Causal Discovery from Mixed Data using Gaussian Copula Models

Proefschrift

ter verkrijging van de graad van doctor
aan de Radboud Universiteit Nijmegen,
op gezag van de rector magnificus prof. dr. J.H.J.M. van Krieken,
volgens besluit van het college van decanen
in het openbaar te verdedigen op dinsdag 22 januari 2019
om 10.30 uur precies

door

Ruifei Cui

geboren op 15 augustus 1989
te Zibo (China)
The research reported in this thesis has been carried out under the auspices of the institute for Computing and Information Science (iCIS) of the Radboud University Nijmegen, and has been sponsored by China National Digital Switching System Engineering & Technological Research Center.
# Contents

Title page

Table of Contents

## 1 Introduction

1.1 Motivation for Causal Discovery
1.2 Causal Discovery from Observational Data
1.3 Graphical Models and Causal Discovery
1.4 The PC Algorithm
1.5 Gaussian Copula Models
1.6 Outline of this Thesis

## 2 Copula PC Algorithm for Causal Discovery from Mixed Data

2.1 Introduction
2.2 Method
   2.2.1 Projected Inverse-Wishart Distribution
   2.2.2 Gibbs Sampler Based on Extended Rank Likelihood
   2.2.3 Estimating Correlation Matrix and Effective Sample Size
   2.2.4 Copula PC Algorithm and Stable Copula PC Algorithm
2.3 Experiments
   2.3.1 Estimation for the Effective Sample Size
   2.3.2 Causal Discovery on Simulations
   2.3.3 Application to Real-World Data
2.4 Conclusion and Discussion
2.A Proof of Theorem 2.1

## 3 Hetcor PC Algorithm for Causal Discovery from Mixed Data

3.1 Introduction
3.2 Method
   3.2.1 Learning Correlation Matrix
   3.2.2 Hetcor PC Algorithm
   3.2.3 Learning Effective Sample Size
3.3 Simulation Study
3.3.1 Setup
3.3.2 Sample Size vs. Local Effective Sample Size
3.3.3 Copula PC and Hetcor PC
3.3.4 MM and Hetcor PC
3.4 Application to Real-world Data
3.5 Conclusion and Discussion
3.A Proof of Lemma 3.1
3.B Proof of Theorem 3.1
3.C Proof of Theorem 3.2
3.D Proof of Theorem 3.3

4 Causal Discovery from Data with Missing Values
4.1 Introduction
4.2 Missingness Mechanism
4.3 Rank PC Algorithm for Data with Missing Values
  4.3.1 Basic Procedure
  4.3.2 Theoretical Analysis
4.4 Copula PC Algorithm for Data with Missing Values
  4.4.1 Gibbs Sampling for Data with Missing Values
  4.4.2 Estimating the Underlying Correlation Matrix
  4.4.3 Estimating the Effective Sample Size
  4.4.4 Consistency of Copula PC Algorithm
4.5 Alternative Approaches
  4.5.1 Listwise Deletion
  4.5.2 Imputation Methods
4.6 Simulation Study
  4.6.1 Setup
  4.6.2 Evaluating Correlation Estimators
  4.6.3 Causal Discovery on Benchmark DAGs
4.7 Application to Real-world Data
  4.7.1 Riboflavin Production Data
  4.7.2 Chronic Fatigue Syndrome Data
4.8 Conclusion and Future Work
4.A Evaluation on Simulated DAGs

5 Gaussian Copula Factor Model and A Novel Inference Approach
5.1 Introduction
5.2 Background
  5.2.1 Missingness Mechanism
  5.2.2 Parameter Estimation in CFA Models
5.3 Gaussian Copula Factor Model
5.4 A Novel Inference Approach for Gaussian Copula Factor Model
  5.4.1 Inference for Gaussian Copula Factor Model
Chapter 1

Introduction

This chapter starts with an introduction to the motivation for our research topic—causal discovery, which hopefully can succeed in answering the question “why do you choose this research topic?”. Then we explain why it is important and crucial to conduct causal discovery from observational data. Next we introduce some key concepts in graphical models, formulate the problem of causal discovery, and review the PC algorithm (a reference causal discovery algorithm) as well as Gaussian copula models that are central to this thesis. The last section gives a brief overview of the research questions that are studied in later chapters.

1.1 Motivation for Causal Discovery

In social science, psychology, and many other sciences, the goal of quite a lot of studies is to understand the underlying mechanism by which variables interact with each other (i.e., the data-generating process), and to predict what the values of variables of interest would be if some external interventions changed the naturally-occurring mechanism. For example, whether and to what extent can a given policy reduce the number of crimes? What is the efficacy of a particular drug in a given population? Whether a dietary habit is a factor that affects life expectancy and what is the effect of changing people’s diets on their life span? In order to answer such questions, the fundamental step is to learn the causal relationships among variables, that is the subject of causal discovery. We next motivate our research by discussing how causation differs from correlation (or more generally association) and analyzing the implication of causal knowledge for traditional machine learning tasks.
Causation vs. correlation

Although much of the conceptual framework and algorithmic tools for causal analysis are now well-established [Pearl, 2009a], how causation differs from correlation is still a problem that confuses many researchers. A straightforward distinction is that causation is asymmetric while correlation is symmetric. For example, we consider the question of what is the relationship between latitude and temperature. Correlation analysis tells us they are significantly correlated, whereas causal analysis suggests that latitude is a direct cause of temperature (not the other way around) as shown in Figure 1.1a. With the result from correlation analysis, one could say that the temperature would be likely to be low when you are in a place with high latitude, and it also holds to say that observing a low temperature gives us more belief of staying at a high latitude position. With causal analysis, we can not only claim the statements above, but also be able to draw new conclusions. For instance, we can further claim that the temperature would go down if we (intentionally) go to a high latitude place, whereas nothing would change to the latitude if we increase the temperature (possibly with external intervention). Therefore, we argue that causal analysis suggests more informative and useful knowledge than correlation analysis, which helps us answer more research questions of interest.

Figure 1.1: Examples that show the distinction between causation and correlation: (a) latitude is a direct cause of temperature; (b) a model that explains the correlation between stork frequency and human birth rate.

Another major distinction between the two concepts is indicated by the slogan “correlation does not imply causation”. A well-known example that explains this statement is the study of correlation analysis between the frequency of storks and human birth rate [Matthews, 2000]. This study shows a statistically significant correlation between stork populations and human birth rates across Europe ($p$-value = 0.008). However, it is obvious that neither stork frequency causes human babies delivery, nor the other way around. In fact, the correlation between the two variables is determined by a hidden confounder that causes both simultaneously, as shown in Figure 1.1b. From another point of view, we may say that it is possible to train a good predictor of the human birth rate which uses the frequency of storks (along with other features) as an input. However, if politicians asked us whether one could boost the birth rate by increasing the number of storks, we would have to tell them that this kind of intervention is not covered by the correlation analysis.
Causation vs. traditional machine learning

A recent study [Schölkopf et al., 2012] depicts how traditional machine learning could benefit from the knowledge of causal structure. The basic argument is that causal knowledge has implications for popular scenarios such as covariate shift, concept drift, transfer learning, and semi-supervised learning, which may facilitate some approaches for a given problem, and rule out others. We consider the simplest setting of two variables as an illustrative example.

**Example 1.1.** Assume the causal model shown in Figure 1.2 is the underlying data generating process. We consider the following two semi-supervised learning tasks:

1. Given training points sampled from the joint distribution $P(X,Y)$, and an additional set of inputs sampled from $P(X)$, what is the conditional distribution $P(Y|X)$? Following the authors, this kind of task is called learning in the causal direction.

2. Given training points sampled from the joint distribution $P(X,Y)$, and an additional set of inputs sampled from $P(Y)$, what is the conditional distribution $P(X|Y)$? This kind of task is called learning in the anticausal direction.

![Figure 1.2: A simple two-variable causal model in (a) graphical form and (b) equation form, where $X$ is the cause, $Y$ is the effect, and $\epsilon_x$ and $\epsilon_y$ are independent noise terms.](image)

For Task 1, since $P(X)$ is independent of $P(Y|X)$ ($\epsilon_x$ and $\epsilon_y$ are independent), $P(X)$ contains no information about $P(Y|X)$. Therefore, the additional points sampled from $P(X)$ does not influence the estimate of $P(Y|X)$. In other words, semi-supervised learning does not work for this scenario.

For Task 2, things go differently. The dependency between $P(Y)$ and $P(X|Y)$ implies that we know something more about $P(X|Y)$ from the additional points of $P(Y)$. In this scenario, semi-supervised learning thus should work better than supervised learning that is only based on the training points from the joint distribution.

This example shows that semi-supervised learning only works in the anticausal direction, in the sense that people could simply rule out semi-supervised learning methods in their analysis if they have the prior knowledge that the current analysis is in causal direction. For more details about the implications of causal knowledge for other machine learning tasks, we refer the reader to Schölkopf et al. [2012]; Janzing et al. [2015].
1.2 Causal Discovery from Observational Data

The golden standard for discovering causal relations is a randomized controlled trial or intervention experiment. Take a typical medical research question as an example: whether a treatment has efficacy on a particular disease in a given population? What researchers usually do is to first recruit some people from the population, and randomly classify them into two groups: the experiment group and the control group. Then everybody in the first group is advised to take the designed treatment while no action on the second group, with all the other conditions equal for both. The observed difference in the occurrence rate of the disease between these two groups may answer the research question. However, such experiments can be impractical, unethical, time-consuming, or even impossible [Rubin, 1974; Hyttinen et al., 2013]. For example, in the study of whether smoking is a cause of lung cancer, it is unethical and impractical to force a randomly-selected group of people to smoke. For a study concerning if gender is a causal risk factor for some disease, it is obviously impossible to change the gender of the respondents. See Maathuis and Nandy [2016] for more examples and analysis. By contrast, observational data can often be collected cheaply and is abundant. Therefore, we study causal discovery from observational data in this thesis.

1.3 Graphical Models and Causal Discovery

This section introduces some necessary graph terminology and briefly reviews the graphical framework of causal discovery.

Graph terminology

A graphical model is a probabilistic model along with a graph $\mathcal{G} = (V, E)$, where the vertices ($X_i : X_i \in V$) denote random variables and the edges $E$ represent dependence structure among the variables. It is a powerful tool to work with families of distributions sharing a set of conditional independence restrictions [Lauritzen, 1996].

A graph is directed if it just contains directed edges and undirected if all edges are undirected. A graph that contains both directed and undirected edges is called a partially directed graph. Graphs without directed cycles (e.g., $X_i \rightarrow X_j \rightarrow X_i$) are acyclic. We refer to a graph as a directed acyclic graph (DAG) if it is both directed and acyclic.

The set of all vertices in $V$ that are adjacent to $X_i$ in the graph $\mathcal{G}$ constitutes the adjacency set of $X_i$, denoted by $\text{adj}(\mathcal{G}, X_i)$. A vertex $X_j$ in $\text{adj}(\mathcal{G}, X_i)$ is called a parent of $X_i$, if $X_j \rightarrow X_i$. A triple $(X_i, X_j, X_k)$ in a graph $\mathcal{G}$ is unshielded if $X_i$ and $X_j$ as well as $X_j$ and $X_k$ are adjacent, but $X_i$ and $X_k$ are not adjacent in $\mathcal{G}$. A v-structure $(X_i, X_j, X_k)$ is an unshielded triple in a graph $\mathcal{G}$ where the edges are oriented as $X_i \rightarrow X_j \leftarrow X_k$. 
Chapter 1. Introduction

Graphical representation to causal relations

A common way to model causal relationships is using a DAG [Spirtes et al., 2000; Pearl, 2009b], in which the vertices denote random variables and an directed edge \( X_i \rightarrow X_j \) represents a causal link from \( X_i \) to \( X_j \). Throughout this thesis, we assume that the data are generated from a causal structure that can be represented by a DAG. Further, we assume causal sufficiency, that is, there are no hidden confounders and selection variables in the system (see Chapter 7 for a discussion about relaxing this assumption).

A multivariate probability distribution over a random vector \( \mathbf{X} \) with \( X_i \in \mathbf{V} \) is said to be Markov w.r.t. a DAG \( \mathcal{G} = (\mathbf{V}, \mathbf{E}) \), if \( \mathbf{X} \) satisfies the causal Markov condition: each variable in \( \mathcal{G} \) is independent of its non-descendants given its parents, which is also implied by so-called \( d \)-separation [Pearl, 2009b]. A distribution is faithful w.r.t. a DAG if there are no conditional independencies in the distribution that are not encoded via \( d \)-separation. If a distribution is both Markov and faithful w.r.t. a DAG \( \mathcal{G} \), the DAG is called a perfect map of the distribution. The causal Markov and faithful assumptions enable us to (partially) reconstruct the causal structure from the joint distribution.

Unfortunately, even with the above assumptions, the causal structure, in general, cannot be identified from the observational data. This is because several DAGs may, via \( d \)-separation, correspond to the same set of conditional independencies. The set of such DAGs is called a Markov equivalence class, which can be represented by a completed partially directed acyclic graph (CPDAG) [Chickering, 2002a]. Arcs in a CPDAG imply a cause-effect relationship between pairs of variables since the same arc appears in all members of the CPDAG. An undirected edge \( X_i - X_j \) in a CPDAG indicates that some of its members contain an arc \( X_i \rightarrow X_j \) while others contain an arc \( X_j \rightarrow X_i \).

Problem formulation Assume that the underlying DAG \( \mathcal{G} = (\mathbf{V}, \mathbf{E}) \) is a perfect map of the joint distribution over \( \mathbf{X} \) with \( X_i \in \mathbf{V} \). Causal discovery aims to learn the Markov equivalence class of the DAG \( \mathcal{G} \) from observations drawn from the joint distribution.

1.4 The PC Algorithm

In this section, we describe the PC algorithm (named after its two inventors, Peter and Clark) [Spirtes et al., 2000], which is a reference algorithm for causal discovery. The sketch of this algorithm is listed in Algorithm 1.1.

Step 1, also called adjacency search, is the central part of the PC algorithm, whose pseudo-code is given in Algorithm 1.2. It starts with a fully connected undirected graph, and then iteratively removes the edges according to conditional independence decisions. Finally, this step outputs the skeleton, denoted by \( \mathcal{C} \), and separation sets where the separation set of \( X_i \) and \( X_j \) is denoted by \( \text{sepset}(X_i, X_j) \).
Algorithm 1.1 Sketch of the PC algorithm

**Require:** Conditional independencies among all variables in $V$.

1: **Step 1**: Adjacency search: search for the skeleton $C$ and separation sets using Algorithm 1.2;
2: **Step 2**: Orient unshielded triples in the skeleton according to the separation sets;
3: **Step 3**: Orient as many of the remaining undirected edges as possible by repeatedly applying the orientation rules R1-R3 (see text);
4: **return** The resulting CPDAG.

Note that the conditional independence queries in this algorithm are organized in ascending order w.r.t. the size of conditioning sets and the size is only up to order $q-1$, where $q$ is the maximum size of the adjacency sets of the nodes in the underlying DAG. This makes this algorithm particularly efficient in computation for large sparse graphs.

Algorithm 1.2 Adjacency search of the PC algorithm (Step 1 in Algorithm 1.1)

**Require:** Conditional independencies among all variables in $V$.

1: Construct a fully connected undirected graph $C$ on the vertex set $V$;
2: Let $\ell = 0$;
3: repeat
   4: repeat
      5: Select a (new) ordered pair of vertices $(X_i, X_j)$ that are adjacent in $C$ and satisfy $|\text{adj}(C, X_i)\backslash\{X_j\}| \geq \ell$;
   6: repeat
      7: Choose a (new) set $S \subseteq \text{adj}(C, X_i)\backslash\{X_j\}$ with $|S| = \ell$;
      8: if $X_i$ and $X_j$ are conditionally independent given $S$ then
         9: Delete edge $X_i - X_j$ from $C$;
      10: Let $\text{sepset}(X_i, X_j) = \text{sepset}(X_j, X_i) = S$;
     end if
   12: until $X_i$ and $X_j$ are no longer adjacent in $C$ or all $S \subseteq \text{adj}(C, X_i)\backslash\{X_j\}$ with $|S| = \ell$ have been considered
13: until all ordered pairs of adjacent vertices $(X_i, X_j)$ in $C$ with $|\text{adj}(C, X_i)\backslash\{X_j\}| \geq \ell$ have been considered
14: Let $\ell = \ell + 1$;
15: until all pairs of adjacent vertices $(X_i, X_j)$ in $C$ satisfy $|\text{adj}(C, X_i)\backslash\{X_j\}| < \ell$
16: **return** The skeleton $C$ and the separation sets.

Step 2 determines the v-structures according to the separation sets. Specifically, it orients an unshielded triple $(X_i, X_j, X_k)$ as a v-structure if and only if $X_j \notin \text{sepset}(X_i, X_k)$. However, this procedure is sensitive to mistakes in conditional independence tests, which may be propagated to the follow-up steps resulting in
more mistakes. In order to improve the robustness of orienting the v-structures, a conservative version of the PC algorithm is proposed [Ramsey et al., 2006a]. In this version, all potential v-structures $X_i - X_j - X_k$ are checked in the following way. It tests whether $X_i$ and $X_k$ are independent conditioning on all subsets of the neighbors of $X_i$ and all subsets of the neighbors of $X_k$. If $X_j$ is in all separating sets or no separating set, no further action is taken and the usual PC is continued. If, however, $X_j$ is in only some separating sets, the triple $X_i - X_j - X_k$ is marked as ‘ambiguous’. Moreover, if no separating set is found among the neighbors, the triple is also marked as ‘ambiguous’. An ambiguous triple is not oriented as a v-structure. Furthermore, no further orientation rule that needs to know whether $X_i - X_j - X_k$ is a v-structure or not is applied. Instead of using the conservative version, which is quite strict towards the v-structures, Colombo and Maathuis [2014] introduce a less strict version for the v-structures called majority rule. In this case, the triple $X_i - X_j - X_k$ is marked as ‘ambiguous’ if and only if $X_j$ is in exactly 50 percent of such separating sets or no separating set was found. If $X_j$ is in less than 50 percent of the separating sets it is set as a v-structure, and if in more than 50 percent it is set as a non v-structure.

Step 3 lastly orients as many of the remaining undirected edges as possible by repeatedly applying the following orientation rules:

- **R1:** Orient $X_j - X_k$ into $X_j \rightarrow X_k$ whenever there is a structure $X_i \rightarrow X_j - X_k$ where $X_i$ and $X_k$ are not adjacent (otherwise a new v-structure is created);

- **R2:** Orient $X_i - X_j$ into $X_i \rightarrow X_j$ whenever there is a chain $X_i \rightarrow X_k \rightarrow X_j$ (otherwise a directed cycle is created);

- **R3:** Orient $X_i - X_j$ into $X_i \rightarrow X_j$ whenever there are two chains $X_i - X_k \rightarrow X_j$ and $X_i - X_l \rightarrow X_j$ where $X_k$ and $X_l$ are not adjacent (otherwise a new v-structure or a directed cycle is created).

**Application to Gaussian data** We notice that the key part of the PC algorithm is to test conditional independence. When a random vector $X$ follows a multivariate Gaussian distribution, i.e., $X \sim \mathcal{N}(0, C)$, the PC algorithm considers the so-called partial correlation, denoted by $\rho_{ij|Q}$, which can be estimated through the correlation matrix $C$ [Anderson, 2003]. Specifically, given observations of $X$ and significance level $\alpha$, classical decision theory yields

$$X_i \perp \!\!\!\!\!\perp X_j | X_Q \Leftrightarrow \sqrt{n - |Q| - 3} \frac{1}{2} \log \left( \frac{1 + \hat{\rho}_{ij|Q}}{1 - \hat{\rho}_{ij|Q}} \right) \leq \Phi^{-1}(1 - \alpha/2),$$

(1.1)

where $i \neq j$, $Q \subseteq \{1, 2, \ldots, p\}\setminus\{i, j\}$. Hence, the PC algorithm requires the sample (Pearson) correlation matrix $\hat{C}$ (to estimate $\rho_{ij|Q}$) and the sample size $n$ as input. High-dimensional consistency of the PC algorithm for Gaussian data is shown under some mild assumptions on the sparsity of the true underlying structure [Kalisch and Bühlmann, 2007].
Chapter 1

Application to nonparanormal data  Harris and Drton [2013] use rank correlations, typically Spearman’s $\rho$ and Kendall’s $\tau$, to replace the Pearson correlations for estimating the correlation matrix, which extends the PC algorithm to the broader class of Gaussian copula models (see next section for the formal definition) but limited to continuous margins, also called nonparanormal models. High-dimensional consistency of the resulting Rank PC algorithm has also been proven.

1.5 Gaussian Copula Models

Definition 1.1 (Gaussian Copula Model). Consider a latent random vector $\mathbf{Z} = (Z_1, \ldots, Z_p)^T$ and an observed random vector $\mathbf{Y} = (Y_1, \ldots, Y_p)^T$, satisfying

$$
\mathbf{Z} \sim \mathcal{N}(0, \mathbf{C}), \\
Y_j = F_j^{-1}[\Phi(Z_j)], \forall j = 1, \ldots, p,
$$

where $\mathbf{C}$ denotes the correlation matrix of $\mathbf{Z}$, $\Phi(\cdot)$ is the standard normal cumulative distribution function, and $F_j^{-1}(t) = \inf\{y : F_j(y) \geq t\}$ is the pseudo-inverse of a cumulative distribution function $F_j(\cdot)$. Then this model is called a Gaussian copula model with correlation matrix $\mathbf{C}$ and univariate margins $F_j(\cdot)$.

This model provides an elegant way to conduct multivariate data analysis for two reasons. First, it raises the theoretical framework in which multivariate associations can be modeled separately from the univariate distributions of the observed variables [Nelsen, 2007]. This is very important in practice, because in many studies people are generally concerned with statistical associations among the variables but not necessarily the scale on which the variables are measured [Hoff, 2007]. Second, the use of copulas is advocated to model multivariate distributions involving diverse types of variables, say binary, ordinal, and continuous [Dobra et al., 2011]. A variable $Y_j$ that takes a finite number of ordinal values \{1, 2, \ldots, $M$\} with $M \geq 2$, is incorporated into our model by introducing a latent Gaussian variable $Z_j$, which complies with the well-known standard assumption for an ordinal variable [Muthén, 1984], i.e.,

$$
Y_j = m, \text{ if } \theta_{m-1} < Z_j \leq \theta_m,
$$

where $m \in \{1, 2, \ldots, M\}$ and $\theta = \{\theta_m\}_{m=0}^M$ are the thresholds ($-\infty = \theta_0 < \theta_1 < \ldots < \theta_M = +\infty$).

Because of these two advantages, recent years have seen wide usage of this model in a variety of research fields, e.g., factor analysis [Murray et al., 2013; Gruhl et al., 2013], undirected graphical modeling [Dobra et al., 2011; Liu et al., 2012; Fan et al., 2017], and causal structure learning [Harris and Drton, 2013]. As an example, Dobra et al. [2011] makes use of a Gaussian-copula-based graphical model to determine the conditional independence relationships in the National Long Term Care Survey functional disability data, which contains 6 binary variables measuring activities of daily living and another 10 binary variables for instrumental activities of daily living. See Dobra et al. [2011] for a detailed description of this example.
Note that an underlying assumption behind the copula model is that the dependencies among observed variables are due to the interactions among their corresponding latents, in the sense that observed variables do not interact directly but via their latents, as shown in Figure 1.3. From a causal perspective, the whole model consists of two parts: the (underlying) causal structure over latent variables, and the causal relations from latents to their corresponding observed variables, i.e., \( Z_j \rightarrow Y_j, \forall j \). Our goal in this thesis is to infer the causal structure among latent variables from observations. The implicit assumption is that possible interventions act on the latent variables, not on the observations themselves, much along the lines of Chapter 10 in Spirtes et al. [2000].

1.6 Outline of this Thesis

In this section, we concisely motivate the research questions addressed in this thesis and summarize our methods proposed in later chapters.

We mainly focus on three questions that arise very often in practice: (1) how to consistently infer causal relations when the system contains both discrete and continuous variables; (2) how to deal with missing values in the data; (3) how to discover the causal structure over latent concepts that cannot be measured directly such as attitude, intelligence, and motivation. We now explain these questions in more details and briefly overview our solutions, respectively.

Mixed data

While the existing causal discovery algorithms are typically designed for either fully continuous data or purely discrete data, it is very prevalent to have both in a practical problem. Tables 1.1 and 1.2 give two real-world examples from the Quality in Acute Stroke Care (QASC) study [Middleton et al., 2011] and the Longitudinal Study of American Youth (LSAY) [Baraldi and Enders, 2010] respectively, providing a summary of part of the variables therein. From the second column of these tables,
we notice that both studies contain diverse types of variables. Therefore, our first research task is to generalize the algorithms to mixed continuous and discrete cases, which is presented in Chapters 2 and 3.

Table 1.1: Summary of partial variables in the QASC.

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Variable Type</th>
<th>Missing Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>binary</td>
<td>0%</td>
</tr>
<tr>
<td>age</td>
<td>continuous</td>
<td>5.89%</td>
</tr>
<tr>
<td>education level</td>
<td>ordinal</td>
<td>15.95%</td>
</tr>
<tr>
<td>ATSI</td>
<td>binary</td>
<td>17%</td>
</tr>
<tr>
<td>time to presentation</td>
<td>continuous</td>
<td>1.69%</td>
</tr>
<tr>
<td>modified Rankin scale</td>
<td>ordinal</td>
<td>9.48%</td>
</tr>
</tbody>
</table>

Table 1.2: Summary of partial variables in the LSAY.

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Variable Type</th>
<th>Missing Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>math encouragement in 9th grade</td>
<td>binary</td>
<td>16.2%</td>
</tr>
<tr>
<td>college encouragement in 9th grade</td>
<td>continuous</td>
<td>14.8%</td>
</tr>
<tr>
<td>academic encouragement in 9th grade</td>
<td>ordinal</td>
<td>16.2%</td>
</tr>
<tr>
<td>grade 12 math scores</td>
<td>binary</td>
<td>33.4%</td>
</tr>
</tbody>
</table>

Chapter 2 proposes the Copula PC algorithm for causal discovery, which is a two-step approach. The first step applies a Gibbs sampler on rank-based data to draw samples of the underlying correlation matrix. These are then translated into an average correlation matrix and an effective sample size (to replace the original sample size to account for information loss incurred by discrete data), which are input to the standard PC algorithm for causal discovery. We also derive a stable version, referred to as Stable Copula PC, which runs PC repeatedly on a number of correlation matrix samples and ensembles the outputs to get a more robust result.

Instead of using Gibbs sampling to estimate the underlying correlation matrix, Chapter 3 proposes a novel heterogeneous estimator, which tests the rank correlation between two continuous variables, the polyserial correlation between a continuous variable and an ordinal variable, and the polychoric correlation between two ordinal variables. The resulting causal discovery algorithm is referred to as the Hetcor PC algorithm. The significance of Hetcor PC over Copula PC is that we prove the convergence rate of the heterogeneous correlation estimator, based on
which we show the probability error bound and high-dimensional consistency of the Hetcor PC algorithm.

**Missing values**

The second research question concerns data with missing values, because all branches of experimental science are plagued by this problem in practice [Little and Rubin, 1987; Poleto et al., 2011], e.g., failure of sensors or drop-outs of subjects in a longitudinal study. See the third column of Tables 1.1 and 1.2 for an intuitive demonstration. Because of its pervasive nature, some methodologists have described missing data as “one of the most important statistical and design problems in research” [Baraldi and Enders, 2010]. We will introduce our solutions to the problem of missing values in Chapter 4, where we show under which conditions and to what extent it is still possible to recover the causal structure when the data are not complete.

**Latent concepts**

In psychology, behavior science and many other sciences, it is very common to come across concepts that cannot be measured directly, such as depression, anxiety, attitude. In order to get a grip on these latent concepts, one commonly-used strategy is to construct a measurement model for such a latent concept, in the sense that domain experts design a set of measurable “items” or survey “questions” that are considered to be indicators of the latent. Figure 1.4 shows an example, where the latent concept ‘intelligence’ is measured by four manifest variables and ‘academic performance’ is measured by another three indicators.\(^1\) Then, the follow-up question researchers are usually interested in is to learn causal relations between latent concepts, e.g., if intelligence is a direct cause of academic performance. More broadly, scientists also want to incorporate some other (explicit) variables into their study, such as gender and age. For example, what is the interacting mechanism among gender, age, intelligence, and academic performance. Our third research task is dedicated to answering this kind of question, which is presented in Chapters 5 and 6.

Chapter 5 introduces the Gaussian copula factor model, a convenient tool to model our research problem, in which a factor can be connected to either one or more observed variables (indicators). Factors with multiple indicators are used to model latent concepts corresponding to psychological traits, and factors with a single indicator are used to model the explicit variables. A novel Bayesian inference approach for this model is subsequently presented in this chapter. On top of the inferred posterior distribution of the correlation matrix over factors (part of the results obtained in Chapter 5), Chapter 6 focuses on learning the causal relationships among latent concepts and explicit variables.

\(^1\)This example is from https://en.wikipedia.org/wiki/Structural_equation_modeling.
Figure 1.4: An example to show how researchers usually measure latent concepts, where the latent concepts are drawn as circles while measured (observed) variables are shown as squares.

We note that Chapters 2 – 5 are self-contained, which can be read in arbitrary order, whereas Chapter 6 relies on the model and inference method proposed in Chapter 5. The R code and research data are publicly available in our Github repository\(^2\) except for the dataset about children with Attention Deficit Hyperactivity Disorder used in Chapter 6. With the code and data, one could reproduce the experimental results presented in this thesis.

\(^2\)https://github.com/cuiruifei
Chapter 2

Copula PC Algorithm for Causal Discovery from Mixed Data

We propose the ‘Copula PC’ algorithm for causal discovery from a combination of continuous and discrete data, assumed to be drawn from a Gaussian copula model. It is based on a two-step approach. The first step applies Gibbs sampling on rank-based data to obtain samples of correlation matrices. These are then translated into an average correlation matrix and an effective sample size, which in the second step are input to the standard PC algorithm for causal discovery. A stable version naturally arises when rerunning the PC algorithm on different Gibbs samples. Our ‘Copula PC’ algorithm extends the ‘Rank PC’ algorithm, which has been designed for Gaussian copula models for purely continuous data. In simulations, ‘Copula PC’ outperforms ‘Rank PC’ in cases with mixed variables, in particular for larger sample size, at the expense of a slight increase in computation time.

2.1 Introduction

Causal discovery, or causal structure learning [Pearl, 2009b], aims to find an underlying directed acyclic graph (DAG), which represents direct causal relations between variables. It is a very popular approach for multivariate data analysis and therefore is widely studied in the past few years, resulting in lots of algorithms. The PC [Spirtes et al., 1993, 2000] algorithm can be considered the reference causal discovery algorithm. It makes use of conditional independence tests to build the underlying DAG from observations. Starting with a complete undirected graph, the PC algorithm removes edges recursively according to the outcome of the con-
ditional independence tests. This procedure yields an undirected graph, also called the skeleton. After applying various edge orientation rules, it finally gives back a partially directed graph to represent the underlying DAGs.

One advantage of the PC algorithm is that it is computationally feasible for sparse graphs even with thousands of variables. Therefore, it is widely used in high-dimensional settings, generating a variety of applications [Maathuis et al., 2010; Stekhoven et al., 2012]. Also, open-source software is available like the R package `pcalg` [Kalisch et al., 2012] and the Tetrad project [Scheines et al., 1998].

When applied to Gaussian models, the PC algorithm tests conditional independence using partial correlation based on Pearson correlations between variables: when the joint distribution is a multivariate Gaussian, pairwise conditional independence is equivalent to the vanishing of the corresponding partial correlation [Lauritzen, 1996]. Following Harris and Drton [2013], we will refer to the PC algorithm for Gaussian models as the ‘Pearson PC’ algorithm. As input it takes the correlation matrix of the observed data and the sample size. The sample size is needed for the conditional independence tests: the higher the sample size, the more reliable the observed correlation matrix as an estimate of the (unknown) true correlation matrix, and the more easily the null hypothesis of conditional independence (given the same value for the partial correlation and the significance level) gets rejected. Under relatively mild assumptions regarding the sparseness of the true underlying DAG, the ‘Pearson PC’ algorithm shows uniform consistency [Kalisch and Bühlmann, 2007].

Harris and Drton [2013] extend the PC algorithm to non-parametric Gaussian (nonparanormal) models, i.e., continuous data assumed to be generated from a Gaussian copula model. They propose to apply the standard PC algorithm, but then replacing the Pearson correlation matrix with rank-based measures of correlation. The so-called ‘Rank PC’ (RPC) algorithm works as well as the ‘Pearson PC’ algorithm on normal data and much better on non-normal data, and is shown to be uniformly consistent in high-dimensional settings.

In this chapter, we aim to generalize the ‘Pearson PC’ and ‘Rank PC’ algorithm to Gaussian copula models that can also handle binary and ordinal variables. The ‘Rank PC’ algorithm is explicitly limited to the continuous situation, where ties appear with probability zero, making ranks well-defined. In the presence of binary and ordinal variables, ties make the rank correlations between observed variables different from those between the corresponding latent variables in the Gaussian copula setting. Ignorance of this difference typically leads to underestimates of the (absolute) correlations [Hoff, 2007].

It is tempting to follow a similar two-step approach as for ‘Rank PC’: first estimate the correlation matrix in the latent space and then use this as input to the standard PC algorithm. This, however, is not as straightforward as it may seem, for two reasons. First, because of the ties, estimating the correlation matrix of Gaussian copula models for mixed data is considerably more complicated. Second, the ties imply a loss of information, which makes that our estimate of the correlation matrix will tend to be less reliable than in the fully continuous case, which
should be accounted for when applying the conditional independence tests in the PC algorithm.

To solve both issues, we propose to make use of a Gibbs sampling procedure, specifically the one derived by Hoff [2007] based on the so-called extended rank likelihood. This procedure is relatively straightforward and easy to implement (see the code in the Appendix of Hoff [2007]). For purely Gaussian data, the correlation matrix samples follow a specific kind of inverse-Wishart distribution [Barnard et al., 2000], which we refer to as the projected inverse-Wishart distribution. Projected inverse-Wishart distributions are characterized by two parameters: the scale matrix and the degrees of freedom; the former relates to the average correlation matrix and the latter to the sample size. As we will show, under the projected inverse-Wishart, the variance of each off-diagonal element of the correlation matrix is an approximate function of its expectation and the degrees of freedom: the more degrees of freedom, the smaller the variance. The idea is now to estimate the scale matrix and degrees of freedom from the Gibbs samples of more general Gaussian copula models on mixed data, as if they were also drawn from a projected inverse-Wishart distribution. The scale matrix is translated into a correlation matrix and the degrees of freedom into a so-called ‘effective sample size’, to take into account the reliability of our estimate of the correlation matrix. These are then input to the standard PC algorithm for causal discovery.

We refer to our two-step procedure as the ‘Copula PC’ (CoPC) algorithm. We also derive a stable version, referred to as ‘Stable Copula PC’ (SCPC), which runs PC repeatedly on a number of Gibbs samples. Experimental results show that both CoPC and SCPC outperform the current ‘Rank PC’ algorithm in mixed databases with discrete and continuous variables.

The rest of this chapter is organized as follows. Section 2.2 proposes an approximate inference method for the correlation matrix and the effective sample size based on the projected inverse-Wishart distribution, and then derives the resulting algorithms CoPC and SCPC. Section 2.3 compares CoPC and SCPC with the ‘Rank PC’ algorithm on simulated data and provides an illustration on real-world data of ADHD patients. Section 2.4 concludes and discusses future work.

2.2 Method

In this section, we introduce an approximate inference approach for the underlying correlation matrix and the effective sample size from mixed data. Section 2.2.1 introduces the projected inverse-Wishart distribution and its application to Gaussian models. Section 2.2.2 discusses how to obtain correlation matrix samples from mixed data using a Gibbs sampling procedure. Section 2.2.3 shows how to use these samples to estimate the two parameters of the projected inverse-Wishart distribution: the scale matrix (as the underlying correlation matrix) and the degrees of freedom (as the effective sample size). Section 2.2.4 derives the resulting Copula PC algorithm and the Stable Copula PC algorithm.
2.2.1 Projected Inverse-Wishart Distribution

Priors on correlation matrices are typically derived by choosing the inverse-Wishart distribution, denoted by $W^{-1}(\Sigma; \Psi_0, \nu)$, as a prior on covariance matrices and then turning the covariance matrices into a correlation matrix to end up with an implied distribution on the correlation matrix. We choose $\Sigma$ from $W^{-1}(\Sigma; \Psi_0, \nu)$ and write

$$P(C) = \mathcal{P}W^{-1}(C; \Psi_0, \nu) \quad (2.1)$$

where $C_{ij} = \Sigma_{ij}/\sqrt{\Sigma_{ii}\Sigma_{jj}}$, $\forall i, j$. Since many covariance matrices possibly correspond to the same correlation matrix, the above process can be considered as a projection from covariance matrices to a correlation matrix. Therefore, we refer to this distribution on correlation matrix $C$ as a projected inverse-Wishart distribution.

For Gaussian models, the projected inverse-Wishart distribution gives exact inference [Murphy, 2007]. Specifically, given data $Z = (z_1, \ldots, z_n)^T$, the posterior reads

$$P(\Sigma|Z) = W^{-1}(\Sigma; \Psi_0 + \Psi, \nu + n) \quad \text{and} \quad P(C|Z) = \mathcal{P}W^{-1}(C; \Psi_0 + \Psi, \nu + n),$$

with $\Psi = Z^T Z$. Also, the projected inverse-Wishart is scale invariant [Barnard et al., 2000; Huang et al., 2013], in the sense that we can make the posterior distribution on correlation matrices independent of the scale of the data by choosing $\Psi_0 = 0$, or perhaps better, $\Psi_0 = \epsilon 1$ in the limit $\epsilon \downarrow 0$.

