LMU
LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

DEPARTMENT OF STATISTICS

KATRIN NEWGER

STATISTICAL MATCHING OF CATEGORICAL DATA WITH MARKOV NETWORKS

MASTER'S THESIS

AUGUST 2018

Statistical Matching of Categorical Data with Markov Networks

Ludwig-Maximilians-Universität München
Department of Statistics

Thesis for the Degree Master of Science (M.Sc.)

Statistical Matching of Categorical Data with Markov Networks

Katrin Newger

Simeonistr. 10
80637 München
Matr. 10580354

Supervisor: Eva Endres

Examiner: Prof. Dr. Thomas Augustin

Submission date: August 31, 2018

List of Abbreviations

BIC	Bayesian information criterion
BN	Bayesian network
CI	conditional independence
CIA	conditional independence assumption
CPD	conditional probability distribution
DAG	directed acyclic graph
D-map	dependence map
d-separation	directed separation
EBIC	extended Bayesian information criterion
i.i.d.	independent and identically distributed
I-map	independence map
MAR	missing at random
MCAR	missing completely at random
MLE	maximum likelihood estimation
MN	Markov network
MNAR	missing not at random
NA	not available
ols	ordinary least squares

Notation Index

Unless indicated otherwise, the notation used throughout this thesis will be as follows. Random variables will be written as upper-case letters. Realizations of random variables will be notated in lower-case. Bold-case letters indicate matrices or vectors, while scalars will be non-bold. Nodes of a graph are notated as upper-case letters, too, as they represent random variables in graphs.

A	a dataset containing $(\mathbf{X}, \mathbf{Y})$
B	a dataset containing $(\mathbf{X}, \mathbf{Z})$
$\hat{\mathsf{A}}$	dataset A plus estimated observations for $\mathbf{Z}$
$\hat{\mathsf{B}}$	dataset B plus estimated observations for $\mathbf{Y}$
$\mathsf{A} \uplus \mathsf{B}$	union of dataset A and B
$\hat{\mathsf{A}} \uplus \hat{\mathsf{B}}$	synthetic data file of estimated datasets $\hat{\mathsf{A}}$ and $\hat{\mathsf{B}}$
n_{A}	number of observations of dataset A
n_{B}	number of observations of dataset B

$\mathbf{X} = (X_1, \ldots, X_p)'$	block of common variables in both A and B
$\mathbf{Y} = (Y_1, \ldots, Y_q)'$	block of specific variables in A
$\mathbf{Z} = (Z_1, \ldots, Z_r)'$	block of specific variables in B
$i = 1, \ldots, p$	index for X
$j = 1, \ldots, q$	index for Y
$k = 1, \ldots, r$	index for Z

$(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$	true and complete data
$P_{\mathbf{XYZ}}$	joint probability distribution of $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$
$\hat{P}_{\mathbf{XYZ}}$	estimated joint probability distribution of $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$
$P(X = x)$	probability that random variable X is observed to be x
$P(Y = y \mid X = x)$	probability of $Y = y$ given $X = x$
$P(x), P(x, y)$	shorthand for $P(X = x)$, $P(X = x, Y = y)$
$P \models (Y \perp Z \mid X)$	Y is conditionally independent of Z given a variable X in P
$(\Omega, \mathcal{F}, P)$	probability space

$L(\cdot)$	likelihood
$\ell(\cdot)$	loglikelihood
I	number of categories for X for maximum likelihood estimation
J	number of categories for Y for maximum likelihood estimation
K	number of categories for Z for maximum likelihood estimation
θ	unknown parameter of a probability distribution

$\mathcal{G}$	directed graph
$\mathcal{H}$	undirected graph
$X \rightarrow Y$	a connection from X to Y in a directed graph
X–Y	a connection between X and Y in an undirected graph
$\mathcal{V}$	set of nodes of a directed or undirected graph
$\mathcal{E}$	set of vertices of a directed or undirected graph
$\mathrm{Pa}(V)$	set of parents of node V
$\mathrm{De}(V)$	set descendants of node V
$\mathrm{Nd}(V)$	set of non-descendants of node V
C	clique in $\mathcal{H}$
$Y \perp_{\mathcal{H}} Z \mid X$	X separates Y and Z in $\mathcal{H}$
$Y \perp_{P} Z \mid X$	Y and Z are independent given X in a probability distribution P
$I_l(\mathcal{H})$	set of local independences in $\mathcal{H}$
$I(\mathcal{H})$	set of global independences in $\mathcal{H}$
$I(\mathcal{G})$	independences in $\mathcal{G}$
$\mathrm{sep}_{\mathcal{H}}(X;Y \mid Z)$	Z separates X and Y in $\mathcal{H}$
$Val(X)$	possible values of X
ϕ	factor in a Markov network
Φ	set of factors
P_{ϕ}	Gibbs distribution parameterized by Φ
$\tilde{P}$	unnormalized measure in a Markov network
ζ	partition function
$\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}$	estimated Markov network based on the union of A and B
$\hat{\mathcal{G}}^{\mathsf{A} \Cup \mathsf{B}}$	estimated Bayesian network based on the union of A and B
$\hat{\mathcal{E}}^{\mathsf{A} \Cup \mathsf{B}}$	estimated edges of a graph based on the union of A and B

ϵ	energy function in a Markov network
$H(x)$	Hamiltonian function
$X \sim Y$	two nodes X and Y are adjacent.
β	adjacent weight in an Ising model
$\mathbf{X}_{-i}$	a vector $\mathbf{X}$ without the i-th element of the original vector $\mathbf{X}$
τ	threshold in an Ising model
η	predictor in an Ising model
λ	penalty parameter for lasso estimation
γ	hyperparameter for the extended Bayesian information criterion

$\mathbf{R}$	correlation matrix
$\mathbf{\Sigma}$	covariance matrix
diag()	elements of the diagonal of a matrix

Contents

List of Figures

List of Tables

Statistical Matching of Categorical Data with Markov Networks

Katrin Newger

Abstract The aim of this thesis is to research how Markov networks can be utilized for statistical matching with categorical data. For this goal, I summarize the theory of statistical matching, and the theory of probabilistic graphical models with a focus on Markov networks. The representation of a joint probability distribution in Markov networks is restated for the aims of statistical matching, thus this thesis offers an equation with which the joint probability distribution of two disjoint datasets can be estimated. Furthermore the theory of the Ising model is explained in detail, as the Ising model is later on used for structure estimation. Since the thesis takes up the research of applying Bayesian networks for statistical matching, I will recap the theory of Bayesian networks, and compare the use of Bayesian networks and Markov networks for statistical matching.

In the practical part of this thesis, 100 disjoint datasets A and B are simulated for applying Markov networks for statistical matching. Both the macro and the micro approach of statistical matching are pursued. I will use and discuss several packages of the statistical programming software R for applying Markov networks. In a final step the results are evaluated.

The results of applying Markov Networks for statistical matching shown in this thesis are promising. Besides discussing limitations, I will ultimately give an outlook for future research in this area.

1 Introduction

Statistical matching is a relevant problem of today's data analysis. The goal of statistical matching is either to aggregate at least two independent datasets A and B, or to estimate the joint probability distribution that generates the observed data. Both datasets have the same population of interest, however, some variables are observed only in A and some other variables only in B. Another thing A and B have in common is a block of variables observed in both A and B.

Several methods are available for this aim, which differ in their assumptions. One assumption that enables statistical matching is that of conditional independence. With conditional independence the problem of matching two datasets becomes solvable.

A theory that decodes joint probabilities by using the independence structure of random variables is probabilistic graphical models. It seems likely that probabilistic graphical models can be a good opportunity for statistical matching. Probabilistic graphical models have a wide theory and offer great options in their application. So why not use them for the aims of statistical matching? A certain variation of probabilistic graphical models are Bayesian networks, which assume a direction of influence. The use of Bayesian networks for statistical matching is being researched and applied at the LMU Munich Department of Statistics.

In this thesis I want to take up what Endres and Augustin (2016) developed for Bayesian networks, and how they can be used for statistical matching. Instead of using Bayesian networks, this thesis will deal with Markov networks, a further variation of probabilistic graphical models. The question of interest is how Markov networks can be utilized for the aims of statistical matching.

The proceedings of this thesis will be as follows: In Chapter 2 I will first explain why statistical matching can be useful, and give a brief summary of the history of statistical matching. Furthermore I will explain the theory of statistical matching and the initial data situation the researcher is confronted with. Beside the assumption of conditional independence, there are several other options for addressing the problem of statistical matching. At the end of this chapter I will turn to categorical data in statistical matching, which are the basis for this thesis.

In Chapter 3 I will explain the theory of probabilistic graphical models. Starting with the basics of graph theory, I will move on to two main variations of probabilistic graphical models already mentioned, i.e. Bayesian networks and Markov networks.

Chapter 4 contains the theory of how probabilistic graphical models can be applied for statistical matching of datasets. At the beginning I will summarize the work of Endres and Augustin (2016) and explain how to utilize Bayesian networks for statistical matching. This is followed by a central part of this thesis: the theory of how Markov networks can be used for statistical matching.

The idea of this thesis is to match categorical data, more specifically binary data. Therefore Chapter 5 deals with the Ising model. The Ising model enables to estimate the structure of a Markov network for binary data. I will also go into more depth about estimating parameters in the Ising model, since I also apply the theory of Markov networks.

Chapter 6 summarizes the procedure of applying Markov networks for statistical matching. As an application I work with simulated binary data that fulfills a certain correlation structure. I will explain how the simulation of the data works, and what to take care for. Next I will explain how to estimate the structure of a Markov network for binary data, and identify adequate software for this aim. Then either the joint probability is estimated (macro approach), or the missing data values for the specific blocks of variables are estimated to obtain a synthetic data file $\hat{\mathsf{A}} \uplus \hat{\mathsf{B}}$ (micro approach). This chapter ends with an evaluation of the results of the macro and micro approach.

Chapter 7 gives a short comparison of Bayesian networks and Markov networks for statistical matching. I will point out how one variation of probabilistic graphical models can be superior, and list the main differences between them when used for statistical matching.

Chapter 8 will sum this thesis up with the results gained in its course. Furthermore I will give a critical discussion of the limitations of Markov networks when used for statistical matching. Beside these limitations, there are still a lot of possibilities Markov networks offer. Thus finally I will give an outlook for further research in this field.

2 Statistical Matching

As the title of this thesis reveals, I will deal with the idea of matching two datasets. But what does ‘matching data’ even mean? Suppose we have two different datasets, created from different units, meaning that a person who was interviewed for the first dataset is unlikely to be interviewed for the second dataset as well. But we do know that the population of interest for both surveys is the same. In this situation it could be interesting to merge the two datasets into one big dataset, in order to gain more profound insights about our target group. Two different approaches of matching are possible: *(1)* exact matching, nowadays often referred to as linking the microdata for identical units. Identical units does not mean that the same person is interviewed, but rather that two persons have identical answers for a block of common variables in our original data. And *(2)*, synthetic or stochastic linking of our original data. That means estimating an overall probability for all variables of the two original datasets, or estimating values to create a new synthetic data file. This approach is often called statistical matching (Okner 1974, pp. 347).

Regardless of the two approaches, the goal of merging data is to get more detailed information of the population of interest, or to fill gaps in the existing data. Furthermore it is obvious that merging data goes along with some problems, regardless of the exact procedure used. First of all, we have to deal with a lot of missing data: while a block of variables is included in both samples, another block of variables is observed in the first sample only, and the last sample contains a block of – again – different variables. Thus the missing data appears blockwise in the two datasets (D’Orazio et al. 2006, p. 1). Figure 1 gives a first impression of the situation.

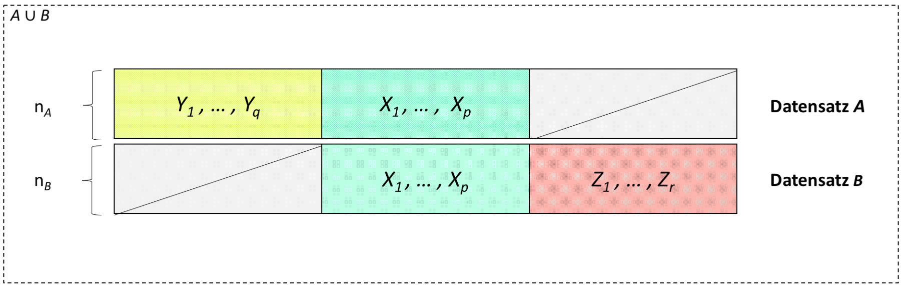

Figure 1: Initial situation of datasets in statistical matching: highlighted in green are the common variables $\mathbf{X}$. The variables $\mathbf{Y}$ are just observed in dataset A (yellow), and the variables $\mathbf{Z}$ are only part of dataset B (red). Depicted from D’Orazio et al. (2006, p. 5).

Furthermore we have to ask the question to what extent our data is matchable at all. Is the block of variables that is included in both datasets comparable in terms of the population definition, or e.g. income concepts? Another problem is the time period which lies between the data collection. If the time gap between the two surveys is too big, perhaps the population changed too much, or something 'game-changing' happened, like a tax reform. A last important problem is the evaluation of the matched data. This is a difficult one, starting with the definition of "good". Is it important to have a *good* match for each unit, or for the mean, the variance, etc.? Or can we concentrate only on some crucial variables? Obviously we cannot compare our matched results to some already existing data, since matching data would then be nonsensical (Okner 1974, pp. 347ff.). The quality of the matched data has been evoking a critical discussion since the beginning of statistical matching in 1972, e.g. by Sims (1972). A very extensive scheme for evaluating the results of statistical matching can be found in Raessler (2002), who evaluates the quality on four levels. The highest quality that can be achieved is to estimate the true missing values right, followed by estimating the true joint probability distribution right. The third level of quality refers to the correlation structure in the data, which is at best obtained by statistical matching, and the fourth level evaluates if the marginal distributions of the variables are estimated right (Raessler 2002, pp. 29ff.). Nonetheless, despite evaluation being complicated, matching data is a promising field.

This thesis will deal with the second approach of matching: statistical matching. It had its beginnings in the mid-1960s. In 1964, Budd et al. (1973) constructed a synthetic data file in order to gain insights about the distribution of income in the U.S. (ibid, pp. 657ff.). Furthermore Okner (1972) produced a synthetic data file with matching, which contained information about socio-demographic variables, but also variables of income and tax return of families. To get this comprehensive synthetic dataset from two independently collected datasets, i.e. the Tax File of 1966 and the Survey of Economic Opportunities of 1967, statistical matching was performed. Ever since, the attention of statistical matching continues (D'Orazio et al. 2006, preface).

In this chapter I want to give a first understanding of statistical matching, including an explanation of the initial situation of data and variables. Moreover I will formulate the two different goals of statistical matching and will introduce the central assumption, which allows to formulate a model that is solvable for the given data. I will also introduce how maximum likelihood estimation can be used in general for the aims of statistical matching.

2.1 Why Statistical Matching?

Confronted with statistical matching, the question arises, why match data at all? In the 1960s – the time statistical matching was applied for the first time – the attention for statistical matching was quite small. A possible reason for this small attention could be the fact that some decades ago the choice of datasets was not that huge. For answering a special research topic, an extra survey was conducted. But times change, and the term *Big Data* can be seen and read everywhere. The access to data is huge, and even bigger are the opportunities to collect data, e.g. over the internet or by tracking mobile data. With the growing amount of data available, the belief to "find something in the data" has also grown, and this in turn entails that more data is gathered. Researchers, governments and also companies record a lot of data. Gathering data became something like a byproduct, characterized by volume, velocity, and variety, often referred to as the three Vs[1] (de Waal 2015, p. 4). But still it remains a problem to get information about variables that are at first sight not connected to each other. E.g. Facebook knows a lot of personal things about people, like hobbies, preferred music, etc., but it is hard for Facebook to evaluate purchase behavior. Thus even Big Data companies suffer from the problem of not having all variables of interest in one dataset. Nevertheless we should use the available data today, and there are techniques which try to bring them in context so that the data has even more potential: case in point, statistical matching. The interest in data and the belief that data of good quality and extensive amount leads to improvements in several fields, could be a reason why the attention for statistical matching grows, even today (de Waal 2015, pp. 4ff.).

But not just the fact that access to data is easier today speaks for statistical matching. It is also the possibility to lower the number of questions in a questionnaire, if the researcher knows that there already is a dataset including relevant questions for this target group. This leads to two advantages: first of of all, participants of a study are likely to give more trustful answers, and their willingness to finish the study is higher for small questionnaires. Secondly the costs are lower if the study has no need to be too extensive (van der Putten et al. 2002, pp. 2ff.).

As we can see, there are several reasons for using statistical matching. In the following, we will go into more detail about this idea.

2.2 Starting Position of Statistical Matching

After a short summary of statistical matching and why to use it, I will focus in the following on the general theory of statistical matching. As the title of this thesis reveals, I will concentrate on categorical data, but of course statistical matching is also possible for continuous variables. A lot of literature is available for this context,

[1] The term 'volume' refers to the tremendous size of the datasets, which is usually too big for a normal computer to handle. The term 'velocity' captures the fact that new data is produced faster and faster, and that it is available for further research nearly right after collecting. The term 'variety' corresponds to the fact that in nearly all areas data is collected, e.g. machinery data, network data, data generated by search engines, etc. (de Waal 2015, p. 4).

see e.g. Raessler (2002) and D'Orazio et al. (2006).

In the following I will notate random variables as upper-case letters, e.g. X represents a random variable. Realizations of random variables will be notated in lower-case, e.g. x is a realization of the random variable X. Bold-case letters indicate matrices or vectors, while scalars will be non-bold.

Data matching problems start with two independent sample surveys,[2] let us call them A and B. Usually these datasets are produced entirely independently of each other, with the aim to examine different questions. The observations in both datasets are independent and identically distributed (i.i.d.). Nevertheless they have something in common:

1. A set of variables $\mathbf{X}$ is included in both samples A and B.

2. The target group of both samples is the same. Consequently the observed data in both datasets was generated by the same joint probability distribution $P_{\mathbf{XYZ}}$.

These two commonalities are the basis of statistical matching and have further consequences for the statistical setting of our problem. Thus we have two datasets $\mathsf{A} \in \mathbb{R}^{n_\mathsf{A} \times (p+q)}$ and $\mathsf{B} \in \mathbb{R}^{n_\mathsf{B} \times (p+r)}$, of which A includes a set of realizations of variable $\mathbf{Y} = (Y_1, \ldots, Y_q)'$ and a set of realizations of variable $\mathbf{X} = (X_1, \ldots, X_p)'$; and B includes a set of realizations of the variable $\mathbf{Z} = (Z_1, \ldots, Z_r)'$, and also $\mathbf{X} = (X_1, \ldots, X_p)'$ like in A. In our case all variables are categorical. Since we assume that all variables come from the same population, the random variable $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ is generated by the probability $P_{\mathbf{XYZ}}$. We now assume that all observations in A and B are generated by the joint probability distribution $P_{\mathbf{XYZ}}$, but remember that $\mathbf{Z}$ is missing in A, and $\mathbf{Y}$ is missing in B. Figure 1 above illustrates the initial situation of the datasets.

The aim of statistical matching is now to estimate the joint probability distribution $\hat{P}_{\mathbf{XYZ}}$, even though joint information of all three variable blocks $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ does not exist. This framework leads to a dual problem: The fact that $\mathbf{Z}$ is missing in A and $\mathbf{Y}$ is missing in B can be seen as a missing data problem, and further we cannot revert to any joint information about $P(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{Z} = \mathbf{z})$, thus there is no obvious solution to finding the joint probability distribution $P_{\mathbf{XYZ}}$ (D'Orazio et al. 2006, pp. 3ff.). D'Orazio et al. (2006) explains that we can assume a 'missing completely at random' mechanism (MCAR) for the missing realizations $\mathbf{y}$ and $\mathbf{z}$, meaning that both the observed and the unobserved units are independent from the

[2] It is also possible to match data of more than two datasets, but for reasons of simplicity I will formulate the theory for matching two datasets. The procedure for several datasets is entirely analogous (D'Orazio et al. 2006, p. 2).

missing entries. In Appendix A I will give a short explanation why MCAR can be assumed.[3]

In statistical matching we can distinguish between two different approaches, although they are not disjoint (D'Orazio et al. 2006, pp. 2ff.):

- **Micro approach**: The aim of this approach is to construct a complete synthetic data file $\hat{\mathsf{A}} \uplus \hat{\mathsf{B}}$, in which the missing observations of $\mathbf{Y}, \mathbf{Z}$ are estimated – with uncertainty. This fact explains the term *synthetic*, since parts of $\mathbf{Y}, \mathbf{Z}$ are not observed directly in the survey, but are synthetically added by estimation.

- **Macro approach**: Here the aim is to estimate the joint probability $P_{\mathbf{XYZ}}$, but only with our datasets A and B, thus without having joint information.

To tackle the problems which come along with statistical matching, D'Orazio et al. (2006) gives three ideas for general solutions:

- Assumption of conditional independence: $\mathbf{Y}$ and $\mathbf{Z}$ are independent given $\mathbf{X}$. Under this assumption an estimation model for the joint probability $P_{\mathbf{XYZ}}$ is identifiable, even though we have to deal with missing data. By using the so-called chain rule we can estimate the joint probability (ibid., p. 13). Chapter 2.4 will deal with this assumption in more detail.

- Use of auxiliary information: As D'Orazio et al. (2006) points out, conditional independence of $\mathbf{Y}$ and $\mathbf{Z}$ given $\mathbf{X}$ is vague and doubtable. The use of auxiliary information – contained in a further dataset (besides A and B) – can solve the problem and lead to parameter estimates that do not suffer from a possibly incorrect assumption (ibid., p. 65).

- Involvement of uncertainty: It could always be that conditional independence (CI) is at least questionable and that auxiliary information is not available. If both is the case, the former explained methods are not appropriate. Rather we need to try to work with our data situation by taking into account this uncertainty in the model for the joint probability of $P_{\mathbf{XYZ}}$. Standard techniques, e.g. Maximum Likelihood Estimation (MLE) are not applicable anymore: we cannot estimate unit values for the parameters of the probability distribution. Nevertheless it is possible to estimate a set or interval of possible parameters (macro approach), or sets/intervals of possible observations for our missing observations of $\mathbf{Y}$ and $\mathbf{Z}$ (micro approach) (ibid., p. 97).

[3] More general information about missing mechanisms and their consequences for statistical matching can be found in D'Orazio et al. (2006), and generally in Rubin (1976).

2.3 Matching Categorical Data

This thesis has the aim to match categorical data, thus I want to take a detailed look at the distributions of categorical data. First of all, categorical data are discrete features, meaning that the variables have a limited number of possible values. A simple example is the variable *marital status*. Possible values are 'single', 'married', or 'widowed'. Furthermore it would be without meaning to construct a ranking between the values, thus we cannot say that being single is better than being married (Fahrmeir et al. 2011, pp. 16ff.). A suitable distribution for categorical data is the multinomial distribution.[4] If we deal with a categorical variable X with n observations, distribution of counts for X given a certain number of observations follows a multinomial distribution. A special case of the multinomial distribution is the binomial distribution, in which just two cases are possible – e.g., the variable *smoker* has two values, "yes" or "no". In practice the binary variable can have two values, which are coded 0/1. The value 1 occurs with probability π, and hence 0 occurs with probability $1 - \pi$ (Agresti 2007, pp. 4ff.).

