of pictures are shown shortly to the participant while they are looking at a central point on the screen. Each pair consists of a stressful picture and a neutral picture. The participant has to pick the most eye catching or the most threatening picture depending on the condition.

Stroop color word test (SCWT)
The classical stroop test consists of three tasks; indentification of colornames, identification of the color of the word and indentification of the physical color instead of the semantic color names. This test measures the capability to keep the attention on the color of the word itself instead of the semantic meaning of the colored word. The relevance of the test can be found in (Pham, Vanderstukken, Philippot, & Vanderlinden, 2003), (Hiatt & Newman, 2008) and (Enticott, Ogloff, Bradshaw, & Fitzgerald, 2008).

Signal Detection Task (SDT)
During each trial an array of stimuli is presented shortly and the subject has to decide as fast as possible whether there is an odd stimuli present. The task assesses aspects of basic decision making and can inform about perceptual sensitivities underlying higher order decision making processes.

Social value test (SWT)
The subject divides hypothetical money between a fictive other and oneself. The outcome indicates the extend to which one is lead by individualism or cooperation.

Trail making task TMT
During this task the subject has to connect numbers, letters or both in numerical/alphabetical order. This measures cognitive flexibility, i.e. the ability to shift concepts. The relevance is shown in (Mortimer, JJoyce, Balasubramaniam, Choudhary, & Saleem, 2007) and (Smith, Arnett, & Newman, 1992).

Moral Judgement Sorting Task (MJST)
During this task the participant is presented with a moral dilemma. Next the participant is presented with nine different statements which he or she has to order with regard to how sensible they are. There are three such dilemmas. Relevance of this task is shown in (Blair R. , 2007) and (Ashkar & Kenny, 2007).

Intradimensional/extradimensional set shifting task (IDED)
During this task the subject is confronted with a rule in each block that has to be learned through trial and error. Stimuli can be simple (one dimension, e.g. color) or complex (multiple dimensions). The shift of learning rule are initially intra-dimensional (e.g. color remains the relevant dimension), then later extra-dimensional (the relevant dimension changes). This task indexes response reversal performance. Relevance is shown in (Jazbec, Pantelis, Robbins, Weickert, Weinberger, & Goldberg, 2007) and (Blair R. , 2001).

Casino (CASINO)
During this task the participant needs to predict whether or not the player of a card game will win or lose. The strategy of the dealer is deducible but is reversed once the participant has learned the previous strategy. This task researches stimulus outcome learning, as opposed to most learning tasks which focus on stimulus-response learning. This task measures inhibition of an prepotent response (impulsivity).

Stop Signal Task (STOP)
During this task the subject needs to respond to the direction of the arrow stimuli presented on a screen. If direction of the arrow determines the desired response. However, if around the moment the stimulus is presented a sound is heard, no response should be given to the stimulus.

A.3 Questionnaires

Psychopathic personality inventory (PPI)
This is a self report assessment of psychopathy.

Behavioral inhibition system/ behavioral activation system scale (BB)
It is argued that two general motivational systems underlie behavior. One system is believed to regulate appetitive motives; behavioral activation system. The goal of this system is to move to something desired. The other system regulates aversive motives; behavioral inhibition system. This

system regulates the avoidance of something unpleasant. This scale assesses individual differences in the sensitivity of these systems.

Interpersonal Reactivity Inventory (IRI)
This questionnaire is used to measure empathy. It measures both cognitive and emotional empathy.

Kirby questionnaire (Kirby)
This questionnaire measures the preference for either smaller immediate rewards or a larger delayed reward.

Sensitivity to punishment and sensitivity to reward questionnaire (SPSRQ)
This questionnaire measures the subjects sensitivity to punishment and the sensitivity to reward.

State trait anger expression inventory (ZAV)
This questionnaire is designed to measure anger in a self–report questionnaire. It measures both state anger, the emotional state of anger, and trait anger, which is anger as a stable personality quality.

State trait anxiety inventory (ZBV)
This questionnaire is designed to measure anxiety in a self–report questionnaire. It measures both state anxiety, the emotional state of anxiety, and trait anxiety, which is anxiety as a stable personality quality.