For Gaussian copula models, although there is no analytical expression, we still expect that the posterior $P(C|Y)$ can be approximated through a projected inverse-Wishart distribution, i.e., $P(C|Y) \approx \mathcal{P}W^{-1}(C; \Psi, \nu)$ for some $\Psi$ and $\nu$.

2.2.2 Gibbs Sampler Based on Extended Rank Likelihood

Hoff [2007] describes an elegant procedure to obtain samples from $P(C|Y)$ for a Gaussian copula model. The essence is that we only consider the ranks among observations, hence the name extended rank likelihood, ignoring the actual variables. Since the cumulative distribution functions $F_j(Y_j)$ are non-decreasing, observing $y_{i_1,j} < y_{i_2,j}$ implies that $z_{i_1,j} < z_{i_2,j}$, where $y_{i_1,j}$ denotes the $i_1^{th}$ observation of the $j^{th}$ component of random vector $Y$. To be precise, observing $Y = (y_1, \ldots, y_n)^T$ tells us that $Z = (z_1, \ldots, z_n)^T$ must lie in the set

$$\left\{ Z \in \mathbb{R}^{n \times p} : \max \{z_{k,j} : y_{k,j} < y_{i,j}\} < z_{i,j} < \min \{z_{k,j} : y_{i,j} < y_{k,j}\} \right\} .$$

Strong posterior consistency for $C$ under the extended rank likelihood has been proved in the situation with both discrete and continuous marginal distribution functions [Murray et al., 2013].

An off-the-shelf sampling algorithm based on the extended rank likelihood is full Gibbs sampling [Hoff, 2007]. The code of this sampling algorithm is provided in the Appendix of Hoff [2007]. In this algorithm, each component of $Z$ is initialized
according to the rank information of the corresponding component of $Y$, after which each component is resampled alternatively. Here we propose a slight modification by just resampling the discrete components instead of all of them. Experimental tests reveal that the results of this faster sampling approach are indistinguishable from Hoff’s original Gibbs sampler. Although this modification is quite straightforward, it significantly reduces computation time because sampling continuous variables is far more time-consuming than sampling discrete ones in Hoff’s Gibbs sampler. We will refer to this modified sampling algorithm as $\text{SamplingAlgo}$.

So, given the observed data $Y$, samples on the underlying correlation matrix, denoted by $\{C^{(1)}, \ldots, C^{(m)}\}$, can be obtained using $\text{SamplingAlgo}$.

### 2.2.3 Estimating Correlation Matrix and Effective Sample Size

This section aims to estimate the underlying correlation matrix and the effective sample size from the obtained samples.

Theorem 2.1 suggests a procedure to estimate the parameters $\Psi$ and $\nu$ from samples of a projected inverse-Wishart distribution $\mathcal{PW}^{-1}(C; \Psi, \nu)$.

**Theorem 2.1.** If the correlation matrix $C$ follows a projected inverse-Wishart distribution with parameters $\Psi$ ($\Psi_{ii} = 1$) and $\nu$, i.e.,

$$P(C) = \mathcal{PW}^{-1}(C; \Psi, \nu),$$

then for each off-diagonal element $C_{ij}$ ($i \neq j$) and large $\nu$, we have

$$\mathbb{E}[C_{ij}] \approx \Psi_{ij} \quad \text{and} \quad \text{Var}[C_{ij}] \approx \frac{(1 - (\Psi_{ij})^2)^2}{\nu}.$$ 

**Proof.** See Appendix 2.A. \hfill \Box

According to Theorem 2.1, the mean over samples of $C$ is a good approximation of $\Psi$. As for $\nu$, we have,

$$\nu \approx \frac{(1 - (\mathbb{E}[C_{ij}])^2)^2}{\text{Var}[C_{ij}]}.$$  \hfill (2.2)

The idea now is to apply the same estimates, as if the samples obtained by Gibbs sampling the Gaussian copula model on mixed data also (approximately) follow a projected inverse-Wishart distribution. Specifically, for the effective sample size $\hat{n}$, we propose to take the average over all $p(p - 1)/2$ estimates on $\nu$ that can be computed by applying (2.2) to each upper triangular element of a $p$-dimensional correlation matrix $C$. 
2.2.4 Copula PC Algorithm and Stable Copula PC Algorithm

Now, we turn the previous results into a working algorithm. The two key input arguments of the ‘Pearson PC’ algorithm are the correlation matrix and the sample size. In the general Gaussian copula model, we take the mean over \( \{C^{(1)}, \ldots, C^{(m)}\} \) and the mean over \( p(p - 1)/2 \) estimates on \( \nu \) as the two arguments respectively, resulting in the Copula PC algorithm.

Next, we introduce a stable version of the Copula PC algorithm. We take \( l \) instances from all the \( m \) samples. For each instance, a corresponding graph can be obtained via the ‘Pearson PC’ algorithm using the earlier estimated effective sample size, by which a collection of \( l \) graphs can be generated, denoted by \( \{\tilde{G}_1, \ldots, \tilde{G}_l\} \). We keep those edge marks that emerge with a probability higher than a pre-defined threshold \( \beta \) and remove the others, leading to a resulting graph. Since this resulting graph seemingly contains only ‘stable’ edge marks, we call this method stable Copula PC algorithm (SCPC). The size of \( l \) has a linear influence on running time because choosing \( l \) means the ‘Pearson PC’ algorithm would run \( l \) times. As for \( \beta \), a small value means keeping more edge marks and vice versa. The Copula PC algorithm and its stable version are summarized in Algorithm 2.1.

2.3 Experiments

In this section, we first verify the property of the projected inverse-Wishart distribution described by Equation (2.2) and check whether it still holds in the presence of discrete variables. Then, we compare the proposed CoPC and SCPC with the ‘Rank PC’ algorithm on simulated data and give an illustration on real-world data of ADHD patients.

Following Kalisch and Bühlmann [2007], we simulate random DAGs and draw samples from the distributions faithful to them. Firstly, we generate an adjacency matrix \( A \), whose entries are zero or in the interval \([0.1, 1]\). There exists a directed edge from \( i \) to \( j \) in the corresponding DAG, if \( i < j \) and \( A_{ji} \neq 0 \). The DAGs generated in this way have the property \( \mathbb{E}(N_i) = s(p - 1) \), where \( N_i \) is the number of neighbors of node \( i \), and \( s \) is the probability that there is an edge between any two nodes, called the sparseness parameter. Then, the samples of a random vector \( Z \) are drawn through

\[
Z_j = \sum_{i<j} A_{ji} Z_i + \epsilon_j ,
\]

with \( \epsilon_j \sim \mathcal{N}(0, 1), \forall j \). The data generated in this way follow a multivariate Gaussian distribution.
Algorithm 2.1 Copula PC algorithm (CoPC) and Stable Copula PC algorithm (SCPC)

1: **Input**: Observations $Y$; Initialized parameters $m, l, \beta$
2: **Output**: Causal graph $G_c$ by CoPC, $G_s$ by SCPC
3: $C^{(1)}, \ldots, C^{(m)} = \text{SamplingAlgo}(Y)$
4: for all $C_{ij}$ with $i < j$ (upper triangular elements) do
5: Compute and store $\nu_k = \frac{(1 - (\mathbb{E}[C_{ij}])^2)}{\text{Var}[C_{ij}]}$
6: end for
7: $\hat{n} =$ the average over $\{\nu_1, \ldots, \nu_{p(p-1)/2}\}$
8: if CoPC then
9: $\hat{C} = \frac{1}{m} \sum_{j=1}^{m} C^{(j)}$
10: $G_c = \text{PC}(\hat{C}, \hat{n})$
11: else
12: Choose $l$ ($l < m$) instances from $C^{(1)}, \ldots, C^{(m)}$
13: for $i = 1 : l$ do
14: Compute and store $\tilde{G}_i = \text{PC}(C^{(l_i)}, \hat{n})$
15: end for
16: for all edge marks do
17: $e =$ the number of graphs containing the current edge mark
18: if $e/l > \beta$ then
19: keep the edge mark
20: end if
21: $G_s =$ all kept edge marks among $\{\tilde{G}_1, \ldots, \tilde{G}_l\}$.
22: end for
23: end if

2.3.1 Estimation for the Effective Sample Size

As argued in Section 2.2.3, the expectation and variance of the elements of correlation matrices drawn from a projected inverse-Wishart distribution are strongly related. To check this relationship, we proceed as follows: 1) we generate a random $p$-dimensional correlation matrix $\Psi$; 2) we draw 500 samples from a projected inverse-Wishart distribution with parameters $\Psi$ and $\nu$; 3) for each upper triangular element, we plot its variance against its expectation.

The left panel in Figure 2.1 shows a typical result for $p = 20$ and $\nu = 1000$. We see that almost all pairs are distributed around the theoretical curve (solid line) especially when the expectation is far from zero, which indicates that it is indeed possible to infer $\nu$ of a projected inverse-Wishart distribution via the expectation and variance of off-diagonal elements.

Next, we study how our inference method works for estimating $\hat{n}$ in different cases. We first generate $n$ samples of $Z$ using Equation (2.3) and discretize some of the variables to obtain the simulated samples of the observed random vector $Y$. 
Figure 2.1: The relationship between the expectation and the variance of the elements of sampled correlation matrices. Left panel: The samples are drawn from a given projected inverse-Wishart distribution. Right panel: The samples are drawn via SamplingAlgo, with circles for binary cases, triangles for ordinal cases with 4 levels, and squares for continuous cases.

Then, we run SamplingAlgo to get samples of the underlying $C$. The results for $p = 20$ and $n = 1000$ for different cases are shown in Figure 2.1 (right panel), where ‘bins=2’ means that all variables are binary, ‘bins=4’ means that all variables are ordinal with 4 levels and ‘continuous’ means that all variables are kept continuous. We take $(1 - (E[C_{ij}])^2)^2$ for the $x$-axis and $n \times \text{Var}[C_{ij}]$ for the $y$-axis, so that all data points are expected to be distributed around a straight line with slope $n/\hat{n}$. For purely continuous variables, a straight line with slope 1 gives an almost perfect fit, as expected. For ordinal and binary variables, we still find a clear trend, but mild deviations from a perfect straight line, indicating that the projected inverse-Wishart distribution is a fine, but not perfect approximation of the exact posterior. The stronger the discretization, the larger the slope $n/\hat{n}$ and thus the lower our estimated effective sample size.

More extensive experiments (not shown) done with different numbers of variables, data points, Gibbs samples and sparseness parameters, reveal that these hardly influence the general picture, as long as the sample size and the number of Gibbs samples are both at least 100.

### 2.3.2 Causal Discovery on Simulations

In this section, we compare CoPC and SCPC with the ‘Rank PC’ [Harris and Drton, 2013] algorithm. The implementation of the standard PC algorithm is from the R package pcalg.

We first generate multivariate normal data ($p$ variables) via Equation (2.3). After that, 25% of all $p$ variables are discretized into binary variables, and another 25% of them are discretized into ordinal variables with 5 levels. In this way, we simulate the observations of $Y$ which are generated from a Gaussian copula model with both discrete and continuous margins.
Three measures are used to test the performance: 1) percentage of correct edges in the resulting skeleton, usually called true positive rate (TPR); 2) percentage of spurious edges, usually called false positive rate (FPR); 3) Structural Hamming Distance (SHD), counting the number of edge insertions, deletions, and flips in order to transfer the estimated CPDAG into the correct CPDAG [Tsamardinos et al., 2006]. The first two measures are for the skeleton while SHD is for the CPDAG. A smaller SHD indicates better performance.

Next, we compare the performance of three versions of the PC algorithm, RPC, CoPC, and SCPC in terms of TPR, FPR, and SHD. We restrict the significance level to $\alpha = 0.01$, which has been shown to yield the best overall SHD [Kalisch and Bühlmann, 2007]. For CoPC, we drop the first 20 Gibbs samples and save the next 100 samples ($m = 100$). For SCPC, we take $l = 20$ equidistant samples, so $\{C^{(1)}, C^{(6)}, \ldots, C^{(96)}\}$, and choose $\beta$ such that the TPR for SCPC is more or less equal to that of RPC, which amounts to $\beta = 0.4$ for sparse graphs with 10 nodes, $\beta = 0.45$ for sparse graphs with 50 nodes, and $\beta = 0.3$ for dense graphs. The remaining parameters are set as follows: $p \in \{10, 50\}$, $n \in \{500, 1000, 2000, 5000\}$, and $\mathbb{E}[N] \in \{2 \text{ (Sparse)}, 5 \text{ (Dense)}\}$.

The comparative results in Figure 2.2 (10 nodes) and Figure 2.3 (50 nodes) provide the mean over 100 repeated experiments and errorbars representing 95% confidence intervals. First, for sparse graphs (both small and large graphs), the three algorithms get nearly the same results w.r.t. TPR, but CoPC and SCPC show a large advantage over RPC w.r.t. FPR and SHD except SCPC with large graphs, which becomes more prominent with increasing sample size. Second, for dense graphs, the advantage of CoPC and SCPC over RPC still exists w.r.t. FPR, although seemingly CoPC performs a little worse than SCPC and RPC w.r.t. TPR. Third, we note that the performance of RPC deteriorates seriously w.r.t. FPR with the increase in sample size, while CoPC and SCPC are very stable. Apparently, using sample size as the effective sample size, RPC incurs more false positives especially for larger sample sizes. Overall, CoPC and SCPC clearly outperform RPC, especially in the sparse cases with larger sample sizes.

2.3.3 Application to Real-World Data

In this section, we give an illustration on a real-world dataset about children with Attention Deficit Hyperactivity Disorder (ADHD) [Cao et al., 2006]. It contains 23 variables for 245 subjects. We focus on nine variables as in Sokolova et al. [2014], but keep all subjects with missing values since these are easily handled by the Gibbs sampler. The nine variables considered are: gender (G), attention deficit level (AD), hyperactivity/impulsivity level (HI), verbal IQ (VIQ), performance IQ (PIQ), full IQ (FIQ), aggressive behavior (Agg), medication status (Med), handedness (HN), where four of them (G, Agg, Med, HN) are binary.

We run CoPC and SCPC ($l = 30, \beta = 0.4$) on the dataset and consider prior knowledge that no variable can cause gender. The resulting graphs are shown in
Figure 2.2: Performance of Rank PC, Copula PC, and Stable Copula PC for 10 nodes, showing the mean of TPR, FPR, and SHD over 100 experiments together with 95% confidence intervals. The first row represents the results with sparse graphs ($\mathbb{E}[N] = 2$) while the second row represents those with dense graphs ($\mathbb{E}[N] = 5$).

Figure 2.3: Performance of Rank PC, Copula PC, and Stable Copula PC for 50 nodes, showing the mean of TPR, FPR, and SHD over 100 experiments together with 95% confidence intervals.
Figure 2.4: The resulting graphs by CoPC (left panel) and SCPC (right panel) on ADHD dataset.

Figure 2.4. The graphs suggest that gender has an effect on attention deficit level, which then causes hyperactivity/impulsivity level. This point has been confirmed by many studies [Bauermeister et al., 2007; Willcutt et al., 2000]. It is common that AD and Agg cause patients to take medicine. Also, VIQ, PIQ, and FIQ are connected to each other by bi-directed edges. This indicates that the causal sufficiency assumption is violated, i.e., that there should be a latent common cause related to IQ, as also suggested in Sokolova et al. [2014].

2.4 Conclusion and Discussion

In this chapter we introduced a novel two-step approach for estimating the causal structure underlying a Gaussian copula model on mixed data. The essence is to estimate the correlation matrix in the latent space, which can then be given to any causal discovery algorithm to search for its underlying structure. Ties between the discretized observations incur information loss, making the estimate of correlation matrix less reliable than in fully continuous cases. For this, we introduced the notion of ‘effective sample size’ that can be estimated from the expectation and variance of the correlation matrix elements. Our approach, based on ranks and correlation matrices, is fully scale invariant and has a natural uninformative setting when choosing a uniform distribution over pairwise correlations, which can be adjusted to account for different assumptions.

We like to think of our two-step approach as a general principle, where for each of the two steps one could plug in one’s favorite choice: e.g., a different MCMC method [Kalaitzis and Silva, 2013] or a MAP approach along the lines of Abegaz and Wit [2014] for estimating the correlation matrix and its reliability, and another method, like FCI [Spirtes et al., 2000] or BCCD [Claassen and Heskes, 2012], for causal structure learning. Having generated samples, running the PC algorithm several times to gain an insight into the reliability of structure estimates is an obvious thing to do. Similar procedures have been proposed, e.g., by bootstrapping the original dataset [Dai et al., 2004; Entner and Hoyer, 2010]. In our simulations, the Gibbs sampler appears to converge quite fast, which makes Gibbs sampling cheap compared to running the PC algorithm, in particular for models with many
variables. Our choice to only resample the discrete random variables and not the continuous ones, here also helps. Being fully Bayesian about structure learning as well may be very nice in theory [Friedman and Koller, 2003], but is computationally infeasible in practice for any reasonable number of variables. Altogether, our Bayesian approach to sample correlation matrices in combination with a more frequentist approach towards structure learning attempts to combine the best of both worlds.

Our methods require the setting of just a few parameters: the significance level $\alpha$ to be used in the PC algorithm (typically 0.01 or 0.05), the number of Gibbs samples and burn-in (the more, the better), and for SCPC, the number of instances $l$ in the ensemble (the more, the better), and the threshold $\beta$ (the higher, the more conservative).

Our estimate of the ‘effective sample size’ appears to work nicely in practice, but can and perhaps should be further improved. Instead of considering the variance of the elements of the correlation matrix, one may come up with another, more direct estimate, for example the entropy of the distribution and translate that into an effect sample size. Preliminary attempts in that direction failed by being typically much less robust than the one described in this chapter. Our current estimate gives a single, global value for the effective sample size. Future work may consider estimating a different value for each conditional independence test, since each test only relies on a local structure, involving only part of the variables. Such estimates then can be integrated into the causal discovery algorithm itself. Another line of future research concerns the theoretical analysis of CoPC and SCPC, where it can be studied to what extent and under which conditions consistency can be proven. Our conjecture here is that consistency of our two-step procedure follows from the proven consistency of the two separate steps: Gibbs sampling to estimate the correct correlation matrix $C$ [Murray et al., 2013] and the PC algorithm to arrive at the correct causal structure [Kalisch and Bühlmann, 2007].

2.A Proof of Theorem 2.1

**Theorem 2.1.** If the correlation matrix $C$ follows a projected inverse-Wishart distribution with parameters $\Psi$ ($\Psi_{ii} = 1$) and $\nu$, i.e.,

$$P(C) = \mathcal{PW}^{-1}(C; \Psi, \nu),$$

then for each off-diagonal element $C_{ij}(i \neq j)$ and large $\nu$, we have

$$\mathbb{E}[C_{ij}] \approx \Psi_{ij} \quad \text{and} \quad \text{Var}[C_{ij}] \approx \frac{(1 - (\Psi_{ij})^2)^2}{\nu}.$$

**Proof.** Consider partitioning the matrix $\Sigma$ and $\Psi$ as

$$\Sigma = \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix} \quad \text{and} \quad \Psi = \begin{bmatrix} \Psi_{aa} & \Psi_{ab} \\ \Psi_{ba} & \Psi_{bb} \end{bmatrix}.$$
Then, if $P(\Sigma) = W^{-1}(\Sigma; \Psi, \nu)$, we have

$$
P(\Sigma_{aa}) = W^{-1}(\Sigma_{aa}; \Psi_{aa}, \nu - \dim(b)) ,
$$

(2.4)

$$
P(\Sigma_{bb|a}) = W^{-1}(\Sigma_{bb|a}; \Psi_{bb|a}, \nu) ,
$$

$$
P(\Sigma_{aa}^{-1}\Sigma_{ab}|\Sigma_{bb|a}) = N(\Sigma_{aa}^{-1}\Sigma_{ab}; \Psi_{aa}^{-1}\Psi_{ab}, \Sigma_{bb|a} \otimes \Psi_{aa}^{-1}) ,
$$

where $\dim(b)$ is the dimension of $\Sigma_{bb}$ and $\Sigma_{bb|a} = \Sigma_{bb} - \Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{ab}$ [Eaton, 2007].

Without loss of generality, we restrict our analysis to a two-dimensional system and suppose that we draw

$$
\Sigma \sim W^{-1}\left(\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}, \nu\right) .
$$

Then, according to (2.4), we have

$$
\Sigma_{11} \sim W^{-1}(1, \nu - 1) , \quad \Sigma_{22|1} \sim W^{-1}(1 - \rho^2, \nu) , \quad \Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22|1} \sim N(\rho, \Sigma_{22|1}) .
$$

Rewriting the resulting $\hat{\rho}$ in terms of these variables, we obtain

$$
\hat{\rho} = \frac{\Sigma_{12}}{\sqrt{\Sigma_{11}\Sigma_{22}}} = \frac{(\Sigma_{11}^{-1}\Sigma_{12})\sqrt{\Sigma_{11}}}{\sqrt{\Sigma_{22|1} + \Sigma_{11}(\Sigma_{11}^{-1}\Sigma_{12})^2}} .
$$

(2.5)

Since for large $\nu$,

$$
\mathbb{E}[\Sigma_{11}] = \frac{1}{\nu - 3} \approx \frac{1}{\nu} , \quad \mathbb{E}[\Sigma_{22|1}] = \frac{1 - \rho^2}{(\nu - 2)} \approx \frac{1 - \rho^2}{\nu} ,
$$

$$
\text{Var}[\Sigma_{11}] = \frac{2}{(\nu - 3)^2(\nu - 5)} \approx \frac{2}{\nu^3} , \quad \text{Var}[\Sigma_{22|1}] = \frac{2(1 - \rho^2)^2}{(\nu - 2)^2(\nu - 4)} \approx \frac{2(1 - \rho^2)^2}{\nu^3} .
$$

we can approximate,

$$
\Sigma_{11} \approx \frac{1}{\nu} \left(1 + \sqrt{\frac{2}{\nu}} x\right) , \quad \Sigma_{22|1} \approx \frac{1 - \rho^2}{\nu} \left(1 + \sqrt{\frac{2}{\nu}} y\right) ,
$$

$$
\Sigma_{11}^{-1}\Sigma_{12} \approx \rho + \sqrt{\frac{1 - \rho^2}{\nu}} z ,
$$

(2.6)

where $x$, $y$, and $z$ are independent random variables, all with mean zero and unit variance. Indeed, for large $\nu$, all noise terms scale with $\sqrt{1/\nu}$ relative to the mean, and can hence be ignored when computing the expectation, to yield, as expected,

$$
\mathbb{E}[\hat{\rho}] \approx \rho .
$$

(2.7)

To estimate the variance, we substitute (2.6) into (2.5), and compute (in leading order, and evaluated for $x = y = z = 0$),

$$
\frac{\partial\hat{\rho}}{\partial x} \approx \rho(1 - \rho^2)\sqrt{\frac{1}{2\nu}} , \quad \frac{\partial\hat{\rho}}{\partial y} \approx \rho(1 - \rho^2)\sqrt{\frac{1}{2\nu}} , \quad \frac{\partial\hat{\rho}}{\partial z} \approx (1 - \rho^2)^{3/2}\sqrt{\frac{1}{\nu}} ,
$$

$$
\frac{\partial^2\hat{\rho}}{\partial x^2} \approx \frac{\rho^2(1 - \rho^2)^2}{\nu^2} , \quad \frac{\partial^2\hat{\rho}}{\partial x\partial y} \approx \frac{\rho^2(1 - \rho^2)^2}{\nu^2} , \quad \frac{\partial^2\hat{\rho}}{\partial x\partial z} \approx \frac{\rho^2(1 - \rho^2)^2}{\nu^2} , \quad \frac{\partial^2\hat{\rho}}{\partial y^2} \approx \frac{\rho^2(1 - \rho^2)^2}{\nu^2} .
$$
yielding the variance

\[ \text{Var}[\hat{\rho}] = \left( \frac{\partial \hat{\rho}}{\partial x} \right)^2 + \left( \frac{\partial \hat{\rho}}{\partial y} \right)^2 + \left( \frac{\partial \hat{\rho}}{\partial z} \right)^2 \approx \frac{(1 - \rho^2)^2}{\nu}. \] (2.8)
Chapter 3

Hetcor PC Algorithm for Causal Discovery from Mixed Data

We consider the problem of learning causal structure from mixed continuous and ordinal data, assumed to be generated from a Gaussian copula model. To learn the underlying correlation matrix a heterogeneous correlation estimator is proposed, which tests the rank correlation between two continuous variables, the polyserial correlation between a continuous variable and an ordinal variable, and the polychoric correlation between two ordinal variables. We obtain the convergence rate of this correlation estimator in the large sample limit. Based on this result the error bound and high-dimensional consistency of the resulting ‘Hetcor PC’ algorithm are derived. For explaining the information loss incurred by discrete data, we propose to replace the original sample size with a local effective sample size in the conditional independence test. Experiments on simulations and real-world data show that this replacement improves the performance of causal discovery algorithms. While our interest is in the PC algorithm, the proposed procedure and theoretical results could be applied to other algorithms that test for the vanishing of partial correlations.

3.1 Introduction

Causal models are a powerful tool to work with families of multivariate distributions sharing a set of conditional independence restrictions. A typical representation of causal models is as a directed acyclic graph (DAG). The vertices correspond to random variables and the directed edges represent causal relations. Learning the structure of a causal model, a.k.a. causal discovery [Pearl, 2009b], is to infer the

This chapter is based on the article “Learning Causal Structure from Mixed Continuous and Ordinal Data with the PC Algorithm”, which is under review.
invariant part of the underlying DAG from data. This fundamental problem in causal modeling has been widely studied over the past decade [Spirtes et al., 2000; Colombo et al., 2012; Harris and Drton, 2013; Peters et al., 2014]. The PC algorithm [Spirtes et al., 2000], a reference causal discovery algorithm, uses conditional independence tests to infer the underlying causal structure. Due to its clever search scheme, causal inference with the PC algorithm is computationally feasible for high-dimensional sparse graphs (up to thousands of nodes). Therefore, the PC algorithm attracts extensive attention and generated a variety of recent improvements [Ramsey et al., 2006b; Harris and Drton, 2013; Colombo and Maathuis, 2014].

For Gaussian data, the PC algorithm conducts conditional independence tests using the so-called partial correlation based on standard Pearson correlations: pairwise conditional independence is equivalent to the vanishing of the corresponding partial correlation. Theoretical properties of the algorithm for Gaussian data are investigated by Kalisch and Bühlmann [2007], who show high-dimensional uniform consistency. Harris and Drton [2013] extend the PC algorithm to the broader class of nonparanormal distributions by replacing the standard Pearson empirical correlations with rank-based measures of correlations in the conditional independence tests, and also proved high-dimensional consistency of the resulting ‘Rank PC’ algorithm. For mixed nonparanormal and ordinal data, We propose the ‘Copula PC’ algorithm [Cui et al., 2016], in which the underlying correlation matrix is estimated based on the so-called extended rank likelihood [Hoff, 2007].

However, the lack of theoretical analysis regarding the property of Copula PC lowers its significance. To this end, we propose a novel heterogeneous estimator to learn the underlying correlation matrix, which tests rank correlations between continuous variables, polyserial correlations between continuous and ordinal variables, as well as polychoric correlations between ordinal variables. We derive the convergence rate of this correlation estimator, based on which we derive the probability error bound and high-dimensional consistency of the resulting Hetcor PC algorithm. Another contribution of this chapter is that for explaining the information loss incurred by discrete data, a local effective sample size is proposed to replace the original sample size in the tests for conditional independences. This modification improves the practical performance of Hetcor PC as well as Copula PC, which is empirically shown in the experimental sections.

The rest of this chapter is organized as follows. Section 3.2 presents our heterogeneous correlation estimator, the resulting Hetcor PC algorithm, and the method to learn the local effective sample size. Sections 3.3 and 3.4 provide numerical experimental results. Section 3.5 concludes this chapter and provides some discussions.

### 3.2 Method

In this section, we first introduce our new method to learn the underlying correlation matrix. On its basis we derive the Hetcor PC algorithm and analyze its theoretical
properties. Then we discuss our method to learn the effective sample size to improve the practical performance of Hetcor PC.

3.2.1 Learning Correlation Matrix

For handling mixed continuous and ordinal data, we propose a heterogeneous correlation estimator that consists of three types of correlations: rank-based correlation between two continuous variables, polyserial correlation between a continuous variable and an ordinal variable, and polychoric correlation between two ordinal variables. Throughout this chapter, we use $I^c$ and $I^o$ to denote the indices of continuous variables and ordinal variables respectively.

**Rank-based Correlation**

When both $Y_j$ and $Y_k$ are continuous, i.e., $j, k \in I^c$, we apply the rank-based correlation estimator since ranks have the desirable property that they are invariant with respect to strictly-increasing transformations of the component variables in accordance with the structure of the model [Segers *et al.*, 2014].

From the definition of Gaussian copula models, it is natural to estimate the underlying correlation between two continuous variables via

$$C_{jk} = \text{Corr} \left( \Phi^{-1} \left[ F_j(Y_j) \right], \Phi^{-1} \left[ F_k(Y_k) \right] \right).$$

By plugging in the marginal empirical cumulative distribution function for $Y_j$ and $Y_k$, we obtain the sample version of this estimator

$$\hat{C}_{jk} = \frac{1}{n} \sum_{i=1}^{n} \Phi^{-1} \left( \frac{\text{rank}(y_{ij})}{n+1} \right) \Phi^{-1} \left( \frac{\text{rank}(y_{ik})}{n+1} \right), \text{ if } j, k \in I^c,$$

which is the well-known *normal score rank correlation coefficient* [Klaassen *et al.*, 1997], where rank($y_{ij}$) denotes the rank of $y_{ij}$ among all the $n$ observations of $Y_j$. It has been proven to be an efficient estimator, in the sense that the information for the true correlation contained in a dataset of size $n$ is

$$\mathcal{J}(\hat{C}_{jk}) \approx \frac{n}{(1 - \hat{C}_{jk}^2)^2}, \text{ if } j, k \in I^c,$$

just the same as for the sample Pearson correlation coefficient in bivariate normal cases. One could choose other rank-based estimators, e.g., Kendall’s $\tau$ and Spearman’s $\rho$, which are also widely used in nonparanormal models and show good statistical properties [Liu *et al.*, 2012; Harris and Drton, 2013].
Polyserial Correlation

Let $Y_j$ and $Y_k$ denote a continuous variable and an ordinal variable respectively, i.e., $j \in I^c$, $k \in I^o$. We estimate the correlation between the two variables using a two-step approach.

**Step 1.a** Since the standard polyserial correlation concerns the mixture of a Gaussian variable and an ordinal variable [Olsson et al., 1982], our first step is to estimate the pseudo Gaussian data $\hat{z}_{ij}$: $\{\hat{z}_{ij}\}_{i=1}^n$ (the realization of $Z_j$) from the observed data $y_j = \{y_{ij}\}_{i=1}^n$ (the realization of $Y_j$). For this, we plug-in the empirical cumulative distribution function of $Y_j$, i.e.,

$$\hat{z}_{ij} = \Phi^{-1} \left( \frac{\text{rank}(y_{ij})}{n+1} \right), \forall i,$$

where $\mathbb{1}_E$ is the indicator of event $E$. The second equality holds for continuous data.

**Step 1.b** Assume that the ordinal variable $Y_k$ takes values $\{1, 2, \ldots, M_k\}$ with $M_k \geq 2$, which could vary across variables. In our copula model, each $Y_k$ is supposed to be generated by discretizing the corresponding latent Gaussian variable $Z_k$, i.e.,

$$Y_k = m, \text{ if } \theta_{m-1}^k < Z_k \leq \theta_m^k,$$

where $m \in \{1, 2, \ldots, M_k\}$ and $\theta^k = \{\theta_m^k\}_{m=0}^{M_k}$ are the thresholds for the $k$-th variable ($-\infty = \theta_0^k < \theta_1^k < \theta_2^k < \ldots < \theta_{M_k}^k = +\infty$). Given observations of $Y_k$, we estimate the thresholds as follows

$$\hat{\theta}_m^k = \Phi^{-1} \left( \frac{\sum_{i=1}^n \mathbb{1}_{y_{ik} \leq m}}{n} \right), \forall m \in \{1, \ldots, M_k-1\}, \quad (3.2)$$

and set $\theta_0^k = -\infty$ and $\theta_{M_k}^k = \infty$.

**Step 2** Let $\rho$ be the correlation between $Z_j$ and $Z_k$. The joint distribution of $(Z_j, Y_k)$ reads

$$p(z_j, Y_k = m \mid \rho, \theta^k) = p(z_j)p(\theta_{m-1}^k < Z_k \leq \theta_m^k \mid \rho, z_j)$$

$$= \phi(z_j) [\Phi(\theta_m^k) - \Phi(\theta_{m-1}^k)],$$

where $\phi(\cdot)$ is the standard normal density function (since $Z_j$ is standard normal) and

$$\hat{\theta}_m^k = \frac{\theta_m^k - \rho z_j}{(1 - \rho^2)^{1/2}}.$$ 

Then, with known thresholds $\theta^k$ and the data, we could estimate the underlying correlation by minimizing the negative log-likelihood function, which has the following form

$$f(\rho \mid \theta^k, z_j, y_k) = -\sum_{i=1}^n \log p(z_{ij}, Y_k = y_{ik} \mid \rho, \theta^k). \quad (3.3)$$
However, we cannot observe $Z_j$ directly and the thresholds are generally unknown. Therefore, we replace $z_{ij}$ and $\theta^k$ with $\hat{z}_{ij}$ and $\hat{\theta}^k$ (estimated in Step 1) respectively, and minimize the following function to learn the underlying correlation, i.e.,

$$\hat{C}_{jk} = \arg\min_{\rho} f(\rho \mid \hat{\theta}^k, \hat{z}_{j}, y_{j,k}), \text{ if } j \in I^c, k \in I^o. \quad (3.4)$$

### Polychoric Correlation

When both $Y_j$ and $Y_k$ are ordinal variables, i.e., $j, k \in I^o$, we compute the polychoric correlation [Olsson, 1979] using a two-step approach.

**Step 1** This step is similar to Step 1.b of estimating the polyserial correlation. The difference here is that we need to apply (3.2) to both $Y_j$ and $Y_k$, to obtain $\hat{\theta}^j$ and $\hat{\theta}^k$.

**Step 2** Given the correlation $\rho$ and thresholds $\theta^j$ and $\theta^k$, the probability of $(Y_j, Y_k)$ taking a value $(a, b)$ is

$$P(Y_j = a, Y_k = b \mid \rho, \theta^j, \theta^k) = P(\theta^j_{a-1} < Z_j \leq \theta^j_a, \theta^k_{b-1} < Z_k \leq \theta^k_b \mid \rho).$$

If the thresholds are known, the negative log-likelihood function for $\rho$ given $Y_j$ and $Y_k$ has the form [Suggala et al., 2017]

$$g(\rho \mid \theta^j, \theta^k, y_{j}, y_{k}) = -\sum_{a=1}^{M_j} \sum_{b=1}^{M_k} \frac{n_{ab}}{n} \log P(Y_j = a, Y_k = b \mid \rho, \theta^j, \theta^k), \quad (3.5)$$

where $n_{ab} = \sum_{i=1}^{n} \mathbb{1}_{y_{ij} = a, y_{ik} = b}$ is the number of samples taking the value $(a, b)$. When the thresholds are unknown, we estimate the underlying correlation by minimizing the negative log-likelihood where $\hat{\theta}^j$ and $\hat{\theta}^k$ are replaced with $\hat{\theta}^j$ and $\hat{\theta}^k$, i.e.,

$$\hat{C}_{jk} = \arg\min_{\rho} g(\rho \mid \hat{\theta}^j, \hat{\theta}^k, y_{j}, y_{k}), \text{ if } j, k \in I^o. \quad (3.6)$$

### Convergence Rate

In this section, we derive an error bound for our estimator with its corresponding rate of convergence.

First of all, we expect that although we employ three types of estimators (the rank correlation between two continuous variables, the polyserial correlation between a continuous variable and an ordinal variable, and the polychoric correlation between two ordinal variables), the polychoric correlation with binary observations, also known as the tetrachoric correlation, constitutes the worst case and we can obtain overall error bounds from an investigation of tetrachoric correlations. This is explicitly stated in Assumption 3.1. Second, we provide the error bound of our correlation estimator when the thresholds are known, which is shown in Theorem 3.1.
Ultimately, we derive the error bound of the estimator when we plug the estimated thresholds into the correlation estimation process (Theorem 3.2).

**Assumption 3.1.** The errors of the rank correlation estimator, the polyserial correlation estimator, and the polychoric correlation estimator are upper-bounded by the error of the tetrachoric correlation estimator, i.e., the polychoric estimator with binary observations.

This assumption intuitively holds since the continuous and ordinal variables can always be transformed into binary observations with less information, resulting in the tetrachoric case. Under Assumption 3.1, from now on, we focus on deriving the convergence rate of our correlation estimator in the tetrachoric case (as the error bound of our overall estimator), in which we have two binary variables $Y_j$ and $Y_k$, and only one threshold for each variable.

Our main tool is to apply Theorem 2 of Mei et al. [2017], for which we define the empirical risk or normalized negative log-likelihood function for $Y_j$ and $Y_k$ as

$$
\hat{R}_{jk}(\rho) = g(\rho \mid \theta_1^j, \theta_1^k, y_j, y_k) - g(\rho_0 \mid \theta_1^j, \theta_1^k, y_j, y_k),
$$

where $g(\cdot)$ is the function from (3.5), $\theta_1^j$ and $\theta_1^k$ are the true thresholds, and $\rho_0$ denotes the true correlation. The corresponding population risk is $R_{jk}(\rho) = \mathbb{E}[\hat{R}_{jk}(\rho)]$, where the expectation is taken over observations of latent variables with correlation $\rho_0$. We may suppress the index $j, k$ for readability and define $p_{a,b}(\rho) = P(Y_j = a, Y_k = b \mid C_{jk} = \rho)$. Then, by (3.5), we have

$$
R(\rho) = \mathbb{E}[\hat{R}(\rho)] = -\sum_{a,b} p_{a,b}(\rho_0) \log \left( \frac{p_{a,b}(\rho)}{p_{a,b}(\rho_0)} \right), \quad (3.7)
$$

For using Theorem 2 of Mei et al. [2017], we further make the following assumption.