Definition 1 ***Binomial probability distribution:***

The probability that a binary variable X with $x_i \in \{0, 1\}^n$ is observed to be 1 exactly x times is then

$$P(X_i = x) = \binom{n}{x} \pi^x (1 - \pi)^{n-x} \tag{2.1}$$

(Agresti 2007, pp. 4ff.).

For our setting with three variable blocks $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$, each variable is binary, thus it holds that for variable X_i of block $\mathbf{X}$ we have $n = n_{\mathsf{A}} + n_{\mathsf{B}}$ observations, with probability π_i for $i = 1, \ldots, p$, thus

$$P(X_i = x) = \binom{n}{x} \pi_i^x (1 - \pi_i)^{n-x}. \tag{2.2}$$

The same holds for variable Y_j of block $\mathbf{Y}$, whereas we only have n_{A} observations and probability π_j for $Y_j = 1$, with $j = 1, \ldots, q$, thus

$$P(Y_j = y) = \binom{n_{\mathsf{A}}}{y} \pi_j^y (1 - \pi_j)^{n_{\mathsf{A}}-y}; \tag{2.3}$$

and for variable Z_k of block $\mathbf{Z}$ a total of n_{B} observations is available, with probability

[4] More information on the multinomial distribution can be found in Agresti (2007).

π_k for $Z_k = 1$, $k = 1, \ldots, r$, thus

$$P(Z_k = z) = \binom{n_{\mathsf{B}}}{z} \pi_k^z (1 - \pi_k)^{n_{\mathsf{B}} - z}. \tag{2.4}$$

In short, we can say that $X_i \sim B(n, \pi_i)$, $Y_j \sim B(n_{\mathsf{A}}, \pi_j)$ and $Z_k \sim B(n_{\mathsf{B}}, \pi_k)$.

2.4 The Conditional Independence Assumption

In research concerning statistical matching an important assumption is often made: the conditional independence assumption (CIA). In the following I will explain what conditional independence means, and we will see the consequences of the CIA for the definition of the joint probability $P_{\mathbf{XYZ}}$ in statistical matching.

In statistical theory, independence of two or more variables is a central concept, which defines if random events are standing in any relation to each other. To illustrate independence, we have two random variables X and Z of the probability space $(\Omega, \mathcal{F}, P)$, with Ω being the sample space, $\mathcal{F}$ being the set of events, and P being a probability measure.[5] Our matter of interest is the probability $P(Y)$, which tells us the probability that Y occurs. Someone informs us that Z already has occurred. If this leads to a re-evaluation of $P(Y)$, the two variables Y and Z are not independent of each other, and it holds that:

Definition 2 ***Dependence of two variables:***

$$P(Y) \;\neq\; P(Y \mid Z), \tag{2.5}$$

in which $P(Y \mid Z)$ can be interpreted as the probability of Y if Z already happened (Meintrup and Schäffler 2005, pp. 119ff.). If the incidence of Z does not lead to a re-evaluation of $P(Y)$, we can say that Y is independent of Z, and it holds that:

Definition 3 ***Probability of two independent variables:***

$$P(Y) \;=\; P(Y \mid Z). \tag{2.6}$$

So far this was a short explanation of independence and conditional probability. Now we will turn to conditional independence. In our former setting we just had to deal with two variables Y and Z. To work with conditional independence, we have to consider at least three variables Y, Z, and X of the probability room $(\Omega, \mathcal{F}, P)$. We now assume that Y and Z are not independent, thus $P(Y) \neq P(Y \mid Z)$ meaning

[5] For reasons of simplicity in the following I will not work with blocks of variables like I do in the statistical matching context.

that if we are informed that Z occurs, we re-evaluate the probability of Y. However the situation changes: Now X occurs, and with this information, the probability for Y will not change anymore, regardless of Z. Thus Z and Y do not influence each other anymore, given X.

Definition 4 ***Conditional Independence:***

A variable Y is conditionallly independent of a variable Z given a variable X in P, which is denoted by

$$P \models (Y \perp Z \mid X),$$

if $P(Y \mid Z, X) = P(Y \mid X)$ or if $P(Z, X) = 0$ (Koller and Friedman 2009, p. 24).

If $P \models (Y \perp Z \mid X)$ holds, it follows that

$$P(Y, Z \mid X) = P(Y \mid X) \cdot P(Z \mid X), \quad \forall \quad \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}, \mathbf{z} \in \mathcal{Z} \tag{2.7}$$

(Studeny 2005, pp. 9ff., and Koller and Friedman 2009, pp. 23ff.).

As I already mentioned, the CIA is of great relevance in statistical matching. Our goal is to estimate the joint probability distribution of our common variables and our specific variables. Since we never observed all variables at once, it is important to use CI of the specific variables given the common variables, to make the joint probability distribution representable in our concrete data situation.[6]

The assumption of conditional independence enables to use the chain rule to make the joint probability distribution identifiable (Meintrup and Schäffler 2005, pp. 125ff.):

Definition 5 ***Chain rule for conditional independence:***

$$P(X, Y, Z) = P(Y \mid X) \cdot P(Z \mid X) \cdot P(X) \tag{2.8}$$

(Koller and Friedman 2009, p. 18).

Equation (2.8) ca be used directly in the statistical matching context, in which we have three blocks of variables $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$. $P(\mathbf{Y} \mid \mathbf{X})$ is observed in dataset A; $P(\mathbf{Z} \mid \mathbf{X})$ is observed in dataset B; and $P(\mathbf{X})$ is the probability observed in both datasets A and B. We can see that the impact of the CIA is great and turns the situation to a solvable one, given our data. In application it would be optimal to test if CIA holds before we build a model based on this assumption. Unfortunately

[6] At this point it should be noted that the common variables ought to be a good predictor for the specific variables (Raessler 2002, p. 19).

this is not possible with our combined datasets A and B, since we do not have a single joint observation. Thus we have to be aware that CIA does not hold at all and we receive misleading and wrong results (D'Orazio et al. 2006, p. 13).

2.5 Parameter Estimation in Statistical Matching

2.5.1 Maximum Likelihood Estimation for Univariate Data in Statistical Matching

In a parametric approach of statistical matching, we assume that $P(\mathbf{X}, \mathbf{Y}, \mathbf{Z}; \boldsymbol{\theta}) \in \mathcal{F}$ and $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^T$ within $T \in \mathbb{N}^+$. $\mathcal{F}$ is a set of parametric distributions and consists of three factors, given CI holds, as can be seen in Equation (2.8). Hence in a parametric approach we can write the probability as follows:

$$\begin{aligned} P_{\mathbf{X},\mathbf{Y},\mathbf{Z}}(\mathbf{x}, \mathbf{y}, \mathbf{z}; \boldsymbol{\theta}) &\overset{\text{i.i.d.}}{=} P_{\mathbf{YZ}|\mathbf{X}}(\mathbf{y}, \mathbf{z} \mid \mathbf{x};\ \boldsymbol{\theta}_{\mathbf{YZ}|\mathbf{X}}) \cdot P_{\mathbf{X}}(\mathbf{x};\ \boldsymbol{\theta}_{\mathbf{X}}) \\ &\overset{\text{CIA}}{=} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y} \mid \mathbf{x};\ \boldsymbol{\theta}_{\mathbf{Y}|\mathbf{X}}) \cdot P_{\mathbf{Z}|\mathbf{X}}(\mathbf{z} \mid \mathbf{x};\ \boldsymbol{\theta}_{\mathbf{Z}|\mathbf{X}}) \cdot P_{\mathbf{X}}(\mathbf{x};\ \boldsymbol{\theta}_{\mathbf{X}}), \end{aligned} \tag{2.9}$$

in which $\boldsymbol{\theta}_{\mathbf{X}} \in \Theta_{\mathbf{X}}$, $\boldsymbol{\theta}_{\mathbf{Y}|\mathbf{X}} \in \Theta_{\mathbf{Y}|\mathbf{X}}$, and $\boldsymbol{\theta}_{\mathbf{Z}|\mathbf{X}} \in \Theta_{\mathbf{Z}|\mathbf{X}}$. To estimate the density function we have to deal with the parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\mathbf{X}}, \boldsymbol{\theta}_{\mathbf{Y}|\mathbf{X}}, \boldsymbol{\theta}_{\mathbf{Z}|\mathbf{X}})$ (D'Orazio et al. 2006, pp. 14ff.). A common statistical approach for estimating parameters is maximum likelihood estimation (MLE). The goal of MLE is to estimate the parameter vector $\boldsymbol{\theta}$ by maximizing the likelihood of $\boldsymbol{\theta}$ given the observations in dataset A and B.

$$\begin{aligned} L(\boldsymbol{\theta} \mid \mathsf{A} \uplus \mathsf{B}) &= \prod_{a}^{n_{\mathsf{A}}} P_{\mathbf{XY}}(\mathbf{x_a}, \mathbf{y_a}; \boldsymbol{\theta}) \cdot \prod_{b}^{n_{\mathsf{B}}} P_{\mathbf{XZ}}(\mathbf{x_b}, \mathbf{z_b}; \boldsymbol{\theta}) \\ &= \prod_{a}^{n_{\mathsf{A}}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y_a} \mid \mathbf{x_a}; \boldsymbol{\theta}_{\mathbf{Y}|\mathbf{X}}) \cdot \prod_{b}^{n_{\mathsf{B}}} P_{\mathbf{Z}|\mathbf{X}}(\mathbf{z_b} \mid \mathbf{x_b}; \boldsymbol{\theta}_{\mathbf{Z}|\mathbf{X}}) \\ &\times \prod_{a}^{n_{\mathsf{A}}} P_{\mathbf{X}}(\mathbf{x_a}; \boldsymbol{\theta}_{\mathbf{X}}) \cdot \prod_{b}^{n_{\mathsf{B}}} P_{\mathbf{X}}(\mathbf{x_b}; \boldsymbol{\theta}_{\mathbf{X}}). \end{aligned} \tag{2.10}$$

It can be seen that Equation (2.10) – the likelihood – is solvable with our two datasets A and B even though missing data for $\mathbf{Y}$ and $\mathbf{Z}$ appears (ibid.). In statistical matching we usually are dealing with a multivariate problem, since we have several X, Y and Z variables. In the follwing I will focus on the case of multivariate discrete data, while D'Orazio et al. (2006, pp. 19ff.) gives an explanation of how to estimate the parameter in a multinomial case.

2.5.2 Maximum Likelihood Estimation for Multivariate Data in Statistical Matching

In our case we assume a binomial distribution for the variables $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$. As we can see in Equation (2.9), we are dealing with three random variable matrices $(\mathbf{Y} \mid \mathbf{X})$, $(\mathbf{Z} \mid \mathbf{X})$ and $\mathbf{X}$, hence we are confronted with a multivariate setting. D'Orazio et al. (2006, pp. 23ff.) argues that a convincing way to deal with this multivariate data is to break them down to univariate data. The authors propose to assume an appropriate saturated loglinear model to handle $(\mathbf{Y} \mid \mathbf{X})$, $(\mathbf{Z} \mid \mathbf{X})$ and $\mathbf{X}$ as univariate $(Y \mid X)$, $(Z \mid X)$ and X (D'Orazio et al. 2006, p. 24). Loglinear models can be used to model cell counts in contingency tables, by specifying what effect the level of the categorical data has on the size of the cell count. Loglinear models are also a flexible and interpretable tool for high-dimensional contingency tables, and for modeling the association structure between the variables. In saturated loglinear models we have no a priori constraints at all (usually some interaction effects are missing), thus all interaction terms are included. More information about loglinear models can be found e.g. in Andreß et al. (1997).

For transforming the multivariate case into a univariate case, we assume that X has in total $I = 2^p$ categories, Y has in total $J = 2^q$ categories, and Z has in total $K = 2^r$ categories. Though instead of dealing with p binary X variables, q binary Y variables, and r binary Z variables, we are confronted with *one* variable X that has now $I > 2$ categories, *one* variable Y that has $J > 2$ categories and *one* variable Z that has $K > 2$ categories. If e.g. $p = q = r = 2$, this leads to $I = J = K = 4$ categories per variable X, Y and Z. Three contingency tables are then needed for estimation: The first is just for X, the second represents $Y \mid X$, and the third $Z \mid X$. The second contingency table for $Y \mid X$ contains the data of dataset A, and has a dimension of $(I \times J)$, wheres the third contingency table contains the data of dataset B and has a dimension of $(I \times K)$. The first contingency table contains all information available for X, thus the data of A and B is summarized, and we have a dimension of $(1 \times I)$.

Furthermore $n^{\mathsf{A}}_{ij.}$ and $n^{\mathsf{B}}_{i.k}$ are defined as the observed marginal tables from datasets A and B, whereas $i, j, k \in \Delta$, and $\Delta = \{(i, j, k) : i = 1, \dots, I;\ j = 1, \dots, J;\ k = 1, \dots, K\}$ and $\boldsymbol{\theta} = \{\theta_{ijk}\}$. Given the likelihood (2.10), the estimated parameters for a univariate setting for the parameter π_{ijk} are:

$$\hat{\theta}_{i..} = \frac{n^{\mathsf{A}}_{i..} + n^{\mathsf{B}}_{i..}}{n_{\mathsf{A}} + n_{\mathsf{B}}}, \quad i = 1, \dots I, \tag{2.11}$$

$$\hat{\theta}_{j|i} = \frac{n^{\mathsf{A}}_{ij.}}{n^{\mathsf{A}}_{i..}}, \quad i = 1, \dots I;\ j = 1, \dots, J, \text{ and} \tag{2.12}$$

$$\hat{\theta}_{k|i} = \frac{n^{\mathsf{B}}_{i.k}}{n^{\mathsf{B}}_{i..}}, \quad i = 1, \ldots I; \; k = 1, \ldots, K \tag{2.13}$$

where the ‘dot’ symbol denotes marginalization of the corresponding variables (D’Orazio et al. 2006, pp. 14ff.).

The authors also give an example of the concept for a better understanding (D’Orazio et al. 2006, pp. 24ff.). Note that there are several more alternatives besides MLE, which are explained for instance in D’Orazio et al. (2006) and Raessler (2002). For applying the theory to data, there is an `R`-package called `StatMatch`, which offers varying functions for statistical matching (D’Orazio 2016).

3 Probabilistic Graphical Models

As the thesis title reveals, two theories come together: statistical matching, which was part of Chapter 2, and Markov networks. The theory of Markov networks belongs to the theory of probabilistic graphical models. In this chapter I will give an idea what probabilistic graphical models are, and what part Markov networks play in this theory.

3.1 What Probabilistic Graphical Models Are

As soon as we deal with real world problems, uncertainty is something we are confronted with all the time. An old but also very deeply elaborated theory is probability theory. Together with the theory of statistics, both are the basis of stochastics. Probability theory has its interest in situations that are random. However, at first sight this seems like an antagonism: isn't the main characteristic of randomness that it is not possible to calculate with? But exactly this is the idea of stochastics, or probability theory: to provide a formalism for describing and working with random events. The essential part of probability theory is to build models and evaluate them for specific situations which are characterized by uncertainty. The role of statistics, which is built on probabilistic models, is to infer from a certain number of observations to the population (Meintrup and Schäffler 2005, Preface). To simplify probabilistic models, a convincing approach is to use knowledge from independence assumptions. And here we are: probability theory, complemented with independence assumptions, yields graphical models. Graphical models illustrate independence assumptions very intuitively. The two main categories of probabilistic graphical models are Bayesian networks and Markov networks (Sucar 2015, pp. 5ff.).

Graphical models are a union of probability theory and graph theory. The aspect of uncertainty is the probabilistic part, whereas the modularity of graphical models comes from graph theory (Sucar 2015, pp. 3ff.). A basic knowledge of graph theory is required to understand the workings of graphical models. For that reason I will introduce the main concepts of graph theory in the following.[7]

3.2 Graph Theory

The main literature for this chapter is Beierle and Kern-Isberner (2006, pp. 451ff.), and Bondy and Murty (1982, ch. 1); all definitions in this chapter come from this literature.

[7] See Meintrup and Schäffler (2005) for more information on probability theory.

The requirement to illustrate situations via diagrams is very popular: it is easy to illustrate, very good to understand, and helps find core information and connections for special situations or problems. Thus graphs are often used because they are able to simplify complex situations (Bondy and Murty 1982, pp. 1ff.). Family trees, the metro net of a city, or also social networks like Facebook, they all can be represented as graphs. Furthermore we can distinguish between directed and undirected graphs. Thus we have to define what exactly a graph is, for which I will first give the definition of a directed graph, followed by the definition of an undirected graph.

Definition 6 ***Directed graph:***

A directed graph $\mathcal{G}$ is an ordered pair of $(\mathcal{V}, \mathcal{E})$, with $\mathcal{V}$ being a set of vertices or nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ a set of pairs of nodes (V, W), which are directed edges of $\mathcal{G}$.

An undirected graph can bee seen as a special case of a directed graph. In this thesis I will distinguish between the two by using $\mathcal{G}$ for a directed graph, and $\mathcal{H}$ for an undirected graph.

Definition 7 ***Undirected graph:***

An undirected graph $\mathcal{H}$ is an ordered pair of $(\mathcal{V}, \mathcal{E})$, with a symmetric relation $\mathcal{E}$, i.e. for all $V, W \in \mathcal{V}$:

$$(V, W) \in \mathcal{E} \Rightarrow (W, V) \in \mathcal{E}$$

Throughout this thesis I will illustrate vertices/nodes as circles. The label of a node is usually written inside the circle. The edges of a directed graph will be illustrated as arrows, whereas the edges of an undirected graph will be represented with just a line between two nodes. An example for both a directed and an undirected graph can be seen in Figure 2.

If two nodes in an undirected graph are linked, they are called adjacent. In Figure 2b) the nodes (X, Z) and (Y, X) are adjacent. A further important definition – for both directed and undirected graphs – is about paths in a graph.

Definition 8 ***Path, Cycle:***

For either a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ or an undirected graph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, a path of length S of two nodes $V, V' \in \mathcal{V}$ is a sequence of nodes

$$V_0, V_1, \dots, V_n$$

so that $V = V_0$ and $V' = V_n$, and for each $s \in \{1, \dots S\}$, $(V_{s-1}, V_s) \in \mathcal{E}$.

A cycle of length n is a path from $V_0, V_1, \ldots, v_n$ with $V_0 = V_n$, i.e. the initial and the final point are the same.

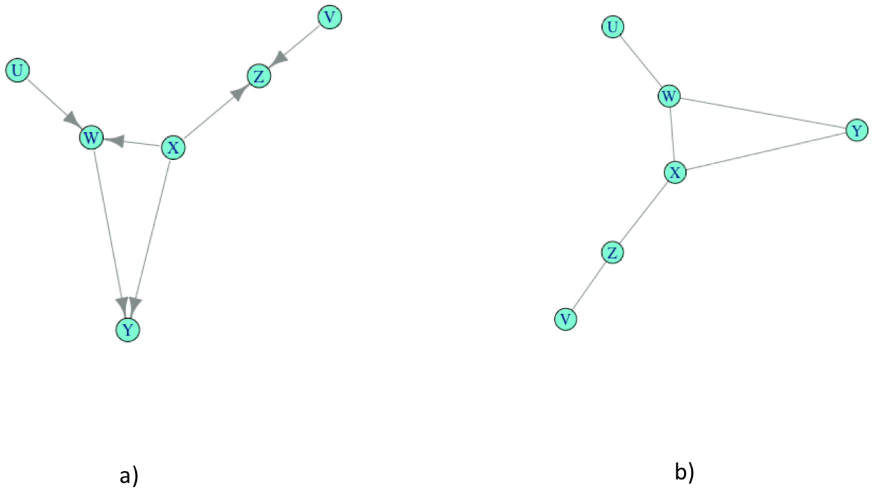

Figure 2: Part a) shows an example of a directed graph with six nodes, b) is an undirected graph with six nodes.

In Figure 2b) the sequence of U–W–X–Z–V is a path, and the sequence W–Y–X–W is a cycle.

Another concept of graph theory which is needed later for Bayesian networks is that of directed acyclic graphs (DAGs). This means that in a directed graph $\mathcal{G}$ there is no cycle at all. Further important terms in directed graphs are *parent* and *child/descendant*.

Definition 9 ***Parent node, Child node:***

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ being a DAG. If for $V, W \in \mathcal{V}$ there is an edge from W to V, thus $(W, V) \in \mathcal{E}$, the node W is called parent, and the node V is called child. The set of parent nodes $Pa(V)$ of node V is written as

$$\mathrm{Pa}(V) = \{W \in \mathcal{V} \mid (W, V) \in \mathcal{E}\}.$$

Figure 2b) includes several parent nodes and child nodes, e.g. $\mathrm{Pa}(Y) = \{X, W\}$. The term *descendant* is a bit more general. For $\mathcal{G}$ being a DAG, and $V, W \in \mathcal{G}$, if there exists a path from W to V in $\mathcal{G}$, then the node V is a descendant of W, or $\mathrm{De}(W) = \{V\}$. In Figure 2a), $\mathrm{De}(U) = \{Y\}$, because there is a path from U over W to Y.

Another tool to describe undirected graphs is the idea of cliques, which is very important for understanding Markov networks. An undirected graph $\mathcal{H}$ is called

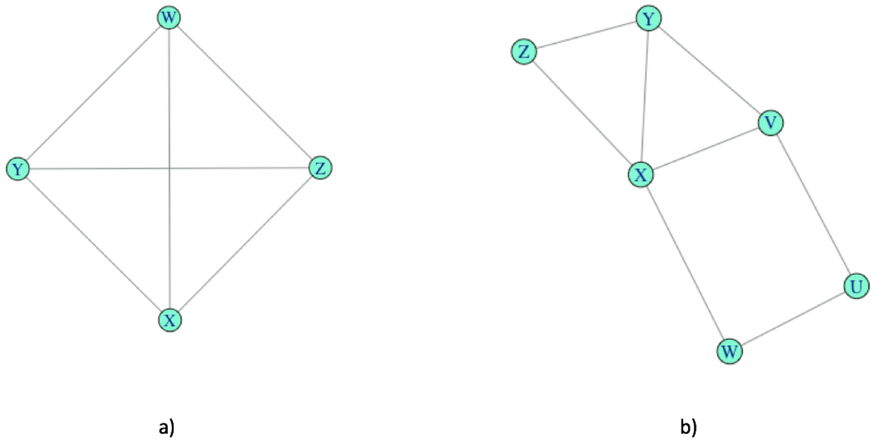

Figure 3: Part a) shows a complete undirected graph. Part b) shows an undirected graph which is not complete.

complete if every pair of nodes in the graph is linked through an edge. Figure 3a) has six pairs of nodes, which are $(X, Y), (X, W), (X, Z), (Y, W), (Y, Z), (Z, W)$, and each of these pairs is linked through an edge; this graph is thus an example of a complete graph. Moreover we can split the graph in Figure 3a) into four *complete subsets*, which are $(X, Y, Z), (Y, Z, W), (X, Z, W)$ and (X, Y, W); each of these subsets is in turn complete.

With this knowledge we can introduce cliques. A clique C is a subset of graph $\mathcal{H}$ such that C is a complete subset. Additionally if there is no node in $\mathcal{H}$ that can be added to the clique C such that C remains a clique, C is called *maximal*.

Thus the complete subset (X, Y, Z) of the graph in Figure 3a) is not a maximal clique, since we can add a further node W to this subset and the subset would still be complete, meaning that every pair of nodes is linked again. If we take a look at the graph in Figure 3b), we can find five maximal cliques, which are $(X, Y, Z), (X, Y, V), (X, W), (V, U)$, and (U, W). If a graph is separated by cliques, then every node of $\mathcal{V}$ is included in at least one clique. As we can see in the graph in Figure 3b), it is also possible that a node is included in several cliques, e.g. the node Y (Sucar 2015, pp. 27ff.).