Social Dysfunction and Aggression Scale (SDAS)
An observer scale for measuring aggressive cognitions and behavior. It consists of nine items: irritability, dysphoric mood, social disturbances, non-directed verbal aggressiveness, negativism, directed verbal aggressiveness, physical violence towards staff, physical violence towards things and physical violence towards persons other than staff.

Appendix B

This appendix contains an overview of which variables are included in the datasets of 30 variables (table 1) and 132 variables (table 2).

Table 1. Variables included in the variable set of 30 variables. The original index refers to the variable index in the original dataset of 1384 variables and the current index to the variable index in this set.

Original index	Current index	Variable Name
1	1	ANAM_Groep
30	2	ANAM_DSM_huidig_AS1_groep1
31	3	ANAM_DSM_huidig_AS1_groep2
32	4	ANAM_DSM_huidig_AS1_groep3
33	5	ANAM_DSM_huidig_AS1_groep4
34	6	ANAM_DSM_huidig_AS1_groep5
35	7	ANAM_DSM_huidig_AS1_groep6
36	8	ANAM_DSM_huidig_AS2_groep1
37	9	ANAM_DSM_huidig_AS2_groep2
38	10	ANAM_DSM_huidig_AS2_groep3
39	11	ANAM_DSM_huidig_AS2_groep4
40	12	ANAM_Huidig_NAO_trekken_clusterA
41	13	ANAM_Huidig_NAO_trekken_clusterB
42	14	ANAM_Huidig_NAO_trekken_clusterC
43	15	ANAM_Huidig_NAO_trekken_overig
44	16	ANAM_Huidig_Group_clusterB1
45	17	ANAM_Huidig_Group_clusterB2
46	18	ANAM_Huidig_Group_clusterB3
49	19	ANAM_PCL_totaal
59	20	ANAM_Groep_indexdelict
74	21	ANAM_Geweldcomponent
75	22	ANAM_Seksuele_component
76	23	ANAM_Vermogenscomponent
78	24	ANAM_Group_eerdere_delicten
160	25	PPI_total
161	26	PPI_F1_benning2003
162	27	PPI_F2_benning2003
243	28	AFFGO_PERC_number_RT_falsenegative
244	29	AFFGO_PERC_number_RT_falsepositive
249	30	AFFGO_PERC_number_RT_NegPos_falsenegative

Table 2. Variables included in the variable set of 132 variables.The original index refers to the variable index in the original dataset of 1384 variables and the current index to the variable index in this set.

Original index	Current index	Variable name
1	1	ANAM_Groep
30	2	ANAM_DSM_huidig_AS1_groep1
31	3	ANAM_DSM_huidig_AS1_groep2
32	4	ANAM_DSM_huidig_AS1_groep3
33	5	ANAM_DSM_huidig_AS1_groep4
34	6	ANAM_DSM_huidig_AS1_groep5
35	7	ANAM_DSM_huidig_AS1_groep6
36	8	ANAM_DSM_huidig_AS2_groep1
37	9	ANAM_DSM_huidig_AS2_groep2
38	10	ANAM_DSM_huidig_AS2_groep3
39	11	ANAM_DSM_huidig_AS2_groep4
40	12	ANAM_Huidig_NAO_trekken_clusterA
41	13	ANAM_Huidig_NAO_trekken_clusterB
42	14	ANAM_Huidig_NAO_trekken_clusterC
43	15	ANAM_Huidig_NAO_trekken_overig
44	16	ANAM_Huidig_Group_clusterB1
45	17	ANAM_Huidig_Group_clusterB2
46	18	ANAM_Huidig_Group_clusterB3
49	19	ANAM_PCL_totaal
59	20	ANAM_Groep_indexdelict
74	21	ANAM_Geweldcomponent
75	22	ANAM_Seksuele_component
76	23	ANAM_Vermogenscomponent
78	24	ANAM_Group_eerdere_delicten
160	25	PPI_total
161	26	PPI_F1_benning2003
162	27	PPI_F2_benning2003
243	28	AFFGO_PERC_number_RT_falsenegative
244	29	AFFGO_PERC_number_RT_falsepositive
249	30	AFFGO_PERC_number_RT_NegPos_falsenegative
250	31	AFFGO_PERC_number_RT_NegPos_falsepositive
255	32	AFFGO_PERC_number_RT_NeuPos_falsenegative
256	33	AFFGO_PERC_number_RT_NeuPos_falsepositive
261	34	AFFGO_PERC_number_RT_PosNeg_falsenegative
262	35	AFFGO_PERC_number_RT_PosNeg_falsepositive
267	36	AFFGO_PERC_number_RT_PosNeu_falsenegative
268	37	AFFGO_PERC_number_RT_PosNeu_falsepositive
273	38	AFFGO_PERC_number_RT_NeuNeg_falsenegative
274	39	AFFGO_PERC_number_RT_NeuNeg_falsepositive
279	40	AFFGO_PERC_number_RT_NegNeu_falsenegative