**Assumption 3.2.** There is a positive constant $\bar{\rho} < 1$ such that $|C_{jk}| \leq \bar{\rho}, \forall j \neq k$. There is a constant $K > 0$ such that $|\theta_1^j| \vee |\theta_1^k| := \max(|\theta_1^j|, |\theta_1^k|) \leq K$, which implies that the likelihood function $p_{a,b}(\rho)$ is strictly positive in the support of $\rho$, i.e., $\exists \gamma > 0$ such that $p_{a,b}(\rho) \geq 1/\gamma, \forall \rho \in [-\bar{\rho}, \bar{\rho}]$.

This assumption states mild conditions to ensure that no two latent variables are perfectly correlated and all categories of the observed ordinal variables have non-zero probabilities.

**Lemma 3.1.** Under Assumption 3.2, the absolute values of the first, second, and third order derivatives of $p_{a,b}(\rho)$ are respectively upper-bounded by constants $L_1$, $L_2$, and $L_3$ that depend on $\bar{\rho}$ and $K$.

**Proof.** See Appendix 3.A, where we show the functional form of $L_1$, $L_2$, and $L_3$ with respect to $\bar{\rho}$ and $K$. \qed

Having the bounds in Lemma 3.1, we may obtain an error bound for our correlation estimator with true thresholds, which is shown in Theorem 3.1.
Theorem 3.1 (Error Bound under True Thresholds). Under Assumption 3.2, which especially implies that the population risk $R(\rho)$ is $(\omega, \eta)$-strongly Morse, there is a universal constant $H$, such that for $\epsilon \leq \left( \frac{\omega}{\eta} \wedge \frac{1}{\tau} \wedge \frac{\eta}{L} \right) := \min \left( \frac{\omega}{\eta}, \frac{1}{\tau}, \frac{\eta}{L} \right)$,
\[
P(\left| \hat{\rho}_n - \rho_0 \right| \geq \epsilon) \leq \bar{\rho} \tau \exp \left( -\frac{\eta^2}{4\tau^2H} \left( n / \log(n) \right) \cdot \epsilon^2 \right),
\]
for the estimator $\hat{\rho}_n$ of $\rho_0$ using the true thresholds. Here $\tau = \tau_1 \vee \tau_2$ with $\tau_1 = 2\gamma L_1$ the constant of sub-Gaussianity of the gradient and $\tau_2 = \gamma L_2 + \gamma^2 L_1^2$ the constant of sub-Gaussianity of the Hessian, and $L = \sup_{|\rho| \leq \bar{\rho}} \left| \frac{d^3}{d\rho^3} R(\rho) \right| = \gamma L_3 + 12\gamma^2 L_1 L_2 + 2\gamma^3 L_1^3$.

Proof. See Appendix 3.B.

Now we begin to derive the error bound of our correlation estimator under the estimated thresholds. We first give the error bound for recovering the thresholds. By applying Lemma 3.4 to the tetrachoric case where $M_k = 2$, $a = 1$, and $\delta(\epsilon) \geq \phi(K) \cdot \epsilon$, we have
\[
P(\left| \hat{\theta}_k - \theta_k \right| \geq \epsilon) \leq 2 \exp \left( -2nK_1^2\epsilon^2 \right), \forall k \in I^o,
\]
where $K_1 := \phi(K)$ with $\phi(\cdot)$ the standard normal density function and $K$ from Assumption 3.2.

Building upon the bound for the estimated thresholds, we deduce the error bound of the correlation estimator, which is shown in Theorem 3.2.

Theorem 3.2 (Error Bound under Estimated Thresholds). Under Assumptions 3.2, which especially implies that the population risk $R(\rho)$ is $(\omega, \eta)$-strongly Morse, there is a universal constant $H$, such that for $\epsilon \leq \left( \frac{\omega}{\eta} \wedge \frac{1}{\tau} \wedge \frac{\eta}{L} \right)$,
\[
P(\left| \hat{\rho}_n - \rho_0 \right| \geq \epsilon) \leq 2 \exp \left( -\frac{K_2^2 \eta^2 \epsilon^2 n}{8K_2^2} \right) + \bar{\rho} \tau \exp \left( -\frac{\eta^2}{16\tau^2H} \left( n / \log(n) \right) \cdot \epsilon^2 \right),
\]
for the estimator $\hat{\rho}_n$ of $\rho_0$ using estimated thresholds. Here $K_1 = \phi(K)$, $K_2$ is a Lipschitz constant for the first order derivative of the risk function from (3.21), $\tau = \tau_1 \vee \tau_2$ with $\tau_1 = 2\gamma L_1$ the constant of sub-Gaussianity of the gradient and $\tau_2 = \gamma L_2 + \gamma^2 L_1^2$ the constant of sub-Gaussianity of the Hessian, and $L = \sup_{|\rho| \leq \bar{\rho}} \left| \frac{d^3}{d\rho^3} R(\rho) \right| = \gamma L_3 + 12\gamma^2 L_1 L_2 + 2\gamma^3 L_1^3$.

Proof. See Appendix 3.C.

3.2.2 Hetcor PC Algorithm

By applying the heterogeneous estimator to learn the underlying correlation matrix to be used in the conditional independence tests, we extend the PC algorithm to mixed continuous and ordinal data, which we refer to as the Hetcor PC algorithm.\(^1\)

\(^1\)The implementation in R and the code used in our experiments are publicly available in https://github.com/cuiruifei/HetcorPC.
In what follows, we derive the error bound of the Hetcor PC algorithm. Following the line of reasoning in Harris and Drton [2013], we rewrite the conditional independence testing criteria to

\[ Z_j \perp \perp Z_k \mid Z_Q \iff |\hat{\rho}_{jk\mid Q}| \leq \alpha, \tag{3.10} \]

where \( \hat{\rho}_{jk\mid Q} \) is the partial correlation between \( Z_j \) and \( Z_k \) given \( Z_Q \) computed from the correlation matrix estimate \( \hat{C} \), and \( \alpha \in [0, 1] \) is the significance level. We use \( \hat{M}_\alpha(\mathcal{G}) \) to denote the output of the Hetcor PC algorithm with significance level \( \alpha \), and \( M(\mathcal{G}) \) to represent the true Markov equivalence class.

For a DAG \( \mathcal{G} = (\mathbf{V}, \mathbf{E}) \) and a correlation matrix \( C \), let \( c_{\min}(C) := \min\{|\rho_{jk}| : \rho_{jk} \neq 0\} \) be the minimal non-zero absolute partial correlation, and \( \lambda_{\min}(C) \) be the minimal eigenvalue. Then for any integer \( q \geq 2 \), let

\[ c_{\min}(C, q) := \min\{c_{\min}(C_{I,I}) : I \subseteq \mathbf{V}, |I| \leq q\}, \]

\[ \lambda_{\min}(C, q) := \min\{\lambda_{\min}(C_{I,I}) : I \subseteq \mathbf{V}, |I| \leq q\} \]

be the minimal non-zero absolute partial correlation and eigenvalue respectively of any principal submatrix of order at most \( q \).

**Theorem 3.3** (Error Bound of the Hetcor PC Algorithm). Let \( y_1, \ldots, y_n \) be independent samples drawn from a Gaussian copula model with correlation matrix \( C \) that is faithful to a DAG \( \mathcal{G} \) with \( p \) nodes. For \( q := \deg(\mathcal{G}) + 2 \) with \( \deg(\mathcal{G}) \) the degree of \( \mathcal{G} \), let \( c := c_{\min}(C, q) \) and \( \lambda := \lambda_{\min}(C, q) \). If \( n > q \), then there exists a threshold \( \alpha \in [0, 1] \) for which

\[ P(\hat{M}_\alpha(\mathcal{G}) \neq M(\mathcal{G})) \leq p^2 \exp\left(-\frac{K_1^2 \eta^2 \lambda^4 c^2 n}{144K_2^2 q^2}\right) + \bar{\rho} \tau p^2 \frac{2}{\exp\left(-\frac{\eta^2 \lambda^4 c^2 n}{576\tau^2 H q^2} \cdot \frac{n}{\log(n)}\right)} , \]

where \( K_1, K_2, \eta, \bar{\rho}, \tau, \) and \( H \) are constants from Theorem 3.2.

**Proof.** See Appendix 3.D. \( \square \)

From the probability error bound in Theorem 3.3, we may derive the high-dimensional consistency of the Hetcor PC algorithm following the same line of reasoning for the Rank PC algorithm [Harris and Drton, 2013].

**Corollary 3.1** (Consistency of the Hetcor PC Algorithm). Let \( (\mathcal{G}_n) \) be a sequence of DAGs, \( p_n \) be the number of nodes of \( \mathcal{G}_n \), and \( q_n = \deg(\mathcal{G}_n) + 2 \). Suppose \( (C_n) \) is a sequence of \( p_n \times p_n \) correlation matrices with \( C_n \) faithful to \( \mathcal{G}_n \). Suppose further that there are constants \( 0 \leq a, b, d, f < 1 \) that govern the growth of the graphs as

\[ \log(p_n) = O(n^a), \quad q_n = O(n^b), \]
and minimal signal strengths and eigenvalues as
\[ c_{\text{min}}(C_n, q_n) = \Omega(n^{-d}), \quad \lambda_{\text{min}}(C_n, q_n) = \Omega(n^{-f}). \]
If \( a + 2b + 2d + 4f < 1 \), then there exists \( \alpha_n \) for which
\[ \lim_{n \to \infty} P(\hat{\mathcal{M}}_{\alpha_n}(G_n) = \mathcal{M}(G_n)) = 1. \]

**Proof.** By Theorem 3.3, for large enough \( n \), the error bound is determined by the second term. Then, we can pick a significance level \( \alpha_n \) such that
\[ P(\hat{\mathcal{M}}_{\alpha_n}(G_n) \neq \mathcal{M}(G_n)) \leq A \exp\left(2n^a - \frac{Bn^{1-2b-2d-4f}}{\log(n)}\right) \]
for constants \( 0 < A, B < \infty \). The bound goes to zero if \( 1 - 2b - 2d - 4f > a \). \( \square \)

Compared to the bound of Rank PC (Theorem 8 in Harris and Drton 2013), we notice that there is an extra factor of \( 1/\log(n) \) in the error bound of our Hetcor PC, which is incurred due to information loss in ordinal variables. However, it is encouraging that this extra factor does not induce stricter conditions to guarantee the consistency result.

Note that in practice we conduct the conditional independence tests in Equation (1.1) instead of the one in Equation (3.10). This does not influence the theoretical results. See Section 5 and Appendix A in Harris and Drton [2013] for detailed analysis.

### 3.2.3 Learning Effective Sample Size

Compared to fully continuous data, the discretization in generating ordinal variables incurs some information loss, which makes the estimated correlations less reliable. To account for this, we propose to learn an effective sample size (typically smaller than the sample size) for each pairwise correlation when at least one ordinal variable is involved, acting as if the correlation is estimated from a smaller size of equivalent continuous data.

**Definition 3.1** (Effective Sample Size). An effective sample size for a population quantity (pairwise correlation here) is a number \( \hat{n} \), with the property that a mixed or fully ordinal data set of size \( n \) contains the same information as a fully continuous data set of size \( \hat{n} \).

**On the polyserial correlation** The information of the polyserial correlation estimator (the one shown in Equation 3.4), denoted by \( J(\hat{C}_{jk}) \), is defined as the second-order derivative of the negative log-likelihood function evaluated at \( \hat{C}_{jk} \), i.e.,
\[ J(\hat{C}_{jk}) = \frac{\partial^2 f(\rho)}{\partial \rho^2}\bigg|_{\rho = \hat{C}_{jk}}. \]
where $f(\rho)$ is the negative log-likelihood function from Equation (3.3). The information for a fully continuous data set of size $n$ is asymptotically $n/(1 - \hat{C}_{jk}^2)^2$ (see Equation 3.1). According to Definition 4.1, the effective sample size for the estimated polyserial correlation $\hat{C}_{jk}$ reads

$$\hat{n}_{jk} = (1 - \hat{C}_{jk}^2)^2 \left[ \frac{\partial^2 f(\rho)}{\partial \rho^2} \bigg|_{\rho = \hat{C}_{jk}} \right], \quad \text{if } j \in I^c, k \in I^o,$$

where $\hat{C}_{jk}$ is from Equation (3.4).

**On the polychoric correlation** Following the same line of reasoning as the polyserial correlation, we have the effective sample size for the estimated polychoric correlation, i.e.,

$$\hat{n}_{jk} = (1 - \hat{C}_{jk}^2)^2 \left[ \frac{\partial^2 g(\rho)}{\partial \rho^2} \bigg|_{\rho = \hat{C}_{jk}} \right], \quad \text{if } j, k \in I^o,$$

where $g(\rho)$ is the negative log-likelihood function from (3.5) and $\hat{C}_{jk}$ is from (3.6).

Note that we estimate the effective sample size for the polyserial and polychoric correlation under the assumption that all thresholds are known, which neglects the uncertainty in estimating the thresholds. Since this fraction of information loss is ignorable [Olsson, 1979], we expect the current procedure to be sufficiently accurate for practical purposes.

**On conditional independence tests** When applying the effective sample size for pairwise correlations to conditional independence tests, we rewrite the testing criteria from Equation (1.1) to

$$Z_u \perp \perp Z_v \mid Z_Q \Leftrightarrow \sqrt{\hat{n}_{uv|Q} - |Q| - 3} \left| \frac{1}{2} \log \left( \frac{1 + \hat{\rho}_{uv|Q}}{1 - \hat{\rho}_{uv|Q}} \right) \right| \leq \Phi^{-1}(1 - \alpha/2), \quad (3.12)$$

where for $\hat{n}_{uv|Q}$ we take

$$\hat{n}_{uv|Q} = \frac{2}{d(d - 1)} \sum_{j,k \in \{u,v\} \cup Q} \hat{n}_{jk},$$

with $d = 2 + |Q|$. We refer to $\hat{n}_{uv|Q}$ as the local effective sample size (LESS), which is the average over the variables involved in the current test.

In Sections 3.3 and 3.4, we will empirically show that the Hetcor PC algorithm with the test in Equation (3.12) outperforms the one with the test in Equation (1.1) for mixed data.

### 3.3 Simulation Study

In this section, we evaluate the Hetcor PC algorithm with simulation studies. Section 4.6.1 introduces the simulation setup, and Section 3.3.2 shows that the Hetcor
PC algorithm can be significantly improved by replacing the (original) sample size with a local effective sample size in the tests for conditional independences. Section 3.3.3 first illustrates that the proposal of local effective sample size can also improve the Copula PC algorithm, and then compares Hetcor PC with Copula PC. Section 3.3.4 demonstrates another approach that can discover causal structure from mixed data, called MM. Since MM relies on a different underlying data-generating casual model from Hetcor PC, we will explore which method is more robust to adversarially generated data.

3.3.1 Setup

Kalisch and Bühlmann [2007] suggests a procedure to generate random DAGs and simulate normally distributed samples that are faithful to the DAGs. It first generates a $p \times p$ adjacency matrix $A$ representing a random DAG: 1) generate a $p \times p$ zero matrix, 2) randomly set entries in the lower-triangle area to one with probability $s$ (measuring the sparseness), 3) change the ones to random weights in the interval $[0.1, 1]$. The DAGs generated in this way have the property $E[N_i] = s(p - 1)$, with $N_i$ the number of neighbors of the $i$-th node. Then, the samples of a Gaussian random vector $Z$ are drawn through

$$Z_j = \sum_{i < j} A_{ji} Z_i + \epsilon_j,$$

with $\epsilon_j \sim N(0, 1), \forall j$. To obtain mixed continuous and ordinal data, we discretize 1/3 of all the $p$ variables (randomly chosen) into ordinal with 2 categories and another 1/3 of them into ordinal with 4 categories.

The true positive rate (TPR) and the false positive rate (FPR) are used to evaluate the estimated skeleton. The structural Hamming distance (SHD), counting the number of edge insertions, deletions, and flips in order to transfer the estimated CPDAG into the correct CPDAG [Tsamardinos et al., 2006] is applied to evaluate the learned CPDAG. A higher TPR, a lower FPR, and a smaller SHD imply better performance.

To conduct causal discovery, we use the order-independent version of the PC algorithm [Colombo and Maathuis, 2014], where the significance level is set to $\alpha = 0.01$ since this yields the optimal SHD overall [Kalisch and Bühlmann, 2007]. For the Gibbs sampler used in the Copula PC algorithm, we discard 50 burn-in samples and use the next 100 samples to learn the correlation matrix and effective sample size. We conduct our experiment for all 24 combinations of the following parameters:

- $n \in \{100, 500, 1000, 2000\}$,
- $p \in \{15, 45, 90\}$,
- $E[N] \in \{2, 5\}$,

where $E[N]$ is the average number of neighbors over all nodes. For each setting, we repeat the experiment 100 times.
3.3.2 Sample Size vs. Local Effective Sample Size

We consider two versions of the Hetcor PC algorithm: 1) heterogeneous correlations with the original sample size, denoted by HetSS; and 2) heterogeneous correlations with the local effective sample size, denoted by HetLESS. The difference is that HetSS tests conditional independences via Equation (1.1) while HetLESS conducts the test via Equation (3.12).

![Graphs showing performance of HetSS and HetLESS](image)

Figure 3.1: Performance of HetSS (heterogeneous correlation + sample size) and HetLESS (heterogeneous correlation + local effective sample size) in causal discovery under various conditions, showing the mean of TPR, FPR, and SHD over 100 experiments with 95% confidence interval when the average neighborhood size is 2 (E[N] = 2).

Figure 3.1 provides the experimental results obtained by HetSS and HetLESS for sparse graphs (E[N] = 2), showing the mean of TPR, FPR, and SHD over 100 experiments with 95% confidence interval under different settings. We see that HetSS shows some advantage over HetLESS w.r.t. TPR, but this superiority
Figure 3.2: Performance of HetSS and HetLESS in causal discovery under various conditions, showing the mean of TPR, FPR, and SHD over 100 experiments with error bars representing 95% confidence interval when the average neighborhood size is 5 ($E[N] = 5$).

is quite small and tends to disappear with an increasing number of sample size. However, on the other hand, HetLESS significantly outperforms HetSS w.r.t. FPR, which becomes more prominent for bigger graphs and larger sample sizes. This suggests that replacing the sample size with a local effective sample size in the conditional independence tests can substantially reduce type-I errors (falsely reject the null hypothesis). Therefore, for the overall metric SHD, HetLESS reports a better performance than HetSS, especially when the graph is big and the sample size is relatively large.

The results for dense graphs ($E[N] = 5$) are shown in Figure 3.2, for the same experiments as sparse graphs. From TPR and FPR, we see that HetLESS still estimates a more accurate skeleton than HetSS, although the advantage is less obvious than in sparse cases. The potential reason for the reduced advantage is
that the denser the graph, the larger the necessary conditioning set, which decreases
the power of the local effective sample size (since we have to take the average over
more variables). For SHD, there is no clear difference between HetSS and HetLESS,
which is probably because the conditional independence test makes more mistakes
when the conditioning set gets larger. These errors are then propagated into the
orientation stage of the PC algorithm, resulting in a less accurate CPDAG.

To conclude, replacing the sample size with an local effective sample size can
improve the power of the Hetcor PC algorithm when the underlying graph is not
too dense, especially for bigger graphs and larger sample sizes.

### 3.3.3 Copula PC and Hetcor PC

![SHD (p=15)](image1)

(a) Two neighbors

![SHD (p=45)](image2)

![SHD (p=90)](image3)

(b) Five neighbors

Figure 3.3: Performance of CopGESS (copula correlations + global effective sample
size ), CopLESS (copula correlations + local effective sample size), and HetLESS
in estimating CPDAG, showing the mean of SHD over 100 experiments with error
bars representing 95% confidence interval when (a) the average neighborhood size
is 2 (\( \mathbb{E}[N] = 2 \)) and (b) the average neighborhood size is 5 (\( \mathbb{E}[N] = 5 \)).

The Copula PC algorithm (see Chapter 2) first applies a Gibbs sampler on rank-
based data to draw correlation matrix samples. The average over these samples is
taken as the underlying correlation matrix estimate (we call it copula correlations).
The variance of pairwise elements over these samples is translated into effective sample sizes for pairwise correlations, the average over which is then taken as an effective sample size to be used in the conditional independence test. We refer to the effective sample size used in Copula PC as the global effective sample size (GESS) since all the conditional independence tests share the same number, and we denote the original Copula PC as CopGESS. Similar to Hetcor PC, the proposal of local effective sample size also applies to Copula PC, which results in a new version of Copula PC, denoted by CopLESS.

Next we evaluate CopGESS and CopLESS. The comparative results are provided in Figure 3.3, showing the mean of SHD over 100 experiments with 95% confidence interval for $E\{N\} = 2$ and $E\{N\} = 5$ respectively. From Figure 3.3, CopLESS reports a better SHD score than CopGESS for sparse graphs ($E\{N\} = 2$) when the underlying graph is not too small and the sample size is suitably large, indicating that the proposal of local effective sample size improves the Copula PC algorithm as well. We also show the performance of HetLESS in Figure 3.3, in which HetLESS and CopLESS are almost indistinguishable.

### 3.3.4 MM and Hetcor PC

Tsagris et al. [2018] recently proposed an approach that can discover causal structure from mixed continuous and ordinal data, denoted by MM following the authors. It tests conditional independences using the likelihood-ratio test, which in turn relies on regression analysis. A major difference of MM from our method is that it is based on different assumptions as to how the data are generated from an underlying causal model. Figure 3.4 provides an example to illustrate such differences. We take the generation of $Y_2$ as an example. For the model underlying MM shown in Figure 3.4a, $Y_1$ is first generated via some functional transformation of a continuous variable $Z_1$, and then we get $Z_2$ that is a linear function of $Y_1$ plus some independent continuous noise, after which $Y_2$ is obtained via the functional transformation of $Z_2$. For the model in Figure 3.4b, see Definition 1.1 for details. Obviously, we expect HetLESS to do best on data generated following its assumptions, and vice versa for MM. The interesting question is which method is most robust to adversarially generated data.

Figure 3.4: An example to show the underlying data-generating causal model for (a) MM and (b) HetLESS, where observed variables are in squares while latent variables are in ovals.
Figure 3.5: A comparison of HetLESS with the method proposed by Tsagris et al. [2018] (denoted by MM following the authors) for $p = 15$ and $\mathbb{E}[N] = 2$, showing the mean of TPR, FPR, and SHD over 100 experiments with error bars representing 95% confidence interval (a) under the assumptions of HetLESS and (b) under the assumptions of MM.

Figure 3.5 summarizes the comparative results of MM and HetLESS for $p = 15$ and $\mathbb{E}[N] = 2$ under their respective assumptions, showing the mean of TPR, FPR, and SHD over 100 experiments with error bars representing 95% confidence interval. We see that HetLESS outperforms MM when the data are generated from the model underlying HetLESS, which becomes more significant for larger sample sizes. By contrast, MM shows less of an advantage over HetLESS when the data are generated from the model underlying MM (HetLESS is even slightly better than MM for small sample sizes). To conclude, in these simulation settings, HetLESS appears to be more robust to model misspecifications than MM.

### 3.4 Application to Real-world Data

We illustrate our approach on the ‘Holzinger & Swineford 1939’ dataset [Holzinger and Swineford, 1939], a classic dataset widely used in the literature and publicly available in the R package lavaan [Rosseel, 2012]. This dataset consists of mental
ability test scores of 301 students, in which we focus on 9 out of the original 26 tests as done in Rosseel [2012]. These tests are ‘Visual perception’, ‘Cubes’, ‘Lozenges’, ‘Paragraph comprehension’, ‘Sentence completion’, ‘Word meaning’, ‘Speeded addition’, ‘Speeded counting of dots’ and ‘Speeded discrimination straight and curved capitals’, which are denoted by “$x_1, x_2, \ldots, x_9$” in the subsequent analysis for simplicity. The numbers of unique values of the 9 variables are individually 

$$\{35, 25, 35, 20, 25, 40, 97, 84, 129\}$$

so the data are approximately continuous. The reason for choosing such a dataset is not because datasets with mixed continuous and ordinal observations are not popular, but because we can take the result on the continuous dataset as a baseline to be used for evaluating our approaches on the datasets with some manually-discretized variables.

We run HetLESS and HetSS on the original (continuous) data, in which the order-independent version of the PC algorithm [Colombo and Maathuis, 2014] is used with significance level 0.05. Both algorithms output the same structure shown in Figure 3.6a. An encouraging result is that “$x_1, x_2, x_3$” are connected to each other, “$x_4, x_5, x_6$” are connected to each other, and “$x_7, x_8, x_9$” are connected to each other as well. This is what we expect, because in the original study “$x_1, x_2, x_3$” are treated as indicators of a latent concept called visual, “$x_4, x_5, x_6$” are indicators of textual, and “$x_7, x_8, x_9$” are indicators of speed. (These 9 variables are essentially classified into the 3 groups: visual, textual, and speed.) We take this structure as the ‘pseudo ground truth’.

Next we discretize variables “$x_2, x_5, x_8$” into ordinal with 2 categories and variables “$x_3, x_6, x_9$” into ordinal with 4 categories while leaving “$x_1, x_4, x_7$” continuous, acting as if we can only observe such a mixed dataset (can be treated as data generated from a Gaussian copula model). The results obtained by HetLESS and HetSS on this mixed dataset are shown in Figures 3.6b and 3.6c, respectively. Compared to the ‘pseudo ground truth’, HetLESS gives entirely the same skeleton while HetSS reports two extra edges ‘$x_5 \rightarrow x_9$’ and ‘$x_5 \rightarrow x_2$’. This result indicates the same conclusion with our simulation study in Section 3.3.2, that is, replacing the sample size with a local effective sample size in the conditional independence tests reduces the number of false positives (extra edges), which in turn improves the performance of the PC algorithm in causal discovery. Note that the graphs in Figures 3.6b and 3.6c are not valid CPDAGs. This might happen for the order-independent (and conservative) version of the PC algorithm, because a triple $a \rightarrow b \leftarrow c$ might be marked as ‘ambiguous’ in the second step of determining v-structures. In such cases, no further orientation rule that needs to know whether $a \rightarrow b \leftarrow c$ is a v-structure or not is applied. For example, in Figure 3.6b, we cannot get $x_9 \rightarrow x_3$ from $x_7 \rightarrow x_9 \leftarrow x_3$, because $x_7 \rightarrow x_9 \leftarrow x_3$ is an ‘ambiguous’ triple (so that Rule 1 in the PC algorithm does not work).
3.5 Conclusion and Discussion

In this chapter, we considered the problem of causal structure learning from mixed continuous and ordinal data. To learn the underlying correlation matrix, a novel heterogeneous estimator was proposed. We proved the convergence rate of this correlation estimator in the large sample limit, based on which we derived the probability error bound and high-dimensional consistency of the resulting Hetcor PC algorithm. Our results extend the theoretical guarantees of the PC algorithm in earlier work [Kalisch and Bühlmann, 2007; Harris and Drton, 2013] from pure continuous data to mixed data. To the best of our knowledge, we are the first to propose a consistent algorithm with strong statistical guarantees that can recover causal structure from mixed data.

We note that the methods used in establishing our results (Theorems 3.1 and 3.2) are fairly general and could be used for bounding the error of correlation estimates derived from polychoric correlations directly (i.e., without use of Assumption 3.1), at least in those situations where the assumptions about the Morse property of the risk can be justified [Suggala et al., 2017]. But considering the estimation error of each threshold separately in this situation will then be insufficient, as then there are many more thresholds and many more of those errors to control. This could be remedied by using a joint error bound like the Dvoretzky-Kiefer-Wolfowitz inequality [Massart, 1990] for the marginal observation probabilities of each discrete value.

In our experiments, we empirically showed that the Hetcor PC algorithm with the local effective sample size outperforms the one with the original sample size, on both synthetic and real-world data. Apparently, giving a different (local) effective sample size to different tests significantly reduces the false positives in the conditional independence tests, which subsequently improves the power of the PC algorithm. Another recent work [Tsagris et al., 2018] that concerns causal discov-
ery from mixed data, uses the cumulative log-odds ratio models to handle ordinal variables, which relies on different assumptions from our model. In our simulations, their method fails under our assumptions, whereas our method is just slightly worse under their assumptions. This result coincides with Suggala et al. [2017], where the authors compared the power of two models for ordinal data in undirected graphical modeling. One possible explanation might be that the node-conditional distributions based on cumulative log-odds ratio models do not lead to a consistent joint distribution [Suggala et al., 2017].

While we focused on the PC algorithm for causal discovery in this chapter, the proposed procedure and theoretical results can be similarly applied to other algorithms, e.g., FCI [Spirtes et al., 2000] for handling potential confounders and selection bias, or some recent work [Claassen and Heskes, 2012; Hyttinen et al., 2014; Zhang et al., 2016].

3.A Proof of Lemma 3.1

**Lemma 3.1.** Under Assumption 3.2, the absolute values of the first, second, and third order derivatives of \( p_{a,b}(\rho) \) are respectively upper-bounded by constants \( L_1, L_2, \) and \( L_3 \) that depend on \( \bar{\rho} \) and \( K \).

**Proof.** Since the standard bivariate normal distribution function

\[
\Phi(x, y, \rho) = P(Z_j \leq x, Z_k \leq y \mid \rho)
\]

for a pair of variables \((Z_j, Z_k)\), the likelihood function in the tetrachoric case \( p_{a,b}(\rho) = P(Y_j = a, Y_k = b \mid \rho) \) has the form

\[
\begin{align*}
p_{1,1}(\rho) & = \Phi(x, y, \rho), \\
p_{2,1}(\rho) & = \Phi(y) - \Phi(x, y, \rho)), \\
p_{1,2}(\rho) & = \Phi(x) - \Phi(x, y, \rho)), \\
p_{2,2}(\rho) & = 1 - \Phi(x) - \Phi(y) + \Phi(x, y, \rho),
\end{align*}
\]

where \( \Phi(\cdot) \) is the standard normal cumulative distribution function, and we set thresholds for the two variables to \( \theta^j_1 = x \) and \( \theta^k_1 = y \) respectively for simplicity.

Then, the absolute value of the first order derivative of the likelihood reads

\[
|p'_{a,b}(\rho)| = \left| \frac{d}{d\rho} \Phi(x, y, \rho) \right| = \frac{1}{2\pi \sqrt{1 - \rho^2}} \exp \left( -\frac{x^2 + y^2 - 2\rho xy}{2(1 - \rho^2)} \right),
\]

\( \forall a, b \in \{1, 2\} \), where \( \phi(x, y, \rho) \) is the density of the standard bivariate normal distribution, and the second equality is from Plackett’s formula [Plackett, 1954].

Since \( |\rho| \leq \bar{\rho}, |x| < K \) and \( |y| < K \) from Assumption 3.2 and \( x^2 + y^2 - 2\rho xy > 0 \), we have

\[
|p'_{a,b}(\rho)| < \frac{1}{2\pi \sqrt{1 - \rho^2}} \leq \frac{1}{2\pi \sqrt{1 - \rho^2}} =: L_1;
\]
for the second order derivative, we have

\[
|p''_{a,b}(\rho)| = \left| \frac{d}{d\rho} \left[ \frac{1}{2\pi \sqrt{1-\rho^2}} \exp \left( -\frac{x^2 + y^2 - 2\rho xy}{2(1-\rho^2)} \right) \right] \right|
\]

\[
= \left| \frac{(\rho^3 - \rho + \rho x^2 - \rho^2 xy - xy + \rho y^2) \exp \left( -\frac{x^2 + y^2 - 2\rho xy}{2(1-\rho^2)} \right)}{2\pi(\rho - 1)(\rho + 1)(1-\rho^2)^{3/2}} \right|
\]

\[
< \frac{(1 + 1 + 1 \cdot K^2 + 1 \cdot K \cdot K + K \cdot K + 1 \cdot K^2) \cdot 1}{2\pi(1-\bar{\rho})(1-\bar{\rho}^2)^{3/2}}
\]

\[
= \frac{1 + 2K^2}{\pi(1-\bar{\rho})(1-\bar{\rho}^2)^{3/2}} =: L_2;
\]

and for the third derivative, we have

\[
|p'''_{a,b}(\rho)| = \left| \frac{d}{d\rho} \left[ \frac{(\rho^3 - \rho + \rho x^2 - \rho^2 xy - xy + \rho y^2) \exp \left( -\frac{x^2 + y^2 - 2\rho xy}{2(1-\rho^2)} \right)}{2\pi(\rho - 1)(\rho + 1)(1-\rho^2)^{3/2}} \right] \right|
\]

\[
= \left| \frac{2\rho^6 - 3\rho^4 + \rho^2 x^4 - 2\rho^3 x^3 y + 2\rho x^3 y + 5\rho^4 x^2 - 4\rho^2 x^2 + \rho^4 x^2 y^2 + 4\rho^2 x^2 y^2 + x^2 y^2 - x^2 - 2\rho^3 xy - 2\rho xy - 2\rho^5 y^2 - 2\rho^2 y^2 - y^2 + 1}{2\pi(\rho - 1)^2(\rho + 1)^2(1-\rho^2)^{5/2}} \right|
\]

\[
= \frac{3 + 18K^2 + 8K^4}{\pi(1-\bar{\rho})^2(1-\bar{\rho}^2)^{5/2}} =: L_3;
\]

which completes our proof. $\square$

3.B Proof of Theorem 3.1

Theorem 3.1 (Error Bound under True Thresholds). Under Assumption 3.2, which especially implies that the population risk $R(\rho)$ is $\omega, \eta$-strongly Morse, there is a universal constant $H$, such that for $\epsilon \leq \left( \frac{\omega}{\eta} \wedge \frac{1}{\tau} \wedge \frac{\eta}{\tau} \right) := \min \left( \frac{\omega}{\eta}, \frac{1}{\tau}, \frac{\eta}{\tau} \right)$,

\[
P(|\hat{\rho}_n - \rho_0| \geq \epsilon) \leq \bar{\rho} \tau \exp \left( -\frac{\eta^2}{4\tau H} (n/\log(n)) \cdot \epsilon^2 \right), \quad (3.14)
\]

for the estimator $\hat{\rho}_n$ of $\rho_0$ using the true thresholds. Here $\tau = \tau_1 \vee \tau_2$ with $\tau_1 = 2\gamma L_1$ the constant of sub-Gaussianity of the gradient and $\tau_2 = \gamma L_2 + \gamma^2 L_1^2$ the constant of sub-Gaussianity of the Hessian, and $L = \sup_{|\rho| \leq \bar{\rho}} \left| \frac{d^3}{d\rho^3} R(\rho) \right| = \gamma L_3 + 12\gamma^2 L_1 L_2 + 2\gamma^3 L_1^3$. 
**Proof.** Our proof proceeds by applying Theorem 2 of Mei et al. [2017] for which we need to check four conditions. The first three follow along the same arguments of reasoning as Suggala et al. [2017] while the fourth one is shown in Lemma 3.3.

1. Since \( p_{a,b}(\rho) \geq 1/\gamma \) from Assumption 3.2 and \( |p'_{a,b}(\rho)| \leq L_1 \) from Lemma 3.1, we have
   \[
   |\hat{R}'(\rho)| = \sum_{a,b} \frac{n_{ab}}{n} \frac{|p'_{a,b}(\rho)|}{p_{a,b}(\rho)} \leq \gamma L_1.
   \]
   Similarly we have \( |\mathbb{E}[\hat{R}'(\rho)]| < \gamma L_1 \), so \( \hat{R}'(\rho) - \mathbb{E}[\hat{R}'(\rho)] \) is bounded by \([-2\gamma L_1, 2\gamma L_1]\). Then, by Hoeffding’s lemma, the gradient of our empirical risk is sub-Gaussian with parameter \( \tau_1 = 2\gamma L_1 \).

2. The second order derivative of the empirical risk has the property that
   \[
   |\hat{R}''(\rho)| = \sum_{a,b} \frac{n_{ab}}{n} \frac{|p''_{a,b}(\rho)p_{a,b}(\rho) - [p'_{a,b}(\rho)]^2|}{p_{a,b}^2(\rho)} \leq \gamma L_2 + \gamma^2 L_1^2.
   \]
   Hence, the Hessian of our empirical risk is sub-Gaussian with parameter \( \tau_2 = \gamma L_2 + \gamma^2 L_1^2 \), which implies that the Hessian is also sub-exponential with parameter \( \tau_2 \).

3. Following the same line of reasoning, the third order derivative \( |\hat{R}'''(\rho)| \leq \gamma L_3 + 12\gamma^2 L_1 L_2 + 2\gamma^3 L_1^3 \), so the second order derivative \( \hat{R}''(\rho) \) is Lipschitz with parameter \( L = \gamma L_3 + 12\gamma^2 L_1 L_2 + 2\gamma^3 L_1^3 \).

4. Finally by Lemma 3.3 the tetrachoric bound of the population risk is \((\omega, \eta)\)-strongly Morse for a pair of constants \( \omega, \eta > 0 \).

Therefore we can apply Theorem 2 of Mei et al. [2017], where in our case \( p = 1 \) (and therefore \( c_h = 0 \) can be chosen in Assumption 3 in Mei et al. [2017], (increasing \( \tau \) if necessary), \( r = \bar{\rho}, \theta = \rho, L \) as above, \( \tau = \tau_1 \vee \tau_2 \), \( \epsilon = \omega \), and there is only one critical point.