This brief summary of graph theory should suffice as a basis for explaining Markov networks and Bayesian networks, which come next.

3.3 Markov Networks

When working with uncertainty in complex systems, probability distributions are a great way to tackle arising problems. Nevertheless if the system gets too complex, the size of the probability distribution grows exponentially. Moreover probability distributions themselves are complex too, and not very intuitive. Even for experts it is hard to find important relations between variables. Hence we have to find a way to work with probability distributions in a more intuitive way. Many experts have a good understanding and knowledge which components of a system influence each other and in which way. Besides, humans have a good understanding for maps and graphs; both help to simplify context. If we put together both probabilistic inference and graph theory, we get a powerful tool. And one way to work with probabilistic models is undirected probabilistic models, or Markov networks (MNs) (Sucar 2015, pp. 8ff.).

From the viewpoint of graph theory, a MN is an undirected graph $\mathcal{H} = \{\mathcal{V}, \mathcal{E}\}$ like in Definition 7. The nodes in the graph $\mathcal{H}$ represent random variables $\mathbf{X} = (X_1, \ldots, X_p)'$. At this point I will ignore the data context of statistical matching, and will just use $\mathbf{X}$, nevertheless the set of nodes $\mathcal{V}$ can be expanded with further nodes. Thus if we want to build a MN for p variables, the MN will have p nodes, one for each variable. The notation for nodes will be the same as for the random variables, thus $\mathcal{V} = \{X_1, \ldots, X_p\}$. The goal of a MN is to decode the independence structure between $\mathbf{X}$. A MN contains three types of dependences and independences:

1. variables that stand in direct relation, i.e. influence each other directly,

2. variables that stand in indirect relation, i.e. influence each other through a set of further variables, and

3. variables that are independent of each other.

In a MN one can imagine that the probabilistic influence *flows* through the graph, and with this picture in mind it is easier to understand independences represented in a MN (Koller and Friedman 2009, pp. 71ff.). In probabilistic graphical models we put together two ideas: (1) the concept of conditional independence, and (2) the theory of graph separation.

Conditional independence comes from probability theory and is already explained in Chapter 2.4. Graph separation however is part of graph theory: for three disjunct subsets[8] $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathcal{V}$, we say that $\mathbf{Z}$ separates $\mathbf{X}$ and $\mathbf{Y}$, or

$$\mathbf{X} \perp_{\mathcal{H}} \mathbf{Y} \mid \mathbf{Z}, \tag{3.1}$$

iff each path from the nodes in $\mathbf{X}$ to the nodes in $\mathbf{Y}$ passes at least one node in $\mathbf{Z}$; this is denoted by $\perp_{\mathcal{H}}$. Graph separation enables to represent direct dependences and also

[8] Or blocks of random variables, which are represented by three disjoint sets of nodes. For this reason I will use bold-case letters $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$, to retain the connection to statistical matching.

conditional independences in a MN. The ideal case is now to represent conditional independences of the joint probability distribution P via graphical separation, thus

$$\mathbf{X} \perp_{\mathcal{H}} \mathbf{Y} \mid \mathbf{Z} \quad \Leftrightarrow \quad \mathbf{X} \perp_{P} \mathbf{Y} \mid \mathbf{Z}, \tag{3.2}$$

with $\perp_P$ denoting independences in the probability distribution (Beierle and Kern-Isberner 2006, pp. 458ff.).

3.3.1 Markov Property and Independences in Markov Networks

We are still dealing with an undirected graph $\mathcal{H} = \{\mathcal{V}, \mathcal{E}\}$, and furthermore P is a probability distribution. We can distinguish between three types of MN:

- $\mathcal{H}$ is called dependence map (D-map) for P, if the conditional independence in P is represented in $\mathcal{H}$:

$$\mathbf{X} \perp_{P} \mathbf{Y} \mid \mathbf{Z} \quad \Rightarrow \quad \mathbf{X} \perp_{\mathcal{H}} \mathbf{Y} \mid \mathbf{Z}; \tag{3.3}$$

- $\mathcal{H}$ is called independence map (I-map) for P, if sets of nodes that are separated in the graph are indeed conditionally independent in P:

$$\mathbf{X} \perp_{\mathcal{H}} \mathbf{Y} \mid \mathbf{Z} \quad \Rightarrow \quad \mathbf{X} \perp_{P} \mathbf{Y} \mid \mathbf{Z}; \tag{3.4}$$

- $\mathcal{H}$ is called perfect graph for P iff $\mathcal{H}$ is both I-map and D-map for P:

$$\mathbf{X} \perp_{\mathcal{H}} \mathbf{Y} \mid \mathbf{Z} \quad \Leftrightarrow \quad \mathbf{X} \perp_{P} \mathbf{Y} \mid \mathbf{Z} \tag{3.5}$$

(Beierle and Kern-Isberner 2006, pp. 364ff.).

In turn, a MN can represent three types of conditional independences of variables:

1. pairwise independence: every strictly positive probability distribution P has a unique MN $\mathcal{H} = \langle \mathcal{V}, \mathcal{E} \rangle$ with

$$(X, Y) \notin \mathcal{E} \quad \text{iff} \quad X \perp_P Y \mid (\mathcal{V} - \{X, Y\}), \tag{3.6}$$

   with $(\mathcal{V} - \{X, Y\})$ indicating that the elements X and Y are excluded from the set of nodes $\mathcal{V}$;

2. local independence $\mathcal{I}_l(\mathcal{H})$: for a given graph $\mathcal{H}$ the neighbors of X are defined as $Nei(X)$ in $\mathcal{H}$. We define the local independences associated with X in $\mathcal{H}$

to be:

$$\mathcal{I}_l(\mathcal{H}) = \{(X \perp \mathcal{X} - \{X\} - Nei_{\mathcal{H}}(X) \mid Nei_{\mathcal{H}}(X) : X \in \mathcal{X}\}; \tag{3.7}$$

3. global independence $\mathcal{I}(\mathcal{H})$: a set of nodes $\mathbf{Z}$ separates $\mathbf{X}$ and $\mathbf{Y}$ in a MN $\mathcal{H}$, denoted $\mathrm{sep}_{\mathcal{H}}(\mathbf{X};\mathbf{Y} \mid \mathbf{Z})$, if there is no path between any node $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$ given $\mathbf{Z}$. We define the global independences associated with $\mathcal{H}$ to be:

$$\mathcal{I}(\mathcal{H}) = \{(\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}) : \mathrm{sep}_{\mathcal{H}}(\mathbf{X};\mathbf{Y} \mid \mathbf{Z})\}. \tag{3.8}$$

These three types of independences are the Markov property for MNs (Frydenberg 1990, pp. 337ff.). In general the types of independence are not equal, but for the case of strictly positive probability distributions, all are equivalent (Koller and Friedman 2009, pp. 118ff.).

3.3.2 The Joint Probability Distribution of a Markov Network

Since we want to use MNs for statistical matching, the independence structure in a MN is important for the application. Nevertheless the actual question is how the probability $P(\mathbf{X})$ can be expressed with a MN. If we want to parameterize a MN, we are not working with conditional probability distributions directly (like in a Bayesian network), but rather with so-called factors. Factors represent the symmetric relation between linked nodes, and capture the *affinities* between variables. Note that factors are no probabilities at all, rather factors can be seen as a degree of conformity. This makes them hard to understand and not very intuitive in their interpretation.

Definition 10 ***Factor:***

Let $\mathbf{D}$ *be a set of random variables. A factor* ϕ *is defined to be a function from possible values,* $Val(\mathbf{D})$ *to* $\mathbb{R}$*. A factor is non-negative if all its entries are non-negative. The set of variables* $\mathbf{D}$ *is called the scope of the factor, and is denoted* $Scope[\phi]$*.*

This thesis restricts itself to nonnegative factors, thus as we see in Definition 10, a factor can be any positive number and is not a limited number between $[0, 1]$. By using factors a MN can be parameterized; in MNs the factors have a similar role as conditional independence distributions have in Bayesian networks; this I will explain later. To parameterize the MN we divide the MN in several subsets of variables. For this we need another definition, which gives us more opportunities for factors (Koller and Friedman 2009, pp. 107ff.).

Definition 11 ***Factor product:***

Let $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ be three disjoint sets of variables, and let $\phi_1(\mathbf{X}, \mathbf{Y})$ and $\phi_2(\mathbf{Y}, \mathbf{Z})$ be two factors. The factor product $\phi_1 \times \phi_2$ is defined to be a factor $\psi : Val(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) \mapsto \mathbb{R}$ as follows:

$$\psi(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \phi_1(\mathbf{X}, \mathbf{Y}) \times \phi_2(\mathbf{Y}, \mathbf{Z}). \tag{3.9}$$

It is important to understand that if two factors are multiplied they have to fit, meaning that if $\mathbf{Y}$ is part of both factors, the value of $\mathbf{Y}$ has to be the same for the factors multiplied.

With this basis we can move on to undirected parameterization of a MN, which is done with the Gibbs distribution.

Definition 12 ***Gibbs distribution***

A distribution P_Φ is a Gibbs distribution parameterized by a set of factors $\Phi = \{\phi_1(\mathbf{D_1}), \phi_2(\mathbf{D_2}), \ldots, \phi_k(\mathbf{D_k}\}$ if it is defined as follows:

$$P_\Phi(X_1, \ldots, X_n) = \frac{1}{\zeta} \tilde{P}_\Phi(X_1, \ldots, X_n), \tag{3.10}$$

with

$$\tilde{P}_\Phi(X_1, \ldots, X_n) = \phi_1(\mathbf{D_1}) \times \phi_2(\mathbf{D_2}) \times \cdots \times \phi_m(\mathbf{D_m})$$

being an unnormalized measure, and

$$\zeta = \sum_{X_1, \ldots, X_n} \tilde{P}_\Phi(X_1, \ldots, X_n)$$

being a normalizing constant called the partition function.

Note that the partition function is very important since the factors themselves are not normalized: $\tilde{P}_\Phi \notin [0, 1]$. Moreover we can see that each factor contributes a part to the joint probability distribution (Koller and Friedman 2009, pp. 108ff.).

A question that has not been answered yet is how to choose the factors. For the parameterization of a MN it is important that the actual structure $\mathcal{H}$ of a MN is captured through the Gibbs distribution. If two nodes are linked, the Gibbs distribution should reflect this. To reach this goal it is important that each node is included at least once in a factor, and furthermore we need factors for direct connections between nodes. These requirements lead to the characteristics of *cliques*, which I explained in Chapter 3.2. Remember that if a graph is separated in its maximal cliques, each node is in at least one clique, and cliques are *complete subsets* of the graph. This allows us to factorize the joint probability distribution via the (maximal) cliques of the MN.[9] By just using cliques for the factors, we can reduce the total number of factors in the parameterization of the MN:

$$P_\Phi(X_1, \ldots, X_n) = \frac{1}{\zeta} \prod_{C \in Cliques(\mathcal{H})} \phi_C(X_C), \tag{3.11}$$

with ζ being the partition function (Koller and Friedman 2009, pp. 106ff.).

With these principles of Markov networks in mind, we can move on to Bayesian networks. Even though the focus of this thesis is on statistical matching with MN, the research and the idea of this thesis originates in statistical matching with Bayesian networks.

3.4 Bayesian Networks

For introducing Bayesian networks (BNs) I will give a short example that demonstrates the difference between BNs and MNs. The example originally comes from Beierle and Kern-Isberner (2006).

Coin toss example:

$L \in \{head,\ number\}$ and $M \in \{head,\ number\}$ are two random variables both describing the toss of a fair coin. It is obvious that the two variables are independent of each other. There is also the random variable G: “ringing of a bell”, with $G \in \{ringin,\ not\ ringing\}$, which happens if both coins have the same result, hence $(l, m) \in \{(head, head), (number, number)\}$. As soon as one takes into consideration the third variable G, L and M are not conditionally independent. We can write this situation as

$$L \perp_P M \mid \emptyset, \quad \text{but not} \quad L \perp_P M \mid G. \tag{3.12}$$

To represent this situation in a MN is impossible, since there is a obvious direction of influence. Nevertheless another graphical representation is possible: BNs represent such a structure in P. In the coin toss example, the result of the two coins influences the result of the bell. This is not a symmetric relation anymore, rather a directional component is involved. The corresponding BN for this situation is shown in Figure 4.

In terms of graph theory, a BN is a directed acyclic graph. As with MNs, also a

[9] It is not necessary to use maximal cliques, as we will see in Chapter 5.1.

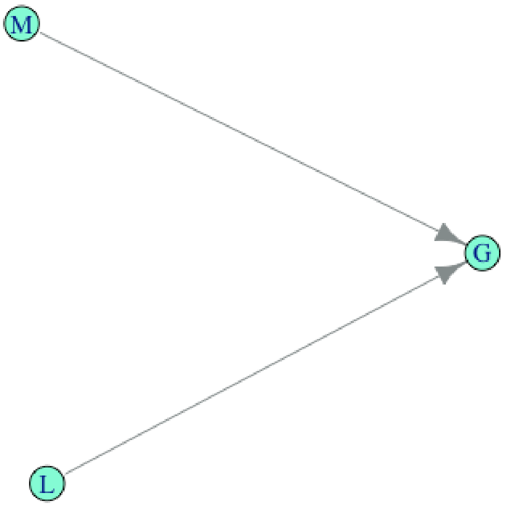

Figure 4: Bayesian network for the coin toss example. The influence goes from L and M to G. This is demonstrated with directed edges.

BN represents a set of random variables $\mathbf{X} = (X_1, \dots, X_p)$ via nodes. Furthermore nodes are connected with directed edges.

$\mathrm{Pa}(X_i) \subseteq \mathcal{V}$, with $i = 1, \dots, p$ is the set of parent nodes; $\mathrm{De}(X_i) \subseteq \mathcal{V}$ is the set of descendants; and $\mathrm{Nd}(X_i) \subseteq \mathcal{V}$ is the set of all non-descendants of X. A BN encodes the following assumption of independence, also called local independences, noted as

$$\text{For each variable } X_i : (X_i \perp_{\mathcal{G}} \mathrm{Nd}(X_i) \mid \mathrm{Pa}(X_i)). \tag{3.13}$$

Thus given the parents of a node, this very node is independent of all its non-descendants.

At this point we can already see why BNs can be a powerful tool for statistical matching: the commonality of conditional independences. In statistical matching we assume CI to make our model identifiable, and BNs represent CI in a natural way. Another thing statistical matching and BN have in common is the use of factorization to model a probability distribution. Since a BN implies CI structures of the variables, we are allowed to factorize a distribution P of $\mathcal{G}$.[10] Factorization in BNs means that the probability for any set of values of the joint probability distribution can be computed with a factor[11] for each value. This leads to great flexibility and also less computation capacity, since we do not have to take into consideration the whole joint probability distribution to calculate a certain probability. The definition of the

[10] A proof why we can use factorization in BNs can be found in Koller and Friedman (2009, p. 62).

[11] In this context a factor is not to be understood as in Equation (10), but rather as a factor of a product.

chain rule of BNs is the following (Koller and Friedman 2009, p. 62):

Definition 13 ***Chain rule for Bayesian networks:***

Let $\mathcal{G}$ be a BN graph over the variables $X_1, \dots, X_p \in \mathcal{V}$. We say that a distribution P over the same space factorizes according to $\mathcal{G}$ if P can be expressed as a product

$$P(X_1, \dots, X_p) = \prod_{i=1}^{p} P(X_i \mid \mathrm{Pa}(X_i)^{\mathcal{G}}). \tag{3.14}$$

This equation is called the chain rule for Bayesian networks. The individual factors $P(X_i \mid \mathrm{Pa}(X_i)^{\mathcal{G}})$ are called conditional probability distributions (CPDs).

The definition of the chain rule leads directly to the Markov assumption, which states that each node is conditionally independent of its non-descendants, given its parents.

We yet have to focus on independence structures presented in $\mathcal{G}$. Even though directed graphs are an intuitive tool to present distributions and dependences, it is not obvious at first sight whether variables are independent given another variable. Nevertheless it is important to understand the graph structure in more detail, since the CI is the commonality between statistical matching and BN. Thus the question is, in what kind of representation in $\mathcal{G}$ does $(X \perp Y \mid Z)$ hold? To find a solution to that question we have to talk about directed separation, in short d-separation.

In the two-variable case, there are two obvious options: (1) X and Y are not connected to each other in any way. That implies that X and Y are marginally independent of each other. (2) X and Y are directly connected with each other, e.g. $X \rightarrow Y$, meaning that X influences Y directly and thus X and Y are dependent.

As soon as a third variable Z is represented in $\mathcal{G}$ it is getting more complicated. For three variables X, Y and Z there are four ways to be connected to each other. The four options can be seen in Figure 5.

The question is now if $(X \perp Y \mid Z)$ holds. For that aim I want to revert to an example of Koller and Friedman (2009, pp. 70ff.):

Imagine a situation in which the student's *intelligence* influences the student's *grade*. Thus for highly intelligent students it is more likely to get good grades. Moreover we know that students with good grades are more likely to get a *letter* of recommendation. Thus we have a situation like in Figure 5 a), with *intelligence* → *grade* → *letter*. As long as we do not know about the student's grades, it is obvious that intelligence and letter are dependent. But as soon as we know about the student's grades (and they are good), we do not need any information on their intelligence anymore to tell if they get a letter of recommendation. This is of course no proof, but it helps to understand the situation of d-separation. In the case of

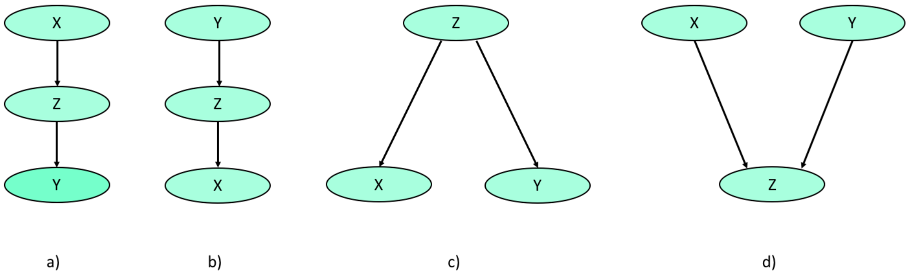

Figure 5: The four possibilities how three variables X, Y and Z can be connected in a Bayesian network structure $\mathcal{G}$. Graph a) is called indirect causal effect, trail b) is an indirect evidential effect, c) is called a common cause, and trail d) is a common effect (Koller and Friedman 2009, p. 70).

indirect causal effect, we can say that $(X \perp Y \mid Z)$. Since conditional independence is symmetric, Figure 5 b) is trivial, and also $(Y \perp X \mid Z)$ holds.

For Figure 5 c) holds that $(X \perp Y \mid Z)$. Nevertheless I will illustrate this again with an example of Koller and Friedman (2009, p. 70): Suppose we still have the variables of a student's *intelligence* and *grade*. It still holds that students with high intelligence are likely to have good grades. We furthermore know that students with high intelligence are also likely to have high *SAT* scores. Thus we are confronted with a situation like this: *grade* ← *intelligence* → *SAT*. As long as we do not know about the student's intelligence, *grade* and *SAT* score are correlated. As soon as we get informed about the student's intelligence, this correlation disappears, since actually it is the intelligence that influences both, the *grade* and the *SAT* score. Again it holds that $(X \perp Y \mid Z)$.

For the graph of Figure 5 d) we have to consider a further variable: the *difficulty* of the tests the student wrote. It is intuitive that the less difficult a test is, the better the grades are. Still also the *intelligence* influences the student's *grade*. Hence we have to deal with this situation: *intelligence* → *grade* ← *difficulty*. As long as we cannot observe *grade*, *intelligence* and *difficulty* are independent (we can also get to this conclusion with the graph). Thus the common effect graph is different from the previous ones: If we cannot observe Z, then X and Y are independent. If we observe the *grade*, and it is good, it is likely that the student is intelligent. But if we now get informed that the test was also easy, the probability that the student has high intelligence decreases, since the easy test has influence on the grade as well.

At this point I will introduce a new term: When influence can flow from X to Y via Z, we define this trail $X \rightleftharpoons Z \rightleftharpoons Y$ as *active*. To transmit this to our graphs in Figure 5, we can summarize that for a) to c) the trails are active iff Z is not observed, and for d) the trail is active iff either Z or one of Z's descendants is observed.

This explanation of dependence structures in BNs should suffice as proof for now. For more information I refer to Koller and Friedman (2009), since more technical knowledge is not necessary at this point.

The definition of d-separation is given below:[12]

Definition 14 ***D-separation:***

Let $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ *be three sets of nodes in* $\mathcal{G}$. *We say that* $\mathbf{X}$ *and* $\mathbf{Y}$ *are d-separated given* $\mathbf{Z}$*, denoted* $\text{d-sep}_{\mathcal{G}}(\mathbf{X}; \mathbf{Y} \mid \mathbf{Z})$*, if there is no path between any node* $X \in \mathbf{X}$ *and* $Y \in \mathbf{Y}$ *given* $\mathbf{Z}$. *We use* $\mathcal{I}(\mathcal{G})$ *to denote the set of independences that correspond to d-separation:*

$$\mathcal{I}(\mathcal{G}) = \{(X \perp Y \mid Z) : \text{d-sep}_{\mathcal{G}}(\mathbf{X}; \mathbf{Y} \mid \mathbf{Z})\}$$

As we can see, the concept of d-separation for BNs may not be as intuitive as the independences in MNs, but we have several more opportunities to model independences in BNs. Furthermore BNs are capable of illustrating other dependences not possible in MNs, and the other way around. The main difference regarding dependence structure in MNs and BNs is that in MNs we are dealing with symmetric influence, whereas in BNs the influence has a direction: the parent node influences the child node.

This chapter paves the way for understanding of this thesis. It summarizes the theory of probabilistic graphical models, starting with the fundamentals of graph theory. The two main variations of probabilistic graphical models explained in this chapter, Markov networks and Bayesian networks, are used in the context of statistical matching. The next chapter will bring both theories together: probabilistic graphical models, and statistical matching.

[12] Again, I will use bold notation $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$, since they represent blocks of random variables.

4 Statistical Matching with Probabilistic Graphical Models

Probabilistic graphical models are a powerful tool for breaking down independence structures in an intuitive way, and even the joint probability distribution is comprehensible. These two factors are needed for statistical matching: computing a joint probability distribution, and assuming conditional independence. Among the first to use probabilistic graphical models for the aim of statistical matching were Eva Endres and Thomas Augustin (2016). I will begin this chapter with explaining their work in this field.

4.1 Matching Discrete Data with Bayesian Networks

So far I have given a basic introduction into Bayesian networks. Endres and Augustin (2016) were among the first to use them for statistical matching of discrete data, and they did formal work to utilize BNs for this goal.

To bring BNs and statistical matching together, we have to remember the framework of statistical matching: we are dealing with three blocks of variables. The common variables $\mathbf{X}$ which are observed in both datasets A and B. And we have our partly missing variables $\mathbf{Y} \in \mathsf{A}$ and $\mathbf{Z} \in \mathsf{B}$. Since a BN implies conditional independences, in its graph we have to use the CIA for our statistical matching scenario, in order to bring the two concepts together. Since we are confronted with missing data, we have no other option than to make sure that influence flows from $\mathbf{Y}$ to $\mathbf{Z}$ via $\mathbf{X}$. To guarantee this CI structure in a BN we have to restrict the directed edges in a form such that $\mathbf{Y} \rightleftharpoons \mathbf{X} \rightleftharpoons \mathbf{Z}$. Now two or three steps are following, depending on whether a macro or micro approach is pursued.