Original index	Current index	Variable name
280	41	AFFGO_PERC_number_RT_NegNeu_falsepositive
284	42	BB_BIS
288	43	BB_BAS
290	44	CPT_gemiddelde_RT_totaal
294	45	CPT_PERC_number_GO_incorrect
375	46	FACES_hit_rate_totaal
376	47	FACES_false_alarm_rate_totaal
377	48	FACES_dprime_totaal
378	49	FACES_beta_ratio_totaal
380	50	FACES_false_alarm_rate_BlijNeu
384	51	FACES_false_alarm_rate_BoosNeu
388	52	FACES_false_alarm_rate_NeuBoos
392	53	FACES_false_alarm_rate_NeuBlij
405	54	EMOS_agressie_2bl_gemiddelde_RT_totaal
410	55	EMOS_agressie_2bl_PERC_number_incorrect
414	56	EMOS_agressie_2bl_STROOPeffect_correct
415	57	EMOS_agressie_2bl_STROOPeffect
451	58	EMOS_angst_2bl_gemiddelde_RT_totaal
456	59	EMOS_angst_2bl_PERC_number_incorrect
460	60	EMOS_angst_2bl_STROOPeffect_correct
461	61	EMOS_angst_2bl_STROOPeffect
487	62	GERT_RT_totaal
507	63	GERT_perc_incorrect_angstig70
527	64	GERT_perc_incorrect_bedroefd70
547	65	GERT_perc_incorrect_boos70
562	66	GERT_perc_incorrect_i70
607	67	GERT_perc_incorrect_neutraal
627	68	GERT_perc_incorrect_verrast70
647	69	GERT_perc_incorrect_vrolijk70
707	70	GERT_perc_incorrect_walging70
708	71	IRI_Total_Fantasy
709	72	IRI_Total_Perspective_taking
710	73	IRI_Total_Empathic_concern
711	74	IRI_Total_Personal_distress
712	75	IRI_Total
713	76	Kirby_maximum_k_LARGE
714	77	Kirby_maximum_k_MEDIUM
715	78	Kirby_maximum_k_SMALL
724	79	PDG_Cooperatiepunten
725	80	PDG_Competitiepunten