It follows that there is \( H > 0 \) such that with probability \( 1 - \delta \), for
   \[
   n \geq 4H \log(\bar{\rho} \tau / \delta) \log(n) / \eta_*^2
   \]
   where \( \eta_*^2 = \frac{\omega^2}{\tau^2} \wedge \frac{\eta^2}{\tau^2} \wedge \frac{\eta^4}{L^2 \tau^2} \), it holds for \( H > 0 \) big enough that
   \[
   |\hat{\rho}_n - \rho_0| \leq \frac{2\tau}{\eta} \sqrt{H \log(\bar{\rho} \tau / \delta) \log(n) / n} =: \epsilon,
   \]
   (note that Mei et al. 2017 use a different constant which hides the dependence on \( \delta \)) This implies, that for
   \[
   P(|\hat{\rho}_n - \rho| \geq \epsilon) \leq \delta,
   \]
   where we now determine \( \delta \) as function of \( \epsilon \), hence
   \[
   \log(\bar{\rho} \tau / \delta) = \frac{\eta^2}{4\tau^2 H (n / \log(n))} \cdot \epsilon^2,
   \]
   (3.16)
therefore such a value of $\delta$ is given by

$$
\delta = \bar{\rho} \tau \exp \left( - \frac{\eta^2}{4\tau^2 H} (n/ \log(n)) \cdot e^2 \right),
$$

and we have

$$
P( |\hat{\rho}_n - \rho | \geq \epsilon ) \leq \bar{\rho} \tau \exp \left( - \frac{\eta^2}{4\tau^2 H} (n/ \log(n)) \cdot e^2 \right).
$$

Note that, by plugging (3.16) into (3.15), we have

$$
n \geq 4H \frac{\eta^2}{4\tau^2 H} (n/ \log(n)) \cdot e^2 \log(n) / \eta^2,
$$
or simply

$$
\epsilon \leq \frac{\eta^2}{\eta} = \frac{\omega}{\eta} \wedge \frac{1}{\tau} \wedge \frac{\eta}{L}.
$$

□

**Lemma 3.2.** $\rho_0$ is the unique minimiser of $R(\rho)$.

**Proof.** By Gibbs’ inequality

$$
- \sum_{a,b} p_{a,b}(\rho_0) \log \left( \frac{p_{a,b}(\rho)}{p_{a,b}(\rho_0)} \right) \geq 0
$$

and equality holds if and only if $p_{a,b}(\rho) = p_{a,b}(\rho_0)$ for all $a, b$, which happens only if $\rho = \rho_0$.

□

**Lemma 3.3.** Under Assumption 3.2, there is $\omega > 0$ such that the population risk $R(\rho)$ is $(\omega, \eta)$-strongly Morse in $[-\bar{\rho}, \bar{\rho}]$ in the tetrachoric case, that is for $\rho \in [-\bar{\rho}, \bar{\rho}]$

$$
|R'(\rho)| \leq \omega \Rightarrow |R''(\rho)| \geq \eta,
$$

with $\eta = \inf_{x,y: |x| \leq K, |y| \leq K} R''(\rho_0)/2 > 0$ and $\bar{\rho}, K$ from Assumption 3.2.

**Proof.** In the tetrachoric case, by combining (3.7) and (3.13), the risk function can be written as

$$
-R(\rho) = \Phi(x, y, \rho_0) \log(\Phi(x, y, \rho)) + (\Phi(x) - \Phi(x, y, \rho_0)) \log(\Phi(x) - \Phi(x, y, \rho))
$$

$$
+ (\Phi(y) - \Phi(x, y, \rho_0)) \log(\Phi(y) - \Phi(x, y, \rho))
$$

$$
+ (1 - \Phi(x) - \Phi(y) + \Phi(x, y, \rho_0)) \cdot \log(1 - \Phi(x) - \Phi(y) + \Phi(x, y, \rho))
$$

$$
+ \text{const}.
$$

(3.17)
So by Plackett’s formula [Plackett, 1954] \( \frac{d}{d \rho} \Phi(x, y, \rho) = \phi(x, y, \rho) \). Using this, and rewriting against in terms of \( p_{a,b} \),

\[
R'(\rho) = \phi(x, y, \rho) \left( \sum p_{a,b}(\rho_0) \frac{(-1)^{1+a-b}}{p_{a,b}(\rho)} \right). \tag{3.18}
\]

Because \( \phi(x, y, \rho) > 0 \), \( R'(\rho) = 0 \) if and only if the sum above is \( = 0 \). As each summand is strictly increasing, necessarily \( R'(\rho) \neq 0 \) for all but a unique value of \( \rho \) which by Lemma 3.2 is \( \rho_0 \). Then using the product rule, using \( \left( \sum p_{a,b}(\rho_0)(\frac{(-1)^{1+a-b}}{p_{a,b}(\rho)}) \right) = 0 \),

\[
R''(\rho_0) = \phi(x, y, \rho_0) \left( \sum p'_{a,b}(\rho_0) \frac{(-1)^{1+a-b}}{p_{a,b}(\rho_0)} |_{\rho=\rho_0} \right),
\]

and therefore

\[
p'_{a,b}(\rho_0)(-1)^{a-b} = \phi(x, y, \rho_0) > 0
\]

also \( R''(\rho_0) > 0 \). It follows by continuity of \( R''(\rho) \) and compactness of the domain of \( \rho \) and compactness of \( \{ x, y : |x| \leq K, |y| \leq K \} \) that \( R(\rho) \) is \( (\omega, \eta) \)-strongly Morse, for \( \eta = \inf_{x,y: |x| \leq K, |y| \leq K} R''(\rho_0)/2 \) and small enough \( \omega > 0 \). \( \square \)

### 3.3 Proof of Theorem 3.2

**Theorem 3.2 (Error Bound under Estimated Thresholds)**. Under Assumptions 3.2, which especially implies that the population risk \( R(\rho) \) is \( (\omega, \eta) \)-strongly Morse, there is a universal constant \( H \), such that for \( \epsilon \leq \left( \frac{\omega}{\eta} \wedge \frac{1}{\tau} \wedge \frac{n}{L} \right) \),

\[
P(|\hat{\rho}_n - \rho_0| \geq \epsilon) \leq 2 \exp \left( -\frac{K_1^2 \eta^2 n \epsilon^2}{8 K_2^2} \right) + \bar{\rho} \tau \exp \left( -\frac{\eta^2}{16 \tau^2 H} (n/ \log(n)) \cdot \epsilon^2 \right), \tag{3.19}
\]

for the estimator \( \hat{\rho}_n \) of \( \rho_0 \) using estimated thresholds. Here \( K_1 = \phi(K) \), \( K_2 \) is a Lipschitz constant for the first order derivative of the risk function from (3.21), \( \tau = \tau_1 \vee \tau_2 \) with \( \tau_1 = 2 \gamma L_1 \) the constant of sub-Gaussianity of the gradient and \( \tau_2 = \gamma L_2 + \gamma^2 L_1^2 \) the constant of sub-Gaussianity of the Hessian, and \( L = \sup_{|\rho| \leq \bar{\rho}} | \frac{d^2}{d \rho^2} R(\rho) | = \gamma L_3 + 12 \gamma^2 L_1 L_2 + 2 \gamma^3 L_1^3 \).

**Proof.** Under the estimated thresholds \( \hat{\theta}^j_1 \) and \( \hat{\theta}^k_1 \), we denote the population risk function by \( R^* \), which is explicitly

\[
R^*(\rho) = R(\hat{\theta}^j_1, \hat{\theta}^k_1, \rho),
\]

since \( R \) is also the function of the thresholds (recall Equation 3.17). In the same sense, the population risk under true thresholds can be explicitly written as

\[
R(\rho) = R(\theta^j_1, \theta^k_1, \rho).
\]
Let $\rho_0$ be the underlying true correlation coefficient and let $\rho_0^\star$ be the critical point at which $R^*(\rho)$ takes the unique minima, i.e., $\rho_0^\star = \text{argmin } R^*$.

By Lemma 3.3, $R(\rho)$ is $(\omega, \eta)$-strongly Morse, i.e., $|R'(\rho)| \leq \omega \Rightarrow |R''(\rho)| \geq \eta$. Since $R'(\rho_0) = 0$ and $\rho_0^\star$ is trivially close to $\rho_0$, we have $|R'(\rho)| \leq \omega$ for $\rho \in [\rho_0 \wedge \rho_0^\star, \rho_0 \vee \rho_0^\star]$, which implies

$$|\rho_0 - \rho_0^\star| \leq 1/\eta \cdot |R'(\rho_0) - R'(\rho_0^\star)|. \quad (3.20)$$

Equation (3.18) shows that $R'(\theta_1^i, \theta_1^k, \rho) := \frac{d}{d\rho} R(\theta_1^i, \theta_1^k, \rho)$ is continuously differentiable with respect to the first and second coordinates in their respect compact domain, i.e., $\theta_1^i, \theta_1^k \in [-K, K]$ with $K$ from Assumption 3.2. Therefore $R'(\theta_1^i, \theta_1^k, \rho)$ is Lipschitz, that is, there exists a constant $K_2$ such that $\forall \rho \in [-\bar{\rho}, \bar{\rho}]$,

$$|R'(\rho) - R^*(\rho)| = |R'(\theta_1^i, \theta_1^k, \rho) - R'(\hat{\theta}_1^i, \hat{\theta}_1^k, \rho)| \leq K_2(|\hat{\theta}_1^i - \theta_1^i| + |\hat{\theta}_1^k - \theta_1^k|). \quad (3.21)$$

Building upon (3.20) and (3.21), we have

$$|\rho_0 - \rho_0^\star| \leq 1/\eta \cdot |R'(\rho_0) - R'(\rho_0^\star)| = 1/\eta \cdot |R'(\rho_0^\star) - R'(\rho_0)| \leq K_2/\eta \cdot |\hat{\theta}_1^i - \theta_1^i| + |\hat{\theta}_1^k - \theta_1^k|,$$

where the equality holds because $R'(\rho_0^\star) = R^*(\rho_0^\star) = 0$.

By Lemma 3.4, the error bound for estimated thresholds in the tetrachoric case reads

$$P(|\hat{\theta}_1^k - \theta_1^k| \geq \epsilon) \leq 2 \exp(-2n\delta(\epsilon)^2) \leq 2 \exp(-2nK_1^2\epsilon^2), \quad (3.22)$$

where the second inequality holds because

$$\delta(\epsilon) \geq \inf_{|t| \leq K} \Phi'(t) \cdot \epsilon = \phi(K) \cdot \epsilon = K_1 \epsilon,$$

with $K_1 := \phi(K)$ and $K$ from Assumption 3.2. Note that the bound in (3.22) also holds for $\hat{\theta}_1^i$ and $\theta_1^i$. Therefore, we have

$$P(|\rho_0 - \rho_0^\star| \geq \epsilon) \leq 2 \exp\left(-\frac{K_1^2 \eta^2 n \epsilon^2}{2K_2^2}\right), \quad (3.23)$$

Finally, since

$$|\hat{\rho}_n - \rho_0| \leq |\rho_0 - \rho_0^\star| + |\rho_0^\star - \hat{\rho}_n|,$$

we bound the first term by $\mu \epsilon$ as in (3.23) and bound the second term by $(1 - \mu)\epsilon$ through an application of our Theorem 3.1 to misspecified risk (the risk with estimated thresholds) where $\mu \in [0, 1]$ is a balancing constant, resulting in

$$P(|\hat{\rho}_n - \rho_0| \geq \epsilon) \leq 2 \exp\left(-\frac{K_1^2 \eta^2 n \mu^2 \epsilon^2}{2K_2^2}\right) + \bar{\rho}_\epsilon \exp\left(-\frac{(1 - \mu)^2 \eta^2}{4\tau^2 H}(n/\log(n)) \cdot \epsilon^2\right).$$

For convenience, we intuitively choose $\mu = 1/2$, resulting in the claimed error bound

$$P(|\hat{\rho}_n - \rho_0| \geq \epsilon) \leq 2 \exp\left(-\frac{K_1^2 \eta^2 n \epsilon^2}{8K_2^2}\right) + \bar{\rho}_\epsilon \exp\left(-\frac{\eta^2}{16\tau^2 H}(n/\log(n)) \cdot \epsilon^2\right).$$
**Lemma 3.4** (Error bound of the Threshold Estimator). Let $\theta_a^k$ be the true threshold and $\hat{\theta}_a^k$ be its estimate by (3.2), we have

$$P(|\hat{\theta}_a^k - \theta_a^k| \geq \epsilon) \leq 2 \exp(-2n\delta(\epsilon)^2), \forall a \in \{1, \ldots, M_k - 1\}, k \in I^c,$$

where $\delta(\epsilon) = (\Phi(\theta_a^k + \epsilon) - \Phi(\theta_a^k)) \wedge (\Phi(\theta_a^k) - \Phi(\theta_a^k - \epsilon)).$

**Proof.** Let $X_i := 1_{Y_i \leq a}$, so $X_1, \ldots, X_n$ constitute a sequence of independent random variables bounded by the interval $[0, 1]$. The estimated threshold can then be written as

$$\Phi(\hat{\theta}_a^k) = \bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i,$$

with $E[\bar{X}] = \Phi(\theta_a^k)$ where $\theta_a^k$ is the underlying true threshold.

Then, by Hoeffding’s inequality, we have

$$P \left( \left| \Phi(\hat{\theta}_a^k) - \Phi(\theta_a^k) \right| \geq \delta(\epsilon) \right) \leq 2 \exp(-2n\delta(\epsilon)^2).$$

Furthermore, when we define $\delta(\epsilon) := (\Phi(\theta_a^k + \epsilon) - \Phi(\theta_a^k)) \wedge (\Phi(\theta_a^k) - \Phi(\theta_a^k - \epsilon))$, we have

$$\left\{ |\hat{\theta}_a^k - \theta_a^k| \geq \epsilon \right\} \subseteq \left\{ \left| \Phi(\hat{\theta}_a^k) - \Phi(\theta_a^k) \right| \geq \delta(\epsilon) \right\}. $$

Therefore

$$P(|\hat{\theta}_a^k - \theta_a^k| \geq \epsilon) \leq P \left( \left| \Phi(\hat{\theta}_a^k) - \Phi(\theta_a^k) \right| \geq \delta(\epsilon) \right) \leq 2 \exp(-2n\delta(\epsilon)^2),$$

which completes the proof. \qed

### 3.3 Proof of Theorem 3.3

**Theorem 3.3** (Error Bound of the Hetcor PC Algorithm). Let $y_1, \ldots, y_n$ be independent samples drawn from a Gaussian copula model with correlation matrix $C$ that is faithful to a DAG $G$ with $p$ nodes. For $q := \deg(G) + 2$ with $\deg(G)$ the degree of $G$, let $c := c_{\min}(C, q)$ and $\lambda := \lambda_{\min}(C, q)$. If $n > q$, then there exists a threshold $\alpha \in [0, 1]$ for which

$$P(\hat{M}_\alpha(G) \neq M(G)) \leq p^2 \exp \left( - \frac{K_1^2 \eta^2 \lambda \epsilon^2 n}{144 K_2^2 q^2} \right) + \bar{\rho} \tau p^2 \exp \left( - \frac{\eta^2 \lambda^2 c^2}{576 \tau H q^2} \cdot \frac{n}{\log(n)} \right),$$

where $K_1, K_2, \eta, \bar{\rho}, \tau$, and $H$ are constants from Theorem 3.2.

**Proof.** The proof of Theorem 3.3 follows the same line of reasoning as Theorem 8 in Harris and Drton [2013], which is listed as follows.

1. Suppose our correlation matrix estimate $\hat{C} = (\hat{\rho}_{jk})$ satisfies

$$\|\hat{C} - C\|_\infty := \max_{j,k} |\hat{C}_{jk} - C_{jk}| < \epsilon := \frac{\lambda^2 c/2}{(2 + c/2)q + \lambdaqc/2}. \tag{3.24}$$
Choose any two nodes \( j, k \in \{1, \ldots, p\} \) and a set \( Q \subseteq \{1, \ldots, p\} \setminus \{j, k\} \) with \(|Q| \leq \deg(G) = q - 2\), and let \( I = \{Z_j, Z_k\} \cup Z_Q\). By applying Lemma 4 in Harris and Drton [2013] to the \(|I| \times |I|\) submatrix of \(C\) and \(\hat{C}\), we have

\[
|\hat{\rho}_{jk|Q} - \rho_{jk|Q}| < c/2.
\] (3.25)

2. We now show that under Condition (3.25), the test in Equation (3.10) makes a correct decision when we choose \(\alpha = c/2\). For this, we need to check two cases.

- When the true partial correlation is 0, i.e., \(\rho_{jk|Q} = 0\), Condition (3.25) reduces to \(|\hat{\rho}_{jk|Q}| < c/2\), which implies the correct decision – accept the null hypothesis.
- When the true partial correlation is not 0, we have \(\rho_{jk|Q} \geq c\) since the absolute value of all non-zero partial correlations are bounded away from zero by \(c\) (recall Equation 3.11). Then, we have

\[
|\hat{\rho}_{jk|Q}| = |\hat{\rho}_{jk|Q} - \rho_{jk|Q} + \rho_{jk|Q}| \geq |\rho_{jk|Q}| - |\hat{\rho}_{jk|Q} - \rho_{jk|Q}| > c/2,
\]

which also implies the correct decision – reject the null hypothesis.

3. Therefore, \(\|\hat{C} - C\|_\infty < \epsilon\) indicates that the test (3.10) decides all conditional independences correctly, which in turn suggests that the Hetcor PC algorithm outputs a correct Markov equivalence class.

Next, using a union bound and the error bound of the correlation estimator in Theorem 3.2, we have that

\[
P(\hat{M}_\alpha(G) \neq M(G)) \leq P(|\hat{\chi}_{jk} - \chi_{jk}| \geq \epsilon \text{ for some } j,k) \\
\leq \frac{p(p-1)}{2} P(|\hat{\mu} - \mu_0| \geq \epsilon) \\
\leq p^2 \exp \left( -\frac{K^2_1 n^2 \epsilon^2}{8K^2_2} \right) + \frac{\tilde{\rho} \tau p^2}{2} \exp \left( -\frac{\eta^2}{16\tau^2 H \log(n)} \cdot \epsilon^2 \right)
\]

Plugging in the definition of \(\epsilon\) in Equation (3.24) gives the claimed inequality

\[
P(\hat{M}_\alpha(G) \neq M(G)) \leq p^2 \exp \left( -\frac{K^2_1 n^2 \lambda^4 c^2 n}{144K_2^2 q^2} \right) + \frac{\tilde{\rho} \tau p^2}{2} \exp \left( -\frac{\eta^2 \lambda^4 c^2}{576\tau^2 H q^2} \cdot \frac{n}{\log(n)} \right),
\]

where we apply the following inequality

\[
\epsilon = \frac{\lambda^2 c/2}{(2 + c/2)q + \lambda qc/2} \geq \frac{\lambda^2 c}{6q},
\]

which holds because \(c \leq 1\) and \(\lambda \leq 1\). The inequality \(c \leq 1\) holds trivially since \(c\) is one of the partial correlations that are bounded by \([-1, 1]\). The inequality \(\lambda \leq 1\) holds because the sum of all the \(q\) eigenvalues of a \(q \times q\) correlation matrix is \(q\), and \(\lambda\) is the minimal eigenvalue.

\(\square\)
Chapter 4

Causal Discovery from Data with Missing Values

We consider the problem of causal structure learning from data with missing values, assumed to be drawn from a Gaussian copula model. First, we extend the ‘Rank PC’ algorithm, designed for Gaussian copula models with purely continuous data (so-called nonparanormal models), to incomplete data by applying rank correlation to pairwise complete observations and replacing the sample size with an effective sample size in the conditional independence tests to account for the information loss from missing values. When the data are missing completely at random (MCAR), we provide an error bound on the accuracy of ‘Rank PC’ and show its high-dimensional consistency. However, when the data are missing at random (MAR), ‘Rank PC’ fails dramatically. Therefore, we propose a Gibbs sampling procedure to draw correlation matrix samples from mixed data that still works desirably under MAR. These samples are translated into an average correlation matrix, and an effective sample size, resulting in the ‘Copula PC’ algorithm for incomplete data. Simulation study shows that: 1) ‘Copula PC’ estimates a more accurate correlation matrix and causal structure than ‘Rank PC’ under MCAR and, even more so, under MAR; 2) the usage of the effective sample size significantly improves the performance of ‘Rank PC’ and ‘Copula PC’. We illustrate our methods on two real-world datasets: riboflavin production data and chronic fatigue syndrome data.

This chapter is based on Cui et al. [2018b] published in Statistics and Computing, which extended an earlier work [Cui et al., 2017] published at IEEE International Conference on Data Mining (ICDM).
4.1 Introduction

Causal structure learning [Pearl, 2009b], or causal discovery, aims to learn underly-
ing directed acyclic graphs (DAG), in which the vertices denote random variables
and the edges represent causal relations among the variables. It is a useful tool
for multivariate analysis and has been widely studied in the past decade [Spirtes
et al., 2000; Colombo et al., 2012; Harris and Drton, 2013; Peters et al., 2014].
Constraint-based methods, e.g., the PC (named by its two inventors, Peter and
Clark) algorithm and the FCI algorithm [Spirtes et al., 2000], have attracted ex-
tensive attention and generated many recent improvements [Colombo et al., 2012;
Claassen et al., 2013; Harris and Drton, 2013], yielding better search strategies and
interpretability. Since all these algorithms share the adjacency search of the PC
algorithm as a common first step, any improvements to PC can be directly trans-
ferred to the others. Therefore, we focus our analysis on the PC algorithm in this
chapter.

The adjacency search of the PC algorithm starts with a completely connected
undirected graph, and then iteratively removes the edges according to conditional
independence decisions. For testing the conditional independence, the PC algo-
rithm requires the correlation matrix and the sample size as input. The sample size
is necessary: the higher the sample size, the more reliable the estimated correla-
tion matrix, and the more easily the null hypothesis of conditional independence
gets rejected (see Equation 1.1). When applied to Gaussian data, the standard PC
algorithm estimates the correlation matrix based on Pearson correlations between
variables. Harris and Drton [2013] extend the PC algorithm to nonparanormal
models, i.e., Gaussian copula models with purely continuous marginal distribu-
tions, by replacing the Pearson correlations with rank-based correlations. We further extend
the PC algorithm to mixed discrete and continuous data assumed to be drawn from
a Gaussian copula model, resulting in the Copula PC algorithm (Chapter 2) and
the Hetcor PC algorithm (Chapter 3). However, all these approaches are based on
the assumption that the data are fully observed.

In practice, all branches of experimental science are plagued by data with missing
values [Little and Rubin, 1987; Poleto et al., 2011], e.g., failure of sensors or drop-
outs of subjects in a longitudinal study. In this chapter, we target to generalize
the PC algorithm to settings where the data are still assumed to be drawn from a
Gaussian copula model, but with some missing values. For this, we need to estimate
the underlying correlation matrix and the ‘effective sample size’ from incomplete
data. The notion ‘effective sample size’, typically smaller than or equal to the
sample size, was proposed in Chapter 2 to account for the information loss incurred
by discrete variables. In this chapter, we use it to account for the information loss
incurred by missing values, acting as if the estimated correlations on incomplete
data are in fact estimated from a smaller size of equivalent complete data.

A variety of methods have been developed for estimating correlation matrices
from Gaussian [Städler and Bühlmann, 2012; Kolar and Xing, 2012; Lounici, 2014]
or conditional Gaussian [Didelez and Pigeot, 1998] data with missing values in the context of undirected graphical models. In nonparanormal cases, Wang et al. [2014] propose to apply rank correlation to pairwise complete observations for estimating the correlation matrix, which is then plugged into existing procedures for inferring the underlying graphical structure. The convergence rate of this rank-based correlation estimator has been derived in the presence of missing values. In this chapter, we transfer this idea to causal structure learning, where this estimator is used for the correlation matrix and the number of pairwise complete observations is taken as the effective sample size. This extends the ‘Rank PC’ algorithm to incomplete data. We carry over the error bound of ‘Rank PC’ to nonparanormal data with missing values as well.

Although we will provide theoretical guarantees of the ‘Rank PC’ algorithm for incomplete data, these only apply to nonparanormal data under missingness completely at random (MCAR), which is a pretty strong assumption [Rubin, 1976]. By contrast, we prefer an approach that is valid for both nonparanormal and mixed data under a less restrictive assumption, missingness at random (MAR) [Rubin, 1976; Schafer and Graham, 2002]. To this end, we propose a Gibbs sampling procedure to draw correlation matrix samples from the posterior distribution given mixed continuous and discrete data with missing values. Then, following the idea of the ‘Copula PC’ algorithm, these Gibbs samples are translated into an average correlation matrix and an effective sample size, which are input to the standard PC algorithm for causal discovery. The difference is that now the effective sample size accounts for information loss incurred by both missing values and discrete variables.

The rest of this chapter is organized as follows. Section 4.2 reviews necessary background knowledge. Sections 4.3 and 4.4 present the ‘Rank PC’ algorithm and the ‘Copula PC’ algorithm for incomplete data respectively, while Section 4.5 introduces alternative approaches. Section 4.6 compares ‘Rank PC’, ‘Copula PC’ with alternative approaches, and evaluates the justification of the usage of the effective sample size in causal discovery on simulated data, whereas Section 4.7 provides an illustration on two real-world datasets. Section 4.8 concludes this chapter and gives potential extensions.

4.2 Missingness Mechanism

Following Rubin [1976], let $Y = (y_{ij}) \in \mathbb{R}^{n \times p}$ be a data matrix with the rows representing independent samples, and $R = (r_{ij}) \in \{0, 1\}^{n \times p}$ be a matrix of indicators, where $r_{ij} = 1$ if $y_{ij}$ was observed and $r_{ij} = 0$ otherwise. $Y$ consists of two parts, $Y_{obs}$ and $Y_{miss}$, where $Y_{obs}$ contains the observed elements in $Y$ and $Y_{miss}$ the missing elements. When the missingness does not depend on the observed values, i.e., $P(R|Y, \theta) = P(R|\theta)$ with $\theta$ denoting unknown parameters, the data are said to be missing completely at random (MCAR), which is a special case of a more realistic assumption called missing at random (MAR). MAR allows the dependency between missingness and observed values, i.e., $P(R|Y, \theta) = P(R|Y_{obs}, \theta)$. For ex-
ample, all people in a group are required to take a blood pressure test at time point 1, while only those whose values at time point 1 lie in the abnormal range need to take the test at time point 2. This results in some missing values at time point 2 that are MAR. A third missingness mechanism is called missing not at random (MNAR), which states that the missingness may be dependent on missing values, namely, \( P(R|Y, \theta) = P(R|Y_{\text{obs}}, \theta) \) no longer holds. For instance, all the people in the example above are required to take the test at time point 2, but the doctor only records those lying in the abnormal range, leaving others missing.

### 4.3 Rank PC Algorithm for Data with Missing Values

In this section, we first introduce the basic procedure of the ‘Rank PC’ algorithm for incomplete data, and then derive the convergence rate of the rank-based correlation estimator as well as the probability error bound of ‘Rank PC’ in the presence of missing values.

#### 4.3.1 Basic Procedure

Our procedure consists of three steps: 1) estimate rank correlations based on pairwise complete observations; 2) estimate the underlying correlation matrix and the effective sample size; 3) plug these into the standard PC algorithm for causal discovery. All analysis in this section is based on nonparanormal data under MCAR if not explicitly stated otherwise.

Since the two typical rank correlations, Kendall’s \( \tau \) and Spearman’s \( \rho \), are similar in our analysis, we focus our attention on Kendall’s \( \tau \) in this chapter. Given the data matrix \( Y \) and indicator matrix \( R \), we compute the Kendall’s \( \tau \) between \( Y_j \) and \( Y_k \) on samples which have observed values for both the two variables, i.e.,

\[
\hat{\tau}_{jk} = \frac{2}{\hat{n}_{jk}(\hat{n}_{jk} - 1)} \sum_{1 \leq i < i' \leq n} r_{ij} r_{ik} r_{i'j} r_{i'k} K(y_i, y_{i'}) ,
\]

(4.1)

where \( K(y_i, y_{i'}) = \text{sign}((y_{ij} - y_{i'j})(y_{ik} - y_{i'k})) \) and \( \hat{n}_{jk} = \sum_{i=1}^{n} r_{ij} r_{ik} \), which is the number of pairwise complete observations for variables \( Y_j \) and \( Y_k \).

Then, we estimate the underlying correlation matrix. For nonparanormal data, the following proposition connects the Kendall’s \( \tau \) to the underlying Pearson correlation.

**Proposition 4.1** (refer to Kendall 1948; Kruskal 1958). Assuming \( X \) follows a nonparanormal distribution with correlation matrix \( C \), we have \( C_{jk} = \sin \left( \frac{\pi}{2} \hat{\tau}_{jk} \right) \).
Motivated by this proposition, we consider the estimator \( \hat{S}^\tau = (\hat{S}_{jk}^\tau) \) for the underlying correlation matrix:

\[
\hat{S}_{jk}^\tau = \sin \left( \frac{\pi}{2} \hat{\tau}_{jk} \right).
\]

When translating the number of pairwise complete observations \( \hat{n}_{jk} \) (see Equation 4.1) into an effective sample size to be used in the conditional independence tests of the PC algorithm, we compare two schemes.

**Scheme 1** We take the average over all the \( \hat{n}_{jk} \)'s, i.e.,

\[
\hat{n} = \frac{2}{p(p-1)} \sum_{1 \leq j < k \leq p} \hat{n}_{jk}.
\]

We refer to this estimator \( \hat{n} \) as the global effective sample size (GESS). In this scheme, all the conditional independence tests share the same effective sample size.

**Scheme 2** We give a different effective sample size to different conditional independence tests, since each test relies on a local structure involving only part of the variables. In this case, we rewrite the conditional independence testing criteria to

\[
X_u \perp \perp X_v | X_Q \Leftrightarrow \sqrt{\hat{n}_{uv|Q} - |Q| - 3} \left\{ \frac{1}{2} \log \left( \frac{1 + \hat{\rho}_{uv|Q}}{1 - \hat{\rho}_{uv|Q}} \right) \right\} \leq \Phi^{-1}(1 - \alpha/2),
\]

where \( \hat{n}_{uv|Q} \) is defined as

\[
\hat{n}_{uv|Q} = \frac{2}{d(d-1)} \sum_{j,k \in \{u,v\} \cup Q} \hat{n}_{jk},
\]

with \( d = 2 + |Q| \). We refer to \( \hat{n}_{uv|Q} \) as the local effective sample size (LESS).

In the last step, we take the estimated correlation matrix \( \hat{S}^\tau \) and the global (or local) effective sample size as input to the standard PC algorithm for causal discovery.

### 4.3.2 Theoretical Analysis

**Convergence Rate of Estimator \( \hat{S}^\tau \)**

When all values in \( Y \in \mathbb{R}^{n \times p} \) are missing with probability \( \delta \), i.e., \( \forall i,j \ P(r_{ij} = 0) = \delta \), Wang et al. [2014] prove the convergence rate of \( \hat{S}^\tau \), which is shown in Theorem 4.1.

**Theorem 4.1.** For any \( n \geq 1 \), any \( m > 0 \), and any \( 0 < \epsilon < 1 \), with probability at least \( (1 - \frac{1}{p^m})(1 - \exp(-\epsilon^2(1 - \delta)^2 n/2 - 2 \log p))) \), we have

\[
\sup_{jk} |\hat{S}_{jk}^\tau - C_{jk}| \leq \frac{\pi}{1 - \delta} \sqrt{\frac{m + 2}{1 - \epsilon}} \sqrt{\frac{\log p}{n}}.
\]
Chapter 4

Error Bound of Rank PC for Incomplete Data

Since $\hat{\tau}_{jk}$ is unbiased, i.e., $E[\hat{\tau}_{jk}] = \tau_{jk}$, we have

$$P(|\hat{S}_{\tau_{jk}} - C_{\tau_{jk}}| > t)$$

$$= P\left(\left|\sin\left(\frac{\pi}{2}\hat{\tau}_{jk}\right) - \sin\left(\frac{\pi}{2}\tau_{jk}\right)\right| > t\right)$$

$$\leq P\left(|\hat{\tau}_{jk} - \tau_{jk}| > \frac{2}{\pi}t\right)$$

$$\leq 2\exp\left(-\frac{2|n_{jk}/2| t^2}{\pi^2}\right)$$

$$\leq 2\exp\left(-\frac{n'^2 t^2}{\pi^2}\right), \quad (4.2)$$

where the second inequality follows from the Hoeffding bound for one-sample U-statistics [Hoeffding, 1963], $\lfloor n_{jk}/2 \rfloor$ is the largest integer contained in $n_{jk}/2$, and $n' = \min\{2|n_{jk}/2| : \forall j, k\}$.

Building upon the result in Equation (4.2), we will now derive the error bound of Rank PC for incomplete data following the same line of reasoning as in Harris and Drton [2013].

For a DAG $G = (V, E)$ and a correlation matrix $C$, let

$$c_{\min}(C) := \min\{||\rho_{jk}|_{\mathcal{Q}}| : \rho_{jk}|_{\mathcal{Q}} \neq 0\}$$

be the minimal non-zero absolute partial correlation, and $\lambda_{\min}(C)$ be the minimal eigenvalue. Then for any integer $q \geq 2$, let

$$c_{\min}(C, q) := \min\{c_{\min}(C_{I, I}) : I \subseteq V, |I| \leq q\}, \quad (4.3)$$

$$\lambda_{\min}(C, q) := \min\{\lambda_{\min}(C_{I, I}) : I \subseteq V, |I| \leq q\}$$

be the minimal non-zero absolute partial correlation and eigenvalue respectively of any principal submatrix of order at most $q$.

**Theorem 4.2** (Error Bound of Rank PC under MCAR). Let $y_1, \ldots, y_n$ be independent samples with some MCAR missing values drawn from a nonparanormal distribution with correlation matrix $C$ that is faithful to a DAG $\mathcal{G}$ with $p$ nodes. For $q := \deg(\mathcal{G}) + 2$ with $\deg(\mathcal{G})$ the degree of $\mathcal{G}$, let $c := c_{\min}(C, q)$ and $\lambda := \lambda_{\min}(C, q)$. If $n' > q$, then there exists a threshold $\alpha \in [0, 1]$ for which

$$P(\hat{\mathcal{M}}_\alpha(\mathcal{G}) \neq \mathcal{M}(\mathcal{G})) \leq p^2 \exp\left(-\frac{\lambda^4 n'^2 \epsilon^2}{36 \pi^2 q^2}\right), \quad (4.4)$$

where $\hat{\mathcal{M}}_\alpha(\mathcal{G})$ and $\mathcal{M}(\mathcal{G})$ are the estimated and true Markov equivalence class respectively, and $n'$ is from Equation (4.2).

**Proof.** The proof directly follows from the proof of Theorem 8 in Harris and Drton [2013]. \hfill $\square$
From the probability error bound in Theorem 4.2, one could deduce the high-dimensional consistency of the Rank PC algorithm under MCAR. For a large enough \( n \) (thus a large enough \( n' \)), the left-handed term in Equation (4.4) goes to zero under some conditions that govern the growth rate of \( c, \lambda, q, p, \) and \( n' \). See Corollary 9 in Harris and Drton [2013] for more details.

Note that the procedure for Rank PC proposed in this section can be naturally applied to the Hetcor PC algorithm (Chapter 3), in which the likelihood functions in Equations (3.3) and (3.5) are then adjusted to be computed over pairwise complete observations.

4.4 Copula PC Algorithm for Data with Missing Values

In this section, we extend the ‘Copula PC’ algorithm to incomplete data. It includes three steps: 1) apply a Gibbs sampler to draw correlation matrix samples from the posterior distribution given data with missing values (Section 4.4.1); 2) use these samples to estimate the underlying correlation matrix (Section 4.4.2) and the effective sample size (Section 4.4.3); 3) plug the estimated correlation matrix and effective sample size into the standard PC algorithm for causal discovery. All analysis in this section is under the MAR assumption, unless explicitly stated otherwise.

4.4.1 Gibbs Sampling for Data with Missing Values

We choose \( \Sigma \) from an inverse-Wishart distribution, denoted by \( \mathcal{W}^{-1}(\Sigma; \Psi, \nu) \), and write

\[
P(C) = \mathcal{PW}^{-1}(C; \Psi, \nu),
\]

where \( C = (C_{jk}) \) with \( C_{jk} = \Sigma_{jj}/\sqrt{\Sigma_{jj}\Sigma_{kk}} \). Then this distribution on correlation matrix \( C \) is called a projected inverse-Wishart distribution with scale matrix \( \Psi \) and degrees of freedom \( \nu \) (see Chapter 2). In Bayesian inference, this distribution is the conjugate prior of correlation matrices for Gaussian models. Specifically, when we choose the prior \( P(C) = \mathcal{PW}^{-1}(C; \Psi_0, \nu_0) \), the posterior given data \( Z = (z_1, \ldots, z_n)^T \) reads

\[
P(C|Z) = \mathcal{PW}^{-1}(C; \Psi_0 + Z^T Z, \nu_0 + n).
\] (4.5)

For Gaussian copula models with missing values, we cannot observe the random vector \( Z \) directly (refer to Definition 1.1), but an idea is to first obtain the Gaussian pseudo-data from the observed data (i.e., \( Y \)) and then do inference for \( C \). We use a Gibbs sampling procedure to implement this idea.