1. The graph structure:

Like always in graphical models we can use expert knowledge to establish a graph. If expert knowledge is not available, we only use our data for estimation. We will start with an estimation of a DAG just on the basis of our common variables $\mathbf{X}$ of A and B. This estimation, based on all observations of $\mathbf{X}$, is used to create a graph $\hat{\mathcal{G}}_{\mathbf{X}}^{\mathsf{A}\Cup\mathsf{B}}$. Still missing in the graph are the variables $\mathbf{Y}$ and $\mathbf{Z}$. The approach for $\mathbf{Y}$ and $\mathbf{Z}$ is the same, thus I will explain it only for $\mathbf{Y}$. To include $\mathbf{Y}$ in a DAG, we have to use all edges $\hat{\mathcal{E}}_{\mathbf{X}}^{\mathsf{A}\Cup\mathsf{B}}$ of $\hat{\mathcal{G}}_{\mathbf{X}}^{\mathsf{A}\Cup\mathsf{B}}$ as prior knowledge to maintain the already estimated BN structure of $\mathbf{X}$. With this prior knowledge, a DAG for $\mathbf{X}$ and $\mathbf{Y}$ is estimated on the observations from A, shortly $\hat{\mathcal{G}}_{\mathbf{XY}}^{\mathsf{A}}$. The same procedure is used for $\mathbf{Z}$ to get an estimation $\hat{\mathcal{G}}_{\mathbf{XZ}}^{\mathsf{B}}$.

Since the structure of $\hat{\mathcal{G}}^{\mathsf{A}\Cup\mathsf{B}}_{\mathbf{X}}$ is included in both $\hat{\mathcal{G}}^{\mathsf{A}}_{\mathbf{XY}}$ and $\hat{\mathcal{G}}^{\mathsf{B}}_{\mathbf{XZ}}$, it is easy to combine these two DAGs into one DAG that contains all variables $\mathbf{XYZ}$. These are represented as nodes, while the edges are the union $\hat{\mathcal{E}}^{\mathsf{A}\Cup\mathsf{B}}_{\mathbf{X}} \cup \hat{\mathcal{E}}^{\mathsf{A}}_{\mathbf{XY}} \cup \hat{\mathcal{E}}^{\mathsf{B}}_{\mathbf{XZ}}$.

Endres and Augustin (2016) also worked out a second procedure for the estimation of DAG bypassing the estimation of $\hat{\mathcal{G}}^{\mathsf{A}\Cup\mathsf{B}}_{\mathbf{X}}$, where two graphs $\hat{\mathcal{G}}^{\mathsf{A}}_{\mathbf{XY}}$ and $\hat{\mathcal{G}}^{\mathsf{B}}_{\mathbf{XZ}}$ are estimated and combined. More information on the second procedure can be found in Endres and Augustin (2016, p. 163).

2. Estimation of local parameters and the joint probability distribution:

For the estimation of the local parameters and the joint probability distribution in a BN, we use the chain rule for BNs from Equation (3.14). For this, we have to take into consideration which observations are used in the BN; in all other respects we can directly transfer Equation (3.14). Thus the joint probability distribution, which is fully described by its probability mass distribution, of a BN in a statistical matching context is

$$\begin{aligned}\hat{P}^{\mathsf{A}\Cup\mathsf{B}}(\mathbf{X}=\mathbf{x},\mathbf{Y}=\mathbf{x},\mathbf{Z}=\mathbf{z}) = &\prod_{j=1}^{q} \hat{p}_{\hat{\mathcal{G}}^{\mathsf{A}}_{\mathbf{XY}}}(y_j \mid \mathrm{Pa}(Y_j)) \cdot \prod_{k=1}^{r} \hat{p}_{\hat{\mathcal{G}}^{\mathsf{B}}_{\mathbf{XZ}}}(z_k \mid \mathrm{Pa}(Z_k)) \\ &\cdot \prod_{i=1}^{p} \hat{p}_{\hat{\mathcal{G}}^{\mathsf{A}\Cup\mathsf{B}}_{\mathbf{X}}}(x_i \mid \mathrm{Pa}(X_i)).\end{aligned} \tag{4.1}$$

This joint probability distribution is identifiable for the data situation in statistical matching, just as in the likelihood in Equation (2.10), since we are distinguishing between the different datasets.

As mentioned before, for the macro approach we need two steps. If this is our aim, we have reached their end with Equation (4.1). If we furthermore want to construct a synthetic data file by estimating values for the missing $\mathbf{Y}$ and $\mathbf{Z}$ observations, we need an additional third step.

3. Imputation of the missing values:

The aim of this step is to create a synthetic data file, with no missing values at all (micro approach). For that reason we have to estimate the non-observed realizations of $\mathbf{Z}$ in A, and also the non-observed realizations of $\mathbf{Y}$ in B. The approach is rather straightforward. In the second step we obtained estimations for our posterior distributions, which we will use now. For the goal of imputation we draw values from $\hat{P}^{\mathsf{A}\Cup\mathsf{B}}(\mathbf{Y}=\mathbf{y} \mid \mathbf{X}=\mathbf{x})$ and $\hat{P}^{\mathsf{A}\Cup\mathsf{B}}(\mathbf{Z}=\mathbf{z} \mid \mathbf{X}=\mathbf{x})$ for the synthetic values of Y_j, $j=1,\dots,q$, and Z_k, $k=1,\dots,r$, given the realization of $\mathbf{X}$. With them we fill our missing entries (ibid., pp. 162ff.).

Applying BNs as a tool for statistical matching, with data from the Leibniz Institute for the Social Sciences, Endres and Augustin (2016) gained some promising results. The authors note that undirected graphical models, or MNs, can be a promising approach for statistical matching, too.

MNs also imply CI, but without any directions between the nodes. This has two advantages: first, the direction which is implied by a BN between nodes is not always reasonable. In MNs a symmetric connection between the nodes is represented. Second, with undirected edges there is no differentiation of the graphs presented in Figure 5 c) and d), since without the arrows the two graphs are the same. Hence by application of MNs for statistical matching, we do not need to restrict the estimated graph in this regard. For this reason the thesis at hand will work with Markov networks as a tool for statistical matching.

4.2 Matching Categorical Data with Markov Networks

Up to now I presented the theory of both statistical matching and graphical models. The aim of this chapter is to actually combine statistical matching with Markov networks. I will start with the idea of how to estimate such a MN with lots of missing data, and how the CIA will help bring both theories together.

The idea of using Markov networks for statistical matching is straightforward, and is built on applying Bayesian networks for statistical matching in Chapter 4.1. First we have to take the special data situation into consideration: We are dealing with three blocks of variables which are $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$. $\mathbf{X}$ is observed in both data surveys A and B, however $\mathbf{Y}$ is only observed in A, and $\mathbf{Z}$ only in B. The assumption that $(\mathbf{Y} \perp \mathbf{Z} \mid \mathbf{X})$ brings statistical matching and MN together, thus we have to guarantee that the MN makes exactly this assumption hold. As I explained in Chapter 3.3, we can model this if every path $\mathbf{Y}$—$\mathbf{Z}$ is passing at least one X_i of $\mathbf{X}$, such that $\mathbf{Y}$—$\mathbf{X}$—$\mathbf{Z}$. For the estimation of the joint probability distribution, we have to implement two steps.

1. The graph structure:

To ensure that $\mathbf{Y}$—$\mathbf{X}$—$\mathbf{Z}$, we will start with modeling a MN for our variables $\mathbf{X} = (X_1, \ldots, X_p)'$, for which we have $n_{\mathsf{A}} + n_{\mathsf{B}}$ observations. The resulting graph will be abbreviated with $\hat{\mathcal{H}}_{\mathbf{X}}^{\mathsf{A}\Cup\mathsf{B}} = (\mathcal{V}_{\mathbf{X}}, \hat{\mathcal{E}}_{\mathbf{X}})$, with $\mathcal{V}_{\mathbf{X}} = \{X_1, \ldots, X_p\}$. The estimation of $\mathcal{H}_{\mathbf{X}}^{\mathsf{A}\Cup\mathsf{B}}$ is based on all observations $\mathbf{X}$ in dataset A and dataset B. The graph structure of $\hat{\mathcal{H}}_{\mathbf{X}}^{\mathsf{A}\Cup\mathsf{B}}$ will be used in the following as previous knowledge.

Now we deal with our specific variables $\mathbf{Y}$ and $\mathbf{Z}$, while it does not matter which one is modeled first; I will start with $\mathbf{Y}$. We have to model a MN for the variables $\mathbf{Y} = (Y_1, \ldots, Y_q)'$ and $\mathbf{X} = (X_1, \ldots, X_p)'$ with the observations of dataset A. The graph structure of $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{X}}$ will be retained and treated as previous knowledge. We are interested especially in the edges within $\mathbf{Y}$ itself, and of course between $\mathbf{Y}$ and $\mathbf{X}$. The received graph will be abbreviated with $\hat{\mathcal{H}}^{\mathsf{A}}_{\mathbf{XY}} = (\mathcal{V}_{\mathbf{XY}}, \hat{\mathcal{E}}_{\mathbf{XY}})$, with $\mathcal{V}_{\mathbf{XY}} = \{Y_1, \ldots, Y_q, X_1, \ldots, X_p\}$.

The same procedure has to be repeated for $\mathbf{Z} = (Z_1, \ldots, Z_r)'$ and $\mathbf{X} = (X_1, \ldots, X_p)'$ on the basis of dataset B, where the structure of $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{X}}$ will again be used as previous knowledge. The graph $\hat{\mathcal{H}}^{\mathsf{B}}_{\mathbf{XZ}} = (\mathcal{V}_{\mathbf{XZ}}, \hat{\mathcal{E}}_{\mathbf{XZ}})$, with $\mathcal{V}_{\mathbf{XZ}} = \{Z_1, \ldots, Z_r, X_1, \ldots, X_p\}$, will now contain edges within $\mathbf{Z}$, between $\mathbf{Z}$ and $\mathbf{X}$, and of course the edges within $\mathbf{X}$. Hence we recieve three MN structures $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{X}}$, $\hat{\mathcal{H}}^{\mathsf{A}}_{\mathbf{XY}}$ and $\hat{\mathcal{H}}^{\mathsf{B}}_{\mathbf{XZ}}$.

What is special about these three networks is that they all have the same MN structure in $\mathbf{X}$, since we used this structure as previous knowledge. At this point we can assemble the $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{X}}$, $\hat{\mathcal{H}}^{\mathsf{A}}_{\mathbf{XY}}$ and $\hat{\mathcal{H}}^{\mathsf{B}}_{\mathbf{XZ}}$ to one MN structure $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{XYZ}}$, while the structure of $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{XYZ}}$ is characterized by the fact that all influence flows from $\mathbf{Y}$—$\mathbf{X}$—$\mathbf{Z}$. Thus our assumption that $(\mathbf{Y} \perp \mathbf{Z} \mid \mathbf{X})$ in statistical matching is fulfilled in the MN structure $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{XYZ}}$.

2. Estimation of the joint probability distribution:

For estimation of the joint probability distribution $\hat{P}^{A \Cup B}(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{Z} = \mathbf{z})$ through $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{XYZ}}$, we have to factorize it via maximal cliques C of the MN $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{XYZ}}$:

$$\hat{P}^{A \Cup B}(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{Z} = \mathbf{z}) = \frac{1}{\zeta} \cdot \prod_{C \in \hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{XYZ}}} \hat{\phi}_C(xyz), \tag{4.2}$$

with C being a maximal clique in $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{XYZ}}$.

$\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{XYZ}}$ is constructed artificially, since statistical matching is characterized by blockwise missing data. Thus there will never be a (maximal) clique within (XYZ), but of course within (XZ) or (XY). By considering this fact, we can split Equation (4.2) into at least three factors.[13][14] Maximal cliques are possible between XY and XZ, and of course within $\mathbf{X} = (X_1, \ldots, X_p)'$, $\mathbf{Y} = (Y_1, \ldots, Y_q)'$, and $\mathbf{Z} = (Z_1, \ldots, Z_r)'$. Furthermore we have to take into consideration the special data situation of statistical matching, and ensure that we distinguish between the two datasets A and B. The joint probability distribution of a MN structure in the context of statistical matching can be expressed as follows:

[13] If more than two datasets are matched, Equation 4.2 also has more factors.

[14] The term factor relates to a factor in a product.

$$\hat{P}^{A\Cup B}(\mathbf{X}=\mathbf{x},\mathbf{Y}=\mathbf{y},\mathbf{Z}=\mathbf{z}) = \frac{1}{\zeta}\cdot \prod_{C\in\hat{\mathcal{H}}^{\mathsf{A}}_{\mathbf{XY}}\setminus C\in\hat{\mathcal{H}}^{\mathsf{A\Cup B}}_{\mathbf{X}}} \hat{\phi}_C(xy) \cdot \prod_{C\in\hat{\mathcal{H}}^{\mathsf{B}}_{\mathbf{XZ}}\setminus C\in\hat{\mathcal{H}}^{\mathsf{A\Cup B}}_{\mathbf{X}}} \hat{\phi}_C(xz) \cdot \prod_{C\in\hat{\mathcal{H}}^{\mathsf{A\Cup B}}_{\mathbf{X}}} \hat{\phi}_C(x), \tag{4.3}$$

with C being a maximal clique in $\hat{\mathcal{H}}^{\mathsf{A\Cup B}}_{\mathbf{XYZ}}$.

In Equation (4.3), ζ is the partition function and C has to be a maximal clique in $\hat{\mathcal{H}}^{\mathsf{A\Cup B}}_{\mathbf{XYZ}}$. The second factor contains all maximal cliques between $\mathbf{X}$ and $\mathbf{Y}$, and those within $\mathbf{Y}$ itself. The third factor contains all maximal cliques between $\mathbf{X}$ and $\mathbf{Z}$, and those within $\mathbf{Z}$ itself. The fourth factor contains all maximal cliques within $\mathbf{X}$ itself. It is possible that the last factor $\prod_{C\in\hat{\mathcal{H}}^{\mathsf{A\Cup B}}_{\mathbf{X}}} \hat{\phi}_C(x)$ of Equation (4.3) is omitted, since C has to be a maximal clique in $\hat{\mathcal{H}}^{\mathsf{A\Cup B}}_{\mathbf{XYZ}}$ and thus $C \in \hat{\mathcal{H}}^{\mathsf{A\Cup B}}_{\mathbf{X}}$ can be $\emptyset$.

For better understanding of Equation (4.3), I will give two posssible examples and explain what the single factors of Equation (4.3) are. Figures 6 and 7 show two possible MNs $\hat{\mathcal{H}}^{\mathsf{A\Cup B}}_{\mathbf{XYZ}}$.

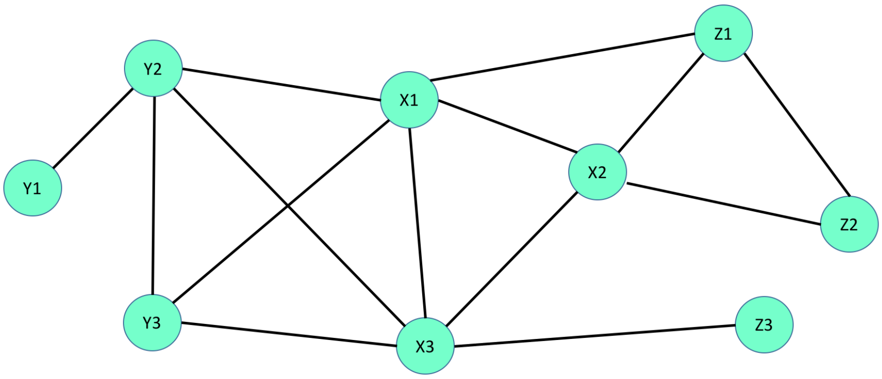

Figure 6: Possible Markov network structure for $\hat{\mathcal{H}}^{\mathsf{A\Cup B}}_{\mathbf{XYZ}}$, with nodes $\mathbf{X} = (X_1, X_2, X_3)'$, $\mathbf{Y} = Y_1, Y_2, Y_3)'$ and $\mathbf{Z} = (Z_1, Z_2, Z_3)'$.

The maximal cliques C in $\hat{\mathcal{H}}^{\mathsf{A\Cup B}}_{\mathbf{XYZ}}$ of Figure 6 are the following: $(Y_1, Y_2), (Y_2, Y_3, X_1, X_3), (X_1, X_2, X_3), (X_1, X_2, Z_1), (X_2, Z_1, Z_2), (X_3, Z_3)$, while

- $\hat{\mathcal{H}}^{\mathsf{A}}_{\mathbf{XY}}$ contains: $(Y_1, Y_2), (Y_2, Y_3, X_1, X_3)$,
- $\hat{\mathcal{H}}^{\mathsf{B}}_{\mathbf{XZ}}$ contains: $(X_1, X_2, Z_1), (X_2, Z_1, Z_2), (X_3, Z_3)$, and
- $\hat{\mathcal{H}}^{\mathsf{A\Cup B}}_{\mathbf{X}}$ contains: (X_1, X_2, X_3).

Even though there was just one edge added to the graph of Figure 6, the situation changes a lot in Figure 7 . The maximal cliques C in $\hat{\mathcal{H}}^{\mathsf{A\Cup B}}_{\mathbf{XYZ}}$ of Figure 7 are the follow-

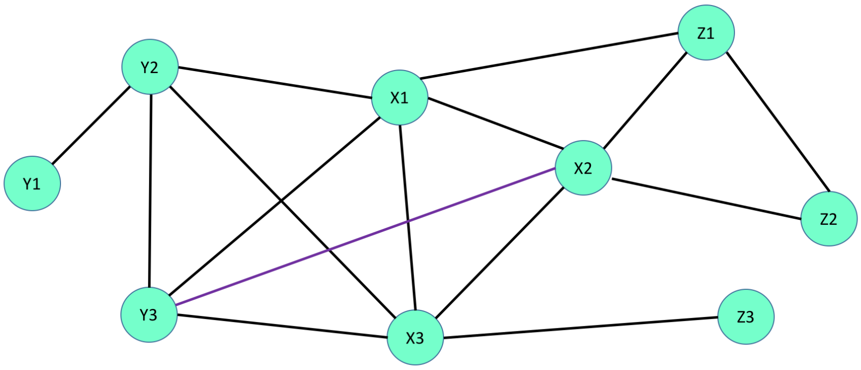

Figure 7: Possible Markov network structure for $\hat{\mathcal{H}}^{\mathsf{A}\Cup\mathsf{B}}_{\mathbf{XYZ}}$, with nodes $\mathbf{X} = (X_1, X_2, X_3)'$, $\mathbf{Y} = Y_1, Y_2, Y_3)'$ and $\mathbf{Z} = (Z_1, Z_2, Z_3)'$. Compared to Figure 6 this MN has one more edge, which is between Y_3 and X_2 and highlighted in purple.

ing: $(Y_1, Y_2), (Y_2, Y_3, X_1, X_3), (Y_3, X_1, X_2, X_3), (X_1, X_2, Z_1), (X_2, Z_1, Z_2), (X_3, Z_3)$, while

- $\hat{\mathcal{H}}^{\mathsf{A}}_{\mathbf{XY}}$ contains: $(Y_1, Y_2), (Y_2, Y_3, X_1, X_3), (Y_3, X_1, X_2, X_3)$,
- $\hat{\mathcal{H}}^{\mathsf{B}}_{\mathbf{XZ}}$ contains: $(X_1, X_2, Z_1), (X_2, Z_1, Z_2), (X_3, Z_3)$,
- $\hat{\mathcal{H}}^{\mathsf{A}\Cup\mathsf{B}}_{\mathbf{X}}$ contains: $\emptyset$.

These two examples illustrate what Equation (4.3) means for estimating the joint probability. Furthermore I showed that the last factor of Equation (4.3) indeed can be irrelevant.

This chapter is a central part of this thesis: It shows how probabilistic graphical models can be utilized for the aim of statistical matching. In the first part of this chapter I presented the work of Endres and Augustin (2016), which is built on the theory of BNs. We have seen that the concept of d-separation in BNs fits the CIA in statistical matching, and that we can estimate a joint probability distribution with BNs even though missing data appears. In the second part I moved on to utilizing MNs for the aim of statistical matching. MNs present certain independence structures as well, which also fit the assumptions of statistical matching. We have seen that it is possible to construct a MN that fulfills the CIA in statistical matching, and that we can represent a joint probability distribution in MNs identifiable with the blocks of missing values in dataset A and B. Note that the CIA is the brick that everything is built on, and which brings probabilistic graphical models and statistical matching together.

In Chapter 7 of this thesis I will give a discussion of using probabilistic graphical models for statistical matching, and also compare BNs and MNs for this aim. For now, I will move on to the estimation of the structure of MNs.

5 Estimation of Markov Networks for Binary Variables

The aim of this chapter is to present the Ising model, which enables fitting a MN for binary data. This model is very relevant for this thesis, since later on statistical matching will be performed with binary data. After explaining the Ising model, I will go on with estimation of the model's parameters, and selecting the best model via the extended Bayesian information criterion.

For reasons of simplicity, in this chapter I will restrict myself to just one block of random variables $\mathbf{X} = (X_1, \ldots, X_p)'$, with $X_i \in \{1, 0\}, i = 1, \ldots, p$. Furthermore $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ is an undirected graph with a set of vertices $\mathcal{V} = \{X_1, \ldots, X_p\}$. $\mathcal{E}$ is a set of edges, whose elements are unordered pairs of nodes, indicated by tuples of two nodes $(X_i, X_h),\ i \neq h$, with $X_i, X_h \in \mathcal{V}$ and $i, h = 1, \ldots, p$.

5.1 The Ising Model

Most times in analyzing real data, the graph structure of $\mathcal{H}$ is unknown and has to be explored via an appropriate data model. Especially for binary data, finding the neighborhood of each node is not obvious.[15] Nevertheless, by fitting an Ising model the neighborhood of each node can be found step by step, and thus the complete graph structure is revealed.

The Ising model was one of the first types of MNs, and has its origin in statistical physics. In the original setting, the Ising model is used to describe ferromagnetic processes in solids; the two possible values of each node are $\{-1, +1\}$, which tell the direction of an atom's spin. Even though the original Ising model was constructed for ferromagnetism, it generalizes to all kinds of binary variables (Ravikumar et al. 2010, pp. 1287ff., and Koller and Friedman 2009, p. 126).[16]

In our case of binary data $X_1, \ldots, X_p \in \{0, 1\}$ we actually work with the energy of the so-called Boltzmann distribution. In literature the theory of the Ising model and Boltzmann distribution is often mixed, as some authors work with the original Ising model assuming that $X_1, \ldots, X_p \in \{-1, +1\}$, and others work directly with the Boltzmann machine which assumes $X_1, \ldots, X_p \in \{0, 1\}$. In the end, both variants have the same energy.[17]

[15] In the case of Gaussian data, the covariance matrix has to be inverted. This inversion is called P, the precision matrix. Entries that aren't zero point out an edge between the respective vertices.