Original index	Current index	Variable name
726	81	PDT_RT_stress
727	82	PDT_RT_neutraal
728	83	PDT_RT_totaal
729	84	PDT_PD
730	85	PDT_RL
733	86	SCWT_RT_totaal_gemiddeld
746	87	SCWT_PERC_number_incorrect
753	88	SCWT_STROOP_effect
755	89	SDT_RT_totaal_gemiddeld
780	90	SDT_PERC_number_false_alarm
781	91	SDT_PERC_number_miss
794	92	SDT_hit_rate_totaal
795	93	SDT_false_alarm_rate_totaal
796	94	SDT_dprime_totaal
797	95	SDT_beta_ratio_totaal
806	96	SPSRQ_Total_punishment
807	97	SPSRQ_Total_reward
829	98	STOP_RT_total
832	99	STOP_PERC_number_Blok14_RT_falsenegative
833	100	STOP_PERC_number_Blok14_RT_falsepositive
838	101	STOP_PERC_number_Blok23_RT_falsenegative
839	102	STOP_PERC_number_Blok23_RT_falsepositive
854	103	SWT_winstZelf
855	104	SWT_winstAnder
856	105	SWT_gemiddelde_RT
860	106	TMT_totaal_aantal_fouten
874	107	TMT_Inferentiescore_correct_gemiddeld
875	108	TMT_Inferentiescore_incorrect_gemiddeld
876	109	ZAV_total_state
877	110	ZAV_total_trait
878	111	ZBV_total_state
879	112	ZBV_total_trait
909	113	MJST_MEAN_RT
913	114	MJST_MEAN_CORR_D123
918	115	IDED_RT_SD_totaal_zonder_eerste
932	116	IDED_aantal_SD_incorrect_tot_correct
952	117	IDED_aantal_SR_incorrect_tot_correct
972	118	IDED_aantal_CD_incorrect_tot_correct
992	119	IDED_aantal_CDS_incorrect_tot_correct
1011	120	IDED_aantal_CR_incorrect_tot_correct

Original index	Current index	Variable name
1031	121	IDED_aantal_IDS_incorrect_tot_correct
1051	122	IDED_aantal_IDR_incorrect_tot_correct
1073	123	IDED_aantal_EDS_incorrect_tot_correct
1094	124	IDED_aantal_EDR_incorrect_tot_correct
1151	125	CASINO_RT_totaal__totaal
1154	126	CASINO_RT_reward__totaal
1157	127	CASINO_RT_punishment__totaal
1270	128	CASINO_PERC_number_totaal__incorrect
1272	129	CASINO_PERC_number_reward__incorrect
1281	130	CASINO_PERC_number_nsp__incorrect
1289	131	CASINO_PERC_number_SWI__incorrect
1298	132	CASINO_PERC_number_REV__incorrect

Appendix C

This appendix contains the pseudocode for imputation and discretization for clarity of the exact procedure. For the conditional independence test the actual MATLAB code is provided.

C.1 Discretization

$[V, S]$ is the dataset with V the collection of numerical variables and S the collection of subjects, such that x_{ij} is the value of variable v_i and subject s_j.
$[V^{New}, S^{new}]$ is the new dataset with x_{ij}^{new} as the new discrete value of variable v_i and subject s_j.

N is the number of bins.
B is the set of $N + 1$ boundaries where b_i is the lower boundary of bin i, and b_{i+1} the upper boundary.
C is the set of categorical values with a value for each bin, where c_k corresponds to the k^{th} bin.

For each $v_i \epsilon V$
  Determine minimum, min_i , and maximum, max_i , of v_i
  $difference_i = max_i - min_i$
  $width_i = difference_i / N$
  $b_1 = -\infty$
  For each $b_k \epsilon B$ $with$ $k = 2 \dots N$
    $b_k = min_i + (i \cdot width_i)$
  End
  $b_{N+1} = \infty$
  For each $x_{ij} \epsilon [v_i, S]$
    Determine for which bin $b_k < x_{ij} \leq b_{k+1}$ and fill x_{ij}^{new} with c_k
  End
end

C.2 Imputation

$[V, S]$ is the dataset with V the collection of numerical variables and S the collection of subjects, such that x_{ij} is the value of variable v_i and subject s_j.
N_S is the number of subjects.

N is the number of bins
B is the set of $N + 1$ boundaries where b_i is the lower boundary of bin i, and b_{i+1} the upper boundary.
C is the set of values with a value for each bin, where c_k corresponds to the k^{th} bin.

F is the set of frequencies with the frequency of each bin, where f_l corresponds to the l^{th} bin.
D is the distribution for the bins of the variable and d_m is the cumulative probability of value c_m.