Let \( Z = (z_{ij}) \in \mathbb{R}^{n \times p} \) be the Gaussian pseudo-data implied by \( Y \), thus \( Z \) has two parts as well, \( Z_{\text{obs}} \) and \( Z_{\text{miss}} \). As initialization of our Gibbs sampling procedure, we propose to obtain the Gaussian pseudo-data of non-missing values
Algorithm 4.1: Gibbs sampler for nonparanormal data under MCAR

1: Step 1: $Z_{\text{miss}} \sim P(Z_{\text{miss}}|Z_{\text{obs}}, C)$.
2: for $j \in \{1, \ldots, p\}$ do
3: $v^T = C_{[-j,-j]} C_{[-j,-j]}^{-1}$
4: $\sigma^2_j = C_{[j,j]} - v^T C_{[-j,-j]}$
5: for $i$ such that $r_{i,j} = 0$ do
6: $\mu_{i,j} = Z_{i,-j} \times v$
7: Draw $z_{i,j}$ from $N(\mu_{i,j}, \sigma^2_j)$
8: end for
9: end for
10: Step 2: $C \sim P(C|Z_{\text{miss}}, Z_{\text{obs}})$.
11: Draw $C$ from $\mathcal{P}W^{-1}(C; \Psi_0 + Z^T Z, \nu_0 + n)$

$Z_{\text{obs}}$. For this, we substitute the empirical cumulative distribution function based on non-missing data $Y_{\text{obs}}$:

$$z_{ij} = \Phi^{-1}\left[ \frac{\sum_{d=1}^n r_{dj} \mathbb{1}(y_{dj} < y_{ij})}{\sum_{d=1}^n r_{dj} + 1} \right], \text{ if } r_{ij} = 1,$$

(4.6)

where $\mathbb{1}(\cdot)$ is the indicator function.

For nonparanormal data with missing values completely at random, each marginal distribution of $Z_{\text{obs}}$ can approximately represent the underlying true distribution. Then we iterate the following two steps to impute missing values (step 1) and draw correlation matrix samples from the posterior (step 2):

1. $Z_{\text{miss}} \sim P(Z_{\text{miss}}|Z_{\text{obs}}, C)$;
2. $C \sim P(C|Z_{\text{obs}}, Z_{\text{miss}})$.

This procedure generates a Markov chain that has its stationary distribution equal to $P(C|Y)$, which can be easily implemented via the Gibbs scheme in Algorithm 4.1.

However, for mixed data under MAR, the initialization shown in Equation (4.6) is no longer sufficient for two reasons: 1) tied observations may occur, making the ranks no longer well-defined; 2) the missing values in one variable may depend on the values of others. These differentiate the obtained marginal distributions from the underlying true distributions. Hence, we need an additional strategy to obtain $Z_{\text{obs}}$ to leverage the sampling scheme in Algorithm 4.1.

For this, we borrow the idea of the so-called extended rank likelihood [Hoff, 2007], derived as follows. Since the transformation $Y_j = F_j^{-1}(\Phi(Z_j))$ is non-decreasing, observing $y_j = (y_{1,j}, \ldots, y_{n,j})^T$ implies a partial ordering on $z_j = (z_{1,j}, \ldots, z_{n,j})^T$, i.e., $z_j$ must lie in

$$\mathcal{D}(y_j) = \{z_j \in \mathbb{R}^n : y_{i,j} < y_{k,j} \Rightarrow z_{i,j} < z_{k,j}\}.$$

Therefore, observing $Y$ suggests that $Z$ must be in

$$\mathcal{D}(Y) = \{Z \in \mathbb{R}^{n \times p} : z_j \in \mathcal{D}(y_j), \forall j = 1, \ldots, p\}.$$
Algorithm 4.2 Gibbs sampler for mixed data under MAR

1. **Step 1**: \( Z_{\text{obs}} \sim P(Z_{\text{obs}} | Z_{\text{obs}} \in D(Y_{\text{obs}}), C) \).
2. **for** \( j \in \{1, \ldots, p\} \) **do**
3. \( v^T = C_{\lfloor j, -j \rfloor} C_{\lfloor -j, -j \rfloor}^{-1} \)
4. \( \sigma_j^2 = C_{\lfloor j, j \rfloor} - v^T C_{\lfloor -j, -j \rfloor} \)
5. **for** \( y \in \text{unique}\{y_{1,j}, \ldots, y_{n,j}\} \) **do**
6. \( z_l = \max\{z_{i,j} : y_{i,j} < y\} \)
7. \( z_u = \min\{z_{i,j} : y < y_{i,j}\} \)
8. **for** \( i \) such that \( y_{i,j} = y \) **do**
9. \( \mu_{i,j} = Z_{\lfloor i, -j \rfloor} \times v \)
10. Draw \( u_{i,j} \) from \( \mathcal{U}(\Phi[z_l - \mu_{i,j}] / \sigma_j, \Phi[z_u - \mu_{i,j}] / \sigma_j) \)
11. \( z_{i,j} = \mu_{i,j} + \sigma_j \times \Phi^{-1}(u_{i,j}) \)
12. **end for**
13. **end for**
14. **end for**
15. **Step 2**: \( Z_{\text{miss}} \sim P(Z_{\text{miss}} | Z_{\text{obs}}, C) \) as in Algorithm 4.1.
16. \( Z = (Z^T - \mu)^T \), with \( \mu \) the mean vector of \( Z \).
17. **Step 3**: \( C \sim P(C | Z_{\text{miss}}, Z_{\text{obs}}) \) as in Algorithm 4.1.

Taking the occurrence of this event as the data, one can compute the following likelihood

\[
P(Z \in D(Y) | C, F_1, \ldots, F_p) = \int_{D(Y)} p(Z | C) dZ = P(Z \in D(Y) | C),
\]

which is independent of the margins \( F_j \). Then inference for \( C \) proceeds by iterating the following two steps:

1. \( Z \sim P(Z | Z \in D(Y), C) ; \)
2. \( C \sim P(C | Z) \).

The strong posterior consistency for \( C \) under the extended rank likelihood has been proven in Murray et al. [2013]. We now use this method to obtain \( Z_{\text{obs}} \) from \( Y_{\text{obs}} \) and embed it into our procedure in Algorithm 4.1, resulting in the Gibbs sampler in Algorithm 4.2. Note that line 16 in Algorithm 4.2 needs to relocate the data such that the mean of each coordinate of \( Z \) is zero. This is necessary for the algorithm to be sound because the mean may shift when missing values depend on the observed data (MAR). For clarity, we list step 1 and step 2 separately in Algorithm 4.2, but the actual implementation takes these together to avoid repeated computation of lines 3 and 4. This Gibbs sampler can be implemented using the function \texttt{sbgcop.mcmc} in the R package \texttt{sbgcop} [Hoff, 2010], where the equivalent of line 16 in Algorithm 4.2 should be added to guarantee that the procedure also works under MAR.\(^1\)

\(^1\)The code is available in https://github.com/cuiruiifei/CausalMissingValues.
4.4.2 Estimating the Underlying Correlation Matrix

By iterating the steps in Algorithm 4.1 (or 4.2), we can draw samples of the correlation matrix, denoted by \( \{C^{(1)}, \ldots, C^{(m)}\} \). The mean over all the samples is a natural estimate of the underlying correlation matrix \( \hat{C} \), i.e.,

\[
\hat{C} = \frac{1}{m} \sum_{i=1}^{m} C^{(i)}.
\]  

(4.7)

We refer to the estimator in Equation (4.7) as the copula estimator for the correlation matrix.

Since Kendall’s \( \tau \) is a U-statistic and can be treated as the sum of a set of bounded variables \( K(y_i, y_i') \) in Equation 4.1 is bounded by the interval \([-1,1]\), Hoeffding’s inequalities can be used to prove its convergence rate, as we did in Section 4.3.2. Such analysis of the copula estimator, on the other hand, is much more complicated (see Hoff 2007; Hoff et al. 2014 for recent achievements). Nevertheless, intuitively, one would expect the Gibbs sampler to yield better convergence rates than Kendall’s \( \tau \), in particular in the case of missing values, because it more efficiently makes use of all available data instead of restricting itself to independent estimation of the individual elements of the correlation matrix based on pairwise complete observations. We will check this empirically in Section 4.6.2.

4.4.3 Estimating the Effective Sample Size

While it is straightforward to estimate the effective sample size for the pairwise deletion method (the one we used in Section 4.3), a different strategy in the current case is needed.

The projected inverse-Wishart distribution has a property that is summarized in Theorem 4.3 (see Theorem 2.1 in Chapter 2), showing the relationship between the mean, variance and degrees of freedom.

**Theorem 4.3.** Consider a \( p \)-dimensional random matrix \( C \) that follows from a projected inverse-Wishart distribution, i.e., \( P(C) = \mathcal{PW}^{-1}(C; \Psi, \nu) \), we have

\[
\text{Var}[C_{jk}] \approx \frac{(1 - (\mathbb{E}[C_{jk}])^2)^2}{\nu},
\]

for each off-diagonal element \( C_{jk} \) and large \( \nu (\gg p) \).

In Equation (4.5), since generally \( \nu_0 \ll n \), the posterior degrees of freedom \( \nu_0 + n \approx n \). From Theorem 4.3, the variance of each estimated correlation by our copula estimator for an \( n \)-size fully observed and continuous data set is

\[
\text{Var}[C_{jk}] \approx \frac{(1 - (\mathbb{E}[C_{jk}])^2)^2}{n}, \forall j \neq k.
\]

However, this does not hold any longer when the observational data set of size \( n \) is mixed and contains some missing values. Specifically, there will be some additional variance (or reduced information) in the correlation matrix samples incurred by missing values and ties in discrete variables.
Definition 4.1 (Effective Sample Size). The effective sample size for a population quantity (pairwise correlation here) is a number \( \hat{n} \), with the property that a mixed data set of size \( n \) with missing values contains the same information (thus variance) as a fully observed and continuous data set of size \( \hat{n} \).

Note that here we redefine the ‘effective sample size’ to incorporate both discrete data and missing values while the definition in Section 3.2.3 only considers discrete data. According to Definition 4.1, the effective sample size for the correlation \( C_{jk} \) (denoted by \( \hat{n}_{jk} \) for clarity since it can vary for different combinations of \( j \) and \( k \)) reads

\[
\hat{n}_{jk} = \frac{(1 - (\mathbb{E}_n[C_{jk}])^2)^2}{\text{Var}_n[C_{jk}]}, \forall j \neq k,
\]

where \( \mathbb{E}_n[C_{jk}] \) and \( \text{Var}_n[C_{jk}] \) denote respectively the mean and variance estimated through the correlation matrix samples drawn from a mixed data set of size \( \hat{n} \) with missing values.

When applying the effective sample size to conditional independence tests, we also compare the two different schemes discussed in Section 4.3.1: the same effective sample size for all conditional independence tests or a separate local effective sample size for each test.

4.4.4 Consistency of Copula PC Algorithm

Theorem 4.4 (Consistency of Copula PC under MCAR). Let \( y_1, \ldots, y_n \) be independent samples with some missing values drawn from a Gaussian copula model with correlation matrix \( C \) and univariate margins \( F_j \). Suppose (1) \( C \) is faithful to a DAG \( \mathcal{G} \); (2) the data are missing completely at random. Then

\[
\lim_{n \to \infty} P(\hat{\mathcal{M}}_n(\mathcal{G}) = \mathcal{M}(\mathcal{G})) = 1,
\]

where \( \hat{\mathcal{M}}_n(\mathcal{G}) \) and \( \mathcal{M}(\mathcal{G}) \) are the estimated and true Markov equivalence class respectively.

Proof. The proof follows two separate steps: Gibbs sampling to estimate the correct underlying correlation matrix and the PC algorithm to reach the correct causal structure. The first step directly follows from the proof of Theorem 1 in Murray et al. [2013], with the additional observation that the estimation of ordinary and polychoric/polyserial correlations from pairwise complete data is still consistent under MCAR. The second step has been proven in Kalisch and Bühlmann [2007]. □

While it is straightforward to prove the consistency of our Gibbs sampling procedure under MCAR, a theoretical proof that it is still consistent under MAR, is much more difficult. Hence, we will empirically show in Section 4.6.2 that, our procedure still works favorably while the rank-based estimator fails under MAR.
4.5 Alternative Approaches

In this section, we describe some alternative approaches for handling missing values.

4.5.1 Listwise Deletion

A simple widely-used approach for missing values is the so-called listwise deletion (LD), also known as case deletion or complete-case analysis. It excludes all records with missing information, so the analyses are restricted to cases that have complete data. This approach is consistent under MCAR and can produce a complete data set, which in turn allows for the use of standard analysis techniques. However, the drawbacks of this approach are numerous. For example, it dramatically reduces the total sample size, particularly for data sets with a large proportion of missing data or many variables. Suppose that we have $p$ variables and let $\delta_j$ denote the proportion of missing values in the $j$-th variable. We randomly draw $\delta_j$ from a uniform distribution with mean $\beta$, e.g.,

$$\delta_j \sim U(0, 2 \times \beta), \forall j = 1, \ldots, p.$$

Then, the expected percentage of complete cases under MCAR in such a dataset reads:

$$\mathbb{E} \left[ \prod_{j=1}^{p} (1 - \delta_j) \right] = \prod_{j=1}^{p} \mathbb{E} (1 - \delta_j)$$

$$= \prod_{j=1}^{p} (1 - \beta)$$

$$= (1 - \beta)^p.$$

Figure 4.1 shows the relationship between the percentage of complete cases and the number of variables for different expected proportions of missing values ($\beta$). We can see that the percentage of complete cases decreases dramatically with the increase of variables, which becomes more serious for a bigger $\beta$. Therefore, our conjecture is that a causal discovery algorithm with listwise deletion for handling missing values would output a very sparse or even empty graph, especially when the underlying graph has many vertices and the data contains many missing values. We will check this conjecture in Section 4.6 and 4.7.

4.5.2 Imputation Methods

Instead of discarding the entire record with missing information, a potentially more efficient method is to replace the missing items with plausible values and proceed with the desired analysis. A common procedure is called mean substitution (MS), in which missing values are replaced with the average of observed values for that variable. MS keeps the mean of that variable but ignores the variance. Another
option in wide use is called \textit{hot deck} (HD), in which the missing items are randomly drawn from the observed values of that variable. HD keeps the whole distribution of the variable, but incurs distortions of the covariance with other variables. In what follows, we use a simple example to illustrate how MS and HD influence correlations between variables, since the correlations are parameters of interest in causal discovery.

Without loss of generality, we consider a zero-mean (we can always relocate the mean of a distribution to be zero subject to an unchanged correlation) bivariate distribution \((X,Y)\) with correlation \(\rho\), i.e.,

\[
\rho = \frac{\mathbb{E}[XY]}{\sqrt{\mathbb{E}[X^2]\mathbb{E}[Y^2]}}.
\]

Let \((X_1, Y_1), \ldots, (X_n, Y_n)\) be independent samples drawn from the population distribution, where \(X\) is fully observed while \(Y\) contains MCAR missing values with proportion \(\delta\). Under MS, since all the imputed values are zeros in large sample limit, the covariance for such data reads \((1 - \delta) \mathbb{E}[XY]\) and the variance of \(Y\) is \((1 - \delta) \mathbb{E}[Y^2]\). Thus the correlation in this case reads:

\[
\rho_{MS} = \frac{(1 - \delta) \mathbb{E}[XY]}{\sqrt{\mathbb{E}[X^2](1 - \delta) \mathbb{E}[Y^2]}} = \sqrt{1 - \delta} \rho.
\]

Under HD, the covariance is also \((1 - \delta) \mathbb{E}[XY]\) since \(X_i \perp Y_j, \forall i, j\) (independent
draws). The variance of each univariate margin remains the same as the population value. Thus the correlation for HD reads:

$$
\rho_{\text{HD}} = \frac{(1 - \delta) \mathbb{E}[XY]}{\sqrt{\mathbb{E}[X^2]\mathbb{E}[Y^2]}} = (1 - \delta) \rho.
$$

We see that both MS and HD tend to diminish the correlation especially for a large proportion of missing values although they keep the same sample size as the original data, and they are not consistent for estimating correlations even under MCAR. A simulation study regarding the behavior of correlation estimators with different missing value strategies is provided in Section 4.6.2.

There are other procedures for imputation, like maximum likelihood and multiple imputation [Schafer and Graham, 2002], but they usually assume multivariate normality that is obviously violated in our case. Therefore, we do not consider these approaches in our analysis.

### 4.6 Simulation Study

In this section, we compare the proposed methods with alternative approaches through simulation studies. Section 4.6.1 introduces the simulation setup. Section 4.6.2 and 4.6.3 evaluate the performance of these approaches in correlation estimation and in causal discovery respectively.

#### 4.6.1 Setup

We choose two well-known DAGs from the Bayesian network repository\(^2\) for evaluating our approaches:

- Asia network [Lauritzen and Spiegelhalter, 1988]: this network contains 8 nodes, 5 arcs, and 3 undirected edges in its Markov equivalence class. It describes the effect of visiting Asia and smoking behavior on the probability of contracting tuberculosis, cancer or bronchitis. The Asia network is depicted in Figure 4.2.

- Alarm network [Beinlich et al., 1989]: this network contains 37 nodes, 46 arcs, and 4 undirected edges in the CPDAG of the equivalence class. It was originally designed to help interpret monitoring data to alert anesthesiologists to various situations in the operating room. The Alarm network is depicted in Figure 4.3.

Given a DAG, we simulate normally distributed samples that are faithful to the DAG, following the procedure of Kalisch and Bühlmann [2007]: 1) obtain a lower-triangle adjacency matrix \( A \) to represent the DAG where \( \text{ones} \) and \( \text{zeros} \) denote directed edges and absence of edges respectively; 2) change the \( \text{ones} \) to be random

\(^2\)http://www.bnlearn.com/bnrepository/
weights in the interval $[0, 1]$. Then, the samples of a random vector $Z$ are drawn through

$$Z_j = \sum_{i<j} A_{ji}Z_i + \epsilon_j,$$

with $\epsilon_j \sim \mathcal{N}(0, 1)$. The data generated in this way follows a multivariate Gaussian distribution with mean zero and covariance matrix $\Sigma = (\mathbb{I} - A)^{-1}(\mathbb{I} - A)^{-T}$, where $\mathbb{I}$ is the identity matrix. In the last step, we scale the data such that each coordinate follows a standard normal distribution, to simulate the random vector $Z$ in Definition 1.1. The implementation of this process and the standard PC algorithm.
is based on the R package pcalg [Kalisch et al., 2010].

Missing values with a certain proportion $\delta_j$ in a variable (the $j$th variable) are created following the procedure in Kolar and Xing [2012]:

- Under MCAR, $\forall i, j, z_{i,j}$ is missing if $r_{i,j} = 0$ where $r_{i,j} \sim Bern(1 - \delta_j)$.
- Under MAR, for $j = 1, \ldots, \lfloor p/2 \rfloor$, $i = 1, \ldots, n$: $z_{i,2* j}$ is missing if $z_{i,2* j - 1} < \Phi^{-1}(\delta_j)$.

Motivated by the two real-world data sets shown in Table 1.1 and 1.2, we give a different missing rate to different variables. Specifically, we randomly draw $\delta_j$ from a uniform distribution as shown in Equation (4.8).

For recovering the causal structure, we consider the order-independent version of PC [Colombo and Maathuis, 2014] as our standard algorithm, in which the significance level is set to $\alpha = 0.01$. For the Gibbs sampling step, we abandon the first 500 samples (burn-in) and save the next 500 for estimating the underlying correlation matrix and the effective sample size.

### 4.6.2 Evaluating Correlation Estimators

Section 4.6.2 illustrates how different missing value strategies behave in correlation estimation. Section 4.6.2 aims to empirically show that the copula estimator has a better convergence rate than the estimator based on Kendall’s $\tau$ whose convergence rate was shown theoretically.

#### Consistency

We now empirically check the behavior of correlation estimators with different strategies for handling missing values through a simple example. We consider a zero-mean bivariate normal distribution with a population correlation $\rho$, in which the first coordinate is fully observed (no missing values). Under MCAR, we randomly set 50% of values in the second coordinate to be missing. Under MAR, the second coordinate is forced to be missing provided that the observations of the first is negative (thus also 50% missing values). A first strategy for missing data is the listwise deletion that reduces to pairwise deletion in bivariate cases, thus it is equivalent to the method proposed in Section 4.3, denoted by ‘Tau’. Another two alternative approaches are based on the mean substitution and hot deck, denoted by ‘MS’ and ‘HD’ respectively for simplicity. A fourth method involved is our copula correlation estimator, denoted by ‘Cop’.

Figure 4.4 shows the results obtained by the four approaches under (a) MCAR and (b) MAR, providing the mean over 100 experiments with error bars representing one standard deviation for different sample sizes $n \in \{100, 500, 1000\}$ and different population correlations $\rho \in \{0, 0.3, 0.6, 0.9\}$, where the dotted horizontal lines denote the true correlations. Under MCAR, we see that estimates of ‘Tau’ and ‘Cop’ are consistently around the true values, which confirms our theoretical results.
Figure 4.4: The true correlations (dotted horizontal line) and the correlations learned by methods based on mean substitution (MS), hot deck (HD), pairwise deletion (Tau), and the copula estimator (Cop) for different sample sizes, showing the mean over 100 experiments with error bars representing one standard deviation, under (a) MCAR and (b) MAR.
in Section 4.3 and 4.4. By contrast, MS and HD report clearly-biased results when the true $\rho$ is not zero (more serious for HD), which is identical to the analysis in Section 4.5.2. Under MAR, the most encouraging result is that our copula estimator can still consistently estimate the correlations while ‘Tau’ fails and MS as well as HD perform even worse than MCAR cases. This compensates the theoretical analysis in Section 4.4.4. When $\rho = 0$, ‘Tau’ goes back to be unbiased because MAR reduces to MCAR in this case.

**Convergence Rate**

While we have shown the convergence rate of the estimator based on Kendall’s $\tau$ in Theorem 4.1, it is difficult to analyze the copula estimator theoretically. Therefore, we empirically compare the convergence rate of the two estimators to get an insight into the finite-sample behavior of the copula estimator. We first randomly generate a $p = 20$ dimensional correlation matrix, under which normally distributed samples are drawn. Then, we fill in some missing values to these samples, to which we apply
the two correlation estimators to learn the correlation matrix. The Supremum (SUP) and Correlation Matrix Distance (CMD) \cite{Herdin et al., 2005} are used to measure the distance between learned and true correlation matrices:

\[ \text{SUP} = \sup_{jk} |\hat{C}_{jk} - C_{jk}|, \quad \text{CMD} = 1 - \frac{\text{tr}(\hat{C} C)}{\|\hat{C}\|_f \|C\|_f}, \]

where \(\text{tr}(\cdot)\) is matrix trace and \(\|\cdot\|_f\) is the Frobenius norm.

Figure 4.5 shows the convergence property of the two estimators for different sample sizes under (a) MCAR and (b) MAR when the expected percentage of missing values \(\beta = 0.25\), providing the mean of SUP and CMD over 100 experiments, where ‘Tau’ and ‘Cop’ denote the estimator based on Kendall’s \(\tau\) and the copula estimator respectively. We see that the copula estimator reports a smaller SUP and CMD for all sample sizes, showing better convergence than the rank-based estimator under both MCAR and MAR. Figure 4.6 provides the results over different proportions of missing values when the sample size \(n = 1000\), for the same experimental setting as in Figure 4.5. It suggests that the copula estimator substantially outperforms the rank-based estimator: the more missing values, the bigger the ad-
vantage. More extensive experiments (not shown) done for different numbers of variables reveal a similar picture. To conclude, the copula correlation estimator is at least bounded by the error bound of the Kendall’s $\tau$ based estimator that is shown in Theorem 4.1.

4.6.3 Causal Discovery on Benchmark DAGs

In this subsection, we evaluate the ‘Rank PC’ (RPC) and ‘Copula PC’ (CoPC), and assess the justification of the usage of the effective sample size in causal discovery on the two benchmark DAGs: the Asia network and the Alarm network. A first alternative is the listwise deletion based approach, in which we first perform listwise deletion and then apply the standard PC algorithm for causal discovery, denoted by ‘PC + LD’. A second alternative considers the mean substitution based approach, denoted by ‘PC + MS’. We do not incorporate the hot deck based approach because, from the previous analysis (Sections 4.5.2 and 4.6.2), we know that MS is better than HD in correlation estimation and they share the same sample size, thus MS should naturally outperform HD in causal discovery.

Three metrics are used to evaluate the algorithms: the true positive rate (TPR) and the false positive rate (FPR), which are defined as

$$TPR = \frac{TP}{|E|} \quad \text{and} \quad FPR = \frac{FP}{p(p-1)/2 - |E|}$$

with $|E|$ the number of edges in the true skeleton, as well as the structural Hamming distance (SHD), counting the number of edge insertions, deletions, and flips in order to transfer the estimated CPDAG into the correct CPDAG [Tsamardinos et al., 2006]. The TPR and FPR evaluate the estimated skeleton while SHD is an overall measure for evaluating the estimated CPDAG. A higher TPR, a lower FPR, and a smaller SHD imply better performance. We consider different proportions of missing values $\beta \in \{0.1, 0.2, 0.3\}$, and different sample sizes $n \in \{100, 500, 1000\}$ for the Asia network and $n \in \{500, 1000, 2000\}$ for the Alarm network.

Figures 4.7 and 4.8 show the results on nonparanormal data generated by the Asia network under MCAR and MAR respectively, providing the mean of TPR, FPR, and SHD over 100 experiments and errorbars representing the 95% confidence interval, where SS, GESS, and LESS represent the original sample size, global effective sample size, and local effective sample size, respectively. Thus, ‘RPC + SS’ denotes the Rank PC with the sample size, ‘RPC + GESS’ denotes the Rank PC with the global effective sample size, etc. We see that, compared to other approaches, ‘PC + LD’ deteriorates dramatically w.r.t. TPR as the percentage of missing values increases regardless of the sample sizes and missingness types. This is due to the sharp decrease of the number of complete cases in the listwise deletion method, as shown in Figure 4.1. ‘PC + MS’, on the other hand, scales well w.r.t. TPR, but reports a very bad result w.r.t. FPR for large sample sizes. Our analysis is that the sample size used in ‘PC + MS’, usually much larger than the number
Figure 4.7: Performance of causal discovery algorithms on nonparanormal data generated by the Asia network under MCAR, showing the mean of TPR, FPR, and SHD over 100 experiments with 95% confidence interval, where ‘PC + LD’ and ‘PC + MS’ denote the standard PC algorithm with listwise deletion and mean substitution, ‘RPC + SS’ denotes the Rank PC with the sample size, ‘RPC + GESS’ denotes the Rank PC with the global effective sample size, etc. The three rows represent the results when the sample sizes are 100, 500, 1000, respectively.

of complete cases used in ‘PC + LD’, makes the conditional independence tests rejected more easily and thus incurs more edges in the resulting graph. Therefore, both ‘PC + LD’ and ‘PC + MS’ give a bad overall performance especially for a
Figure 4.8: Performance of causal discovery algorithms on nonparanormal data generated by the Asia network under MAR, for the same experiments as in Figure 4.7.

larger sample size. By contrast, RPC and CoPC can be seen to be relatively robust to the increase of missing values, where the group of CoPC (with SS, GESS, or LESS) shows an advantage over the group of RPC.

The results for the Alarm network on nonparanormal data are shown in Figure 4.9 under MCAR and Figure 4.10 under MAR, for the same experiments as in Figure 4.7. We do not consider ‘PC + LD’ here, because there are only very few complete records left (2% even when $\beta = 0.1$). From Figures 4.9 and 4.10, we see that RPC and CoPC substantially outperform ‘PC + MS’, as expected. In terms
Figure 4.9: Performance of causal discovery algorithms on nonparanormal data generated by the Alarm network under MCAR. The three rows in each subfigure represent the results when the sample sizes are 500, 1000, 2000, respectively.

of the comparison of Rank PC and Copula PC, we have that, both approaches are indistinguishable under MCAR w.r.t SHD: RPC is slightly better for small sample sizes with many missing values while CoPC shows a small advantage over RPC for larger sample sizes. However, CoPC significantly outperforms RPC w.r.t. all the three metrics under MAR, which becomes even more prominent for larger sample sizes. This is mainly because the Gibbs sampler in CoPC still works quite well in correlation estimation while RPC gives a biased estimate under MAR, as shown in
Next, we analyze whether the effective sample size improves causal discovery. Although a decrease in TPR appears for both CoPC and RPC when SS is replaced with GESS or LESS, we see a bigger improvement in FPR. Thus, w.r.t. the overall metric SHD, the PC algorithms with GESS and LESS perform substantially better than with SS. Also, we notice that LESS can yield more accurate results than GESS: indistinguishable TPR, but better FPR and SHD. Overall, we conclude...
that: 1) compared to the sample size, the usage of an effective sample size (both GESS and LESS) significantly reduces the number of false positives, which thus leads to a better CPDAG; 2) the local effective sample size is a better choice in the conditional independence tests. More experiments (not shown) done for networks with more variables indicate that: the more the variables, the bigger the advantage of LESS over GESS and SS.

Apart from the experiments on the two known DAGs, we also evaluate the algorithms on randomly-simulated DAGs and mixed data. These results that can be found in the Appendix 4.A confirm the above conclusions.

4.7 Application to Real-world Data

In this section, we illustrate our approaches on two real-world datasets: riboflavin production data and chronic fatigue syndrome data. The first contains no missing values while the second contains only a few. The reason why we choose such two datasets is not because datasets with many missing values are not popular, but because we can take the result on the (almost) complete dataset as a baseline to be used for evaluating our approaches on the datasets with some manually-added missing values.

4.7.1 Riboflavin Production Data

Our first application to real-world data considers the dataset of riboflavin production by Bacillus subtilis, which is publicly available in the R package hdci [Dezeure et al., 2015]. It contains 71 continuously-measured observations of 4088 predictors (gene expressions) and a one-dimensional response. For the ease of reproduction, we choose the 10 genes with largest empirical variance as our experimental data, denoted by riboflavinV10\(^3\), as done in Bühlmann et al. [2014]. The resulting graph on all the 71 available observations by the conservative version of ‘Rank PC’ or ‘Copula PC’ with significance level 0.05 is shown in Figure 4.11, which we take as the ‘pseudo ground truth’ to be used for evaluating resulting graphs of the algorithms on data with missing values. The algorithms do not orient any edges, mainly because the number of observations is very small and we use the conservative version of the standard PC algorithm. Then, we manually fill in a specific proportion of missing values (measured by \(\beta\)) to riboflavinV10 following the procedure in Section 4.6.1 and run our algorithms on the resulting incomplete data. The number of ‘Missing edges’ (edges that appear in the true skeleton but not in the learned one) and ‘Extra edges’ (edges that appear in the learned skeleton but not in the true one) are used to evaluate the skeleton, while SHD evaluates the learned CPDAG.

Table 4.1 shows the mean of ‘Missing edges’, ‘Extra edges’, and SHD over 50 experiments with an indication of the number of perfect solutions (‘Missing edges’

\(^3\)The data and code is also available in https://github.com/cuiruifei/CausalMissingValues.
= 0, ‘Extra edges’ = 0, SHD = 0) over these trials, for different proportions of added missing values. ‘PC + LD’ for $\beta = 0.2$ and 0.3 under MCAR leaves only a few complete records and hence fails. It still works under MAR, on the other hand, because here only even-indexed variables contain missing values (see Section 4.6.1). From Table 4.1, we first see that, despite a good performance of ‘PC + LD’ in incurring extra edges, it leads to more missing edges at the same time especially for a larger proportion of missing values, which thus yields a worse SHD than other approaches. Second, MS shows a better performance than LD for handling missing values in causal discovery, which is because the usage of original sample size (much larger than the number of complete records) obtains a better balance between ‘Missing edges’ and ‘Extra edges’. Most importantly, CoPC substantially outperforms RPC and ‘PC + MS’ w.r.t. all the metrics regardless of the proportions of missing values, which becomes more significant under MAR. In addition, we do not see clear difference between ‘CoPC + SS’, ‘CoPC + GESS’, and ‘CoPC + LESS’, which is mainly because the small sample size (only 71 available observations) and small number of variables (only 10) make SS, GESS, and LESS almost indistinguishable.

### 4.7.2 Chronic Fatigue Syndrome Data

In this subsection, we consider a dataset about chronic fatigue syndrome (CFS) of 183 subjects [Heins et al., 2013], which originally comes from a longitudinal study with five time slices. In this chapter, we focus only on one time slice representing the subjects after the first treatment as done in Rahmadi et al. [2017], resulting in a subset of the original data, denoted by $CFS_1$. This dataset contains 6 ordinal variables: 1) fatigue severity assessed with the subscale fatigue severity of the checklist individual strength, denoted by ‘fatigue’; 2) the sense of control over fatigue assessed with the self-efficacy scale, denoted by ‘control’; 3) focusing on symptoms measured with the illness management questionnaire, denoted by ‘focusing’; 4) the objective activity of the patient measured using an actometer, denoted by ‘oActivity’; 5) the subject’s perceived activity measured with the subscale activity of the checklist individual strength, denoted by ‘pActivity’; and 6) physical functioning measured with subscale physical functioning of the medical outcomes survey, denoted by ‘functioning’. For a detailed description of the questionnaires, the actometer, and other information, we refer the readers to Heins et al. [2013].
In CFS1, there are only a few missing values: 2 in ‘fatigue’, 2 in ‘control’, 2 in ‘focusing’, 21 in ‘oActivity’, 2 in ‘pActivity’, and 2 in ‘functioning’. We run the conservative version of ‘Hetcor PC’ (HPC) and ‘Copula PC’ (CoPC) with significance level 0.05 on CFS1. Due to the small number of missing values, both HPC and CoPC output the same structure shown in Figure 4.12(a), regardless of using SS, GESS or LESS. We take this structure as the ‘pseudo ground truth’. Then, we manually add more missing values to CFS1 as follows: 1) set ‘oActivity’ to be missing when ‘pActivity’ is smaller than the 37th smallest observation (that is, since 20% × 183 = 36.6, we add about 20% missing values to ‘oActivity’ depending on ‘pActivity’); 2) set ‘fatigue’ to be missing provided that ‘functioning’ is smaller than the 37th smallest observation; and 3) set ‘control’ to be missing given ‘focusing’ under the same condition. We refer to the resulting dataset as CFS1_0. The datasets CFS1 and CFS1_0, as well as the code are publicly available.\footnote{https://github.com/cuiruifei/CausalMissingValues}

The learned graphs of running the causal discovery approaches on CFS1_0 are shown in Figure 4.12 from (b) to (f), in which ‘HPC + GESS’ and ‘HPC + LESS’ output the same structure shown in (e) while CoPCs with SS, GESS, and LESS output the same structure shown in (f). Compared to the ‘pseudo ground truth’, ‘PC + LD’ reports the absence of three edges, in correspondence with what we hypothesised in Section 4.5 and the empirical results in Section 4.6. ‘PC + MS’ gives a very bad result: four missing edges and two extra edges. ‘HPC + SS’ indicates one missing edge, two extra edges, and some extra orientations while ‘HPC + GESS or LESS’ suggests two missing edges and one extra edge. By contrast, it is very encouraging that the Copula PC algorithm only implies one missing edge, showing better performance than the other approaches.

4.8 Conclusion and Future Work

In this chapter, we extended the ‘Rank PC’ algorithm to incomplete data by applying rank correlations to pairwise complete observations and taking the number of pairwise complete observations as an effective sample size. Similar idea could be used to extend the ‘Hetcor PC’ algorithm to incomplete data by applying the heterogeneous correlation estimator to pairwise complete observations. Despite theoretical guarantees, this naive pairwise-deletion-based approach has several limitations. First, it only works under MCAR, which is a strong assumption that is quite difficult to justify. Departures from MCAR may lead to a biased analysis and a possibly distorted conclusion. Third, it is hard to compute standard errors or other measures of uncertainty since parameters are estimated from different sets of units. See Schafer and Graham [2002] for more information about the limitations of pairwise complete case analysis.

To solve these limitations, we proposed a novel Bayesian approach, in which a Gibbs sampler is designed to draw correlation matrix samples from the posterior
distribution given incomplete data. These are then translated into the underlying correlation matrix and the effective sample size for causal discovery. One highlight of this approach is that it works for mixed data under MAR, a less restrictive assumption, and even if MAR fails, Bayesian methods like ours can still show strong robustness [Schafer and Graham, 2002]. Another highlight is that the approach uses an elegant way to carry over the additional uncertainty from missing values to con-
ditional independence tests. From the experiments, the Gibbs sampler used in our approach showed good scalability over the network size, in the sense that the burn-in period (number of iterations before convergence) hardly grows as the number of variables increases. In addition, one could plug in some available optimizations of this step [Kalaitzis and Silva, 2013] to reduce the time complexity.

For both ‘Rank PC’ and ‘Copula PC’, we replaced the sample size with an effective sample size in the tests for conditional independence when that data contains missing values, which significantly improves the performance of the PC algorithm. Especially, a local effective sample size for each conditional independence test makes much sense in particular when some variables contain more missing values than others. While we considered the PC algorithm for estimating the underlying causal structure, the idea of using the (local) effective sample size can be applied to other standard algorithms like FCI [Spirtes et al., 2000], in particular for handling potential confounders and selection bias, GES [Chickering, 2002b], or their state-of-the-art variants [Claassen et al., 2013; Triantafillou and Tsamardinos, 2015; Magliacane et al., 2016].

Although our interest in this chapter is in causal structure estimation, the proposed technique for handling missing values in Section 4.4.1 can serve as a general tool for other tasks, e.g., factor analysis [Murray et al., 2013; Gruhl et al., 2013] and undirected graphical models [Dobra et al., 2011; Fan et al., 2017]. Our method can not only give a quite good estimate for the underlying correlation matrix under MAR, but also provides an uncertainty measure for this estimate, which is especially important in analyses based on incomplete data.