[16] For $X \in \{0, 1\}$ the distribution is often referred to as *Boltzmann distribution.* What is special about the Ising model, or the Boltzmann distribution, is the fact that it is not working with maximal cliques but rather with pairwise nodes (often referred to as pairwise Markov networks). More information about the Ising model and the Boltzmann distribution can be found in Appendix C.

[17] Appendix C shows the connection of both.

I will start with estimating a MN for binary data, for which it is essential to understand the procedure of the Ising model. A popular notation while working with MNs is the so-called energy function ϵ. A factor $\phi(X_i, X_h)$, with $i, h = 1, \ldots, p$ and $i \neq h$ can also be expressed as energy ϵ:

$$\phi(X_i, X_h) = \exp(-\epsilon(X_i, X_h)), \quad \text{with } \epsilon(X_i, X_h) = -\ln(\phi(X_i, X_h)). \tag{5.1}$$

Formulating this as the product over all factors results in

$$\prod_{x_i \sim x_h} \phi(x_i, x_h) = \exp\left(- \sum_{(x_i, x_h) \in \mathcal{E}} \epsilon(x_i, x_h) \right). \tag{5.2}$$

For the Boltzmann machine the overall energy is defined by the Hamiltonian function $H(x)$ (Loh and Wainwright 2013, pp. 3024ff., and van Borkulo and Epskamp 2014, pp. 1ff.), which is

$$H(x) = -\left(\sum_{i}^{p} \tau_i x_i + \sum_{(x_i, x_h) \in \mathcal{E}} \beta_{ih} x_i x_h \right). \tag{5.3}$$

By putting together the definition of the energy in Equation(5.1), the definition of the factor product in Equation(5.2), overall energy of the Boltzmann machine in Equation (5.3), and the definition of the Gibbs distribution in Equation (3.10), this leads to

$$\begin{aligned} \tilde{P}(X) &= \prod_{x_i \sim x_h} (\exp(-\epsilon(x_i, x_h))) \\ &= \exp(-(H(x)) \\ &= \exp\left(\sum_{i}^{p} \tau_i x_i + \sum_{(x_i, x_h) \in \mathcal{E}} \beta_{ih} x_i x_h \right). \end{aligned} \tag{5.4}$$

Since in the literature some steps are usually skipped to reach the unnormalized measure, I start at the very beginning and develop it step by step. As $\tilde{P}(\mathbf{X})$ is only an unnormalized measure for now, we need to consider the partition function ζ next, which sums up all possible compositions of $\mathbf{X}$. Furthermore ζ is a function of our parameters τ and β, which are summarized in a matrix $\mathbf{\Theta} \in \mathbb{R}^{p \times p}$:

$$P_{\mathbf{\Theta}}(\mathbf{X}) = \frac{1}{\zeta} \exp\left(\sum_{i}^{p} \tau_i x_i + \sum_{(x_i, x_h) \in \mathcal{E}} \beta_{ih} x_i x_h \right). \tag{5.5}$$

A very interesting point about the Ising model is that the factorization can also be represented in terms of an exponential family,[18] which is associated with the clique structure of $\mathcal{H}$. For binary $X_i \in \{0, 1\}, i = 1, \ldots, p$, one can associate with each clique C – whether maximal or not – a sufficient statistic $\mathbb{I}_C C(X) := \prod_{X_i \in C} X_i$. Only if $X_i = 1$ for all $X_i \in C$, $\mathbb{I}_C C(X) = 1$. In form of the exponential family, the sufficient statistic is weighted by the natural parameter $\theta_C \in \mathbb{R}$, and allows to rewrite Equation (5.5) to

$$P(\mathbf{X}) = \exp\left\{\sum_{C \in \mathcal{C}} \theta_C \ \mathbb{I}_C C(X) - \ln(\theta)\right\}, \tag{5.6}$$

with $\mathcal{C}$ being the set of all cliques in $\mathcal{H}$, and $\ln(\theta)$ being the normalization constant. In the case of pairwise interaction, Equation (5.6) reduces to Equation (5.5), the Ising model, since each $C(X)$ consists of two nodes (Loh and Wainwright 2013, pp. 3025ff.).

At this point we are confronted with the problem that the computation can very quickly get very extensive. Since ζ is the sum of all possible configurations, it is hard to calculate, as the set of all possible configurations increases exponentially – this is challenging especially for high-dimensional data. E.g. for 12 variables, ζ has to sum up $2^{12} = 4096$ possible configurations. And 12 variables is still a small number.

A time saving and convincing concept is to work with conditional probabilities, and estimation of the 'pseudo likelihood' (Besag 1972), which is the likelihood of the conditional probability X_i given all other nodes $\mathbf{X}_{-i}$.[19] ζ now has to sum up only two possible states of X_i, which are 0 and 1. Hence by working with the conditional probability $P(X_i \mid \mathbf{X}_{-i})$, for ζ we get

$$\begin{aligned}
\zeta(\mathbf{\Theta}) &= \sum_{x_i \in \{0,1\}} \exp\left(\tau x_i + \sum_{(x_i, x_h) \in \mathcal{E}} \beta_{ih} x_i x_h\right) \\
&= \exp\left(\tau_0 \cdot 0 + \sum_{(x_i, x_h) \in \mathcal{E}} \beta_{0h} \cdot 0 \cdot x_h\right) \\
&\quad + \exp\left(\tau_1 \cdot 1 + \sum_{(x_i, x_h) \in \mathcal{E}} \beta_{1h} \cdot 1 \cdot x_h\right) \\
&:= 1 + \exp\left(\tau_i \sum_{(x_i, x_h) \in \mathcal{E}} \beta_{ih} x_h\right).
\end{aligned} \tag{5.7}$$

[18] More information about the exponential family in general can be found in (Brown 1986).

[19] The notation $-i$ in the index implies that the i-th node is excluded.

With this partition function ζ the whole computation of the joint probability distribution simplifies tremendously. We now have

$$P_{\boldsymbol{\Theta}}(X_i \mid \mathbf{X}_{-i}) = \frac{\exp\left(\tau_i \sum_{(x_i,x_h)\in\mathcal{E}} \beta_{ih}x_h\right)}{1+\exp\left(\tau_i \sum_{(x_i,x_h)\in\mathcal{E}} \beta_{ih}x_h\right)}; \tag{5.8}$$

if we substiute $\eta = (\tau_i \sum_{(x_i,x_h)\in\mathcal{E}} \beta_{ih}x_h)$, Equation (5.8) can be rewritten to

$$P_{\boldsymbol{\Theta}}(X_i \mid \mathbf{X}_{-\mathbf{i}}) = \frac{\exp(\eta)}{1+\exp(\eta)}, \tag{5.9}$$

which is just the popular logistic regression model[20] (Fahrmeir et al. 2009, p. 192).

But how can we obtain the structure of a MN? The logistic regression model of Equation (5.8) has to be fitted $p = |\mathcal{V}|$ times, thus each variable is used once as response variable, with all remaining variables as predictors. This is the key to finding the structure of a MN for binary data. For $\beta_{ih} = \beta_{hi} = 0$, X_i and X_h are conditionally independent. Thus in a MN graph $\mathcal{H}$, the nodes X_i and X_h are not connected. If this is not the case we have to distinguish between several options: if both $\beta_{ih} \neq 0$ and $\beta_{hi} \neq 0$ it is obvious that X_i and X_h are connected in $\mathcal{H}$. If just one of the two β_{ih} and β_{hi} is unequal to zero, it is up to the applicant to decide. The AND-rule dictates that both βs have to be unequal to zero for nodes X_i and X_h to be connected in $\mathcal{H}$, whereas the OR-rule requires only one of the β_{ih}, β_{hi} to be unequal to zero. With this procedure we obtain the neighborhood for each node of $\mathcal{V}$, and thus a complete MN graph $\mathcal{H}$ (van Borkulo 2017, pp. 22ff.). As we are in the context of MN theory, and the independence structure holds also for the Ising model, we can state that $P_{\boldsymbol{\Theta}}(X_i \mid \mathbf{X}_{-\mathbf{i}}) = P_{\boldsymbol{\Theta}}(X_i \mid X_{Nei(i)})$.

As mentioned in the context of Equation (5.5), $\boldsymbol{\Theta}$ contains $\boldsymbol{\tau}$ and $\boldsymbol{\beta}$. The interpretation of β_{ih} is straightforward: it is a value for the effect of variable X_h on variable X_i. More precisely, in the MN context it can be interpreted as the connection strength between nodes X_i and X_h. τ_i is a so-called threshold of variable X_i. For our purposes, the interpretation of the threshold is not very meaningful, as it actually is the threshold for activation of a neuron. Only if the variable X_i reaches an energy of $-\tau_i$ the variable can be activated. Thus the interpretation is meaningful in areas like physics or chemistry, but not necessarily elsewhere.

The pseudo likelihood approach is a good option for estimation of the graph structure. Furthermore Gidas (1988) showed the consistency of the pseudo likelihood estimation for Gibbs distributions, which are generally used for MNs. Murphy (2012) summarizes that the pseudo likelihood approach is applicable for most kinds of data

[20] A short explanation of logistic regression models in general can be found in Appendix D.

(there are no restrictions on decomposable/chordal graphs, etc.),[21] except hidden MNs with hidden variables.

5.2 Lasso Estimation for Ising Models

Most times in analyzing data, the real MN structure is unknown. For that reason a logistic regression is fitted for each variable X_i, with the remaining variables as predictors (see Chapter 5.1). For estimation of $\mathbf{\Theta}$, the ℓ_1-regularized logistic regression, or *lasso*,[22] is used for the pseudo likelihood of Equation (5.8).[23] Thus also for parameter estimation, we benefit from the fact that the partition function is greatly simplified by using conditional probabilities.

Before discussing the actual estimation equation, I want to explain the idea of lasso estimation, which is used to optimize the neighborhood selection by shrinkage of the β-parameters towards zero. Lasso is a shrinkage method[24] for the estimation of the coefficients of a model. The estimation of β is restricted by the ℓ_1 penalty term

$$\sum_{x_h \in \mathcal{V}_{-i}} |\beta_{ih}| \leq t. \tag{5.10}$$

For t being small enough, some of the coefficients will be shrunk towards zero. For optimization we first need the likelihood of all nodes X_i over all independent observations $s = 1, \ldots, n$:

$$\begin{aligned} L(x_i \mid \mathbf{x}_{-i}; \mathbf{\Theta}_{\mathbf{X_i}}) &= \prod_{s=1}^{n} \frac{\exp(\eta_{is})}{1+\exp(\eta_{is})} \\ &= \frac{\prod_{s=1}^{n} \exp(\eta_{is})}{\prod_{s=1}^{n} 1+\exp(\eta_{is})} \\ &= \frac{\exp(\sum_{s=1}^{n} \eta_{is})}{\prod_{i=1}^{n} 1+\exp(\eta_{si})}. \end{aligned} \tag{5.11}$$

Thus for the log-likelihood with lasso regularization, we get

$$l_i^{lasso}(x_i \mid \mathbf{x}_{-i}; \mathbf{\Theta}_{\mathbf{X_i}}) = \sum_{s=1}^{n} \eta_{is} - \sum_{s=1}^{n} \ln\left(1+\exp(\eta_{is})\right) + \lambda \sum_{x_h \in \mathcal{V}_i} \beta_{ih}. \tag{5.12}$$

[21] In a chordal graph all cycles with at least four nodes have an additional edge that connects two nodes of the cycle. Note that the additional edge is not part of the cycle (Koller and Friedman 2009, p. 38).

[22] Least Absolute Shrinkage and Selection Operator.

[23] The *lasso* for MNs was proposed by Ravikumar et al. (2010), although first popularized by Besag (1972; 1974).

[24] More information on lasso estimation can be found in Appendix E.

In this equation, $\lambda > 0$ is the penalty parameter, which ensures shrinkage of the coefficients (Tibshirani 1996, pp. 268ff.). For choosing the best penalty parameter, the implementation e.g. with the `R`-package `glmnet` (Friedman et al. 2018) can be used. `glmnet` uses a range of 100 penalty parameters at most. It is obvious that with varying penalty parameters we obtain different vectors for $\boldsymbol{\beta}$ and $\boldsymbol{\tau}$, and thus different neighborhoods for each node. Nevertheless we have to decide for one $\boldsymbol{\beta}$ and $\boldsymbol{\tau}$ and hence for one graph structure $\mathcal{H}$ of the MN (van Borkulo and Epskamp 2014, pp. 4ff., and Ravikumar et al. 2010, pp. 1309ff.). A criterion for deciding on the best neighborhood is the extended Bayesian information criterion (EBIC), which I will introduce in the next chapter.

5.3 The Extended Bayesian Information Criterion

Beside the often used Akaike information criterion, another often used criterion is the Bayesian information criterion (BIC) by Schwarz (1978). The BIC's goal is to find the right dimension of a model, given a set of observations. The BIC solves this problem with approximate Bayesian statistics. Chen and Chen (2008) point out that for most applications of MNs the ordinary BIC is too liberal though, resulting in an optimal model that has too many spurious covariates. This is the case especially for high-dimensional data with $p \gg n$. Even though I will later implement a MN for the aim of statistical matching with $p < n$, I need to explain the EBIC in more detail. The criterion is still a good choice for model selection, while the `R`-package `IsingFit` (van Borkulo et al. 2016) I use for implementation works also with the EBIC.

In the context of MNs for binary data, the model that minimizes

$$\text{EBIC}_{\gamma}(h) = -2\ell(\hat{\boldsymbol{\Theta}}_{\mathbf{i}}) + |Nei(X_i)| \cdot \ln(n) + 2\gamma|Nei(X_i)| \cdot \ln(p-1) \tag{5.13}$$

is optimal. In this equation we are confronted with several new notations. $\ell(\hat{\boldsymbol{\Theta}}_{\mathbf{i}})$ is the well known log-likelihood for Equation (5.8), which is

$$\ell(x_i|\mathbf{x}_{-i}; \boldsymbol{\Theta}_{\mathbf{X_i}}) = \sum_{s=1}^{n} \eta_{is} - \sum_{s=1}^{n} \ln\left(1 + \exp(\eta_{is})\right). \tag{5.14}$$

Furthermore $|Nei(X_i)|$ is the number of neighbors of node X_i, which is selected by the logistic regression model for varying λ. And γ is a hyperparameter, with $\gamma \in [0, 1]$.

In Equation (5.13) it can be seen that the term $2\gamma|Nei(X_i)| \cdot \ln(p-1)$ penalizes for both a growing number of covariates $p-1$, and a growing number of adjacent nodes for X_i. Thus γ determines the strength of prior information on the size of the model space. For $\gamma = 0$ the network will be maximally extensive, resulting in relatively many connections. Vice versa, for $\gamma = 1$ the model will have far less, if any connections between the nodes (van Borkulo 2017, pp. 5ff.).

Note that for $\gamma = 0$ the ordinary BIC is obtained, since the BIC is a special case of the EBIC. Thus for the following implementation, in which $p < n$, we would also have the option to set γ to zero and work with the ordinary BIC.

The EBIC is an appropriate option to choose the best neighborhood. In recent studies, the EBIC has proven good trade-off of positive selection rates (proportions of true selected edges), and false discovery rates (proportions of false positives among the selected edges) for the Ising model (Barber and Drton 2015, pp. 9ff.).

In this chapter I prepared the theory for the implementation of MNs. Since I will apply statistical matching for binary data, I presented the Ising model (or the Boltzmann machine), which is an appropriate choice for MNs for binary data. As literature is not very consistent in this regard, I developed Equation (5.8) step by step from the beginning. I also gave a short explanation of how to derive the structure of a MN from the Ising model, and therefore presented the AND-rule and OR-rule. Moreover I explained how to estimate the parameter of the model with lasso estimation, which is a shrinkage method. Since we can choose between a set of possible shrinkage parameters, I also introduced the EBIC, an extended version of the ordinary and popular BIC.

6 Applying Markov Networks for Statistical Matching

An important part of this thesis is the application of statistical matching with MNs. This chapter has the aim of explaining how the application can take place, in which order one has to proceed for the application, and the appropriate software that deals with MNs. Before starting with the application, we need data to work with. In this thesis I will not work with already existing data, but rather I will simulate data to review the results. For application I use the statistical programming software `R` (R Core Team 2008) in the latest available version 3.5.1.

6.1 Data Simulation

For applying MNs for statistical matching I will work with categorical, more precisely binary data. Because of the data situation of statistical matching we have to deal with three blocks of variables $\mathbf{X} = (X_1, \ldots, X_p)'$, $\mathbf{Y} = (Y_1, \ldots, Y_q)'$ and $\mathbf{Z} = (Z_1, \ldots, Z_r)'$. It holds that $X_i \sim \text{Bin}(n = n_\mathsf{A} + n_\mathsf{B}, \pi_i)$ with $i = 1, \ldots, p$, $Y_j \sim \text{Bin}(n_\mathsf{A}, \pi_j)$ with $j = 1, \ldots, q$, and $Z_k \sim \text{Bin}(n_\mathsf{B}, \pi_k)$ with $k = 1, \ldots, r$. For the simulation I will work with four variables for each block, thus $p = 4$, $q = 4$ and $r = 4$.

A main challenge for simulation of the data is the assumption that $\mathbf{Y} \perp \mathbf{Z} \mid \mathbf{X}$, since the theory I presented up to this point is built on CI. Hence I cannot simulate independent binary values for $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$, but rather I have to consider a special correlation structure that guarantees CI. For simulating the data we first have to construct a correlation matrix $\mathbf{R}$.

6.1.1 The Correlation Matrix R

Generally a correlation matrix $\mathbf{R}$ is not just a $p \times p$ matrix for p variables, which contains values between $[-1, 1]$. It rather has to fulfill certain algebraic characteristics:

- a correlation matrix $\mathbf{R}$ has to be symmetric, thus $\mathbf{R} = \mathbf{R}^\top$,
- the elements of the diagonal all equal 1, since the diagonal elements contain the correlation of variable X_i, X_i, with $i = 1, \ldots, p$, and
- $\mathbf{R}$ has to be a positive definite matrix: a symmetric matrix $\mathbf{R}$ is positive definite iff all eigenvalues of the matrix are greater than zero. A correlation matrix has to be positive definite because the correlation matrix is calculated

from the covariance matrix $\mathbf{\Sigma}$, by

$$\mathbf{R} = (\mathrm{diag}(\mathbf{\Sigma}))^{-\frac{1}{2}} \cdot \mathbf{\Sigma} \cdot (\mathrm{diag}(\mathbf{\Sigma}))^{-\frac{1}{2}}, \tag{6.1}$$

with diag() being the diagonal elements of $\mathbf{\Sigma}$. Equation (6.1) holds only if $\mathbf{R}$ is positive definite (Bilodeau and Brenner 1999, pp. 8ff.).

Only if all these characteristics are fulfilled we can work with $\mathbf{R}$ for further variable simulation.

But we have further constraints for $\mathbf{R}$: With the help of $\mathbf{R}$ we want to simulate data that fulfill the CIA. For that reason we need some special, blockwise structure in $\mathbf{R}$:

- the variables of block $\mathbf{X}$ should by highly correlated with the variables of block $\mathbf{Y}$,
- the variables of block $\mathbf{X}$ should be highly correlated with the variables of block $\mathbf{Z}$, and
- the variables of block $\mathbf{Y}$ should be barely correlated with the variables of block $\mathbf{Z}$.

It is not trivial to combine all these requirements for the correlation matrix. In my own application, after trying to construct a correlation matrix with values between $[-1, 1]$ without success,[25] I limited the values of correlation to positive values, thus each element of $\mathbf{R}$ is a value of the interval $[0, 1]$. With this constraint it was much easier to create a correlation matrix with the required structure and that also fulfills the algebraic characteristics of a correlation matrix, especially being positive definite. I constructed the correlation matrix with the statistical software `R` (R Core Team 2008), with Table 1 showing the result for $\mathbf{R}$.

As we can see, all variables $X_1, \ldots, X_4$ have a correlation of 0.4, furthermore $\mathbf{X}$ is correlated with both $\mathbf{Y}$ and $\mathbf{Z}$ with 0.8, and also importantly the correlation between $\mathbf{Y}$ and $\mathbf{Z}$ is very close to zero with a value of 0.02.

6.1.2 Simulation of Correlated Variables $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$

The data simulation is done via `R`, and the `R` package `bindata` (Leisch et al. 2011). The main function of this package is `rmvbin()`. This function works in my case with three arguments. The first argument `n` determines how many realizations of the variables should be simulated. In my case I required 500 realizations for each

[25] To simulate the special structure of the correlation matrix was not the problem, but as soon as I transformed it to a positive definite matrix, the correlation structure was gone.

	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	Z_1	Z_2	Z_3	Z_4
X_1	1.0	0.4	0.4	0.4	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
X_2	0.4	1.0	0.4	0.4	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
X_3	0.4	0.4	1.0	0.4	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
X_4	0.4	0.4	0.4	1.0	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
Y_1	0.8	0.8	0.8	0.8	1.0	0.4	0.4	0.4	0.02	0.02	0.02	0.02
Y_2	0.8	0.8	0.8	0.8	0.4	1.0	0.4	0.4	0.02	0.02	0.02	0.02
Y_3	0.8	0.8	0.8	0.8	0.4	0.4	1.0	0.4	0.02	0.02	0.02	0.02
Y_4	0.8	0.8	0.8	0.8	0.4	0.4	0.4	1.0	0.02	0.02	0.02	0.02
Z_1	0.8	0.8	0.8	0.8	0.02	0.02	0.02	0.02	1.0	0.4	0.4	0.4
Z_2	0.8	0.8	0.8	0.8	0.02	0.02	0.02	0.02	0.4	1.0	0.4	0.4
Z_3	0.8	0.8	0.8	0.8	0.02	0.02	0.02	0.02	0.4	0.4	1.0	0.4
Z_4	0.8	0.8	0.8	0.8	0.02	0.02	0.02	0.02	0.4	0.4	0.4	1.0

Table 1: Correlation matrix **R**, which is used to simulate data and which has the required structure of CI. Columns and rows 1 to 4 represent **X**, columns and rows 5 to 8 represent **Y**, and columns and rows 9 to 12 represent **Z**. Green values indicate a high correlation, red values indicate low correlation. It can be seen that the correlation structure is fulfilled.

variable. The second argument is `bincor`, which is a matrix of $p \times p$, with p being the number of variables. Since I have in total 12 variables, four for each block of variables, `bincor` is a 12×12 matrix which determines the correlation structure. Thus I set `bincor = `**R**. The last argument of the function `rmvbin()` to be defined is `margprob`. `margprob` is a vector of length p, in our case of length 12. The i-th element of this vector contains the probability of X_i being 1, which is the marginal probability. In my correlation matrix there is a range of $[0.02, 0.8]$ for correlation. To find a marginal probability that fits this range of correlation, I used the function `corrcheck()` of the package `GenOrd` (Barbiero and Ferrari 2015). Since the variance of a binomial distribution, which is $n\pi(1-\pi)$, is maximal for $\pi = 0.5$, a large range of possible correlation is obtained for a marginal probability of e.g. $\pi = 0.6$. I checked this with the function `corrcheck()`. Thus I produced a vector of length 12 with the value 0.6 on each entry and used this vector for the argument `margprob`.

The result of function `rmvbin()` was a data matrix of dimension 500×12, containing 500 realizations for each variable. This simulation procedure was repeated 100 times, each time with a different starting value for simulation, which was defined in `R` with the function `set.seed()`. As result I got an `R`-object of type `list`, in which each element was a dataset with 500 realizations for each of the 12 variables. However, the correlation structure used for simulation was the same for every dataset, as mapped in Table 1.