For each $v_i \epsilon V$
  Determine B as described in D*iscretization*
  For each $c_k \epsilon$C
    c_k is the average of b_k and b_{k+1}, except for c_1 and c_N which are respectively $b_{k+1} - bin\ width/2$ and $b_k + bin\ width/2$
  End
  For each $x_{ij} \epsilon [v_i, S]$
    Determine for which bin $b_k < x_{ij} \leq b_{k+1}$ and raise f_k by one
  End
  $d_1 = 0$
  For each $d_m \epsilon D\ with\ m = 2 \ldots N$
    $d_m = d_{m-1} + f_m / N_S$
  End
  $d_{N+1} = 1$
  For each $x_{ij} \epsilon [v_i, S]$ which is missing
    Draw a random probability r between 0 and 1
    Determine for which bin m, $d_m < r \leq d_{m+1}$ and insert c_m in x_{ij}
  End
end

This is the implementation for numerical variables. For categorical variables the imputation is done the same way except for the fact that it is not necessary to discretise and use the average value of each bin for the distribution, instead the distribution for the original categorical values is determined and used.

C.3 Conditional independence test

```
function [ independent ] = ConditionalIndependenceTest
( x, y, Z, alpha, data )
% x & y are variables you wish to test for independence
% Z conditional variables
% Data contains the data with subjects in rows and variables in columns
% Returns boolean for conditional independence

independent = false;
varx = data(:,x);
vary = data(:,y);
varZ = data(:,Z);
total = [varx,vary,varZ];

%Maximum of variable is the number of values for
%that variable (forced when discretized)
maximaX = max(varx);
maximaY = max(vary);
maximaZ = max(varZ);
%initialize tables for observed and expected values
observed = zeros([maximaX, maximaY, maximaZ]);
expected = zeros([maximaX, maximaY, maximaZ]);

%Fill table with observed value frequencies
for sub = 1:size(total,1)
    observed(calcLinearIndex(size(observed),total(sub,:))) =
observed(calcLinearIndex(size(observed),total(sub,:)))+1;
end

%Fill table with expected frequenties
nrCells = numel(observed);
nrEmptyEntries = 0;
%For each cell
for linIndex = 1:nrCells
    %If the cell has a zero entry
    if observed(linIndex) == 0
        nrEmptyEntries = nrEmptyEntries+1;
    end
    %Calculate rowtotal, column total and table total
    subscripts = calcSubscripts(size(observed), linIndex);
    rowTotal = calcTableTotal(observed, linIndex, subscripts, 1, maximaX);
    columnTotal = calcTableTotal(observed, linIndex, subscripts, 2,
maximaY);
    generalTotal = calcTableTotal(observed, linIndex, subscripts, 3,
[maximaX,maximaY]);
    %Calculate expected value
    expected(calcLinearIndex(size(expected), subscripts)) =
(rowTotal*columnTotal)/generalTotal;
end

%Calculate G2 value
G2mid = 0;
%For each cell
for linIndex = 1:nrCells
    %If the cell is not a zero entry
    if observed(linIndex)~=0
        %Add G2 value for cell to total
        G2mid = G2mid + observed(linIndex) *
log(observed(linIndex)/expected(linIndex));
```

```
    end
end
G2 = G2mid*2;
%Calculate degrees of freedom
df = calcDegreesFreedom(maximaX,maximaY,maximaZ, nrEmptyEntries);
%If the degrees of freedom <= 0
if df <= 0
    %X and Y are dependent
    independent = false;
else
    %calculate p-value with function from BNT
    %This function returns P(<=G2|df)
    p = chisquared_prob(G2, df);
    %If P(>G2|df) > alpha
    if (1-p)>alpha
        %X and Y are independent
        independent = true;
    end
end


function [ df ] = calcDegreesFreedom
(nrValuesX,nrValuesY,nrValuesZ,nrEmptyEntries )
%Function calculates the degrees of freedom based on the number of
%values for X, Y and conditional variables and the number of empty entries
if size(nrValuesZ,2)~=0
    df = (nrValuesX - 1) * (nrValuesY-1) * prod(nrValuesZ) -
nrEmptyEntries;
else
    df = (nrValuesX - 1) * (nrValuesY-1) - nrEmptyEntries;

end
```