While the extended rank likelihood (the basis of our Gibbs sampler) is justifiable for ordinal and continuous variables, it cannot meaningfully handle numeric values for nominal variables (categorical variables without ordering). To include such nominal variables in our copula model, we may consider a multinomial probit model. The main idea is to relate a nominal variable to a vector of latent variables that can be thought of as the unnormalized probabilities of choosing each of the categories, as done in Wang et al. [2017a]. Also, we consider extending our work to MNAR cases, which can be done under some additional assumptions, e.g., that none of the missingness indicators causally affect each other in the underlying causal graph [Strobl et al., 2017].

4.4A Evaluation on Simulated DAGs

In Section 4.6, we showed the experimental results on two well-known benchmark networks. In order to test our algorithms on more networks, we randomly simulate DAGs following the procedure of Kalisch and Bühlmann [2007] that is implemented via the function randomDAG in the R package pcalg [Kalisch et al., 2010]. We restrict the number of variables to \( p = 20 \) and set the sparseness parameter in generating DAGs to \( s = 2/(p - 1) \), such that the average neighbors of each node is two [Kalisch and Bühlmann, 2007]. For each experiment, we obtain a random DAG
Figure 4.13: Performance of causal discovery algorithms on nonparanormal data generated by randomly-simulated DAGs under MCAR, showing the mean of TPR, FPR, and SHD over 100 experiments with 95% confidence interval. The three rows in each subfigure represent the results when the sample sizes are 500, 1000, 2000, respectively.

that is used to generate nonparanormal data, on which we evaluate our algorithms. The mean of TPR, FPR, and SHD over 100 experiments with 95% confidence interval are shown in Figure 4.13 under MCAR and Figure 4.14 under MAR.

In order to evaluate the performance of Copula PC on mixed data, we generate data as follows: 1) generate Gaussian data and fill in some missing values (as we
Figure 4.14: Performance of causal discovery algorithms on nonparanormal data generated by randomly-simulated DAGs under MAR, showing the mean of TPR, FPR, and SHD over 100 experiments with 95% confidence interval. The three rows in each subfigure represent the results when the sample sizes are 500, 1000, 2000, respectively.

The results in Figures 4.13, 4.14, 4.15 and 4.16 confirm our conclusions: 1) both
Figure 4.15: Performance of causal discovery algorithms on mixed data generated by randomly-simulated DAGs under MCAR. The three rows in each subfigure represent the results when the sample sizes are 500, 1000, 2000, respectively.

Rank PC and Copula PC substantially outperform a simple data-interpolation-based method ‘PC + MS’; 2) the Copula PC algorithm shows a significant advantage over the Rank (or Hetcor) PC algorithm under MAR; 3) the PC algorithm with the local effective sample size performs better than with the global effective sample size, which in turn outperforms the one with the original sample size.
Figure 4.16: Performance of causal discovery algorithms on mixed data generated by randomly-simulated DAGs under MAR. The three rows in each subfigure represent the results when the sample sizes are 500, 1000, 2000, respectively.
Table 4.1: Results obtained by various causal discovery algorithms on the riboflavinV10 dataset with different proportions of missing values (β), showing the mean of missing edges, extra edges, and SHD over 50 repeated experiments with an indication of the number of perfect solutions (the corresponding metric is 0) over these trials.

<table>
<thead>
<tr>
<th>Missing edges</th>
<th>Extra edges</th>
<th>SHD</th>
</tr>
</thead>
<tbody>
<tr>
<td>β 0.1</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td>MCAR</td>
<td>PC + LD</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>PC + MS</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td>RPC + SS</td>
<td>4.0</td>
</tr>
<tr>
<td></td>
<td>CoPC + SS</td>
<td>5.0</td>
</tr>
<tr>
<td>MAR</td>
<td>PC + LD</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>PC + MS</td>
<td>1.7</td>
</tr>
<tr>
<td></td>
<td>RPC + SS</td>
<td>1.3</td>
</tr>
<tr>
<td></td>
<td>CoPC + SS</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>MAR</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>PC + MS</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>RPC + SS</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>CoPC + SS</td>
<td>0.8</td>
</tr>
</tbody>
</table>

The number of perfect solutions (the corresponding metric is 0) over these trials.
We consider the problem of learning parameters of latent variable models from mixed (continuous and ordinal) data with missing values. We propose a novel Bayesian Gaussian copula factor (BGCF) approach that is consistent under certain conditions and that is quite robust to the violations of these conditions. In simulations, BGCF substantially outperforms two state-of-the-art alternative approaches. An illustration on the ‘Holzinger & Swineford 1939’ dataset indicates that BGCF is favorable over the so-called robust maximum likelihood (MLR) even if the data match the assumptions of MLR.

5.1 Introduction

In psychology, social sciences, and many other fields, researchers are usually interested in “latent” concepts that cannot be measured directly, e.g., depression, anxiety, or intelligence. To get a grip on these latents, one commonly-used strategy is to construct a measurement model for such a latent concept, in the sense that domain experts design multiple “items” or “questions” that are considered to be indicators of the latent concept. For exploring evidence of construct validity in theory-based instrument construction, confirmatory factor analysis (CFA) has been widely studied [Jöreskog, 1969; Castro et al., 2015; Li, 2016]. In CFA, researchers start with several hypothesised latent variable models that are then fitted to the data individually, after which the one that fits the data best is picked to explain
the observed phenomenon. In this process, the fundamental task is to learn the parameters of a hypothesised model from observed data, which is the focus of this chapter. For convenience, we simply refer to these hypothesised latent variable models as CFA models from now on.

The most common method for parameter estimation in CFA models is maximum likelihood (ML), because of its attractive statistical properties (consistency, asymptotic normality, and efficiency). The ML method, however, relies on the assumption that observed variables follow a multivariate normal distribution [Jöreskog, 1969]. When the normality assumption is not deemed empirically tenable, ML may not only reduce the accuracy of parameter estimates, but may also yield misleading conclusions drawn from empirical data [Li, 2016]. To this end, a robust version of ML is introduced for CFA models when the normality assumption is slightly or moderately violated [Kaplan, 2008], but still requires the observations to be continuous. In the real world, the indicator data in questionnaires are usually measured on an ordinal scale (resulting in a bunch of ordered categorical variables, or simply ordinal variables) [Poon and Wang, 2012], in which neither normality nor continuity is plausible [Lubke and Muthén, 2004]. In such cases, diagonally weighted least squares (DWLS in LISREL; WLSMV or robust WLS in Mplus) has been suggested to be superior to the ML method and is usually considered to be preferable over other methods [Barendse et al., 2015; Li, 2016].

However, there are two major issues that the existing approaches do not consider. One is the mixture of continuous and ordinal data. As we mentioned above ordinal variables are omnipresent in questionnaires, whereas sensor data are usually continuous. Therefore, a more realistic case in real applications is mixed continuous and ordinal data. A second important issue concerns missing values. In practice, all branches of experimental science are plagued by missing values [Little and Rubin, 1987], e.g., failure of sensors, or unwillingness to answer certain questions in a survey. A straightforward idea in this case is to combine missing values techniques with existing parameter estimation approaches, e.g., performing listwise-deletion or pairwise-deletion first on the original data and then applying DWLS to learn parameters of a CFA model. However, such deletion methods are only consistent when the data are missing completely at random (MCAR), which is a rather strong assumption [Rubin, 1976], and cannot transfer the sampling variability incurred by missing values to follow-up studies. The two modern missing data techniques, maximum likelihood and multiple imputation, are valid under a less restrictive assumption, missing at random (MAR) [Schafer and Graham, 2002], but they require the data to be multivariate normal.

Therefore, there is a strong demand for an approach that is not only valid under MAR but also works for mixed continuous and ordinal data. For this purpose, we propose a novel Bayesian Gaussian copula factor (BGCF) approach, in which a Gibbs sampler is used to draw pseudo Gaussian data in a latent space restricted by the observed data (unrestricted if that value is missing) and draw posterior samples of parameters given the pseudo data, iteratively. We prove that this approach is
consistent under MCAR and empirically show that it works quite well under MAR.

The rest of this chapter is organized as follows. Section 5.2 reviews background knowledge and related work. Section 5.3 gives the definition of a Gaussian copula factor model and Section 5.4 presents a novel inference procedure for this model. Section 5.5 compares our BGCF approach with two alternative approaches on simulated data, and Section 5.6 gives an illustration on the ‘Holzinger & Swineford 1939’ dataset. Section 5.7 concludes this chapter and provides some discussion.

## 5.2 Background

In this section, we review basic missingness mechanisms and related work on parameter estimation in CFA models.

### 5.2.1 Missingness Mechanism

Following Rubin [1976], let $Y = (y_{ij}) \in \mathbb{R}^{n \times p}$ be a data matrix with the rows representing independent samples, and $R = (r_{ij}) \in \{0, 1\}^{n \times p}$ be a matrix of indicators, where $r_{ij} = 1$ if $y_{ij}$ was observed and $r_{ij} = 0$ otherwise. $Y$ consists of two parts, $Y_{\text{obs}}$ and $Y_{\text{miss}}$, representing observed and missing elements in $Y$ respectively. When the missingness does not depend on the data, i.e., $P(R|Y, \theta) = P(R|\theta)$ with $\theta$ denoting unknown parameters, the data are said to be missing completely at random (MCAR), which is a special case of a more realistic assumption called missing at random (MAR). MAR allows the dependency between missingness and observed values, i.e., $P(R|Y, \theta) = P(R|Y_{\text{obs}}, \theta)$. For example, all people in a group are required to take a blood pressure test at time point 1, while only those whose values at time point 1 lie in the abnormal range need to take the test at time point 2. This results in some missing values at time point 2 that are MAR.

### 5.2.2 Parameter Estimation in CFA Models

When the observations follow a multivariate normal distribution, maximum likelihood (ML) is the mostly-used method. It is equivalent to minimizing the discrepancy function $F_{\text{ML}}$ [Jöreskog, 1969]:

$$F_{\text{ML}} = \ln|\Sigma(\theta)| + \text{tr} \left[ SS^{-1}(\theta) \right] - \ln|S| - p,$$

where $\theta$ is the vector of model parameters, $\Sigma(\theta)$ is the model-implied covariance matrix, $S$ is the sample covariance matrix, and $p$ is the number of observed variables in the model. When the normality assumption is violated either slightly or moderately, robust ML (MLR) offers an alternative. Here parameter estimates are still obtained using the asymptotically unbiased ML estimator, but standard errors are statistically corrected to enhance the robustness of ML against departures from normality [Kaplan, 2008; Muthén, 2010]. Another method for continuous nonnormal data is the so-called asymptotically distribution free method, which is a weighted...
least squares (WLS) method using the inverse of the asymptotic covariance matrix of the sample variances and covariances as a weight matrix [Browne, 1984].

When the observed data are on ordinal scales, Muthén [1984] proposes a three-stage approach. It assumes that a normal latent variable $x^*$ underlies an observed ordinal variable $x$, i.e.,

$$x = m, \text{ if } \tau_{m-1} < x^* < \tau_m,$$

where $m (= 1, 2, ..., c)$ denotes the observed values of $x$, $\tau_m$ are thresholds $(-\infty = \tau_0 < \tau_1 < \tau_2 < ... < \tau_c = +\infty)$, and $c$ is the number of categories. The thresholds and polychoric correlations are estimated from the bivariate contingency table in the first two stages [Olsson, 1979; Jöreskog, 2005]. Parameter estimates and the associated standard errors are then obtained by minimizing the weighted least squares fit function $F_{WLS}$:

$$F_{WLS} = (s - \sigma(\theta))^T W^{-1} (s - \sigma(\theta)),$$

where $\theta$ is the vector of model parameters, $\sigma(\theta)$ is the model-implied vector containing the nonredundant vectorized elements of $\Sigma(\theta)$, $s$ is the vector containing the estimated polychoric correlations, and the weight matrix $W$ is the asymptotic covariance matrix of the polychoric correlations. A mathematically simple form of the WLS estimator, the unweighted least squares (ULS), arises when the matrix $W$ is replaced with the identity matrix. Another variant of WLS is the diagonally weighted least squares (DWLS), in which only the diagonal elements of $W$ are used in the fit function [Muthén et al., 1997; Muthén, 2010], i.e.,

$$F_{DWLS} = (s - \sigma(\theta))^T W_D^{-1} (s - \sigma(\theta)),$$

where $W_D^{-1} = \text{diag}(W)$ is the diagonal weight matrix. Various recent simulation studies have shown that DWLS is favorable compared to WLS, ULS, as well as the ML-based methods for ordinal data [Barendse et al., 2015; Li, 2016].

### 5.3 Gaussian Copula Factor Model

**Definition 5.1 (Gaussian Copula Factor Model).** Consider a latent random (factor) vector $\eta = (\eta_1, \ldots, \eta_k)^T$, a response random vector $Z = (Z_1, \ldots, Z_p)^T$ and an observed random vector $Y = (Y_1, \ldots, Y_p)^T$, satisfying

$$\eta \sim N(0, C), \tag{5.1}$$

$$Z = \Lambda \eta + \epsilon, \tag{5.2}$$

$$Y_j = F_j^{-1}(\Phi[Z_j/\sigma(Z_j)]), \forall j = 1, \ldots, p, \tag{5.3}$$

with $C$ a correlation matrix over factors, $\Lambda = (\lambda_{ij})$ a $p \times k$ matrix of factor loadings ($k \leq p$), $\epsilon \sim N(0, D)$ residuals with $D = \text{diag}(\sigma_1^2, \ldots, \sigma_p^2)$, $\sigma(Z_j)$ the standard deviation of $Z_j$, $\Phi(\cdot)$ the cumulative distribution function (CDF) of the standard Gaussian, and $F_j^{-1}(t) = \inf\{x : F_j(x) \geq t\}$ the pseudo-inverse of a CDF $F_j(\cdot)$. Then this model is called a Gaussian copula factor model.
The model is also defined in Murray et al. [2013], but the authors restrict the factors to be independent of each other while we allow for their interactions. Our model is a combination of a Gaussian factor model (from $\eta$ to $Z$) and a Gaussian copula model (from $Z$ to $Y$). The first part allows us to model the latent concepts that are measured by multiple indicators, and the second part provides a good way to model diverse types of variables (depending on $F_j(\cdot)$ in Equation 5.3, $Y_j$ can be either continuous or ordinal). Figure 5.1 shows an example of the model. Note that we allow the special case of a factor having a single indicator, e.g., $\eta_1 \to Z_1 \to Y_1$, because this allows us to incorporate other (explicit) variables (such as age and income) into our model. In this special case, we set $\lambda_{11} = 1$ and $\epsilon_1 = 0$, thus $Y_1 = F_1^{-1}(\Phi[\eta_1])$.

In the typical design for questionnaires, one tries to get a grip on a latent concept through a particular set of well-designed questions [Martínez-Torres, 2006; Byrne, 2013], which implies that a factor (latent concept) in our model is connected to multiple indicators (questions) while an indicator is only used to measure a single factor, as shown in Figure 5.1. This kind of measurement model is called a pure measurement model (Definition 8 in Silva et al. [2006]). Throughout this chapter, we assume that all measurement models are pure, which indicates that there is only a single non-zero entry in each row of the factor loadings matrix $\Lambda$. This inductive bias about the sparsity pattern of $\Lambda$ is fully motivated by the typical design of a measurement model.

In what follows, we transform the Gaussian copula factor model into an equivalent model that is used for inference in the next subsection. We consider an integrated $(p + k)$-dimensional random vector $X = (Z^T, \eta^T)^T$, which is still multivariate Gaussian, and obtain its covariance matrix

$$
\Sigma = \begin{bmatrix}
\Lambda C \Lambda^T + D & \Lambda C \\
C \Lambda^T & C
\end{bmatrix}, \quad (5.4)
$$

Figure 5.1: Gaussian copula factor model.
and precision matrix

$$\Omega = \Sigma^{-1} = \begin{bmatrix} D^{-1} & -D^{-1}\Lambda \\ -\Lambda^T D^{-1} & C^{-1} + \Lambda^T D^{-1}\Lambda \end{bmatrix}. \tag{5.5}$$

Since $D$ is diagonal and $\Lambda$ only has one non-zero entry per row, $\Omega$ contains many intrinsic zeros. The sparsity pattern of such $\Omega = (\omega_{ij})$ can be represented by an undirected graph $G = (V, E)$, where $(i, j) \notin E$ whenever $\omega_{ij} = 0$ by construction. Then, a Gaussian copula factor model can be transformed into an equivalent model controlled by a single precision matrix $\Omega$, which in turn is constrained by $G$, i.e., $P(X|C, \Lambda, D) = P(X|\Omega_G)$.

**Definition 5.2 (G-Wishart Distribution [Roverato, 2002]).** Given an undirected graph $G = (V, E)$, a zero-constrained random matrix $\Omega$ has a $G$-Wishart distribution, if its density function is

$$p(\Omega|G) = \frac{|\Omega|^{(\nu-2)/2}}{I_G(\nu, \Psi)} \exp \left[ -\frac{1}{2} \text{tr}(\Psi\Omega) \right] \mathbb{1}_{\Omega \in M^+(G)},$$

with $M^+(G)$ the space of symmetric positive definite matrices with off-diagonal elements $\omega_{ij} = 0$ whenever $(i, j) \notin E$, $\nu$ the number of degrees of freedom, $\Psi$ a scale matrix, $I_G(\nu, \Psi)$ the normalizing constant, and $\mathbb{1}(\cdot)$ the indicator function.

The $G$-Wishart distribution is the conjugate prior of precision matrices $\Omega$ that are constrained by a graph $G$ [Roverato, 2002]. That is, given the $G$-Wishart prior, i.e., $P(\Omega|G) = \mathcal{W}_G(\nu_0, \Psi_0)$ and data $X = (x_1, \ldots, x_n)^T$ drawn from $\mathcal{N}(0, \Omega^{-1})$, the posterior for $\Omega$ is another $G$-Wishart distribution:

$$P(\Omega|G, X) = \mathcal{W}_G(\nu_0 + n, \Psi_0 + X^T X).$$

When the graph $G$ is fully connected, the $G$-Wishart distribution reduces to a Wishart distribution [Murphy, 2007]. Placing a $G$-Wishart prior on $\Omega$ is equivalent to placing an inverse-Wishart on $C$, a product of multivariate normals on $\Lambda$, and an inverse-gamma on the diagonal elements of $D$. With a diagonal scale matrix $\Psi_0$ and the number of degrees of freedom $\nu_0$ equal to the number of factors plus one, the implied marginal densities between any pair of factors are uniformly distributed between $[-1, 1]$ [Barnard et al., 2000].

### 5.4 A Novel Inference Approach for Gaussian Copula Factor Model

In this section, we first propose a novel Bayesian inference procedure for Gaussian copula factor models. Then, we theoretically analyze the identifiability and prove the consistency of our procedure.
5.4.1 Inference for Gaussian Copula Factor Model

We first introduce the inference procedure for complete mixed data and incomplete Gaussian data respectively, based on which the procedure for mixed data with missing values is then derived. From this point on, we use \( S \) to denote the correlation matrix over the response vector \( Z \).

Mixed Data without Missing Values

For a Gaussian copula model, Hoff [2007] proposed a likelihood that only concerns the ranks among observations, which is derived as follows. Since the transformation \( Y_j = F_j^{-1}(\Phi[Z_j]) \) is non-decreasing, observing \( y_j = (y_{1,j}, \ldots, y_{n,j})^T \) implies a partial ordering on \( z_j = (z_{1,j}, \ldots, z_{n,j})^T \), i.e., \( z_j \) lies in the space restricted by \( y_j \):

\[
D(y_j) = \{ z_j \in \mathbb{R}^n : y_{i,j} < y_{k,j} \Rightarrow z_{i,j} < z_{k,j} \}.
\]

Therefore, observing \( Y \) suggests that \( Z \) must be in

\[
D(Y) = \{ Z \in \mathbb{R}^{n \times p} : z_j \in D(y_j), \forall j = 1, \ldots, p \}.
\]

Taking the occurrence of this event as the data, one can compute the following likelihood [Hoff, 2007]

\[
P(Z \in D(Y)|S, F_1, \ldots, F_p) = P(Z \in D(Y)|S).
\]

Following the same argumentation, the likelihood in our Gaussian copula factor model reads

\[
P(Z \in D(Y)|\eta, \Omega, F_1, \ldots, F_p) = P(Z \in D(Y)|\eta, \Omega),
\]

which is independent of the margins \( F_j \).

For the Gaussian copula factor model, inference for the precision matrix \( \Omega \) of the vector \( X = (Z^T, \eta^T)^T \) can now proceed via construction of a Markov chain having its stationary distribution equal to \( P(Z, \eta, \Omega|Z \in D(Y), G) \), where we ignore the values for \( \eta \) and \( Z \) in our samples. The prior graph \( G \) is uniquely determined by the sparsity pattern of the loading matrix \( \Lambda = (\lambda_{ij}) \) and the residual matrix \( D \) (see Equation 5.5), which in turn is uniquely decided by the pure measurement models.

The Markov chain can be constructed by iterating the following three steps:

1. **Sample** \( Z \): \( Z \sim P(Z|\eta, Z \in D(Y), \Omega) \);
   
   Since each coordinate \( Z_j \) directly depends on only one factor, i.e., \( \eta_q \) such that \( \lambda_{jq} \neq 0 \), we can sample each of them independently through \( Z_j \sim P(Z_j|\eta_q, z_j \in D(y_j), \Omega) \).

2. **Sample** \( \eta \): \( \eta \sim P(\eta|Z, \Omega) \);

3. **Sample** \( \Omega \): \( \Omega \sim P(\Omega|Z, \eta, G) \).
Gaussian Data with Missing Values

Suppose that we have Gaussian data \( Z \) consisting of two parts, \( Z_{\text{obs}} \) and \( Z_{\text{miss}} \), denoting observed and missing values in \( Z \) respectively. The inference for the correlation matrix of \( Z \) in this case can be done via the so-called data augmentation technique that is also a Markov chain Monte Carlo procedure and has been proven to be consistent under MAR [Schafer, 1997]. This approach iterates the following two steps to impute missing values (Step 1) and draw correlation matrix samples from the posterior (Step 2):

1. \( Z_{\text{miss}} \sim P(Z_{\text{miss}} | Z_{\text{obs}}, S) \);
2. \( S \sim P(S | Z_{\text{obs}}, Z_{\text{miss}}) \).

Mixed Data with Missing Values

For the most general case of mixed data with missing values, we combine the two procedures introduced above into the following four-step inference procedure:

1. \( Z_{\text{obs}} \sim P(Z_{\text{obs}} | \eta, Z_{\text{obs}} \in D(Y_{\text{obs}}), \Omega) \);
2. \( Z_{\text{miss}} \sim P(Z_{\text{miss}} | \eta, Z_{\text{obs}}, \Omega) \);
3. \( \eta \sim P(\eta | Z_{\text{obs}}, Z_{\text{miss}}, \Omega) \);
4. \( \Omega \sim P(\Omega | Z_{\text{obs}}, Z_{\text{miss}}, \eta, G) \).

A Gibbs sampler that achieves this Markov chain is summarized in Algorithm 5.1 and implemented in R.\(^1\) Note that we put Step 1 and Step 2 together in the actual implementation since they share some common computations (lines 2 - 4). The difference between the two steps is that the values in Step 1 are drawn from a space restricted by the observed data (lines 5 - 13) while the values in Step 2 are drawn from an unrestricted space (lines 14 - 17). Another important point is that we need to relocate the data such that the mean of each coordinate of \( Z \) is zero (line 20). This is necessary for the algorithm to be sound because the mean may shift when missing values depend on the observed data (MAR).

By iterating the steps in Algorithm 5.1, we can draw correlation matrix samples over the integrated random vector \( X \), denoted by \( \{\Sigma^{(1)}, \ldots, \Sigma^{(m)}\} \). The mean over all the samples is a natural estimate of the true \( \Sigma \), i.e.,

\[
\hat{\Sigma} = \frac{1}{m} \sum_{i=1}^{m} \Sigma^{(i)} .
\] (5.6)

\(^1\)The code including those used in simulations and real-world applications is provided in https://github.com/cuiruifei/CopulaFactorModel.
Algorithm 5.1 Gibbs sampler for Gaussian copula factor model with missing values

Require: Prior graph $G$, observed data $Y$.

# Step 1 and Step 2:
1: for $j \in \{1, \ldots, p\}$ do
2: \hspace{1em} $q = \text{factor index of } Z_j$
3: \hspace{1em} $a = \frac{\Sigma_{[j,q+p]/\Sigma_{[y+p,q+p]}}}{\Sigma_{[j,j]}}$
4: \hspace{1em} $\sigma^2_j = \Sigma_{[j,j]} - a \times \Sigma_{[y+p,j]}$
5: end for

# Step 1: $Z_{obs} \sim P(Z_{obs} | \eta, Z_{obs} \in D(Y_{obs}), \Omega)$
6: for $y \in \text{unique}\{y_{1,j}, \ldots, y_{n,j}\}$ do
7: \hspace{1em} $z_l = \max \{z_{i,j} : y_{i,j} < y\}$
8: \hspace{1em} $z_u = \min \{z_{i,j} : y < y_{i,j}\}$
9: \hspace{1em} for $i$ such that $y_{i,j} = y$ do
10: \hspace{2em} $\mu_{i,j} = \eta_{[i,q]} \times a$
11: \hspace{2em} $u_{i,j} \sim U(\Phi[z_l - \mu_{i,j}, \sigma^2_j], \Phi[z_u - \mu_{i,j}, \sigma^2_j])$
12: \hspace{2em} $z_{i,j} = \mu_{i,j} + \sigma_j \times \Phi^{-1}(u_{i,j})$
13: \hspace{1em} end for
14: end for

# Step 2: $Z_{miss} \sim P(Z_{miss} | \eta, Z_{obs}, \Omega)$
15: for $i$ such that $y_{i,j} \in Y_{miss}$ do
16: \hspace{1em} $\mu_{i,j} = \eta_{[i,q]} \times a$
17: \hspace{1em} $z_{i,j} \sim N(\mu_{i,j}, \sigma^2_j)$
18: end for
19: $Z = (Z_{obs}, Z_{miss})$
20: $Z = (Z^T - \mu)^T$, with $\mu$ the mean vector of $Z$

# Step 3: $\eta \sim P(\eta | Z, \Omega)$
21: $A = \Sigma_{[\eta,Z]}\Sigma_{[Z,Z]}^{-1}$
22: $B = \Sigma_{[\eta,\eta]} - A\Sigma_{[Z,\eta]}$
23: for $i \in \{1, \ldots, n\}$ do
24: \hspace{1em} $\mu_i = (Z_{[i,:]}A^T)^T$
25: \hspace{1em} $\eta_{[i,:]} \sim N(\mu_i, B)$
26: end for
27: $\eta_{[i,j]} = \eta_{[i,j]} \times \text{sign}(\text{Cov}[^{\eta_{[i,j]}}; Z_{[i,f(j)]}]), \forall j$, where $f(j)$ is the index of the first indicator of $\eta_j$.

# Step 4: $\Omega \sim P(\Omega | Z, \eta, G)$
28: $X = (Z, \eta)$
29: $\Omega \sim W_G(\nu_0 + n, \Psi_0 + X^TX)$
30: $\Sigma = \Omega^{-1}$
31: $\Sigma_{ij} = \Sigma_{ij} / \sqrt{\Sigma_{ii} \Sigma_{jj}}, \forall i, j$

Based on Equations (5.4) and (5.6), we obtain estimates of the parameters of inter-
\[
\hat{C} = \hat{\Sigma}_{[\eta, \eta]}; \\
\hat{\Lambda} = \hat{\Sigma}_{[Z, \eta]} \hat{C}^{-1}; \\
\hat{D} = \hat{S} - \hat{\Lambda} \hat{C} \hat{\Lambda}^T, \quad \text{with} \quad \hat{S} = \hat{\Sigma}_{[Z, Z]}.
\]

We refer to this procedure as a Bayesian Gaussian copula factor approach (BGCF).

### 5.4.2 Theoretical Analysis

**Identifiability of \( C \)** Without additional constraints, \( C \) is non-identifiable [Anderson and Rubin, 1956]. More precisely, given a decomposable matrix \( S = \Lambda C \Lambda^T + D \), we can always replace \( \Lambda \) with \( \Lambda U \) and \( C \) with \( U^{-1} C U^{-T} \) to obtain an equivalent decomposition \( S = (\Lambda U)(U^{-1} C U^{-T})(U^T \Lambda^T) + D \), where \( U \) is a \( k \times k \) invertible matrix. Since \( \Lambda \) only has one non-zero entry per row in our model, \( U \) can only be diagonal to ensure that \( \Lambda U \) has the same sparsity pattern as \( \Lambda \) (see Lemma 5.1 in Appendix 5.A). Thus, from the same \( S \), we get a class of solutions for \( C \), i.e., \( U^{-1} C U^{-T} \), where \( U \) can be any invertible diagonal matrix. In order to get a unique solution for \( C \), we impose two sufficient identifying conditions: 1) restrict \( C \) to be a correlation matrix; 2) force the first non-zero entry in each column of \( \Lambda \) to be positive. See Lemma 5.2 in Appendix 5.A for proof. Condition 1 is implemented via line 31 in Algorithm 5.1. As for the second condition, we force the covariance between a factor and its first indicator to be positive (line 27), which is equivalent to Condition 2. Note that these conditions are not unique; one could choose one’s favorite conditions to identify \( C \), e.g., setting the first loading to 1 for each factor. The reason for our choice of conditions is to keep it consistent with our model definition where \( C \) is a correlation matrix.

**Identifiability of \( \Lambda \) and \( D \)** Under the two conditions for identifying \( C \), factor loadings \( \Lambda \) and residual variances \( D \) are also identified except for the case in which there exists one factor that is independent of all the others and this factor only has two indicators. For such a factor, we have 4 free parameters (2 loadings, 2 residuals) while we only have 3 available equations (2 variances, 1 covariance), which yields an underdetermined system. See Lemmas 5.3 and 5.4 in Appendix 5.A for detailed analysis. Once this happens, one could put additional constraints to guarantee a unique solution, e.g., by setting the variance of the first residual to zero. However, we would recommend to leave such an independent factor out (especially in association analysis) or study it separately from the other factors.

Under sufficient conditions for identifying \( C, \Lambda, \) and \( D \), our BGCF approach is consistent even with MCAR missing values, which is shown in Theorem 5.1.

**Theorem 5.1** (Consistency of the BGCF Approach). Let \( Y_n = (y_1, \ldots, y_n)^T \) be independent observations drawn from a Gaussian copula factor model. If \( Y_n \) is
complete (no missing data) or contains missing values that are missing completely at random, then

\[
\lim_{n \to \infty} P(\hat{C}_n = C_0) = 1 , \\
\lim_{n \to \infty} P(\hat{\Lambda}_n = \Lambda_0) = 1 , \\
\lim_{n \to \infty} P(\hat{D}_n = D_0) = 1 ,
\]

where \(\hat{C}_n\), \(\hat{\Lambda}_n\), and \(\hat{D}_n\) are parameters learned by BGCF, while \(C_0\), \(\Lambda_0\), and \(D_0\) are the true ones.

**Proof.** See Appendix 5.A. \(\square\)

## 5.5 Simulation Study

In this section, we compare our BGCF approach with alternative approaches via simulations.

### 5.5.1 Setup

**Model specification**  Following typical simulation studies on CFA models in the literature [Yang-Wallentin et al., 2010; Li, 2016], we consider a correlated 4-factor model in our study. Each factor is measured by 4 indicators, since Marsh et al. [1998] concluded that the accuracy of parameter estimates appeared to be optimal when the number of indicators per factor was four and marginally improved as the number increased. The interfactor correlations (off-diagonal elements of the correlation matrix \(C\) over factors) are randomly drawn from \([0.2, 0.4]\), which is considered a reasonable and empirical range in the applied literature [Li, 2016]. For the ease of reproducibility, we construct our \(C\) as follows.

```r
set.seed(12345)
C <- matrix(runif(4^2, 0.2, 0.4), ncol=4)
C <- (C*lower.tri(C)) + t(C*lower.tri(C))
diag(C) <- 1
```

In the majority of empirical research and simulation studies [DiStefano, 2002], reported standardized factor loadings range from 0.4 to 0.9. For facilitating interpretability and again reproducibility, each factor loading is set to 0.7. Each corresponding residual variance is then automatically set to 0.51 under a standardized solution in the population model, as done in Li [2016].

**Data generation**  Given the specified model, one can generate data in the response space (the \(Z\) in Definition 5.1) via Equations (5.1) and (5.2). When the observed data (the \(Y\) in Definition 5.1) are ordinal, we discretize the corresponding margins into the desired number of categories. When the observed data are
nonparanormal, we set the $F_j(\cdot)$ in Equation (5.3) to the CDF of a $\chi^2$-distribution with degrees of freedom $df$. The reason for choosing a $\chi^2$-distribution is that we can easily use $df$ to control the extent of non-normality: a higher $df$ implies a distribution closer to a Gaussian. To fill in a certain percentage $\beta$ of missing values (we only consider MAR), we follow the procedure in Kolar and Xing [2012], i.e., for $j = 1, \ldots, \lfloor p/2 \rfloor$, $i = 1, \ldots, n$: $y_{i,2^*j}$ is missing if $z_{i,2^*j-1} < \Phi^{-1}(2^*\beta)$.

**Evaluation metrics** We use average relative bias (ARB) and root mean squared error (RMSE) to examine the parameter estimates, which are defined as

$$\text{ARB} = \frac{1}{r} \sum_{i=1}^{r} \frac{\hat{\theta}_i - \theta_i}{\theta_i}, \quad \text{RMSE} = \sqrt{\frac{1}{r} \sum_{i=1}^{r} (\hat{\theta}_i - \theta_i)^2},$$

where $\hat{\theta}_i$ and $\theta_i$ represent the estimated and true values respectively. An ARB value less than 5% is interpreted as a *trivial* bias, between 5% and 10% as a *moderate* bias, and greater than 10% as a *substantial* bias [Curran et al., 1996]. Note that ARB describes an overall picture of average bias, that is, summing up bias in a positive and a negative direction together. A smaller absolute value of ARB indicates better performance on average.

### 5.5.2 Ordinal Data without Missing Values

In this subsection, we consider ordinal complete data since this matches the assumptions of the diagonally weighted least squares (DWLS) method, in which we set the number of ordinal categories to be 4. We also incorporate the robust maximum likelihood (MLR) as an alternative approach, which was shown to be empirically tenable when the number of categories is more than 5 [Rhemtulla et al., 2012; Li, 2016]. See Section 5.2 for details of the two approaches.

Before conducting comparisons, we first check the convergence property of the Gibbs sampler used in our BGCF approach. Figure 5.2 shows the RMSE of estimated interfactor correlations (left panel) and factor loadings (right panel) over 100 iterations for a randomly-drawn sample with sample size $n = 500$. We see quite a good convergence of the Gibbs sampler, in which the burn-in period is only around 10. More experiments done for different numbers of categories and different random samples show that the burn-in is less than 20 on the whole across various conditions.

Now we evaluate the three approaches. Figure 5.3 shows the performance of BGCF, DWLS, and MLR over different sample sizes $n \in \{100, 200, 500, 1000\}$, providing the mean of ARB (left panel) and the mean of RMSE with 95% confidence interval (right panel) over 100 experiments. From Figure 5.3a, interfactor correlations are, on average, trivially biased (within two dashed lines) for all the three methods that in turn give indistinguishable RMSE regardless of sample sizes. From Figure 5.3b, MLR moderately underestimates the factor loadings, and performs worse than DWLS w.r.t. RMSE especially for a larger sample size, which confirms the conclusion in previous studies [Barendse et al., 2015; Li, 2016]. Most
Figure 5.2: Convergence property of our Gibbs sampler over 100 iterations. Left panel: RMSE of interfactor correlations; Right panel: RMSE of factor loadings.

Figure 5.3: Results obtained by the Bayesian Gaussian copula factor (BGCF) approach, the diagonally weighted least squares (DWLS), and the robust maximum likelihood (MLR) on complete ordinal data (4 categories) over different sample sizes, showing the mean of ARB (left panel) and the mean of RMSE with 95% confidence interval (right panel) over 100 experiments for (a) interfactor correlations and (b) factor loadings, where dashed lines and dotted lines in left panels denote ±5% and ±10% bias respectively.

importantly, our BGCF approach outperforms DWLS in learning factor loadings especially for small sample sizes, even if the experimental conditions entirely match the assumptions of DWLS.
### 5.5.3 Mixed Data with Missing Values

In this subsection, we consider mixed nonparanormal and ordinal data with missing values, since some latent concepts in real-world applications are measured by sensors that usually produce continuous but not necessarily Gaussian data. The 8 indicators of the first 2 factors (4 per factor) are transformed into a $\chi^2$-distribution with $\text{df} = 8$, which yields a slightly-nonnormal distribution (skewness is 1, excess kurtosis is 1.5) [Li, 2016]. The 8 indicators of the last 2 factors are discretized into ordinal with 4 categories.

One alternative approach in such cases is DWLS with pairwise-deletion (PD), in which heterogeneous correlations (Pearson correlations between numeric variables, polyserial correlations between numeric and ordinal variables, and polychoric correlations between ordinal variables) are first computed based on pairwise complete observations, and then DWLS is used to estimate model parameters. A second alternative concerns the full information maximum likelihood (FIML) [Arbuckle, 1996; Rosseel, 2012], which first applies an EM algorithm to impute missing values and then uses MLR to learn model parameters.

Figure 5.4 shows the performance of BGCF, DWLS with PD, and FIML for $n = 500$ over different percentages of missing values $\beta \in \{0\% , 10\% , 20\% , 30\% \}$. First, despite a good performance with complete data ($\beta = 0\%$) DWLS (with PD) deteriorates significantly with an increasing percent of missing values especially for factor loadings, while BGCF and FIML show quite good scalability. Second, our BGCF approach overall outperforms FIML: indistinguishable for interfactor correlations but better for factor loadings.