After I obtained the simulated data, I checked whether the correlation structure of **R** really was transmitted to the simulated data. Table 2 contains the correlation coefficients of Spearman, exemplarily for the simulated data of one dataset. As can

	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	Z_1	Z_2	Z_3	Z_4
X_1	1	0.750	0.713	0.754	0.532	0.588	0.48	0.518	0.512	0.465	0.494	0.568
X_2	0.750	1	0.716	0.708	0.543	0.549	0.506	0.503	0.505	0.491	0.513	0.543
X_3	0.713	0.716	1	0.704	0.507	0.538	0.506	0.510	0.545	0.525	0.553	0.551
X_4	0.754	0.708	0.704	1	0.517	0.563	0.492	0.512	0.521	0.521	0.512	0.538
Y_1	0.532	0.543	0.507	0.517	1	0.601	0.596	0.54	0.247	0.195	0.178	0.255
Y_2	0.588	0.549	0.538	0.563	0.601	1	0.568	0.545	0.262	0.186	0.252	0.288
Y_3	0.480	0.506	0.506	0.492	0.596	0.568	1	0.530	0.174	0.177	0.223	0.203
Y_4	0.518	0.503	0.510	0.512	0.540	0.545	0.530	1	0.232	0.222	0.222	0.231
Z_1	0.512	0.505	0.545	0.521	0.247	0.262	0.174	0.232	1	0.492	0.512	0.553
Z_2	0.465	0.491	0.525	0.521	0.195	0.186	0.177	0.222	0.492	1	0.491	0.509
Z_3	0.494	0.513	0.553	0.512	0.178	0.252	0.223	0.222	0.512	0.491	1	0.542
Z_4	0.568	0.543	0.551	0.538	0.255	0.288	0.203	0.231	0.553	0.509	0.542	1

Table 2: Matrix containing the correlation coefficients of Spearman for one simulated dataset. Each value is rounded to three decimals. By comparing this matrix to the real correlation matrix **R**, we can see that the required structure is fulfilled particularly for blocks with small correlation.

be seen, the small correlation between $\mathbf{Y}$ and $\mathbf{Z}$ was indeed transmitted, but the high correlation between $\mathbf{X}$ and $\mathbf{Y}$, and $\mathbf{X}$ and $\mathbf{Z}$ was slightly scaled down in the simulated data. This is no problem, however, since the blockwise correlation structure is still retained. Moreover this is just one example, and the empirical correlation structure varies among the 100 datasets.

Up to now I have 100 datasets without any missings at all. Since in statistical matching it is assumed that the data is missing completely at random, for each row I randomly deleted either all columns of $\mathbf{Y}$ or of $\mathbf{Z}$. The result is a data frame in which blockwise missings of $\mathbf{Y}$ or $\mathbf{Z}$ occurs. The first 11 rows of one dataset can be seen in Table 3. Now I can construct datasets A and B, where A contains all the rows in which $\mathbf{Z}$ is missing, and B all the rows where $\mathbf{Y}$ is missing, with $n_{\mathsf{A}} = n_{\mathsf{B}} = 250$. Again, this step is repeated for all 100 datasets.

id	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	Z_1	Z_2	Z_3	Z_4
1	1	1	1	1	1	1	1	1	NA	NA	NA	NA
2	0	0	0	0	0	0	0	0	NA	NA	NA	NA
3	1	1	1	1	NA	NA	NA	NA	1	1	1	1
4	0	0	0	0	0	0	0	0	NA	NA	NA	NA
5	1	1	1	1	0	1	1	1	NA	NA	NA	NA
6	1	1	1	1	NA	NA	NA	NA	1	1	1	1
7	1	1	1	1	NA	NA	NA	NA	1	1	1	1
8	0	0	1	1	NA	NA	NA	NA	1	1	1	1
9	1	1	1	1	1	1	1	1	NA	NA	NA	NA
10	0	0	0	0	1	1	1	0	NA	NA	NA	NA
11	1	1	1	1	1	1	1	1	NA	NA	NA	NA

Table 3: Extract of one simulated data frame. In each row either $Y_1, \ldots, Y_4$ is missing, or $Z_1, \ldots, Z_4$. $X_1, \ldots, X_4$ is available in each row, since it is observed in both datasets A and B. The red rows belong to dataset A, the yellow rows to dataset B.

6.2 Estimation of the Structure of Markov Networks with the `R` Package `IsingFit`

With the simulation of the data succeeded, I can move on to estimating the three MNs, which are $\hat{\mathcal{H}}^{\mathsf{A}\Cup\mathsf{B}}_{\mathbf{X}}$, $\hat{\mathcal{H}}^{\mathsf{A}}_{\mathbf{XY}}$ and $\hat{\mathcal{H}}^{\mathsf{B}}_{\mathbf{XZ}}$, and thus $\hat{\mathcal{H}}^{\mathsf{A}\Cup\mathsf{B}}_{\mathbf{XYZ}}$ for the simulated datasets A and B. For estimation of the MNs I used the `R`-package `IsingFit` (van Borkulo et al. 2016). The core function of this package is called like the package itself, `IsingFit()`. This function implements everything I explained in Chapter 5: each vector $\mathbf{x}, \mathbf{y}, \mathbf{z}$ is once used as response variable in Equation (5.8), with the remaining variables as predictors. For estimation of the model parameter $\mathbf{\Theta}$, the ℓ_1 regularization is used to obtain the estimates. In the end the model criterion EBIC is used to find the best model.

The function `IsingFit()` comes along with several arguments. First of all, we have to hand over the relevant data via the argument `x`. Furthermore we have to decide which rule we use to receive the neighborhood for each node, thus we have to decide for the argument `AND` being either `TRUE` or `FALSE`.

For the estimation of the structure I will proceed as explained in Chapter 4.2. Thus I begin with $\hat{\mathcal{H}}_{\mathbf{X}}^{\mathsf{A} \uplus \mathsf{B}}$, which has to be the first step, since we need this graph structure as prior knowledge for the two remaining steps. For all $n = 500$ realizations of $\mathbf{X}$ I received an `IsingFit`-object, which I named `x_structure`. This is repeated for all 100 datasets. For the dataset of which Table 3 showed an extract, the corresponding information object `x_structure` is shown in Table 4.

```
> x_structure
```

```
Estimated network:
```

	X_1	X_2	X_3	X_4
X_1	0.00	2.15	1.48	2.33
X_2	2.15	0.00	1.88	1.42
X_3	1.48	1.88	0.00	1.71
X_4	2.33	1.42	1.71	0.00

```
Estimated Thresholds:
```

X_1	X_2	X_3	X_4
-2.79	-2.41	-2.34	-2.86

```
> x_structure$lambda.values
```

X_1	X_2	X_3	X_4
0.002	0.002	0.002	0.002

Table 4: Values that are included in an object of class `IsingFit`, which is here named `x_structure`. This `R`-output shows the results of the `IsingFit`-function for the variable block $\mathbf{X}$. We can see the estimates of β (estimated network), the estimates of τ (estimated thresholds), and the λ-values that are used by the EBIC for estimation of the structure between $X_1, \ldots, X_4$.

For one of the simulated datasets I want to explain the results in more detail. For the remaining datasets the results can be seen in the programmed code, which is part of the Digital Appendix G.

As the resulting MN structure of $\mathbf{X}$, we obtained for the βs that each $\beta_{ih} > 0$, with $i \neq h$ and $i, h = 1, \ldots, 4$. Thus all nodes for the variables of $\mathbf{X}$ are connected with each other. Furthermore we get the values $\tau_i < -1$, with $i = 1, \ldots, 4$. E.g. for node X_1, $\tau_1 = -2.79$, thus

$$\frac{\exp(-2.79)}{1 + \exp(-2.79)} = 0.058,$$

meaning that π_1 has to be greater than or equal to 0.058, otherwise X_1 cannot be activated.[26] In our case $\hat{\pi}_1 = 0.614$. For the remaining nodes, the activation threshold is about the same size. It is also possible to take a look at the chosen λ-values of the lasso estimation. These values are chosen by the EBIC, with $\gamma = 0.25$. The values of λ can be seen in Table 4 under `x_strucutre$lambda.values`, and minimize Equation (5.13) for this dataset. The network which is constructed by putting together the estimated neighborhoods for each node $X_1, \ldots, X_4$ is presented in Figure 8.

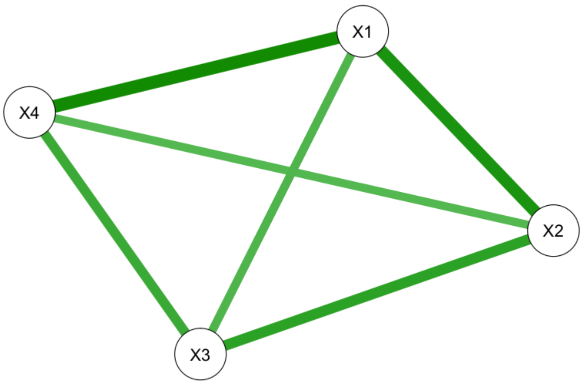

Figure 8: Estimated Markov network for the block of common variables $\mathbf{X}$. Each variable is presented with a node of the same name as the variable. The graph $\hat{\mathcal{H}}_{\mathbf{X}}^{\mathsf{A} \uplus \mathsf{B}}$ contains one maximal clique, which is (X_1, X_2, X_3, X_4).

The structure of $\hat{\mathcal{H}}_{\mathbf{X}}^{\mathsf{A} \uplus \mathsf{B}}$ is used in the following two steps as prior knowledge, and thus the connections between the nodes that represent $\mathbf{X}$ stay the same as in Figure 8. The next MN structure is estimated for both $\mathbf{X}$ and $\mathbf{Y}$, but only with the data of A, to receive graph $\hat{\mathcal{H}}_{\mathbf{XY}}^{\mathsf{A}}$. The result of the `IsingFit`-function is contained in the variable `xy_structure`; Table 5 shows the values of this object.

For the estimate values, several βs beside the diagonal are zero, hence between the respective nodes there are no connections. The MN for these estimates can be seen in Figure 9.

The last step of the estimation of the MN structure is based on dataset B, and

[26] However, this meaning of the interpretation is not very useful in our case – see Chapter 5.1.

```
> xy_structure
```

```
Estimated network:
```

	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
X_1	0.00	1.73	0.90	1.70	0.00	0.98	0.00	0.00
X_2	1.73	0.00	1.61	1.20	0.78	0.00	0.27	0.00
X_3	0.90	1.62	0.00	1.82	0.43	0.00	0.99	0.00
X_4	1.70	1.20	1.82	0.00	0.00	0.42	0.00	1.04
Y_1	0.00	0.78	0.43	0.00	0.00	1.55	1.42	0.61
Y_2	0.98	0.00	0.00	0.41	1.55	0.00	0.90	1.15
Y_3	0.00	0.27	0.99	0.00	1.42	0.90	0.00	1.00
Y_4	0.00	0.00	0.00	1.04	0.61	1.15	1.00	0.00

```
Estimated Thresholds:
```

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
-1.99	-2.67	-2.83	-3.43	-2.23	-2.90	-1.66	-2.19

```
> xy_structure$lambdavalues
```

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
0.039	0.023	0.017	0.003	0.028	0.003	0.014	0.002

Table 5: Values that are included in an object of class `IsingFit`. Estimation with data of A, thus `xy_structure` contains all structure information for a MN with nodes $X_1, \ldots, X_4, Y_1, \ldots, Y_4$.

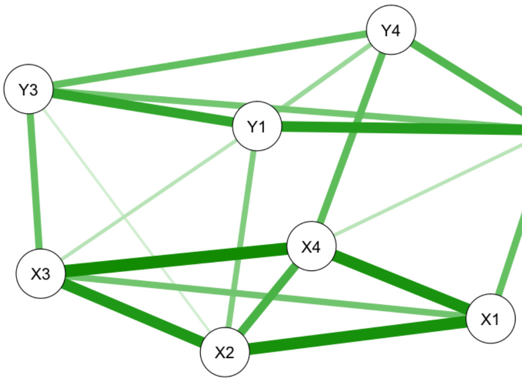

Figure 9: Markov network for the variables of $\mathbf{X}$ and $\mathbf{Y}$ of dataset A. It can be seen that the clique (X_1, X_2, X_3, X_4) is again present in $\hat{\mathcal{H}}^{\mathsf{A}}_{\mathbf{XY}}$. Also for $\hat{\mathcal{H}}^{\mathsf{A}}_{\mathbf{XY}}$ the clique (X_1, X_2, X_3, X_4) stays maximal.

thus creates the graph $\hat{\mathcal{H}}^{\mathsf{B}}_{\mathbf{XZ}}$. Again Equation (5.8) was implemented for the variables $\mathbf{X}$ and $\mathbf{Z}$ via the function `IsingFit`. The values of this function are now retained in a variable named `xz_structure`, and can be seen in Table 6.

```
> xz_structure

Estimated network:
```

	X_1	X_2	X_3	X_4	Z_1	Z_2	Z_3	Z_4
X_1	0.00	1.90	1.53	2.21	0.00	0.00	0.00	0.77
X_2	1.90	0.00	1.17	0.93	0.6	0.00	0.00	0.50
X_3	1.53	1.17	0.00	0.52	0.52	1.47	0.95	0.22
X_4	2.21	0.93	0.52	0.00	0.69	0.84	0.23	0.00
Z_1	0.00	0.60	0.52	0.69	0.00	1.04	1.05	0.74
Z_2	0.00	0.00	1.47	0.84	1.04	0.00	0.68	1.07
Z_3	0.00	0.00	0.95	0.23	1.05	0.68	0 00	1.00
Z_4	0.77	0.50	0.22	0.00	0.74	1.07	1.00	0.00

```
Estimated Thresholds:
```

X_1	X_2	X_3	X_4	Z_1	Z_2	Z_3	Z_4
-3.05	-2.22	-3.44	-2.7	-2.57	-1.93	-1.72	-1.99

```
> xz_structure$lambdavalues
```

X_1	X_2	X_3	X_4	Z_1	Z_2	Z_3	Z_4
0.012	0.011	0.001	0.021	0.002	0.006	0.004	0.002

Table 6: Estimated parameters for the Markov network structure based on data in B, thus `xz_structure` contains all structure information for a MN with nodes $X_1, \ldots, X_4, Z_1, \ldots, Z_4$.

Again we receive all connections between the variables of $\mathbf{X}$, which is important to be able to put together the different graphs in the last step. The corresponding MN $\hat{\mathcal{H}}^{\mathsf{B}}_{\mathbf{XZ}}$ is shown in Figure 10.

The last step that remains is to put together graphs $\hat{\mathcal{H}}^{\mathsf{A}}_{\mathbf{XY}}$ and $\hat{\mathcal{H}}^{\mathsf{B}}_{\mathbf{XZ}}$ to the graph $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{XYZ}}$. This step is rather straightforward, since both graphs have the same structure $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{X}}$ that is the clique (X_1, X_2, X_3, X_4). This procedure guarantees that there is no direct connection between any node of $\mathbf{Y}$ and $\mathbf{Z}$, so that the assumption of conditional independence $\mathbf{Y} \perp \mathbf{Z} \mid \mathbf{X}$ indeed holds. Figure 11 shows the complete graph. Once again, the whole estimation procedure is done for all 100 simulated datasets. The results presented are for the dataset shown in part in Table 3.

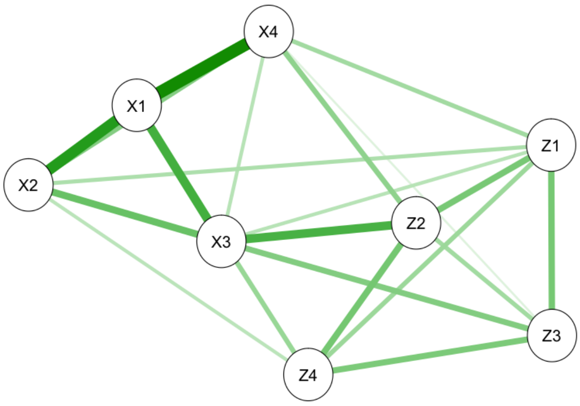

Figure 10: Markov network for the variables of $\mathbf{X}$ and $\mathbf{Z}$ of dataset B. It can be seen that the clique (X_1, X_2, X_3, X_4) is again present in $\hat{\mathcal{H}}^{\mathsf{B}}_{\mathbf{XZ}}$. Again the clique (X_1, X_2, X_3, X_4) stays maximal in $\hat{\mathcal{H}}^{\mathsf{B}}_{\mathbf{XZ}}$.

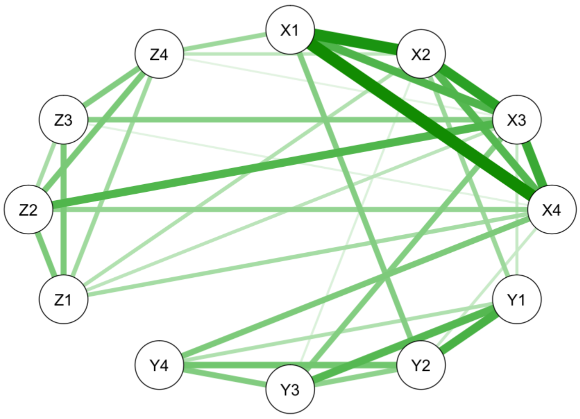

Figure 11: Markov network for the variables of $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ of dataset A and B. The prior knowledge of the clique (X_1, X_2, X_3, X_4) ensures the CIA. In the upper half of the graph are all four nodes of variable $\mathbf{Z}$. In the lower half of the graph are the nodes of variable $\mathbf{Y}$. Each connection between them has to pass at least one node of $\mathbf{X}$.

6.3 The Macro Approach: Estimation of the Joint Probability Distribution

After estimating the network structure it is now possible to estimate all elements of the joint probability distribution, which is described by its probability mass distribution $\hat{P}(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{Z} = \mathbf{z})$ for each dataset. For that aim we use the structure of graph $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{XYZ}}$. This network will be the foundation for the estimation of the joint probability. Up to this point I only have matrices containing the β-values for the structure of $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{X}}$, $\hat{\mathcal{H}}^{\mathsf{A}}_{\mathbf{XY}}$ and $\hat{\mathcal{H}}^{\mathsf{B}}_{\mathbf{XZ}}$. This information enables to construct a matrix containing all β-values of $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{XYZ}}$ though, since there is not any connection Y–Z.

This matrix is necessary to estimate the joint probability distribution. Hence the first task is to merge the matrices containing β-values of Tables 4, 5 and 6 to one matrix for the MN $\hat{\mathcal{H}}^{\mathsf{A} \uplus \mathsf{B}}_{\mathbf{XYZ}}$. This matrix contains β-values for X_i, $i = 1, \ldots, 4$, which can be seen for the example dataset in Table 4, for X_i, Y_j, $j = 1, \ldots, 4$, illustrated in Table 5, and for X_i, Z_k, $k = 1, \ldots, 4$, from Table 6, and additionally β-values between Y_j, Z_k which are all equal to zero, since they are not connected. This matrix is stored in an object I called `xyz_structure` and can be seen in Table 7, under the heading *Estimated network*.

Also required are the thresholds for each variable; Table 4 shows the thresholds for the variables $X_1, \ldots, X_4$, Table 5 shows the thresholds for $Y_1, \ldots, Y_4$, and Table 6 shows the thresholds for $Z_1, \ldots, Z_4$. The thresholds for the joint probability distribution are summarized in Table 7 under the heading *Estimated Thresholds*.

The object `xyz_structure` contains everything needed for the estimation of the joint probability distribution. For estimation of the joint probability function I used the `R`-package `IsingSampler` (Epskamp 2015). With the function `IsingLikelihood()` this package enables to estimate a joint probability distribution for the Ising model. For estimation, `IsingLikelihood` needs a matrix with all the β-values and a vector that contains the thresholds; in my case the object `xyz_structure` contains both. Since we have 12 variables in total, with each being either 0 or 1, the joint probability distribution $P(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{Z} = \mathbf{z})$ has $2^{12} = 4096$ probability components. An extract of the estimated joint probability distribution can be seen in Table 8. The sum of all probability components has to be 1, which is always the case. Again, the estimation of the joint probability distribution is applied for all 100 estimated Markov networks.

6.4 The Micro Approach: Estimation of the Missing Values in Dataset A and B

After estimating the joint probability distribution, the estimation of the missing values in A and B is the final step of the statistical matching procedure. The basis of estimating the missing values is the joint probability function of Chapter 6.3.

There are two ways to obtain estimates: *(1)* drawing concrete values from the joint probability distribution, or *(2)* for a given set of observations (thus for one row in the simulated data) the missing values are estimated by maximizing the probability, given the restriction of the observed data. If the missing observations of dataset A should be estimated, i.e. $\mathbf{z}$, we can draw values from the joint probability distribution with the restriction of the existing observations. For estimating the missing observations of dataset B, missing values for $\mathbf{y}$ have to be estimated, thus again we

```
> xyz_structure
```

```
Estimated network:
```

	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	Z_1	Z_2	Z_3	Z_4
X_1	0.00	2.15	1.48	2.33	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.77
X_2	2.15	0.00	1.88	1.42	0.78	0.00	0.27	0.00	0.60	0.00	0.00	0.50
X_3	1.48	1.88	0.00	1.71	0.43	0.00	0.99	0.00	0.52	1.47	0.95	0.22
X_4	2.33	1.42	1.71	0.00	0.00	0.42	0.00	1.04	0.69	0.84	0.23	0.00
Y_1	0.00	0.78	0.43	0.00	0.00	1.55	1.42	0.61	0.00	0.00	0.00	0.00
Y_2	0.98	0.00	0.00	0.42	1.55	0.00	0.90	1.15	0.00	0.00	0.00	0.00
Y_3	0.00	0.27	0.99	0.00	1.42	0.90	0.00	1.00	0.00	0.00	0.00	0.00
Y_4	0.00	0.00	0.00	1.04	0.61	1.15	1.00	0.00	0.00	0.00	0.00	0.00
Z_1	0.00	0.60	0.52	0.69	0.00	0.00	0.00	0.00	0.00	1.04	1.05	0.74
Z_2	0.00	0.00	1.47	0.84	0.00	0.00	0.00	000	1.04	0.00	0.68	1.07
Z_3	0.00	0.00	0.95	0.23	0.00	0.00	0.00	0.00	1.05	0.68	0.00	1.00
Z_4	0.00	0.50	0.22	0.00	0.00	0.00	0.00	0.00	0.74	1.07	1.00	0.00

```
Estimated Thresholds:
```

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	Z_1	Z_2	Z_3	Z_4
0.002	0.002	0.002	0.002	0.028	0.003	0.014	0.002	0.002	0.006	0.004	0.002

Table 7: *Estimated network* shows a matrix including all β-values for $\hat{\mathcal{H}}^{\mathsf{A} \Cup \mathsf{B}}_{\mathbf{XYZ}}$. The β-values of variables Y_j, Z_k all equal zero, since there are no connections between those variables. *Estimated Thresholds* lists the threshold for all variables $X_1, \ldots, X_4, Y_1, \ldots, Y_4, Z_1, \ldots, Z_4$.

id	prob	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	Z_1	Z_2	Z_3	Z_4
1	0.000216715820612435	0	0	0	0	0	0	0	0	0	0	0	0
2	1.32535870476192e-05	1	0	0	0	0	0	0	0	0	0	0	0
3	1.95535920316079e-05	0	1	0	0	0	0	0	0	0	0	0	0
4	1.02863765627839e-05	1	1	0	0	0	0	0	0	0	0	0	0
5	2.09732319485271e-05	0	0	1	0	0	0	0	0	0	0	0	0
6	5.63674381730984e-06	1	0	1	0	0	0	0	0	0	0	0	0
7	1.24304957039338e-05	0	1	1	0	0	0	0	0	0	0	0	0
8	2.8737199045894e-05	1	1	1	0	0	0	0	0	0	0	0	0
9	1.24062866639541e-05	0	0	0	1	0	0	0	0	0	0	0	0
10	7.79850303261538e-06	1	0	0	1	0	0	0	0	0	0	0	0
11	4.6469255541289e-06	0	1	0	1	0	0	0	0	0	0	0	0
12	2.51262802481008e-05	1	1	0	1	0	0	0	0	0	0	0	0
13	6.6387806129642e-06	0	0	1	1	0	0	0	0	0	0	0	0
14	1.83390937105969e-05	1	0	1	1	0	0	0	0	0	0	0	0

Table 8: An extract of the estimated joint probability distribution $\hat{P}(\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{y}, \mathbf{Z} = \mathbf{z})$, which contains 14 probability components. Under *prob* the probability of a single component is listed. It can be seen that for each component the probability is close to zero. This can be explained by the fact that for data simulation, the marginal probability was relatively symmetric with 0.6 and we deal with 4096 probability mass distributions.

have to restrict the joint probability distribution to the existing observations. For this thesis I estimated the missing values with option *(2)*.