Two more experiments are provided in Appendix 5.B. One concerns incomplete ordinal data with different numbers of categories, showing that BGCF is substantially favorable over DWLS (with PD) and FIML for learning factor loadings, which becomes more prominent with a smaller number of categories. Another one considers incomplete nonparanormal data with different extents of deviation from a Gaussian, which indicates that FIML is rather sensitive to the deviation and only performs well for a slightly-nonnormal distribution while the deviation has no influence on BGCF at all. See Appendix 5.B for more details.

### 5.6 Application to Real-world Data

In this section, we illustrate our approach on the ‘Holzinger & Swineford 1939’ dataset [Holzinger and Swineford, 1939], a classic dataset widely used in the literature and publicly available in the R package lavaan [Rosseel, 2012]. The data consists of mental ability test scores of 301 students, in which we focus on 9 out of the original 26 tests as done in Rosseel [2012]. A latent variable model that is often proposed to explore these 9 variables is a correlated 3-factor model shown in Figure 5.5, where we rename the observed variables to “Y1, Y2, . . . , Y9” for simplicity in visualization and to keep it identical to our definition of observed variables.
Figure 5.4: Results for \( n = 500 \) obtained by BGCF, DWLS with pairwise-deletion, and the full information maximum likelihood (FIML) on mixed nonparanormal (\( df = 8 \)) and ordinal (4 categories) data with different percentages of missing values, for the same experiments as in Figure 5.3.

(Definition 5.1). The interpretation of these variables is given in the following list.

- Y1: Visual perception;
- Y2: Cubes;
- Y3: Lozenges;
- Y4: Paragraph comprehension;
- Y5: Sentence completion;
- Y6: Word meaning;
- Y7: Speeded addition;
- Y8: Speeded counting of dots;
- Y9: Speeded discrimination straight and curved capitals.

The summary of the 9 variables in this dataset is provided in Table 5.1, showing the number of unique values, skewness, and (excess) kurtosis for each variable. From the column of uniques values, we notice that the data are approximately
Figure 5.5: Path diagram for the Holzinger & Swineford data, in which latent concepts are in ovals while observed variables are in squares, bidirected edges between latent concepts denote correlation coefficients (interfactor correlations), directed edges denote factor loadings, and self-referring arrows denote residual variance, respectively. The edge weights in the graph are the model parameters learned by our BGCF approach.

Table 5.1: The number of unique values, skewness, and (excess) kurtosis of each variable in the ‘HolzingerSwineford1939’ dataset.

<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>UNIQUE VALUES</th>
<th>SKEWNESS</th>
<th>KURTOSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y1</td>
<td>35</td>
<td>-0.26</td>
<td>0.33</td>
</tr>
<tr>
<td>Y2</td>
<td>25</td>
<td>0.47</td>
<td>0.35</td>
</tr>
<tr>
<td>Y3</td>
<td>35</td>
<td>0.39</td>
<td>-0.89</td>
</tr>
<tr>
<td>Y4</td>
<td>20</td>
<td>0.27</td>
<td>0.10</td>
</tr>
<tr>
<td>Y5</td>
<td>25</td>
<td>-0.35</td>
<td>-0.54</td>
</tr>
<tr>
<td>Y6</td>
<td>40</td>
<td>0.86</td>
<td>0.84</td>
</tr>
<tr>
<td>Y7</td>
<td>97</td>
<td>0.25</td>
<td>-0.29</td>
</tr>
<tr>
<td>Y8</td>
<td>84</td>
<td>0.53</td>
<td>1.20</td>
</tr>
<tr>
<td>Y9</td>
<td>129</td>
<td>0.20</td>
<td>0.31</td>
</tr>
</tbody>
</table>
We run our Bayesian Gaussian copula factor approach on this dataset. The learned parameter estimates are shown in Figure 5.5, in which interfactor correlations are on the bidirected edges, factor loadings are in the directed edges, and unique variance for each variable is around the self-referring arrows. The parameters learned by the MLR approach are not shown here, since we do not know the ground truth so that it is hard to conduct a comparison between the two approaches.

In order to compare the BGCF approach with MLR quantitatively, we consider answering the question: “What is the value of $Y_j$ when we observe the values of the other variables, denoted by $Y_{\setminus j}$, given the population model structure in Figure 5.5?”

This is a regression problem but with additional constraints to obey the population model structure. The difference from a traditional regression problem is that we should learn the regression coefficients from the model-implied covariance matrix rather than the sample covariance matrix over observed variables.

- For MLR, we first learn the model parameters on the training set, from which we extract the linear regression intercept and coefficients of $Y_j$ on $Y_{\setminus j}$. Then we predict the value of $Y_j$ based on the values of $Y_{\setminus j}$. See Algorithm 5.2 for pseudo code of this procedure.

- For BGCF, we first estimate the correlation matrix $\hat{S}$ over response variables (the $Z$ in Definition 5.1) and the empirical CDF $\hat{F}_j$ of $Y_j$ on the training set. Then we draw latent Gaussian data $Z_j$ given $\hat{S}$ and $Y_{\setminus j}$, i.e., $P(Z_j|\hat{S}, Z_{\setminus j} \in \mathcal{D}(Y_{\setminus j}))$. Lastly, we obtain the value of $Y_j$ from $Z_j$ via $\hat{F}_j$, i.e., $Y_j = \hat{F}_j^{-1}(\Phi[Z_j])$. See Algorithm 5.3 for pseudo code of this procedure. Note that we iterate the prediction stage (lines 7-8) for multiple times in the actual implementation to get multiple solutions to $Y_j^{(new)}$, then the average over these solutions is taken as the final predicted value of $Y_j^{(new)}$. This idea is quite similar to multiple imputation.

Algorithm 5.2 Pseudo code of MLR for regression.

1. **Input:** $Y^{(train)}$ and $Y_{\setminus j}^{(new)}$.
2. **Output:** $Y_j^{(new)}$.
3. **Training Stage:**
   4. Fit the model using MLR on $Y^{(train)}$;
   5. Extract the model-implied covariance matrix from the fitted model, denoted by $\hat{S}$;
   6. Extract regression coefficients $b$ of $Y_j$ on $Y_{\setminus j}$ from $\hat{S}$, that is, $b = \hat{S}_{[\setminus j,\setminus j]}^{-1} \hat{S}_{[\setminus j,j]}$;
   7. Obtain the regression intercept $b_0$, that is, $b_0 = \mathbb{E}(Y_j^{(train)}) - b \cdot \mathbb{E}(Y_{\setminus j}^{(train)})$.

8. **Prediction Stage:**
   9. $Y_j^{(new)} = b_0 + b \cdot Y_{\setminus j}^{(new)}$. 
Figure 5.6: MSE obtained by BGCF and MLR when we take each $Y_j$ as outcome variable (the others as predictors) alternately, showing the mean over 100 experiments (10 times 10-fold cross validation) with error bars representing a standard error.

Algorithm 5.3 Pseudo code of BGCF for regression.

1: Input: $Y_{(train)}$ and $Y_{(new)}$.
2: Output: $Y_{(new)}$.
3: Training Stage:
4: Apply BGCF to learn the correlation matrix over response variables, i.e., $\hat{S} = \hat{\Sigma}_{[Z,Z]}$;
5: Learn the empirical cumulative distribution function of $Y_j$, denoted by $\hat{F}_j$.
6: Prediction Stage:
7: Sample $Z_{j(new)}^{(new)}$ from $P(Z_{j(new)}^{(new)}|\hat{S}, Z_{\setminus j} \in D(Y_{\setminus j}))$;
8: Obtain $Y_{j(new)}^{(new)}$, i.e., $Y_{j(new)}^{(new)} = \hat{F}_{j}^{-1}(\Phi[Z_{j(new)}^{(new)}])$.

The mean squared error (MSE) is used to evaluate the prediction accuracy, where we repeat a 10-fold cross validation for 10 times (thus 100 MSE estimates totally). Also, we take $Y_j$ as the outcome variable alternately while treating the others as predictors (thus 9 tasks totally). Figure 5.6 provides the results of BGCF and MLR for all the 9 tasks, showing the mean of MSE with a standard error represented by error bars over the 100 estimates. We see that BGCF outperforms MLR for Tasks 5 and 6 although they perform indistinguishably for the other tasks. The advantage of BGCF over MLR is encouraging, considering that the experimental conditions match the assumptions of MLR. More experiments are done (not shown) after we make the data moderately or substantially nonnormal, suggesting that BGCF is significantly favorable to MLR, as expected.
5.7 Summary and Discussion

In this chapter, we proposed a novel Bayesian Gaussian copula factor (BGCF) approach for learning parameters of CFA models that can handle mixed continuous and ordinal data with missing values. We analyzed the separate identifiability of interfactor correlations $C$, factor loadings $\Lambda$, and residual variances $D$, since different researchers may care about different parameters. For instance, it is sufficient to identify $C$ for researchers interested in learning causal relations among latent variables [Silva and Scheines, 2006; Silva et al., 2006], with no need to worry about additional conditions to identify $\Lambda$ and $D$. Under sufficient identification conditions, we proved that our approach is consistent for MCAR data and empirically showed that it works quite well for MAR data.

In the experiments, our approach outperforms DWLS even under the assumptions of DWLS. Apparently, the approximations inherent in DWLS, such as the use of the polychoric correlation and its asymptotic covariance, incur a small loss in accuracy compared to an integral approach like the BGCF. When the data follow from a more complicated distribution and contain missing values, the advantage of BGCF over its competitors becomes more prominent. Another highlight of our approach is that the Gibbs sampler converges quite fast, where the burn-in period is rather short. To further reduce the time complexity, a potential optimization of the sampling process is available [Kalaitzis and Silva, 2013].

There are various generalizations to our inference approach. While our focus in this chapter is on the correlated $k$-factor models, it is straightforward to extent the current procedure to other class of latent models that are often considered in CFA, such as bi-factor models and second-order models, by simply adjusting the sparsity structure of the prior graph $G$. Also, one may concern models with impure measurement indicators, e.g., a model with an indicator measuring multiple factors or a model with residual covariances [Bollen, 1989], which can be easily solved with BGCF by changing the sparsity pattern of $\Lambda$ and $D$. Another line of future work is to analyze standard errors and confidence intervals while this chapter concentrates on the accuracy of parameter estimates. Our conjecture is that BGCF is still favorable because it naturally transfers the extra variability incurred by missing values to the posterior Gibbs samples: we indeed observed a growing variance of the posterior distribution with the increase of missing values in our simulations. On top of the posterior distribution, one could conduct further studies, e.g., causal discovery over latent factors [Silva et al., 2006], regression analysis (as we did in Section 5.6), or other machine learning tasks.

5.A Proof of Theorem 5.1

Theorem 5.1 (Consistency of the BGCF Approach). Let $Y_n = (y_1, \ldots, y_n)^T$ be independent observations drawn from a Gaussian copula factor model. If $Y_n$ is
complete (no missing data) or contains missing values that are missing completely at random, then

\[
\lim_{n \to \infty} P(\hat{C}_n = C_0) = 1, \\
\lim_{n \to \infty} P(\hat{\Lambda}_n = \Lambda_0) = 1, \\
\lim_{n \to \infty} P(\hat{D}_n = D_0) = 1,
\]

where \( \hat{C}_n, \hat{\Lambda}_n, \) and \( \hat{D}_n \) are parameters learned by BGCF, while \( C_0, \Lambda_0, \) and \( D_0 \) are the true ones.

**Proof.** If \( S = \Lambda C \Lambda^T + D \) is the response vector’s covariance matrix, then its correlation matrix is

\[
\tilde{S} = V^{-\frac{1}{2}} S V^{-\frac{1}{2}} = V^{-\frac{1}{2}} \Lambda C \Lambda^T V^{-\frac{1}{2}} + V^{-\frac{1}{2}} D V^{-\frac{1}{2}} = \tilde{\Lambda} \tilde{C} \tilde{\Lambda}^T + \tilde{D},
\]

where \( V \) is a diagonal matrix containing the diagonal entries of \( S \). We make use of Theorem 1 from Murray et al. [2013] to show the consistency of \( \tilde{S} \). Our factor-analytic prior puts positive probability density almost everywhere on the set of correlation matrices that have a \( k \)-factor decomposition. Then, by applying Theorem 1 in Murray et al. [2013], we obtain the consistency of the posterior distribution on the response vector’s correlation matrix for complete data, i.e.,

\[
\lim_{n \to \infty} \Pi(\tilde{S} \in \mathcal{V}(\tilde{S}_0) | Z_n \in \mathcal{D}(Y_n)) = 1 \text{ a.s. } \forall \mathcal{V}(\tilde{S}_0), \tag{5.8}
\]

where \( \mathcal{D}(Y_n) \) is the space restricted by observed data, and \( \mathcal{V}(\tilde{S}_0) \) is a neighborhood of the true parameter \( \tilde{S}_0 \). When the data contain missing values that are completely at random (MCAR), we can also directly obtain the consistency of \( \tilde{S} \) by again using Theorem 1 in Murray et al. [2013], with an additional observation that the estimation of ordinary and polychoric/polyserial correlations from pairwise complete data is still consistent under MCAR. That is to say, the consistency shown in Equation (5.8) also holds for data with MCAR missing values.

From this point on, to simplify notation, we will omit adding the tilde to refer to the rescaled matrices \( \tilde{S}, \tilde{\Lambda}, \) and \( \tilde{D} \). Thus, \( S \) from now on refers to the correlation matrix of the response vector. \( \Lambda \) and \( D \) refer to the scaled factor loadings and noise variance respectively.

The Gibbs sampler underlying the BGCF approach has the posterior of \( \Sigma \) (the correlation matrix of the integrated vector \( X \)) as its stationary distribution. \( \Sigma \) contains \( S \), the correlation matrix of the response random vector, in the upper left block and \( C \) in the lower right block. Here \( C \) is the correlation matrix of factors, which implicitly depends on the *Gaussian copula factor model* from Definition 1 of the main paper via the formula \( S = \Lambda C \Lambda^T + D \). In order to render this decomposition identifiable, we need to put constraints on \( C, \Lambda, D \). Otherwise, we can always replace \( \Lambda \) with \( \Lambda U \) and \( C \) with \( U^{-1} CU^{-1} \), where \( U \) is any \( k \times k \) invertible matrix, to obtain the equivalent decomposition \( S = (\Lambda U)(U^{-1} CU^{-T})(U^T \Lambda^T) + D \). However, we have assumed that \( \Lambda \) follows a particular sparsity structure in which there is only a single non-zero entry for each row. This assumption restricts the
space of equivalent solutions, since any $\Lambda U$ has to follow the same sparsity structure as $\Lambda$. More explicitly, $\Lambda U$ maintains the same sparsity pattern if and only if $U$ is a diagonal matrix (Lemma 5.1).

By decomposing $S$, we get a class of solutions for $C$ and $\Lambda$, i.e., $U^{-1}CU^{-1}$ and $\Lambda U$, where $U$ can be any invertible diagonal matrix. In order to get a unique solution for $C$, we impose two identifying conditions: 1) we restrict $C$ to be a correlation matrix; 2) we force the first non-zero entry in each column of $\Lambda$ to be positive. These conditions are sufficient for identifying $C$ uniquely (Lemma 5.2). We point out that these sufficient conditions are not unique. For example, one could replace the two conditions with restricting the first non-zero entry in each column of $\Lambda$ to be one. The reason for our choice of conditions is to keep it consistent with our model definition where $C$ is a correlation matrix. Under the two conditions for identifying $C$, factor loadings $\Lambda$ and residual variances $D$ are also identified except for the case in which there exists one factor that is independent of all the others and this factor only has two indicators. For such a factor, we have 4 free parameters (2 loadings, 2 residuals) while we only have 3 available equations (2 variances, 1 covariance), which yields an underdetermined system. Therefore, the identifiability of $\Lambda$ and $D$ relies on the observation that a factor has a single or at least three indicators if it is independent of all the others. See Lemmas 5.3 and 5.4 for detailed analysis.

Now, given the consistency of $S$ and the unique smooth map from $S$ to $C$, $\Lambda$, and $D$, we obtain the consistency of the posterior mean of the parameter $C$, $\Lambda$, and $D$, which concludes our proof.

Lemma 5.1. If $\Lambda = (\lambda_{ij})$ is a $p \times k$ factor loading matrix with only a single non-zero entry for each row, then $\Lambda U$ will have the same sparsity pattern if and only if $U = (u_{ij})$ is diagonal.

Proof. ($\Rightarrow$) We prove the direct statement by contradiction. We assume that $U$ has an off-diagonal entry that is not equal to zero. We arbitrarily choose that entry to be $u_{rs}, r, s \in \{1, 2, \ldots, k\}, r \neq s$. Due to the particular sparsity pattern we have chosen for $\Lambda$, there exists $q \in \{1, 2, \ldots, p\}$ such that $\lambda_{qr} \neq 0$ and $\lambda_{qs} = 0$, i.e., the unique factor corresponding to the response $Z_q$ is $\eta_r$. However, we have $(\Lambda U)_{qs} = \lambda_{qr} u_{rs} \neq 0$, which means $(\Lambda U)$ has a different sparsity pattern from $\Lambda$. We have reached a contradiction, therefore $U$ is diagonal.

($\Leftarrow$) If $U$ is diagonal, i.e., $U = \text{diag}(u_1, u_2, \ldots, u_k)$, then $(\Lambda U)_{ij} = \lambda_{ij} u_j$. This means that $(\Lambda U)_{ij} = 0 \iff \lambda_{ij} u_j = 0 \iff \lambda_{ij} = 0$, so the sparsity pattern is preserved.

Lemma 5.2 (Identifiability of $C$). Given the factor structure defined in Section 3 of the main paper, we can uniquely recover $C$ from $S = \Lambda C \Lambda^T + D$ if 1) we constrain $C$ to be a correlation matrix; 2) we force the first element in each column of $\Lambda$ to be positive.
Proof. Here we assume that the model has the stated factor structure, i.e., that there is some $\Lambda$, $C$, and $D$ such that $S = \Lambda \Lambda^T + D$. We then show that our chosen restrictions are sufficient for identification using an argument similar to that in Anderson and Rubin [1956].

The decomposition $S = \Lambda \Lambda^T + D$ constitutes a system of $\frac{p(p+1)}{2}$ equations:

$$
\begin{align*}
    s_{ii} &= \lambda_{if(i)}^2 + d_{ii} \\
    s_{ij} &= c_{f(i)f(j)} \lambda_{if(i)} \lambda_{jf(j)}, \ i < j,
\end{align*}
$$

(5.9)

where $S = (s_{ij}), \Lambda = (\lambda_{ij}), C = (c_{ij}), D = (d_{ij})$, and $f : \{1, 2, \ldots, p\} \rightarrow \{1, 2, \ldots, k\}$ is the map from a response variable to its corresponding factor. Looking at the equation system in (5.9), we notice that each factor correlation term $c_{qr}, q \neq r$, appears only in the equations corresponding to response variables indexed by $i$ and $j$ such that $f(i) = q$ and $f(j) = r$ or vice versa. This suggests that we can restrict our analysis to submodels that include only two factors by considering the submatrices of $S, \Lambda, C, D$ that only involve those two factors. To be more precise, the idea is to look only at the equations corresponding to the submatrix $S_{f^{-1}(q)f^{-1}(r)}$, where $f^{-1}$ is the preimage of $\{1, 2, \ldots, k\}$ under $f$. Indeed, we will show that we can identify each individual correlation term corresponding to pairs of factors only by looking at these submatrices. Any information concerning the correlation term provided by the other equations is then redundant.

Let us then consider an arbitrary pair of factors in our model and the corresponding submatrices of $\Lambda, C, D$, and $S$. (The case of a single factor is trivial.) In order to simplify notation, we will also use $\Lambda, C, D$, and $S$ to refer to these submatrices. We also re-index the two factors involved to $\eta_1$ and $\eta_2$ for simplicity. In order to recover the correlation between a pair of factors from $S$, we have to analyze three separate cases to cover all the bases (see Figure 5.7 for examples concerning each case):

1. The two factors are not correlated, i.e., $c_{12} = 0$. (There are no restrictions on the number of response variables that the factors can have.)

2. The two factors are correlated, i.e., $c_{12} \neq 0$, and each has a single response, which implies that $Z_1 = \eta_1$ and $Z_2 = \eta_2$.

3. The two factors are correlated, i.e., $c_{12} \neq 0$, but at least one of them has at least two responses.

Case 1: If the two factors are not correlated (see the example in the left panel of Figure 5.7), this fact will be reflected in the matrix $S$. More specifically, the off-diagonal blocks in $S$, which correspond to the covariance between the responses of one factor and the responses of the other factor, will be set to zero. If we notice this zero pattern in $S$, we can immediately determine that $c_{12} = 0$.

Case 2: If the two factors are correlated and each factor has a single associated response (see the middle panel of Figure 5.7), the model reduces to a Gaussian
Copula model. Then, we directly get \( c_{12} = s_{12} \) since we have put the constraints \( Z = \eta \) if \( \eta \) has a single indicator \( Z \).

Case 3: If at least one of the factors (w.l.o.g., \( \eta_1 \)) is allowed to have more than one response (see the example in the right panel of Figure 5.7), we arbitrarily choose two of these responses. We also require one response variable corresponding to the other factor (\( \eta_2 \)). We use \( \lambda_{i1}, \lambda_{j1}, \) and \( \lambda_{l2} \) to denote the loadings of these response variables, where \( i, j, l \in \{1, 2, \ldots, p\} \). From Equation (5.9) we have:

\[
\begin{align*}
    s_{ij} &= \lambda_{i1} \lambda_{j1} \\
    s_{il} &= c_{12} \lambda_{i1} \lambda_{l2} \\
    s_{jl} &= c_{12} \lambda_{j1} \lambda_{l2} .
\end{align*}
\]

Since we are in the case in which \( c_{12} \neq 0 \), which automatically implies that \( s_{jl} \neq 0 \), we can divide the last two equations to obtain \( \frac{s_{jl}}{s_{il}} = \frac{\lambda_{i1}}{\lambda_{j1}} \). We then multiply the result with the first equation to get \( s_{ij} \frac{s_{jl}}{s_{il}} = \lambda_{i1}^2 \). Without loss of generality, we can say that \( \lambda_{i1} \) is the first entry in the first column of \( \Lambda \), which means that \( \lambda_{i1} > 0 \). This means that we have uniquely recovered \( \lambda_{i1} \) and \( \lambda_{j1} \).

We can also assume without loss of generality that \( \lambda_{l2} \) is the first entry in the second column of \( \Lambda \), so \( \lambda_{l2} > 0 \). If \( \eta_2 \) has at least two responses, we use a similar argument to the one before to uniquely recover \( \lambda_{l2} \). We can then use the above equations to get \( c_{12} \). If \( \eta_2 \) has only one response, then \( d_{ll} = 0 \), which means that \( s_{ll} = \lambda_{l2}^2 \), so again \( \lambda_{l2} \) is uniquely recoverable and we can obtain \( c_{12} \) from the equations above.

Thus, we have shown that we can correctly determine \( c_{qr} \) only from \( S_{f^{-1}(q)f^{-1}(r)} \) in all three cases. By applying this approach to all pairs of factors, we can uniquely recover all pairwise correlations. This means that, given our constraints, we can uniquely identify \( C \) from the decomposition of \( S \).

\[\square\]

Lemma 5.3 (Identifiability of \( \Lambda \)). Given the factor structure defined in Section 5.4 of the main paper, we can uniquely recover \( \Lambda \) from \( S = \Lambda C \Lambda^T + D \) if 1) we constrain \( C \) to be a correlation matrix; 2) we force the first element in each column of \( \Lambda \) to be positive; 3) when a factor is independent of all the others, it has either a single or at least three indicators.
Proof. Compared to identifying $C$, we need to consider another case in which there is only one factor or there exists one factor that is independent of all the others (the former can be treated as a special case of the latter). When such a factor only has a single indicator, e.g., $\eta_1$ in the left panel of Figure 5.7, we directly identify $d_{11} = 0$ because of the constraint $Z_1 = \eta_1$. When the factor has two indicators, e.g., $\eta_2$ in the left panel of Figure 5.7, we have four free parameters ($\lambda_{22}$, $\lambda_{32}$, $d_{22}$, and $d_{33}$) while we can only construct three equations from $S$ ($s_{22}$, $s_{33}$, and $s_{23}$), which cannot give us a unique solution. Now we turn to the three-indicator case, as shown in Figure 5.8. From Equation (5.9) we have:

$$s_{12} = \lambda_{11} \lambda_{21}$$
$$s_{13} = \lambda_{11} \lambda_{31}$$
$$s_{23} = \lambda_{21} \lambda_{31}.$$  

We then have $\frac{s_{12}s_{13}}{s_{23}} = \lambda_{11}^2$, which has a unique solution for $\lambda_{11}$ together with the second constraint $\lambda_{11} > 0$, after which we can naturally get the solutions to $\lambda_{21}$ and $\lambda_{31}$. For the other cases, the proof follows the same line of reasoning as Lemma 5.2.

Lemma 5.4 (Identifiability of $D$). Given the factor structure defined in Section 5.4 of the main paper, we can uniquely recover $D$ from $S = \Lambda C \Lambda^T + D$ if 1) we constrain $C$ to be a correlation matrix; 2) when a factor is independent of all the others, it has either a single or at least three indicators.

Proof. We conduct our analysis case by case. For the case where a factor has a single indicator, we trivially set $d_{ii} = 0$. For the case in Figure 5.8, it is straightforward to get $d_{11} = s_{11} - \lambda_{11}^2$ from $\frac{s_{12}s_{13}}{s_{23}} = \lambda_{11}^2$ (the same for $d_{22}$ and $d_{33}$). Another case we need to consider is Case 3 in Figure 5.7, where we have $\frac{s_{ij}s_{il}}{s_{jl}} = \lambda_{i1}^2$ (see analysis in Lemma 5.2), based on which we obtain $d_{ii} = s_{ii} - \lambda_{i1}^2$. By applying this approach to all single factors or pairs of factors, we can uniquely recover all elements of $D$.

5.B Extended Simulations

This section continues the experiments in Section 5.5 of the main paper, in order to check the influence of the number of categories for ordinal data and the extent
Figure 5.9: Results for \( n = 500 \) and \( \beta = 10\% \) obtained by BGCF, DWLS with PD, and FIML on ordinal data with different numbers of categories, showing the mean of ARB (left panel) and the mean of RMSE with 95\% confidence interval (right panel) over 100 experiments for (a) interfactor correlations and (b) factor loadings, where dashed lines and dotted lines in left panels denote \pm 5\% and \pm 10\% bias respectively.

B1: Ordinal Data with Different Numbers of Categories

In this subsection, we consider ordinal data with various numbers of categories \( c \in \{2, 4, 6, 8\} \), in which the sample size and missing values percentage are set to \( n = 500 \) and \( \beta = 10\% \) respectively. Figure 5.9 shows the results obtained by BGCF (Bayesian Gaussian copula factor), DWLS (diagonally weighted least squares) with PD (pairwise deletion), and FIML (full information maximum likelihood), providing the mean of ARB (average relative bias) and the mean of RMSE (root mean squared error) with 95\% confidence interval over 100 experiments for (a) interfactor correlations and (b) factor loadings. In the case of two categories, FIML underestimates factor loadings dramatically, DWLS obtains a moderate bias, while BGCF just gives trivial bias. With an increasing number of categories, FIML gets closer and closer to BGCF, but still BGCF is favorable.
B2: Nonparanormal Data with Different Non-normality

In this subsection, we consider nonparanormal data, in which we use the degrees of freedom $df$ of a $\chi^2$-distribution to control the extent of non-normality (see Section 5.5.1 of the main paper for details). The sample size and missing values percentage are set to $n = 500$ and $\beta = 10\%$ respectively, while the degrees of freedom varies $df \in \{2, 4, 6, 8\}$.

Figure 5.10 shows the results obtained by BGCF, DWLS with PD, and FIML, providing the mean of ARB (left panel) and the mean of RMSE with 95% confidence interval (right panel) over 100 experiments for (a) interfactor correlations and (b) factor loadings. The major conclusion drawn here is that, while a nonparanormal transformation has no effect on our BGCF approach, FIML is quite sensitive to the extent of non-normality, especially for factor loadings.

Figure 5.10: Results for $n = 500$ and $\beta = 10\%$ obtained by BGCF, DWLS with PD, and FIML on nonparanormal data with different extents of non-normality, for the same experiments as in Figure 5.9.
A common goal in psychometrics, sociology, and econometrics is to uncover causal relations among latent concepts representing hypothetical constructs that cannot be measured directly, such as attitude, intelligence, and motivation. Through measurement models, these constructs are typically linked to measurable indicators, e.g., responses to questionnaire items. This chapter addresses the problem of causal structure learning among such latent concepts and other observed variables. We propose the ‘Copula Factor PC’ algorithm as a novel two-step approach. It first draws samples of the underlying correlation matrix in a Gaussian copula factor model via a Gibbs sampler on rank-based data. These are then translated into an average correlation matrix and an effective sample size, which are taken as input to the standard PC algorithm for causal discovery in the second step. We prove the consistency of our ‘Copula Factor PC’ algorithm, and demonstrate that it outperforms the PC-MIMBuild algorithm and a greedy step-wise approach. We illustrate our method on a real-world data set about children with Attention Deficit Hyperactivity Disorder.

6.1 Introduction

Social scientists, psychologists, and many other scientists are usually interested in learning causal relations between latent concepts that cannot be measured directly, e.g., attitude, intelligence, and motivation (see Kummerfeld and Ramsey [2016]; Silva et al. [2006], and Chapter 10 of Spirtes et al. [2000] for more details). In order
to get a grip on these latent concepts, one commonly-used strategy is to construct a measurement model for such a latent concept, in the sense that domain experts design a set of measurable “items” or survey “questions” that are considered to be indicators of the latent variable. For instance, in the study of Attention Deficit Hyperactivity Disorder (ADHD), 18 questions are designed to measure three latent variables: inattention, hyperactivity, and impulsivity [Ullebø et al., 2012]. In some other cases where it is difficult to design a measurement model due to the absence of domain knowledge or for other reasons, there are some off-the-shelf algorithms, e.g., BPC [Silva et al., 2006] and FOFC [Kummerfeld and Ramsey, 2016], for learning the measurement models from indicator data. In this chapter, we focus on inferring the causal structure among latent variables, assuming that the measurement models are given. We also allow interactions between these latent variables and other (explicit) variables, e.g., subject characteristics like gender and age. Another issue we consider is that there are diverse types of variables in most real-world data: the questionnaire data in a survey is typically ordinal, whereas other variables might be binary, or continuous.

In this chapter, we use a Gaussian copula factor model (see Section 5.3 for the formal definition) to describe such situations, in which a factor can be connected to either one or more observed variables (indicators). Factors with multiple indicators are used to model latent concepts corresponding to psychological traits, such as attitude and intelligence. The copula model provides a good way of analyzing diverse types of variables, where the associations between variables are parameterized separately from their marginal distributions [Hoff, 2007].

We propose the ‘Copula Factor PC’ algorithm for estimating the causal structure among factors of a Gaussian copula factor model, which is based on a two-step approach. The first step draws samples of the underlying correlation matrix, where the Gibbs sampler by Hoff [2007] for Gaussian copula models is extended to Gaussian copula factor models by replacing the Wishart prior with the $G$-Wishart prior and adding a new strategy to sample latent factors. These samples are then translated into an average correlation matrix, and an effective sample size that is used to account for information loss incurred by discrete variables. The second step takes the estimated correlation matrix and effective sample size as input to the standard PC algorithm [Spirtes et al., 2000] for causal discovery.

The rest of this chapter is organized as follows. Section 6.2 describes our ‘Copula Factor PC’ algorithm, while Section 6.3 introduces two alternative approaches: the PC-MIMBuild algorithm [Silva et al., 2006] and a greedy step-wise approach. Section 6.4 compares the ‘Copula Factor PC’ algorithm with the two alternative approaches on simulated data, and Section 6.5 gives an illustration on real-world data of ADHD patients. Section 6.6 concludes this chapter and provides some discussion.


6.2 Copula Factor PC Algorithm

In this section, we derive our Copula Factor PC algorithm, which is a two-step approach.

The first step is based on the BGCF approach, the inference procedure proposed in Chapter 5. By BGCF, we can draw correlation matrix samples over the integrated random vector \( X = (Z^T, \eta^T)^T \) by iterating the steps in Algorithm 5.1. We then extract the submatrix over \( \eta \) to obtain samples of the correlation matrix over latent factors, denoted by \( \{C^{(1)}, \ldots, C^{(m)}\} \).

The second step is to estimate the underlying correlation matrix and an effective sample size from these samples, which are taken as input to the standard PC algorithm for causal discovery. The mean over all these samples is a natural estimate of the underlying correlation matrix, i.e.,

\[
\hat{C} = \frac{1}{m} \sum_{i=1}^{m} C^{(i)}.
\]

In Chapter 5 (Section 5.4.2), we analyzed the identifiability of \( C \), and here we relook this property from a causal prospective. Given a decomposable correlation matrix over the response random vector \( Z \), i.e., \( S = \Lambda CA^T + D \), we get a class of solutions for \( C \), i.e., \( U^{-1}CU^{-1} \) with \( U \) any invertible diagonal matrix, since we can always obtain an equivalent decomposition \( S = (AU)(U^{-1}CU^{-T})(U^TA^T) + D \). However, we find that all members in this class encode the same set of conditional independences (see Lemma 6.1), and therefore imply the same causal structure [Spirtes et al., 2000]. Hence, any solution in this class is appropriate for finding the underlying causal structure among latent factors. In order to get a unique solution for \( C \), we imposed the two sufficient identifying conditions discussed in Section 5.4.2. Note that one could also choose one’s favorite constraints for identifying \( C \), as long as the unique solution belongs to the class \( U^{-1}CU^{-1} \).

**Lemma 6.1.** Consider a random vector \( \eta = (\eta_1, \ldots, \eta_k)^T \) that follows a multivariate normal distribution with population correlation matrix \( C \). Then, for any invertible diagonal matrix \( U = \text{diag}(u_1, u_2, \ldots, u_k) \), the matrix \( \tilde{C} = UCU \) encodes the same set of conditional independencies among \( \eta \) as \( C \).

**Proof.** See Appendix 6.A. \( \square \)

As for the effective sample size \( \hat{n} \), we build upon the idea in Chapter 2, that is, taking the posterior distribution’s degrees of freedom \( \nu \) as an approximation to \( \hat{n} \). Theorem 6.1 suggests a procedure to estimate the degrees of freedom of a G-Wishart distribution.

**Theorem 6.1.** Consider a random matrix \( \Omega \) following a G-Wishart distribution with graph \( G = (V, E) \) as well as parameters \( \nu \) and \( \Psi \), i.e., \( \Omega \sim W_G(\nu, \Psi) \). Let
Σ = Ω⁻¹ and \( \tilde{\Sigma} \) be the normalized matrix of \( \Sigma \), i.e., \( \tilde{\Sigma}_{ij} = \Sigma_{ij} / \sqrt{\Sigma_{ii} \Sigma_{jj}} \). Then, for large \( \nu \), we have

\[
\text{Var} [\tilde{\Sigma}_{ij}] \approx \frac{(1 - (\mathbb{E}[\tilde{\Sigma}_{ij}])^2)^2}{\nu},
\]

(6.1)

for off-diagonal elements \( \tilde{\Sigma}_{ij} \) whenever \((i, j) \in E\).

**Proof.** See Appendix 6.B.

From the theorem, we have that all off-diagonal elements of the latent correlation matrix satisfy Equation (6.1), because the prior subgraph over latent factors is fully connected. Therefore, we estimate \( \hat{n} \) as follows

\[
\hat{n} = \frac{1}{k(k - 1)} \sum_{i \neq j} \nu_{ij}, \quad \text{where} \quad \nu_{ij} = \frac{(1 - (\mathbb{E}[C_{ij}])^2)^2}{\text{Var}[C_{ij}]}.
\]

The ‘Copula Factor PC’ (CFPC) algorithm arises when taking the estimated correlation matrix \( \hat{C} \) and the effective sample size \( \hat{n} \) (to replace the \( n \) in Equation 1.1) as the input to the standard PC algorithm.¹ The CFPC algorithm is consistent, as shown in Theorem 6.2.

**Theorem 6.2 (Consistency of CFPC algorithm).** Let \( Y_n = (y_1, \ldots, y_n)^T \) be independent observations drawn from a Gaussian copula factor model. If 1) the measurement model per factor is known and pure; and 2) the distribution over factors is faithful to a DAG \( \mathcal{G} \), then

\[
\lim_{n \to \infty} P(\hat{\mathcal{M}}_n(\mathcal{G}) = \mathcal{M}(\mathcal{G})) = 1,
\]

where \( \hat{\mathcal{M}}_n(\mathcal{G}) \) is the output of CFPC algorithm and \( \mathcal{M}(\mathcal{G}) \) is the true Markov equivalent class of \( \mathcal{G} \).

**Proof.** The proof follows two separate steps: the BGCF approach to estimate the correct underlying correlation matrix \( C \) (see Theorem 5.1) and the PC algorithm to reach the correct Markov equivalent class with high probability [Spirtes et al., 2000].

---

¹The R code is publicly available in https://github.com/cuiruifei/CopulaFactorModel.

### 6.3 Alternative Approaches

In this section, we introduce two alternative approaches to be compared with our CFPC algorithm: the PC-MIMBuild algorithm and a greedy step-wise approach.

#### 6.3.1 The PC-MIMBuild Algorithm

The original PC-MIMBuild algorithm only works for continuous data. Here, we extend it to mixed cases by learning the correlation matrix of response variables via the
Gibbs sampler by Hoff [2007] and taking it as input to the original PC-MIMBuild. We further generalize the PC-MIMBuild algorithm to handle latent factors with just a single indicator, by replacing the conditional independence testing method designed only for factors with at least two indicators (Theorem 19 in Silva et al. [2006]) with a test based on partial correlation.