I started with the estimation of the missing $\mathbf{z}$ in dataset A. Everything I need for the estimation is the joint probability distribution. For each row of dataset A, $\mathbf{x}$ and $\mathbf{y}$ are given and used as restriction. Given these observations, each probability component of the joint probability distribution that fulfills this restriction is needed. The values of $\mathbf{Z}$ of the probability component with the highest probability to appear is then used as estimates for the missing $\mathbf{z}$ in dataset A. This is done for each row of A, resulting in a dataset $\hat{\mathsf{A}}$ in which the observation $\mathbf{x}$ and $\mathbf{y}$ are identical to A, and the observation $\mathbf{z}$ is estimated. The same procedure is used to estimate the missing $\mathbf{y}$ in dataset B, resulting in an estimated dataset $\hat{\mathsf{B}}$ in which the observation $\mathbf{x}$ and $\mathbf{z}$ are identical to B, and the observation $\mathbf{y}$ is estimated via the joint probability distribution.

This last step ends the application of Markov networks for the aim of matching data. What I have done is simulate 100 datasets, containing 12 binary variables, four for each block. Each dataset contains 500 observations of the variables. Furthermore for simulating the datasets, a special correlation structure was used to guarantee that CIA is fulfilled. For all 100 datasets a Markov network was constructed, as explained in Chapter 4.2. Since the simulated data is binary I used the Ising model (see Chapter 5.1) for structure estimation. After a Markov network has been obtained for all 100 datasets, I estimated the joint probability distribution (macro approach). After this step, the estimation of the missing values is possible, which is done via the estimated joint probability distribution (micro approach). Throughout Chapter 6 I show extracts of the results for a single dataset, while each step is actually applied for all 100 datasets. Since I am working with simulated data, it is possible to evaluate the results of the macro and micro approach, which I will do now.

6.5 Evaluation of Applying Markov Networks for Statistical Matching

In this chapter I will summarize the evaluation of applying MNs for statistical matching. Raessler (2002, pp. 29ff.) lists four quality levels for evaluating the quality of matched data:

1. First level: preserving the individual values,
2. Second level: preserving joint distribution,
3. Third level: preserving correlation structures,
4. Fourth level: preserving marginal distributions.

For the purposes here, I will focus on the first and the fourth level. I will start with the evaluation of the marginal distribution.

For simulating the data I used a fix correlation structure between the variables, and transmitted a fix marginal probability to each variable. In this thesis, for each variable I used a marginal probability of 0.6 for being 1. Hence, for evaluation we would expect estimates for marginal probabilities around 0.6 for each variable.[27]

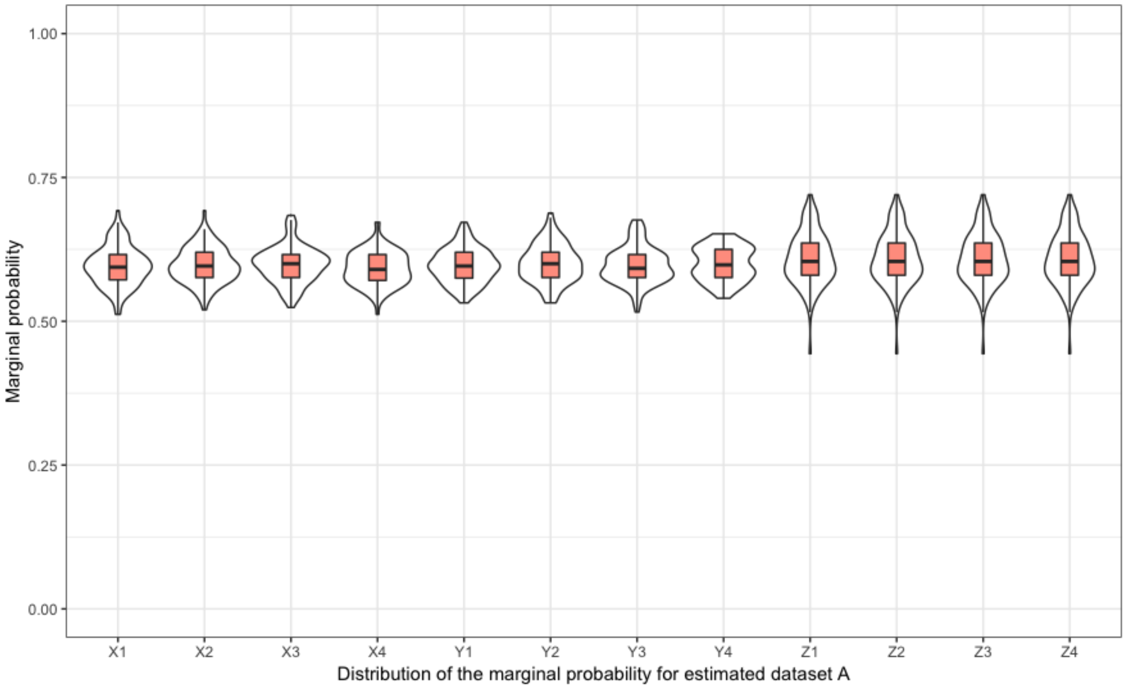

Figure 12: Boxplots of the marginal probability of the 100 estimated datasets $\hat{\mathsf{A}}$ separated by variables. The values $\mathbf{z}$ are estimated and thus the boxplots for variables $Z_1, \ldots, Z_4$ show the results for the marginal probability with the estimations. For the other variables $\mathbf{x}$ and $\mathbf{y}$ the values of A and $\hat{\mathsf{A}}$ are always identical, hence also the boxplots are the same as in Figure 13.

Figure 12 shows boxplots of the marginal probability for all 100 estimated datasets $\hat{\mathsf{A}}$, while for comparison Figure 13 shows boxplots of the marginal probabilities for the true 100 datasets A. In datasets A and $\hat{\mathsf{A}}$ the observations $\mathbf{x}$ and $\mathbf{y}$ are the same, thus also the boxplots are the same for these variables. In Figure 13 we can see that the median of all variables of $\mathbf{Z}$ is very close to 0.6, and also for $\hat{\mathsf{A}}$ the median of the variables of $\mathbf{Z}$ is around 0.6 (see Figure 12). Moreover, in $\hat{\mathsf{A}}$ the distribution of the marginal probabilities for the 100 datasets is much more heavy-tailed as for the original dataset, and also is not unimodal but slightly bimodal.

Also for the 100 datasets B the missing observations for $\mathbf{Y}$ are estimated, and 100 datasets $\hat{\mathsf{B}}$ are created. Figure 14 shows the boxplots for the marginal probabilities of $\mathbf{x}$ and $\mathbf{z}$, which are the same as in Figure 15. Figure 15 shows the boxplots for the

[27] The true original datasets indeed had marginal probabilities of 0.6.

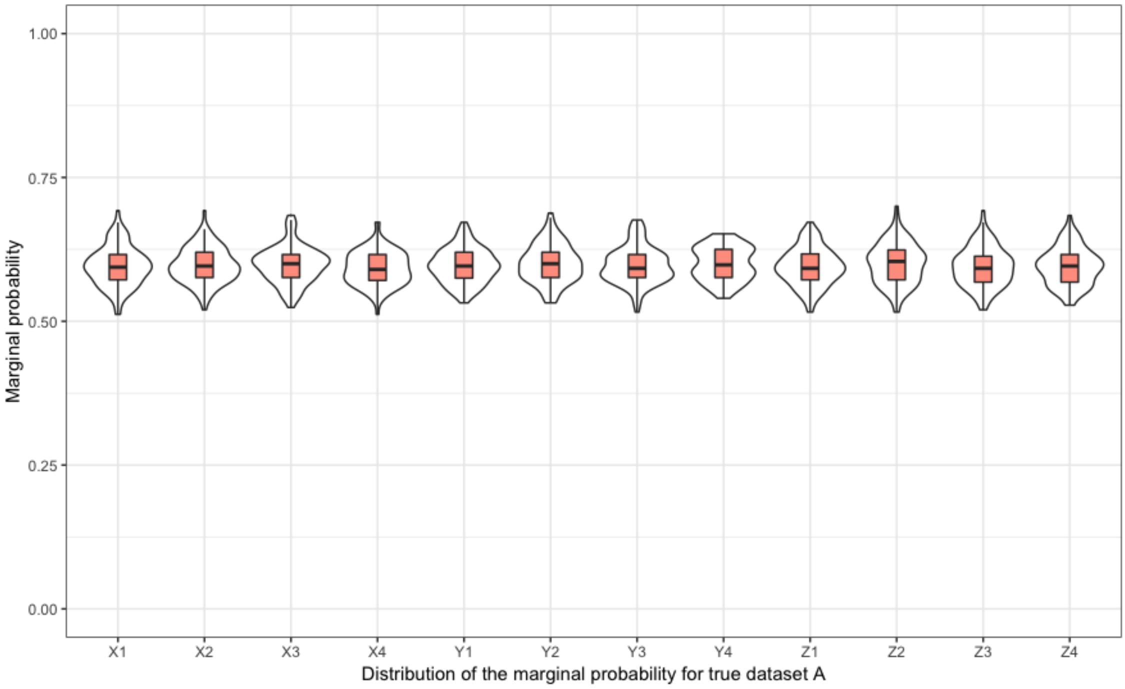

Figure 13: Boxplots for the marginal probabilities of the true 100 datasets A without any missings for $\mathbf{Z}$.

100 true datasets B, which contain no missing observations for $\mathbf{Y}$. It can be seen that for the true datasets B the median of the marginal probabilities is very close to 0.6 for all variables. In Figure 14 we can see that the median for the estimated values of $\mathbf{y}$ is also around 0.6. Nevertheless the distribution of the marginal probabilities for $\mathbf{y}$ are more heavy-tailed.

Also the ratio of correctly estimated values $\mathbf{z}$ in $\hat{\mathsf{A}}$ and $\mathbf{y}$ in $\hat{\mathsf{B}}$ is of interest. For that goal I compared the values in $\hat{\mathsf{A}}$ with the true values in A, and the values in $\hat{\mathsf{B}}$ with the true values in B, again for all 100 simulated datasets. In Figure 16 we can see that the median of not correctly estimated values $\mathbf{z}$ is around 0.22. For the 100 datasets I received for $\hat{\mathsf{A}}$, the best result was a ratio of not correctly estimated values of around 0.14, while worst results were around 0.30. Thus even in the worst case, the results of the micro approach are still much better than just guessing. In Table 9 we can see that the mean for all four variables of not correctly estimated values is around 0.22, which is very close to the median (this can be explained by the symmetric distributions).

In Figure 17 we can see that the median of not correctly estimated values for $\mathbf{y}$ is around 0.23. For the 100 datasets I received for $\hat{\mathsf{B}}$, the best result was a ratio of not correctly estimated values of around 0.14, while the worst results were around 0.30. Even though the results are slightly worse than for the estimation of $\mathbf{z}$, also here the estimation is still much better than just guessing. In Table 10 we can see that also the mean of not correctly estimated values is around 0.23, just like the median.

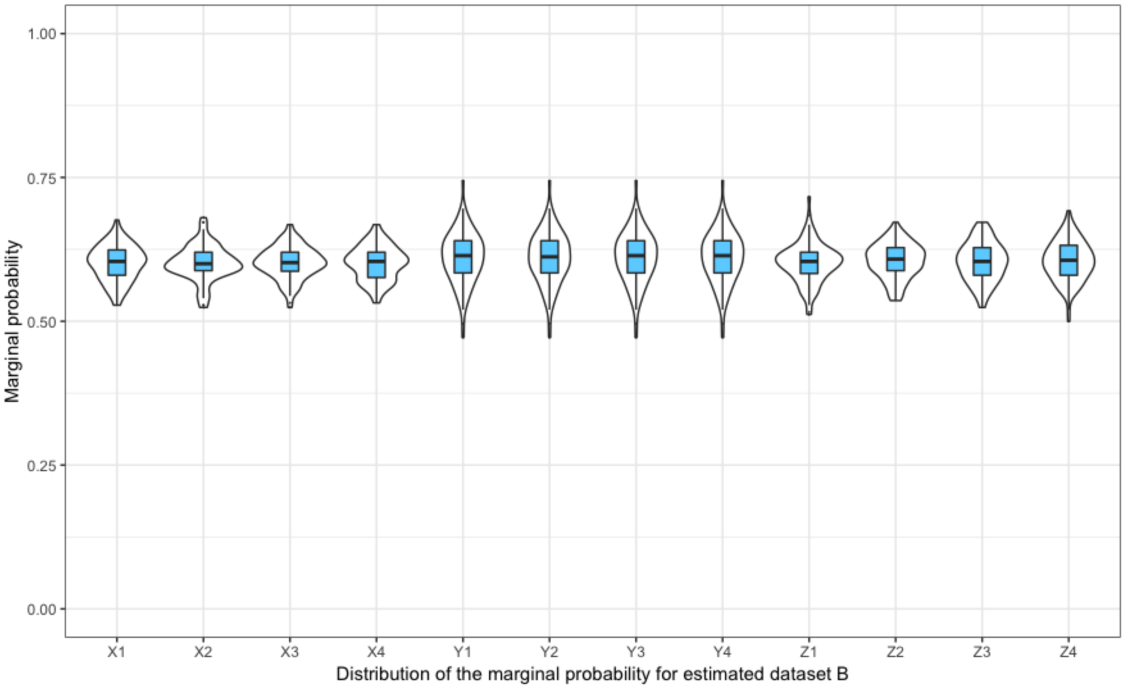

Figure 14: Boxplots of the marginal probability of the 100 estimated datasets $\hat{\mathsf{B}}$ separated by variables. The values $\mathbf{y}$ are estimated and thus the boxplots for variables $Y_1, \ldots, Y_4$ show the results for the marginal probability with the estimations. For the other variable blocks $\mathbf{X}$ and $\mathbf{Z}$ the values of B and $\hat{\mathsf{B}}$ are identical, hence also the boxplots are the same as in Figure 15.

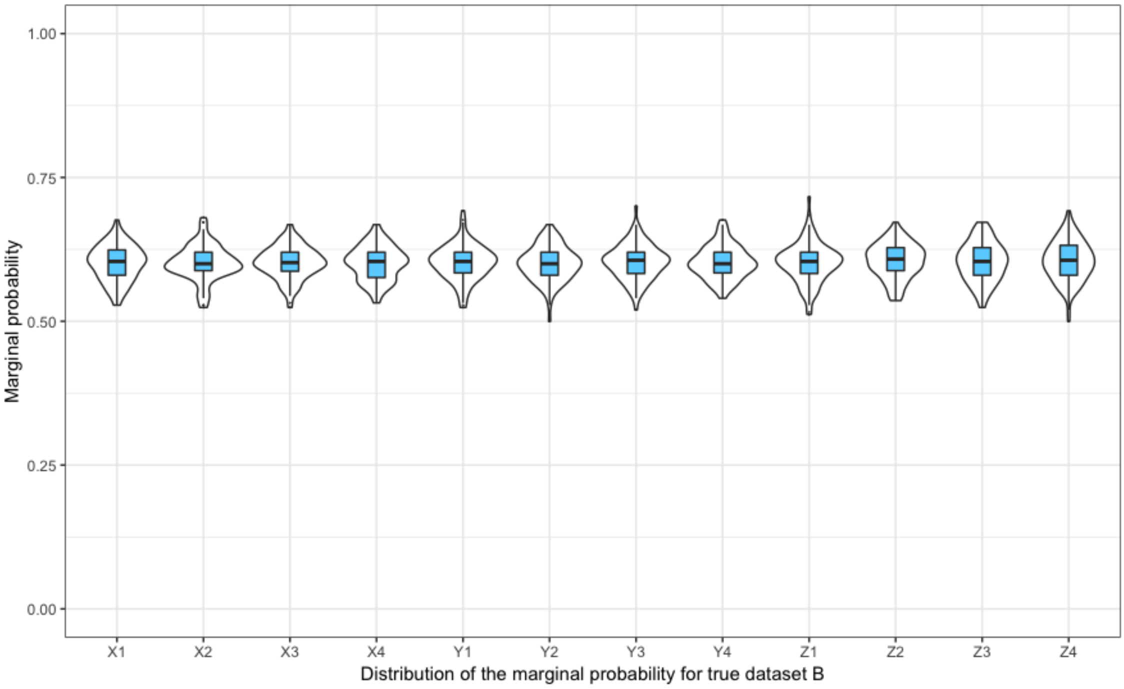

Figure 15: Boxplots for the marginal probabilities of the true 100 datasets B without any missings for $\mathbf{Y}$.

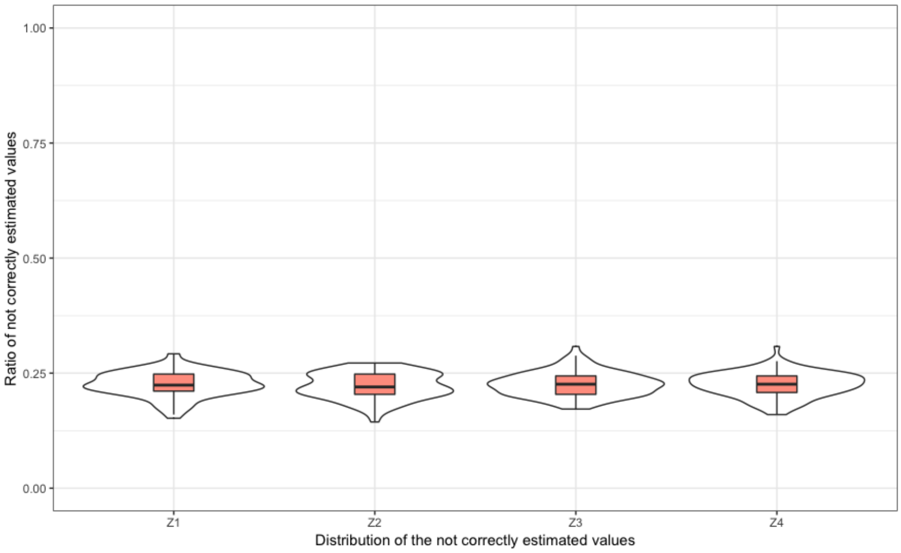

Figure 16: Boxplots for the ratio of not correctly estimated values $\mathbf{z}$ in dataset $\hat{\mathsf{A}}$. The median is about 0.22.

	Z_1	Z_2	Z_3	Z_4
Mean	0.226	0.222	0.226	0.225

Table 9: Mean of not correct estimates for $\mathbf{z}$ for the micro approach.

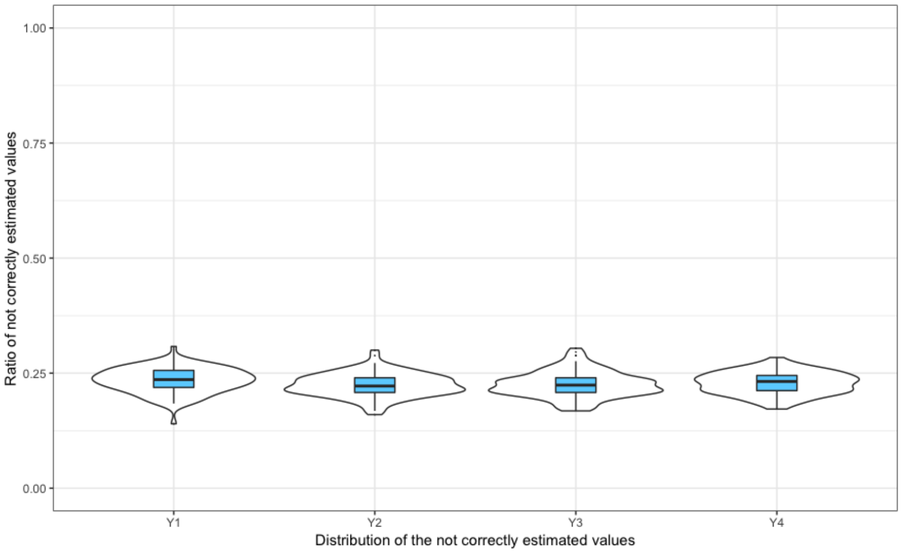

Figure 17: Boxplots for the ratio of not correctly estimated values $\mathbf{y}$ in dataset $\hat{\mathsf{B}}$. The median is around 0.23.

	Y_1	Y_2	Y_3	Y_4
Mean	0.236	0.222	0.224	0.232

Table 10: Mean of not correct estimates for $\mathbf{y}$ for the micro approach.

7 Comparing Bayesian Networks and Markov Networks for Statistical Matching

The aim of this thesis is to continue the work of Endres and Augustin (2016), which focused on BNs for matching data. Instead of using BNs, this thesis researches if MNs are also appropriate for the aims of statistical matching. Both variations of probabilistic graphical models have in common that they simplify a joint probability distribution in their representation by using the (in)dependence structure of the components of the joint probability distribution. The general goal of both is to find a perfect map, meaning that a graph is both I-map and D-map. In statistical matching the problem of blockwise missing data is characteristic and therefore indispensable. The fact that there is not a single observation of $(\mathbf{XYZ})$ implies that we do not know what kind of (in)dependence structure the joint probability distribution has. In fact, it is the CIA that allows to estimate a joint probability distribution. But this also goes along with disadvantage that it is not possible to test if CI is really fulfilled in the data. Hence with having so little information about the joint probability distribution, it is not possible to know what kind of (in)dependence structure is present.

As I explained in Chapter 3, BNs and MNs represent different kinds of structure in the data. For some data only a BN is able to find a perfect map for the joint probability distribution, and the other way around. In the case of chordal graphs, both BNs and MNs can be a perfect map (Koller and Friedman 2009, pp. 139ff.).

Given the fact that in statistical matching we just do not know what kind of structure the joint probability distribution has, it is not possible to make a general statement which variation of probabilistic graphical models is superior for the aims of statistical matching.

The work of both Endres and Augustin (2016) and this thesis was executed with the statistical software `R` (R Core Team 2008). At the moment it seems like the available add-ons for applying BNs in `R` offer more functions and possibilities to work with. However, I want to note that this is just a personal impression, which is only grounded in the use of categorical data. Furthermore this need not not apply to other software, since I did not compose an overview what other programming software, e.g. `Python` (Python Core Team 2015) can offer to simplify application of probabilistic graphical models.

An advantage BNs indeed have compared with MNs is the intuition of the theory. The fact that the chain rule for Bayesian networks, see Equation (3.14), works with conditional probabilities is more intuitive than the definition of factors in Definition 10, which are needed to formulate the joint probability distribution in Equation (3.10). Furthermore the estimation of the joint probability distribution can be quite extensive, since the partition function ζ has to be calculated. This step is not neccesary for BNs, but also for the Ising model the partition function simplifies a lot.

By applying BNs for statistical matching, cycles in the graph structure $\mathcal{G}$ are not allowed. Endres and Augustin (2016) present a second, more individual procedure to estimate the structure of $\mathcal{G}^{\mathsf{A}\Cup\mathsf{B}}_{\mathbf{XYZ}}$ (Endres and Augustin 2016, pp. 163ff.), through which cycles are indeed possible by putting together all information. This is a problem for the estimation of the joint probability that can not occur with MNs, since a MN may well contain cycles. Nevertheless, with the procedure of estimation for $\mathcal{G}$, I explained in Chapter 4 that this problem will not occur either.

After all I cannot give a recommendation what kind of method to use. In some cases expert knowledge may be available that justifies the use of BNs over MNs or vice versa, but as long as this is not the case, both versions are equally appropriate.

8 Discussion

The end of this thesis is a discussion. The goal of this thesis was to research how Markov networks can be utilized for statistical matching, and to apply Markov networks for statistical matching with simulated datasets. For that reason I summarized the theory of statistical matching in Chapter 2, and also the theory of probabilistic graphical models in Chapter 3. In Chapter 4 I explained the work of Endres and Augustin (2016) that deals with Bayesian networks for statistical matching. I then focused on a fundamental part of this thesis: the theory of using Markov networks for statistical matching. The goal was to match categorical, more precisely binary data, thus Chapter 5 covered the problem of structure estimation of a Markov network for binary data. In Chapter 6 I explained how I simulated the datasets, which I matched by utilizing Markov networks. Moreover I summarized the results of the evaluation of matching. In Chapter 7 I briefly compared the two main variations of probabilistic graphical models, being Bayesian networks and Markov networks, and how they differ in their application for statistical matching.

8.1 Summary of Results and Conclusion

This thesis tries to utilize Markov Networks for the aims of statistical matching. Even though this was just a first try, we can see that the assumption of conditional independence, which makes the joint probability distribution solvable in statistical matching, is strongly connected to the independence structure a Markov Network represents. This is the main reason why Markov networks can be used for matching disjoint datasets. Furthermore the theory of Markov networks, especially the representation of the joint probability distribution, can be manipulated in a way that fits the special data situation in statistical matching. Equation (4.3) shows how the representation of the joint probability distribution is still solvable for the statistical matching situation.

By applying Markov networks for statistical matching I worked with 100 simulated datasets. For all of them I used the same marginal probability and correlation structure for simulation. For the application, in each case I matched two datasets A, with $\mathbf{z}$ missing, and B with $\mathbf{y}$ missing. In a situation where the CIA holds (as in my case), the results of matching the datasets were quite good. The estimated joint probability distribution and also the ratio of correctly estimated observations for the missing entries in A and B were good. However, this was a first try and improvement is likely after further research for using Markov networks for statistical matching.

For the case of binary data appropriate software in `R` already exists, and works for the goal of matching data. However, I want to note that the package I used was the only one I could find for the aims of this thesis.

All in all I can state: Markov networks offer promising results when it comes to statistical matching. The representation of the joint probability distribution by graphs is very intuitive, and the theory is suitable for statistical matching.

8.2 Limitations

Even though the application worked well, there are some limitations of applying Markov networks for statistical matching. In this thesis I worked with binary data, but if one is working with differently scaled data a lot changes. For structure estimation I used the Ising model, which is only suitable for binary data. Thus for differently scaled data the applicant has to look which kind of model is appropriate for the data. The use of a different model for structure estimation also goes along with the need of other software, or other packages. The statistical programming software `R` (R Core Team 2008) offers a great variety of packages, but still it is essential to find the right one. Additionally I made the experience during this thesis that the variety for packages that deal with Markov networks is not that big, especially when it comes to categorical data. I also took a short look at what `Python` (Python Core Team 2015) can offer for the application of Markov networks, but this, too, seemed meager. In my opinion software is more manifold for the application of Bayesian networks.

8.3 Outlook

For this thesis I only applied simulated data that fulfills the assumption of conditional independence of statistical matching. This is a strong assumption which is not guaranteed to be fulfilled in practice, and additionally could not be tested for the data situation in statistical matching. Hence it is of great relevance to research what happens if this assumption is not fulfilled. Do Markov networks handle this problem and still perform well?

As mentioned above, the Ising model is just appropriate for binary data, but often the applicant is confronted with differently scaled data or even a mix of differently scaled data. For that reason suitable methods for structure estimation are needed in theory and application.

And of course, the theory and software I used in this thesis can be researched in more detail. There are several more arguments in `R`-functions I used, for which I set a fix value for all 100 datasets. Thus it would be interesting to see how these function-parameters can influence and maybe improve the results. I hope that soon a more complex simulation study will be able to answer these questions.

Appendix

A The Missing Data Problem in Statistical Matching

The goal of statistical matching is to receive either a synthetic dataset $\hat{\mathsf{A}} \uplus \hat{\mathsf{B}}$, or to estimate the joint probability distribution. By simply putting together both datasets, Figure 1 shows that there are blockwise missing values for either realizations of $\mathbf{Y}$ or $\mathbf{Z}$. Thus the dataset $\mathsf{A} \uplus \mathsf{B}$ confronts the researcher with a missing data problem. Rubin (1976) defines three mechanisms that cause the problem of missing data: *Missing completely at random* (MCAR), *Missing at random* (MAR), and *Missing not at random* (MNAR). As soon as dealing with missing data, the first step is to analyze which mechanism provokes the missing data, since this has to be considered in further analysis or inference.

In the context of statistical matching, observed data is generated by a joint probability distribution $P_{\mathbf{XYZ}}$, and the observed data is independent and identically distributed. We can assume that MCAR is the mechanism that causes the missings, since the reason lies in the way the data was collected: we cannot observe ($\mathbf{XYZ}$) simultaneously, because e.g. the questionnaire or the interviewer did not ask for ($\mathbf{XYZ}$), but rather for either ($\mathbf{XY}$) or ($\mathbf{XZ}$). For MCAR the missing values are independent of both the observed realizations of $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$, but also of the unobserved realizations. A more mathematical explanation is given by D'Orazio et al. (2006, pp. 6ff.).

MCAR allows to formulate the likelihood via the observed probability distribution of $\mathsf{A} \uplus \mathsf{B}$ as

$$L(\boldsymbol{\theta} \mid \mathsf{A} \uplus \mathsf{B}) = \prod_{a}^{n_{\mathsf{A}}} P_{\mathbf{XY}}(\mathbf{x_a}, \mathbf{y_a}; \boldsymbol{\theta}) \cdot \prod_{b}^{n_{\mathsf{B}}} P_{\mathbf{XZ}}(\mathbf{x_b}, \mathbf{z_b}; \boldsymbol{\theta}). \tag{A.1}$$

Thus all information to estimate $\boldsymbol{\theta}$ is included in the observed data $\mathsf{A} \uplus \mathsf{B}$ (Raessler 2002, pp. 76ff.).

B The Multinomial Distribution

In statistical matching we are confronted with three blocks of variables, being $\mathbf{X} = (X_1, \ldots, X_p)'$, $\mathbf{Y} = (Y_1, \ldots, Y_q)'$, and $\mathbf{Z} = (Z_1, \ldots, Z_r)'$. For dealing with this multivariate setting, D'Orazio et al. (2006) argues that working with saturated loglinear models reduces the multivariate problem to an univariate one. But instead of working with binomially distributed variables X_i, Y_j, Z_k, with $i = 1, \ldots, p, j = 1, \ldots, q, k = 1, \ldots, r$, the univariate variables X, Y, Z are now multinomially distributed. While for a binomial distribution a Bernoulli experiment has two mutually exclusive outcomes (usually coded with 0/1), the experiment of a multinomial distribution has $A_1, \ldots, A_m$ mutually exclusive outcomes; this experiment is repeated for a certain time n. The outcome A_l has a probability to occur of π_l, with $l = 1, \ldots, m$.

In general a random variable W_l counts the number of occurrences of outcome A_l, with $l = 1, \ldots, m$. This experiment is independently repeated n times, while π_l is the probability that outcome A_l occurs. The joint probability distribution for the W_l is then

$$P(W_1 = w_1, \ldots, W_m = w_m) = \frac{n!}{w_1! \ldots w_m!} \pi_1^{w_1} \ldots \pi_m^{w_m}. \tag{B.1}$$

This distribution is called multinomial distribution and is actually a multivariate distribution. If a random variable $W = (W_1, \ldots, W_m)$ is multinomially distributed, this is denoted

$$W = (W_1, \ldots, W_m) \sim M(n, \boldsymbol{\pi}) \tag{B.2}$$

(Tutz 2000, pp. 13ff.).

C A Short Summary of the Ising Model

Originally, the Ising model was established for statistical physics by the German physician Ernst Ising (1925). It was one of the first models for a Markov random field, or for probabilistic graphical models more generally. The goal of the Ising model is to explain empirically observed facts about ferromagnetic materials. From a one-dimensional perspective a ferromagnetic material can be simplified as a line with a certain number of points. Each point represents a so-called spin, which at any moment can have one of two possible positions: the points x_i of a grid with n spins can be seen as realizations of the random variables W_i, with $n = 1, \ldots, n$ and each $W_i \in \{-1, +1\}$.

The original formulation of the Ising model is for ferromagnetic material with sample space Ω of all sequences

$$\omega = (\omega_0, \omega_1, \ldots, \omega_n), \tag{C.1}$$

with $\omega_i \in \{-1, +1\}$ indicating whether a spin is of positive or negative charge. The energy of the Ising model $\epsilon(\omega)$ is written as

$$\epsilon(\omega) = \underbrace{-J \sum_{i,j} \omega_i \omega_j}_{(1)} \underbrace{- mH \sum_i \omega_i}_{(2)}, \text{ with } i \neq j. \tag{C.2}$$

In Equation (C.2), J is a constant for the property of material being considered. The first term, marked with brace (1) is the energy caused by interaction of two neighboring spins ω_i and ω_j. The second term in brace (2) is the effect of an external magnetic field of intensity H, while $m > 0$ is a constant for the property of material of the external field.

If the constant $J > 0$, the Ising model is in a so-called *active case*, meaning that neighboring spins are aligned to have the same charge. Meanwhile $J < 0$ results in a *repulsive case*, which tends to reinforce pairs in which spins are of opposite charge. Ising (1925) showed that the probability measure on Ω is given by

$$P(\boldsymbol{\omega}) = \frac{1}{\zeta} \exp\left(\frac{1}{kT}(-\epsilon(\boldsymbol{\omega}))\right), \tag{C.3}$$

with k being a universal constant, and T being the temperature (Kindermann and Snell 1980, pp. 1ff.).

The energy of the Ising model can be reformulated in terms of the Ising model with

$$\omega_i = 2x_i - 1 \Leftrightarrow x_i = \frac{\omega_i + 1}{2}. \tag{C.4}$$

D The Logistic Regression Model

If the outcome of response variable y_i is to be modeled via linear regression, the assumption of normal distribution is not appropriate. For binary response variables, logistic regression is suitable.

Suppose a data situation in which the response variable $y_i,\ i = 1, \ldots, n$ is binary and independent given the covariates $x_{i1}, \ldots, x_{ip}$, p being the number of covariates. The goal is to model the probability $\pi_i = P(y_i = 1 \mid x_{i1}, \ldots, x_{ip})$. The linear predictor $\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$ is linked with the response function $h(\cdot)$, so that $h(\eta) \in [0, 1]$. Thus it is possible to model

$$\pi_i = h(\eta_i). \tag{D.1}$$

In a logistic regression model the response function is the logistic function, which is

$$\pi = h(\eta) = \frac{\exp(\eta)}{1 + \exp(\eta)}, \tag{D.2}$$

or the logit link function, which is

$$g(\pi) = \log\left(\frac{\pi}{1 - \pi}\right). \tag{D.3}$$

In Equation (D.3), $g(\cdot)$ is the so called link function (Fahrmeir et al. 2009, pp. 189ff.). It can be seen that for the case of binomial data logistic regression is very convenient. By working with conditional probabilities in the Ising model, the estimation of the structure of a MN is greatly simplified, since this enables to work with the well known logistic regression model.

E Lasso Estimation

I will briefly explain lasso estimation for the popular case of a linear regression model, which is

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{E.1}$$

with $\mathbf{Y} = (Y_1, \ldots, Y_n)'$. $\mathbf{X}$ is a design matrix with n rows and p columns, n being the number of observations and p the number of variables. $\boldsymbol{\epsilon} \sim N(0, \sigma^2)$ is an error term, which is normally distributed with a mean of 0 and variance of 1. The goal is now to estimate $\boldsymbol{\beta}$. A popular method for that is the ordinary least squares estimator $\boldsymbol{\beta}^{ols}$, which is an unbiased estimator with minimal variance compared to other unbiased estimators (Fahrmeir et al. 2013, pp. 178ff.). But one characteristic of the least squares is to tend to overestimate $\boldsymbol{\beta}$, thus $||\boldsymbol{\beta}^{ols}|| \geq \boldsymbol{\beta}$. This can be a disadvantage, since even very small effects translate to the model for Equation (E.1), and the interpretability of the model suffers (Tibshirani 1996, pp. 283).

If the applicant wants to select between several possible models, and also aims for shrinkage of effects towards zero, lasso estimation (also often referred to as ℓ_1-estimation) is a useful version of ordinary least squares estimation for $\boldsymbol{\beta}$ (Efron et al. 2004, p. 409). The characteristic of the lasso estimator is simultaneous shrinkage of the $\boldsymbol{\beta}$ values towards zero. This is achieved by using an additional constraint, which formulates an upper bound for the sum of length of the $\boldsymbol{\beta}$ values. The lasso estimation equation is under the constraint that $\sum_{j=1}^{p} |\beta_j| \leq t$, and $t \geq 0$:

$$\hat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\boldsymbol{\beta}} \left\{ (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j| \right\}, \quad \text{with } \lambda \geq 0. \tag{E.2}$$

The term $\lambda \sum_{j=1}^{p} |\beta_j|$ is a penalty term; for greater values of λ this term has more effect. For estimation varying values for λ are used, which leads to different $\boldsymbol{\beta}$. By using a selection criterion, the λ value that leads to the best result is chosen (Fahrmeir et al. 2013, pp. 208ff.). The parameter t controls the amount of shrinkage; together with $t^{ols} := \sum_{j=1}^{p} |\hat{\beta}_j^{ols}|$ it holds that

- for $t < t^{ols}$: some values of $\boldsymbol{\beta}$ are shrunk exactly to zero, and
- for $t \geq t^{ls}$: $\hat{\boldsymbol{\beta}}^{lasso} = \hat{\boldsymbol{\beta}}^{ols}$

(Fahrmeir et al. 2013, pp. 208ff.).

F List of Software Used

The application of Markov networks for statistical matching was carried out with `R` (R Core Team 2008) in the at this point most recent version 3.5.1. Moreover, several `R`-packages were used for application. I will list the used packages below, although some of them automatically involve several other packages. The directly used packages are:

- `bindata`, version 0.9-19 (Leisch et al. 2011),
- `GenOrd`, version 1.4.0 (Barbiero and Ferrari 2015),
- `igraph`, version 1.2.2 (Csárdi and Nepusz 2006),
- `IsingFit`, version 0.3.1 (van Borkulo et al. 2016),
- `IsingSampler`, version 0.2 (Epskamp 2015),
- `matrix`, version 1.2-14 (Bates et al. 2018),
- `matrixcalc`, version 1.0-3 (Novomestky 2015),
- `qgraph`, version 1.5 (Epskamp et al. 2018).

G Digital Appendix

The digital appendix contains the following files:

- the electronic version of this thesis, in the document "Thesis.pdf",
- programming code for the statistical software R (R Core Team 2008):
  - a document called "READ ME", which contains information about the R-code in general, and about the order of running the R-code,
  - "1. Correlationmatrix.R",,
  - "2. Data Simuation.R",
  - "3. Structure Estimation.R",
  - "4. Estimation of Joint Probability.R",
  - "5. Imputation of the Missing Data.R",
  - "6. Evaluation.R",
- the file "Workspace_final.RData", containing all R objects produced with the above files.

References

Agresti, A. (2007). *An Introduction to Categorical Data Analysis.* Hoboken: John Wiley & Sons, Inc.

Andreß, H.-J., J. A. Hagenaars, and S. Kühnel (1997). *Analyse von Tabellen und kategorialen Daten.* Berlin: Springer.

Barber, R. F. and M. Drton (2015). High-dimensional ising model selection with bayesian information criteria. *Electronic Journal of Statistics 9*(1), 567–607.

Barbiero, A. and P. A. Ferrari (2015). *Simulation of Discrete Random Variables with Given Correlation Matrix and Marginal Distributions.*

Bates, D., M. Maechler, and T. A. Davis (2018). *Sparse and Dense Matrix Classes and Methods.*

Beierle, C. and G. Kern-Isberner (2006). *Methoden wissensbasierter Systeme.* Wiesbaden: Vieweg.

Besag, J. (1972). Nearest-neighbour systems and the auto-logistic model for binary data. *Journal of the Royal Statistical Society. Series B (Methodological) 34*(1), 75–83.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological) 36*(2), 192–236.

Bilodeau, M. and D. Brenner (1999). *Theory of Multivariate Statistics.* New York: Springer.

Bondy, J. A. and U. S. R. Murty (1982). *Graph Theory With Applications.* New York: North Holland.

Brown, L. D. (1986). Fundamentals of statistical exponential families with applications in statistical decision theory. *Lecture Notes–Monograph Series 9.*

Budd, E. C., D. B. Radner, and J. C. Henrichs (1973). Size distribution of family personal income: Methodology and estimates for 1964. *U.S. Bureau of Economic Analysis* (21).

Chen, J. and Z. Chen (2008). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika 95*(3), 759–771.

Csárdi, G. and T. Nepusz (2006). The igraph software package for complex network research. *InterJournal Complex Systems 1695.*

de Waal, A. G. (2015). Statistical matching: Experimental results and future research questions.

D'Orazio, M. (2016). Statmatch: Statistical matching. Technical report.

D'Orazio, M., M. D. Zio, and M. Scanu (2006). *Statistical Matching: Theory and Practice*. Chichester: John Wiley & Sons.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least angle regression. *The Annals of Statistics 32*(2), 407–499.

Endres, E. and T. Augustin (2016). Statistical matching of discrete data by Bayesian networks. In A. Antonucci, G. Corani, and C. P. de Campos (Eds.), *Journal of Machine Learning Research Workshop and Conference Proceedings*, Volume 52, pp. 159–170.

Epskamp, S. (2015). *Sampling Methods and Distribution Functions for the Ising Model.*

Epskamp, S., G. Costantini, J. Haslbeck, A. O. J. Cramer, L. J. Wal-dorp, V. D. Schmittmann, and D. Borsboom (2018). *Graph Plotting Methods, Psychometric Data Visualization and Graphical Model Estimation.*

Fahrmeir, L., T. Kneib, and S. Lang (2009). *Regression. Modelle, Methoden und Anwendungen.* Springer.

Fahrmeir, L., T. Kneib, S. Lang, and B. Marx (2013). *Regression: Models, Methods and Applications.* Springer.

Fahrmeir, L., R. Künstler, I. Pigeot, and G. Tutz (2011). *Statistik: Der Weg zur Datenanalyse.* Berlin: Springer.

Friedman, J., T. Hastie, R. Tibshirani, N. Simon, B. Narasimhan, and J. Qian (2018). Lasso and elastic-net regularized generalized linear models. Technical report.

Frydenberg, M. (1990). The chain graph Markov property. *Scandinavian Journal of Statistics 17*(4), 333–353.

Gidas, B. (1988). Consistency of maximum likelihood and pseudo-likelihood estimators for Gibbs distributions. In G. R. Sell and H. Weinberger (Eds.), *Stochastic Differential Systems, Stochastic Control Theory and Applications*, Volume 10. New York: Springer.

Ising, E. (1925). Beitrag zur Theorie des Ferromagnetismus. *Zeit für Physik 31*, 253–258.

Kindermann, R. and J. L. Snell (1980). *Markov Random Fields and Their Applications.* Providence: Amercian Mathematical Society.

Koller, D. and N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques.* Cambridge, MA: MIT Press.

Leisch, F., A. Weingessel, and K. Hornik (2011). *Generation of Artificial Binary Data.*

Loh, P.-L. and M. J. Wainwright (2013). Structre estimation for discrete graphical models: Generalized covariance matrices and their inverses. *The Annals of Statistics 41*(6), 3022–3049.

Meintrup, D. and S. Schäffler (2005). *Stochastik: Theorie und Anwendungen.* Heidelberg: Springer.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective.* London: MIT Press.

Novomestky, F. (2015). *Collection of Functions for Matrix Calculations.*

Okner, B. A. (1972). Constructing a new data base from existing microdata sets: The 1966 merge file. *Annals of Economic and Social Measurement 1*(3), 325–362.

Okner, B. A. (1974). Data matching and merging: An overview. *Annals of Economic and Social Measurement 3*(2), 347–352.

Python Core Team (2015). *Python: A Dynamic, Open Source Programming Language.* Python Software Foundation.

R Core Team (2008). *R: A Language and Environment for Statistical Computing.* Version 3.5.1. Vienna, Austria: R Foundation for Statistical Computing.

Raessler, S. (2002). *Statistical Matching: A Frequentist Theory, Practical Applications, and Alternative Bayesian Approaches.* New York: Springer.

Ravikumar, P., M. J. Wainwright, and J. D. Lafferty (2010). High-dimensional Ising model selection using l_1-regularized logistic regression. *The Annals of Statistics 38*(3), 1287–1319.

Rubin, D. (1976). Inference and missing data. *Biometrika 63*, 581–592.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*(2), 461–464.

Sims, C. A. (1972). Comments (on okner 1972). *Annals of Economic and Social Measurement 1*(3), 355–357.

Studeny, M. (2005). *Probabilistic Conditional Independence Structures.* London: Springer.

Sucar, L. E. (2015). *Probabilistic Graphical Models: Principles and Application.* London: Springer.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58*(1), 267–288.

Tutz, G. (2000). *Die Analyse kategorialer Daten.* München: Oldenbourg.

van Borkulo, C. D. (2017). *Symptom network models in depression research: From methodological exploration to clinical application.* Ph. D. thesis, University of Groningen.

van Borkulo, C. D. and S. Epskamp (2014). Supplementary information (accompanying a new method for constructing networks from binary data). Technical report, University of Amsterdam.

van Borkulo, C. D., S. Epskamp, and A. Robitzsch (2016). *Fitting Ising Models Using the ELasso Method.*

van der Putten, P., J. N. Kok, and A. Gupta (2002). Data fusion through statistical matching. Technical report, MIT Sloan Working Paper No. 4342-02.

Declaration of Authorship

I hereby confirm to have written this Master's thesis independently and under the exclusive use of the sources listed.

Munich, August 31, 2018

Katrin Newger