Given a pure and correct measurement model involving at least 2 indicators per factor, Spirtes et al. Spirtes et al. [2000] proposed to test independence and conditional independence among the factors, by taking advantage of the following proposition (see also Theorem 19 of Silva et al. [2006]):

**Proposition 6.1** (Conditional Independence Test 1, CIT 1). Let $G$ be a pure linear latent variable model. Let $\eta_1$, $\eta_2$ be two factors in $G$, and $Q$ a set of factors in $G$. Let $Z_1$ be an indicator of $\eta_1$, $Z_2$ be an indicator of $\eta_2$, and $Z_Q$ be a set of indicators of $Q$ containing at least two indicators per factor. Then $\eta_1$ is $d$-separated from $\eta_2$ given $Q$ in $G$ if and only if the rank of the correlation matrix of $\{Z_1, Z_2\} \cup Z_Q$ is less than or equal to $|Q|$ with probability 1 with respect to the Lebesgue measure over the linear coefficients and error variances of $G$.

One way to test if the rank of a covariance matrix in Gaussian models is at most $q$ is to fit a factor analysis model with $q$ latents and assess its significance [Silva et al., 2006]. The PC-MIMBuild algorithm arises when applying ‘CIT 1’ to test conditional independence among latent factors in the PC algorithm.

When a factor only has a single indicator, we propose to test conditional independence by making use of the following proposition:

**Proposition 6.2** (Conditional Independence Test 2, CIT 2). Let $\eta_1$, $\eta_2$ be two factors in $G$, and $Q$ a set of factors in $G$. Let $Z_1$ be one of the indicators of $\eta_1$, $Z_2$ be one of the indicators of $\eta_2$, and $Z_Q$ be all the indicators of $Q$. Then $\eta_1$ is $d$-separated from $\eta_2$ given $Q$ in $G$ if and only if $Z_1$ is independent of $Z_2$ given $Z_Q$ for all $Z_1$ and $Z_2$.

This test can proceed via partial correlations for Gaussian data. By using ‘CIT 1’ or ‘CIT 2’, we generalize the PC-MIMBuild algorithm to the case where a factor has either a single or multiple indicators. Also, we extend the PC-MIMBuild algorithm to mixed continuous and discrete cases by learning the correlation matrix of response variables via the Gibbs sampler by Hoff [2007] and taking it as input to the original PC-MIMBuild. The pseudocode of the extended PC-MIMBuild algorithm is summarized in Algorithm 6.1.

### 6.3.2 A Greedy Step-wise Approach

This approach first extracts the measurement model of a factor with multiple indicators, e.g., the model shown in Figure 6.1, which is the measurement model of $\eta_3$ extracted from Figure 5.1. Then, it uses off-the-shelf techniques [Finney and DiStefano, 2006] to fit such a model and obtain pseudo-data of the factor (factor scores).
Algorithm 6.1 PC-MIMBuild algorithm

1: **Input:** Measurement models and indicator data $Y$.
2: **Output:** Markov equivalent class $\mathcal{M}$ over latent factors.
3: Get correlation matrix of response variables via Gibbs sampler by Hoff [2007] given $Y$;
4: **if** Unconditional independence, i.e., $|Q| = 0$, or all factors in $Q$ have a single indicator. **then**
5: The PC algorithm with CIT 2;
6: **else**
7: The PC algorithm with CIT 1;
8: **end if**
9: Return $\mathcal{M}$.

Algorithm 6.2 Greedy step-wise PC algorithm

1: **Input:** Measurement models (represented by the sparsity pattern of $\Lambda$) and indicator data $Y$.
2: **Output:** Markov equivalent class $\mathcal{M}$ over latent factors.
3: for $i \in \{1, \ldots, k\}$ do
4: Let $Q = \{Y_j : \lambda_{ji} \neq 0\}$, which is the set of indicators of the $i$-th factor;
5: **if** $|Q| = 1$ **then**
6: Take the indicator data as the factor score, i.e., $\eta_i = Q$;
7: **else if** $|Q| = 2$ **then**
8: Take the average of two indicators as the factor score, i.e., $\eta_i = (Q_1 + Q_2)/2$;
9: **else**
10: Fit the measurement model of the $i$-th factor to its indicator data $Q$;
11: Obtain the factor score $\eta_i$ from the fitted model;
12: **end if**
13: **end for**
14: Take pseudo data $\eta = (\eta_1, \ldots, \eta_k)$ as input to Copula PC to get $\mathcal{M}$.

Using pseudo-data for factors with multiple indicators together with real data for factors with a single indicator, the ‘Copula PC’ algorithm is next applied for causal discovery. We refer to this approach as the greedy step-wise PC algorithm, whose pseudo-code is written out step by step in Algorithm 6.2.

Using pseudo-data for factors with multiple indicators together with real data for factors with a single indicator, the ‘Copula PC’ algorithm is next applied for causal discovery. We refer to this approach as the greedy step-wise PC algorithm, whose pseudo-code is written out step by step in Algorithm 6.2.

![Figure 6.1: the measurement model of $\eta_3$ extracted from Figure 5.1.](image-url)
One disadvantage of this approach is that it can overestimate the effective sample size when treating the pseudo-data at the same footing as real data. This might incur many false positives, as we will indeed observe in the experiment section.

6.4 Simulation Study

In this section, we compare our ‘Copula Factor PC’ algorithm (CFPC) with the PC-MIMBuild algorithm (MBPC) and the greedy step-wise PC algorithm (GSPC) on simulated data. Kalisch and Bühlmann [2007] provide a procedure to generate random DAGs and simulate normally distributed samples that are faithful to them. It first generates a $k \times k$ adjacency matrix $A$ representing a random DAG: 1) generate a $k \times k$ zero matrix, 2) randomly set entries in the lower-triangle area to be one with probability $s$ (measuring the sparseness), 3) change the ones to be random weights in the interval $[0,1]$. Given the adjacency matrix $A$, values of a random vector $\eta$ are drawn recursively via

$$\eta_i = \sum_{k<i} A_{ik} \eta_k + \epsilon_i,$$

with each $\epsilon_i \sim \mathcal{N}(0,1)$. Following this procedure, we simulate the factors of a Gaussian copula factor model, i.e., the $\eta$ in Equation (5.1). Then, the edge weights from factors to response variables (non-zero elements of $\Lambda$ in Equation 5.2) are uniformly drawn from the interval $[0,1]$. We next generate response variables using Equation (5.2) together with standard Gaussian noise. After discretizing some response variables, we obtain data following a Gaussian copula factor distribution.

Three metrics are used to evaluate the algorithms: the true and false positive rate (TPR and FPR) for assessing the skeleton, and the structural Hamming distance (SHD), counting the number of edge insertions, deletions, and flips to transfer the estimated CPDAG into the correct CPDAG [Tsamardinos et al., 2006], for assessing the CPDAG. A higher TPR, a lower FPR, and a smaller SHD imply better performance. We set the significance level in the standard PC algorithm to $\alpha = 0.01$ and the sparseness parameter in generating DAGs to $s = 2/(k - 1)$, such that the average neighbors of each node is 2 [Kalisch and Bühlmann, 2007]. For the Gibbs sampler, the first 500 samples (burn-in) are discarded and the next 500 samples are stored. We test the algorithms for different numbers of factors $k \in \{4, 10\}$, and sample sizes $n \in \{500, 1000, 2000\}$.

6.4.1 Evaluation on Gaussian Data

We first consider the case where the observed data are Gaussian and all factors have multiple indicators, since this matches the assumptions of the original PC-MIMBuild algorithm. The number of indicators per factor is randomly chosen from 3 to 10, to mimic typical real-world datasets [Skeem and Cauffman, 2003; Ullebø et al., 2012].
Figure 6.2 shows the results, providing the mean of TPR, FPR, and SHD over 100 repeated experiments with errorbars representing 95% confidence intervals. First, we see that CFPC performs clearly better than MBPC w.r.t. TPR despite an indistinguishable performance w.r.t. FPR (CFPC is slightly better than MBPC for $k = 4$ while the other way around for $k = 10$). Therefore, w.r.t. the overall metric SHD, CFPC significantly outperforms MBPC. Our analysis is that MBPC tests for conditional independencies between all pairs of indicators and claims a dependence between factors even if just one of the pairs fails the test. This multiple testing approach, although elegant in theory, is difficult to make robust for largely varying numbers of indicators and sizes of the conditioning set. Second, while CFPC and GSPC report similar TPR scores, CFPC shows a clear advantage over GSPC w.r.t. FPR (thus a better SHD than GSPC), which becomes more prominent for a larger sample size. This is because the correlations between factors are estimated indirectly through their indicators, which makes the correlations less reliable than those estimated directly through the observed data. The effective sample size used in CFPC naturally incorporates the reduced reliability, whereas GSPC that still uses the original sample size rejects the null hypothesis of conditional independence more easily, resulting in more false positives.
6.4.2 Evaluation on Mixed Data

We now focus on mixed data, in which two cases are considered: 1) all factors have multiple indicators; 2) half of the factors have multiple indicators and half only have a single indicator. When a factor has multiple indicators, the number of indicators per factor is randomly chosen from 3 to 10, and all such indicators are discretized into ordinal variables where the number of levels per variable is randomly chosen from 2 to 5. For factors with a single indicator, we discretize half into ordinal variables (from 2 to 5 levels) and keep the other half continuous.

Figures 6.3 and 6.4 summarize the experimental results, providing the mean of TPR, FPR, and SHD over 100 repeated experiments with 95% confidence intervals. From Figure 6.3a, we first see that GSPC is slightly better than CFPC w.r.t. TPR while GSPC and CFPC show a clear advantage over MBPC. Second, MBPC is rather sensitive to sample sizes in cases with only multiple indicators, where a small sample size incurs a poor performance. Figure 6.3b shows that CFPC is significantly better than GSPC w.r.t. FPR, which becomes more prominent in cases with only multiple indicators and larger sample sizes. This is because the effective sample size in CFPC better than GSPC represents the uncertainty in the partial correlation estimates and then incurs less false positives. CFPC also shows clear advantages over MBPC w.r.t. FPR when the number of factors is 4 \((k = 4)\), whereas MBPC works slightly better than CFPC when \(k = 10\). As for the overall metric SHD shown in Figure 6.4, CFPC and GSPC perform clearly better than MBPC in almost all situations because of the bad performance of MBPC w.r.t. TPR. Meanwhile, we can see that CFPC generates a more accurate CPDAG than GSPC, in particular for larger sample sizes. This is because our proposed inference procedure more accurately estimates the correlation matrix (not shown here) and, through the effective sample size, better represents the uncertainty in the correlation estimates than the greedy step-wise method. In a nutshell, our ‘Copula Factor PC’ algorithm, outperforms its two competitors in almost all situations.

6.5 Application to Real-world Data

In this section, we give an illustration on a real-world dataset collected by van Steijn et al. [2012] that includes 236 children with Attention Deficit Hyperactivity Disorder (ADHD) and 406 controls. We focus on 4 (explicit) variables that are related to ADHD symptoms: gender (Gen), Age, verbal IQ (VIQ), performance IQ (PIQ), as well as 18 questions that are designed to measure three latent concepts: inattention (Inatt), hyperactivity (Hyper), and impulsivity (Impul). The first 9 questions (Q1-Q9) are designed to measure ‘Inatt’, while the next 5 questions (Q10-Q14) and the last 4 questions (Q15-Q18) are used to measure ‘Hyper’ and ‘Impul’ respectively [Ullebø et al., 2012]. All the questions are ordinal with four levels: never (0), sometimes (1), frequently (2), and always (3).

Our task is to infer the causal structure among the 4 variables and 3 latent
Figure 6.3: (a) TPR and (b) FPR of CFPC, GSPC, and MBPC for the case where all factors have multiple indicators (left column) and the case where half of the factors have multiple indicators while the other half have a single indicator (right column), showing the mean over 100 experiments with 95% confidence intervals. The two rows represent the results when the number of factors is 4 and 10 respectively.
Figure 6.4: SHD of CFPC, GSPC, and MBPC, showing the mean over 100 experiments together with 95% confidence intervals, for the same experiments as in Figure 6.3.

concepts from observations of the 4 variables and 18 questions. We run our ‘Copula Factor PC’ algorithm (using the order-independent version of the PC algorithm [Colombo and Maathuis, 2014]) on this dataset and enforce the prior knowledge that no variables cause gender. The resulting graph is shown in Figure 6.5, in which double arrows ‘⇒’ represent the mapping from the three latent concepts to their corresponding questions (known) and other edges are those learned by our algorithm.

First, in the inferred model, we find that ‘Gen’ has a direct causal influence on ‘Inatt’. The finding is in the expected direction, namely males are at an increased risk of inattention, hyperactivity, and impulsivity problems. Meta-analyses in population-based samples suggested that males are 24 times more likely to meet full criteria for ADHD than females [Willcutt, 2012] and in clinically referred ADHD samples, the gender ratio was about 5:1 [Novik et al., 2006].

Second, the causal model implies that there is a significant causal path from inattention to hyperactivity (and subsequently to impulsivity), but not the other way around. It suggests that factors that cause inattention affect hyperactivity/impulsivity downstream of that, whereas those factors that lead to high hyperactivity/impulsivity do not necessarily lead to higher inattention. This causal path was previously observed in this sample and was also confirmed in two independent ADHD samples [Sokolova et al., 2016].

Third, the causal direction of the associations between verbal IQ and inattention as well as impulsivity is not clear from our model. Both interpretations seem
reasonable. Previous studies suggest that ADHD is associated with lower (verbal) IQ, and particularly attention problems have been found to be strong predictors for lower IQ and poorer academic performance [Heutink et al., 2006].

To conclude, using the Copula Factor PC algorithm in an ADHD sample allows us to infer causal relations between the different ADHD traits and generic factors (age, gender, and IQ). This enhances knowledge of the causal structure of ADHD (e.g., by answering the question whether inattention is causing hyperactivity, or vice versa), which may have significant clinical implications, as it may inform therapeutic interventions.

6.6 Conclusion and Discussion

In this chapter, we focused on learning causal relations between latent concepts with pre-designed or pre-fitted measurement models. Our typical use case is that of psychological constructs that are linked to responses on questionnaire items. To the best of our knowledge, we are the first to propose a provably convergent algorithm that is able to recover the underlying causal structure between such factors and other observed variables, which can be both discrete and continuous.

In the experiments, our ‘Copula Factor PC’ algorithm clearly outperformed both the PC-MIMBuild algorithm and the greedy step-wise approach. PC-MIMBuild tests for conditional independencies between all pairs of indicators and concludes that the latent factors are dependent even if just one of the pairs fails the indepen-
dence test. In our experience, this multiple testing approach, although elegant in theory, is difficult to make robust for largely varying numbers of indicators and sizes of the conditioning set. The ‘Copula Factor PC’ algorithm more naturally appears to find the right balance between true positives and false positives under varying conditions. It improves upon the greedy step-wise approach by estimating the full correlation matrix instead of individual sub-parts, which increases the power of the conditional independence tests.

Our approach extends earlier work with various novel and essential ingredients needed to handle latent factors. We replaced the Wishart prior with a $G$-Wishart distribution over factors and indicator variables, whose structure directly follows from the measurement model. The corresponding marginal prior on the factors is then still a Wishart distribution, which can be chosen such that the pairwise correlations are uniformly distributed. As in Harris and Drton [2013], we can prove that our procedure is consistent. We also show that, although the correlation matrix over factors itself is non-identifiable, all characteristics that relate to the identification of the correct causal structure can be consistently recovered.

In this chapter, we focused on so-called pure measurement models [Silva et al., 2006; Kummerfeld and Ramsey, 2016], which is the major simplifying assumption of our procedure. We would argue that this is often satisfied, since it is the way in which questionnaires are typically designed by domain experts and that allows for a specific interpretation of the factors (e.g., a predefined set of items relates to the concept “hyperactivity”, another non-overlapping set of items to the concept “inattention”). If the measurement models are not given, they can be learned using off-the-shelf algorithms, such as BPC [Silva et al., 2006] and FOFC [Kummerfeld and Ramsey, 2016], which output pure measurement models.

Acknowledgements

We gratefully thank Anoek Sluiter-Oerlemans for the profound analysis about the experimental results on the ADHD dataset, and Ioan Gabriel Bucur for valuable discussions on the proof of Theorem 6.2. This research has been partially financed by the Netherlands Organisation for Scientific Research (NWO) under project 617.001.451.

6.A Proof of Lemma 6.1

**Lemma 6.1.** Consider a random vector $\eta = (\eta_1, \ldots, \eta_k)^T$ that follows a multivariate normal distribution with population correlation matrix $C$. Then, for any invertible diagonal matrix $U = \text{diag}(u_1, u_2, \ldots, u_k)$, the matrix $\tilde{C} = UCU$ encodes the same set of conditional independencies among $\eta$ as $C$.

**Proof.** Let $i, j \in \{1, \ldots, k\}$, and $Q \subseteq \{1, \ldots, k\}/\{i, j\}$. In the Gaussian case, $\eta_i$ is
independent of \( \eta_j \) given \( \eta_Q \) if and only if the partial correlation between \( \eta_i \) and \( \eta_j \) given \( \eta_Q \), denoted by \( \rho_{ij|Q}^C \), vanishes, i.e.,

\[
\eta_i \perp \eta_j | \eta_Q \iff \rho_{ij|Q}^C = 0. \tag{6.2}
\]

The partial correlation \( \rho_{ij|Q}^C \) is uniquely determined by the correlation matrix \( C \), that is,

\[
\rho_{ij|Q}^C = -\frac{A_{ij}}{\sqrt{A_{ii}A_{jj}}}, \tag{6.3}
\]

where \( A = (C_{i,j,Q})^{-1} \) is the inverse of the principal submatrix of \( C \) over \( \{i, j, Q\} \).

Similarly, for the matrix \( \tilde{C} \), we have

\[
\rho_{ij|Q}^\tilde{C} = -\frac{B_{ij}}{\sqrt{B_{ii}B_{jj}}}, \tag{6.4}
\]

where

\[
B = (\tilde{C}_{i,j,Q})^{-1} = ((UCU)_{i,j,Q})^{-1} = (U_{i,j,Q}C_{i,j,Q}U_{i,j,Q})^{-1} \quad \text{(since } U \text{ is diagonal)} = (U_{i,j,Q})^{-1}(C_{i,j,Q})^{-1}(U_{i,j,Q})^{-1} = (U_{i,j,Q})^{-1}A(U_{i,j,Q})^{-1}.
\]

Since all the diagonal elements of \( U \) are non-zero, we have

\[
B_{ij} = u^{-1}_i A_{ij} u^{-1}_j = 0 \iff A_{ij} = 0. \tag{6.5}
\]

From Equation (6.3), (6.4), and (6.5), we have

\[
\rho_{ij|Q}^\tilde{C} = 0 \iff \rho_{ij|Q}^C = 0. \tag{6.6}
\]

Therefore, according to Equation (6.2) and (6.6), \( \forall i, j, \) and \( Q \), we have

\[
\eta_i \perp \eta_j | \eta_Q \iff \rho_{ij|Q}^C = 0 \iff \rho_{ij|Q}^\tilde{C} = 0,
\]

which concludes our proof. \( \square \)

### 6.B Proof of Theorem 6.1

**Theorem 6.1.** Consider a random matrix \( \Omega \) following a \( G \)-Wishart distribution with graph \( G = (V, E) \) as well as parameters \( \nu \) and \( \Psi \), i.e., \( \Omega \sim \mathcal{W}_G(\nu, \Psi) \). Let \( \Sigma = \Omega^{-1} \) and \( \tilde{\Sigma} \) be the normalized matrix of \( \Sigma \), i.e., \( \tilde{\Sigma}_{ij} = \Sigma_{ij}/\sqrt{\Sigma_{ii}\Sigma_{jj}} \). Then, for large \( \nu \), we have

\[
\text{Var}[\tilde{\Sigma}_{ij}] \approx \frac{(1 - (E[\tilde{\Sigma}_{ij}])^2)}{\nu},
\]

for off-diagonal elements \( \tilde{\Sigma}_{ij} \) whenever \( (i, j) \in E \).
Proof. If $\Omega$ follows a $G$-Wishart distribution, i.e., $\Omega \sim W_G(\nu, \Psi)$, and $\Sigma = \Omega^{-1}$, then we say that $\Sigma$ follows a hyper inverse-Wishart distribution [Roverato, 2000], denoted by $\Sigma \sim \mathcal{HIW}_G(\nu, \Psi)$.

Lemma 6.2 (see Roverato [2002]). For a graph $G = (V, E)$, assume $\Sigma \sim \mathcal{HIW}_G(\nu, \Psi)$. Then, for any $B \subseteq V$, we have

$$\Sigma_{BB} \sim \mathcal{HIW}_{G_B}(\nu, \Psi_{BB}),$$

where $G_B$ is the subgraph only involving variables in $B$.

Lemma 6.3 (see Theorem 2.1). If $\Sigma$ follows an inverse-Wishart distribution, i.e., $\Sigma \sim W^{-1}(\nu, \Psi)$, and $\tilde{\Sigma} = (\tilde{\Sigma}_{ij})$ with $\tilde{\Sigma}_{ij} = \Sigma_{ij} / \sqrt{\Sigma_{ii}\Sigma_{jj}}$, then for each off-diagonal element $\tilde{\Sigma}_{ij}$ and large $\nu$, we have

$$\text{Var}[\tilde{\Sigma}_{ij}] \approx \frac{(1 - (\mathbb{E}[\tilde{\Sigma}_{ij}])^2)^2}{\nu}.$$  

Suppose $\Sigma \sim \mathcal{HIW}_G(\nu, \Psi)$ with a graph $G = (V, E)$. According to Lemma 6.2, for any subset $B = \{i, j\} \subseteq V$, we have

$$\begin{pmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{pmatrix} \sim \mathcal{HIW}_{G_B}(\nu, \begin{pmatrix} \Psi_{ii} & \Psi_{ij} \\ \Psi_{ji} & \Psi_{jj} \end{pmatrix}).$$  

(6.7)

If there exists an edge between node $i$ and node $j$ in graph $G$, i.e., $(i, j) \in E$, the subgraph $G_B$ is a fully connected graph. Then, the hyper inverse-Wishart distribution in (6.7) reduces to an inverse-Wishart distribution, i.e.,

$$\begin{pmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{pmatrix} \sim W^{-1}(\nu, \begin{pmatrix} \Psi_{ii} & \Psi_{ij} \\ \Psi_{ji} & \Psi_{jj} \end{pmatrix}),$$

when $(i, j) \in E$.

Let $\tilde{\Sigma}_{ij} = \Sigma_{ij} / \sqrt{\Sigma_{ii}\Sigma_{jj}}$, then according to Lemma 6.3, for large $\nu$, we have

$$\text{Var}[\tilde{\Sigma}_{ij}] \approx \frac{(1 - (\mathbb{E}[\tilde{\Sigma}_{ij}])^2)^2}{\nu},$$

whenever $(i, j) \in E$ in graph $G$.  

\qed
In this thesis, we generalized the causal discovery algorithms to more realistic cases: mixed continuous and discrete data (possibly with missing values), and latent concepts that cannot be measured directly. To solve the issue of mixed data, we proposed to use the Gaussian copula model, in which observed variables can be binary, ordinal, and continuous, and the association between these variables is modeled by a latent multivariate Gaussian distribution that is independent of the marginal distribution of observed variables. For the problem of latent concepts, our main tool is the Gaussian factor model, in which factors relate to latent concepts while response variables relate to indicators of the latents. The Gaussian copula factor model, an elegant combination of the Gaussian copula model and the Gaussian factor model, is naturally used when both mixed data and latent concepts arise.

Throughout this thesis, our proposed methods generally followed a two-step framework. The first step is to perform statistical inference for the parameters of interest via either Markov chain Monte Carlo (MCMC) methods (Chapters 2, 4, and 5) or maximum likelihood estimation approaches (Chapter 3), based on which we conduct causal discovery in the second step. We would like to think of this framework as a general principle, where one could plug in one’s favorite choice for each of the two steps. For the first step, one may use a different MCMC method [Kalaitzis and Silva, 2013], a MAP approach along the lines of Abegaz and Wit [2014], or variational inference and pseudo maximum likelihood to reduce time complexity. On top of the inference result, one could carry out a variety of tasks other than causal discovery in the second step, such as undirected graphical modeling [Dobra et al., 2011; Fan et al., 2017], factor analysis [Murray et al., 2013; Gruhl et al., 2013], clustering or various prediction tasks.

While our focus was on the PC algorithm in this thesis, other causal discovery algorithms can be naturally integrated into the second step of our framework, e.g.,
FCI [Spirtes et al., 2000] for handling potential confounders and selection bias, score-based approaches like GES [Chickering, 2002b], methods for cyclic models [Mooij and Heskes, 2013; Mooij et al., 2011], or other algorithms [Claassen et al., 2013; Triantafillou and Tsamardinos, 2015; Magliacane et al., 2016]. To gain an insight into the reliability of structure estimates, we proposed the Stable Copula PC algorithm which runs the PC algorithm several times based on different Gibbs samples and ensembles these results. This idea could be used to our other Gibbs-sampling-based methods like the Copula Factor PC algorithm. One could also try a different ensembling strategy, such as bootstrapping the original dataset and then performing stability selection [Dai et al., 2004; Meinshausen and Bühlmann, 2010; Stekhoven et al., 2012]. In our implementation of the PC algorithm, the conditional independence test is performed based on the Fisher’s z-transformation of the partial correlation coefficient with significance level 0.01 or 0.05. One may choose a different testing approach, such as the standard Student’s t-test for the untransformed partial correlation coefficient [Scutari, 2010], and the \(\chi^2\)-test and a test based on the shrinkage James-Stein estimator for the mutual information [Hausser and Strimmer, 2009]. Instead of specifying a particular testing approach and choosing the usual significance level 0.01 or 0.05, a Bayesian optimization procedure is available to choose these parameters in an automatic way [Córdoba et al., 2018].

Although our attention was on learning causal relations from pure observational data, we strongly encourage the usage of prior knowledge and possible interventional experiments. Prior information like the absence of edges could significantly reduce the search space of conditional independences, which facilitates the computational efficiency. Knowledge regarding causal directions is particularly useful in improving the identifiability of the underlying DAG, in the sense that only a couple of prior directions in a CPDAG may incur many additional causal links according to the orientation rules [Meek, 1995; Perkovic et al., 2017]. Possible interventions in a system (usually a local part of the system) are good ways to obtain such prior knowledge. See Hauser and Bühlmann [2012]; Ness et al. [2016]; Wang et al. [2017b] for details on the importance of and approaches to incorporating interventions into causal discovery.

For handling latent concepts, we proposed the Gaussian copula factor model. In particular, we restricted the measurement model for each factor to be pure, in the sense that an indicator can only be used to measure a single factor or in other words indicators of different factors do not overlap each other [Silva et al., 2006]. This inductive bias is fully motivated by the typical construction of questionnaires in measuring latent traits, in which domain experts usually design a set of questions that are considered to be indicators of a latent concept [Martínez-Torres, 2006; Byrne, 2013]. For example, a predefined set of items relates to the concept “hyperactivity”, another non-overlapping set of items to the concept “inattention”. In case one concerns models with impure measurement indicators, e.g., a model with an indicator measuring multiple factors or a model with residual covariances [Bollen, 1989], the issue can be straightforwardly solved with our ‘Bayesian Gaussian Copula
Factor inference approach by changing the sparsity pattern of the factor loading matrix $\Lambda$ and the residual variance-covariance matrix $D$. Another way to cope with impure models is to plug in a purification step, which can be done with off-the-shelf algorithms, such as BPC [Silva et al., 2006] and FOFC [Kummerfeld and Ramsey, 2016].

We further focused on mixed ordinal (ordered categorical variables) and continuous variables, in which a basic assumption is that an ordinal variable is generated by placing some thresholds on a latent Gaussian variable. This is because while the general copula framework is justifiable for ordinal and continuous variables, it cannot meaningfully handle numeric values for unordered categorical variables (a.k.a. nominal variables), such as color and types of job. To include such nominal variables into our copula model, we may consider a multinomial probit model for each nominal margin. The main idea is to relate a nominal variable to a vector of dummy variables that can be thought of as the unnormalized probabilities of choosing each of the categories, as done in Wang et al. [2017a]. We leave this task as a future work.
Bibliography


M. G. Kendall (1948). “Rank correlation methods.”


E. Kummerfeld and J. Ramsey (2016). “Causal clustering for 1-factor measurement models”. In “International Conference on Knowledge Discovery and Data Mining”, pages 1655–1664. ACM.


Summary

This thesis has focused on the problem of causal discovery from observational data. The main goal is to generalize the standard causal structure learning algorithms to more realistic situations. We have concentrated on three settings: mixed continuous and discrete data, datasets with missing values, and systems with latent concepts (or their possible combinations). Throughout this thesis we took the PC algorithm as an example to illustrate our framework, but other standard causal discovery algorithms could also be integrated to our procedure straightforwardly, as discussed in Chapter 7.

In Chapter 2, we considered the situation of mixed continuous and discrete data, for which we proposed to use the Gaussian copula model. This model provides a good way to conduct multivariate data analysis because associations among variables can be characterized separately from the univariate marginal distributions of observed data that can be diverse types. We applied a Bayesian inference procedure for the association parameters of this model, based on a projected inverse-Wishart distribution (as the prior) and the extended rank likelihood. On top of the inferred posterior distribution, we estimate the underlying correlation matrix and an effective sample size, which are then input to the standard PC algorithm for causal discovery, resulting in the Copula PC algorithm. We also derived a stable version, referred to as Stable Copula PC, which runs PC repeatedly on a number of correlation matrix samples and ensembles the outputs to get a more robust result.

In Chapter 3, we still applied the Gaussian copula model to handle the mixture of continuous and discrete data, but proposed a different inference procedure from Chapter 2. We proposed a novel heterogeneous estimator for pairwise correlations, which tests the rank correlation between two continuous variables, the polyserial correlation between a continuous variable and an ordinal variable, and the polychoric correlation between two ordinal variables. The Hetcor PC algorithm arises when taking the learned heterogeneous correlations as input to the standard PC algorithm. The advantage of Hetcor PC over Copula PC is that we proved the convergence rate of the heterogeneous correlation estimator, based on which we subsequently showed the probability error bound and high-dimensional consistency of the Hetcor PC algorithm. We also introduced a proposal for learning an effective sample size to account for information loss incurred by discrete variables, which improves the empirical performance of the Hetcor PC algorithm.
In Chapter 4, we focused on incomplete data, i.e., data plagued by missing values. Our first solution applies rank correlations to pairwise complete observations to learn the underlying correlation matrix, and takes the number of pairwise complete observations as the effective sample size for the corresponding correlation. Note that the effective sample size here is used to account for information loss from missing values while it is for the information loss incurred by discrete variables in previous chapters. This solution works when the data are drawn from a nonparanormal model and the values are missing completely at random (MCAR). Under these assumptions, we provided an error bound on the accuracy of the resulting Rank PC algorithm and showed its high-dimensional consistency. Given that MCAR is a strong assumption, we proposed our second solution, which works for mixed continuous and discrete data under missingness at random (MAR), a less restrictive assumption than MCAR. This solution first applies a Gibbs sampling procedure to draw correlation matrix samples given incomplete mixed data. These samples are then translated into an average correlation matrix and an effective sample size, based on which the PC algorithm is used for causal discovery.

In Chapters 5 and 6, we addressed the problem of causal structure learning among latent concepts and other well-defined observed variables. To handle this situation, we proposed to use the Gaussian copula factor model, where factors with multiple indicators are used to model latent concepts while factors with single indicators are used to model well-defined explicit variables. Chapter 5 presented a novel Bayesian inference approach for the Gaussian copula factor model, in which a Gibbs sampler is used to draw pseudo Gaussian data in a latent space restricted by the observed data (unrestricted if that value is missing) and draw posterior samples of parameters given the pseudo data, iteratively. We analyzed the identifiability of model parameters and proved the consistency of our approach. Building upon the posterior samples of the correlation matrix over factors obtained in Chapter 5, Chapter 6 continued to learn the causal structure over factors following our general framework, that is, first to estimate the underlying correlation matrix and the effective sample size and then to apply the PC algorithm to conduct causal discovery.

In Chapter 7, we concluded this thesis, discussed possible generalizations of our approaches, and provided some potential future works.
Samenvatting

Dit proefschrift onderzoekt methoden voor het ontdekken van causale verbanden uit geobserveerde data. Het belangrijkste doel van dit proefschrift is het generaliseren van bestaande algoritmen, zodat deze algoritmen beter kunnen omgaan met veelvoorkomende situaties. In dit proefschrift richten we ons op drie specifieke situaties: gemengde continue en discrete data, datasets met ontbrekende waarden, en systemen met latente concepten (of mogelijke combinaties daartussen). In dit proefschrift gebruiken we het PC-algoritme als voorbeeld om ons raamwerk mee te illustreren, maar andere standaard algoritmen voor het ontdekken van causale verbanden kunnen ook eenvoudig in onze methode worden geïntegreerd, zoals besproken in hoofdstuk 7.

In hoofdstuk 2, behandelen we de situatie van gemengde continue en discrete data, en introduceren hiervoor het Gaussische Copula-model. Dit model biedt een goede manier om multivariate data-analyse uit te voeren, omdat verbanden tussen variabelen afzonderlijk van de univariate marginale verdelingen kan worden gekarakteriseerd. We gebruiken een Bayesiaanse inferentieprocedure voor de associatieparameters van dit model, gebaseerd op een geprojecteerde inverse-Wishart-verdeling (a-priori-kansverdeling) en de uitgebreide ranglijst waarschijnlijkheid (extended rank likelihood). Verder leiden wij de a-posteriori-verdeling af, schatten we de onderliggende correlatiematrix, en een effectieve steekproefomvang, die vervolgens worden gebruikt als invoer voor het standaard PC-algoritme voor causale ontdekking, resulterend in het Copula PC-algoritme. Daarnaast hebben we ook een stabiele versie afgeleid, ook wel ‘Stable Copula PC’ genoemd, die het PC-algoritme herhaaldelijk toepast op een aantal correlatiematrix samples en deze combineert tot een robuuster resultaat.

In hoofdstuk 3 passen we het Gaussische Copula model nogmaals toe op gemengde continue en discrete data, maar gebruiken een andere inferentie methode dan hoofdstuk 2. We introduceren een nieuwe heterogene schatter voor paarsgewijze correlaties, die de rangcorrelatie tussen twee continue variabelen test, de polyseriale correlatie tussen een continue variabele en een ordinale variabele, en de polychorische correlatie tussen twee ordinale variabelen. Het Hetcor PC algoritme ontstaat als de geleerde heterogene correlaties worden gebruikt als input voor het standaard PC-algoritme. Het voordeel van Hetcor PC ten opzichte van Copula PC is dat we de convergentiesnelheid van de heterogene correlatiewaarschijnlijkheid kunnen aan-
tonen. Daardoor kunnen we de fout in de berekende waarschijnlijkheid begrenzen en de hoogdimensionale consistentie van het Hetcor PC-algoritme aantonen. Verder introduceren we nog een effectieve steekproefomvang die rekening houdt met informatieverlies door discrete variabelen, wat de empirische prestaties van het Hetcor PC-algoritme nog verbeterd.

In hoofdstuk 4 richten we ons op data met ontbrekende waarden. Onze eerste oplossing past rangcorrelaties toe op paarsgewijze complete observaties om de onderliggende correlatiematrix te leren, en neemt het aantal paarsgewijze volledige observaties als de effectieve steekproefomvang voor de corresponderende correlatie. Merk op dat de effectieve steekproefomvang hier wordt gebruikt voor de informatieverlies veroorzaakt door ontbrekende waarden terwijl het in het vorige hoofdstukken werd veroorzaakt door discrete variabelen. Deze oplossing werkt als de gegevens afkomstig zijn van een nietparanormaal model en de waarden die ontbreken volledig willekeurig ontbreken (MCAR). Onder deze aannames geven we ook een begrenzing van de fout in nauwkeurigheid van het resulterende Rank PC-algoritme en tonen zijn hoogdimensionale consistentie. Omdat MCAR een sterke aanname is, stellen wij ook een tweede oplossing voor, die werkt voor gemengde continue en discrete data onder de MAR assumptie, een minder beperkende aanname dan MCAR. Deze oplossing past eerst een Gibbs-samplingsprocedure toe om correlatiematrix samples te geven voor onvolledige gemengde data. Deze samples worden dan vertaald in een gemiddelde correlatiematrix en een effectieve steekproefomvang, die dan worden gebruikt als input voor het PC-algoritme voor het ontdekken van causale verbanden.

In de hoofdstukken 5 en 6 richten we ons op het leren van causale结构uren tussen latente concepten en andere goed gedefinieerde waargenomen variabelen. Hiervoor introduceren wij het Gaussische Copula factor-model, waar factoren met meerdere indicatoren worden gebruikt om latente concepten te modelleren, en factoren met één indicator worden gebruikt om goed gedefinieerde expliciete variabelen te modelleren. Hoofdstuk 5 introduceert een nieuwe Bayesiaanse inferentieprocedure voor het Gaussische Copula factor-model, waarin een Gibbs-sampler wordt gebruikt om iteratief pseudo Gaussische-data te samplen in een latente ruimte beperkt door de waargenomen gegevens (onbeperkt als die waarde ontbreekt) en posterior samples van de parameters gegeven de pseudo data. We analyseren de identificeerbaarheid van modelparameetrs en bewijzen de consistentie van onze aanpak. Voortbouwend op de posterior samples van de correlatiematrix over factoren verkregen in hoofdstuk 5, gaat hoofdstuk 6 verder met het leren van een causale structuur over factoren die ons algemene raamwerk volgt, i.e., eerst het schatten van de onderliggende correlatiematrix en de effectieve steekproefomvang om die vervolgens als invoer te gebruiken voor het PC-algoritme voor het ontdekken van causale verbanden.

In hoofdstuk 7, ronden we dit proefschrift af met onze conclusies en bespreken we mogelijke generalisaties en uitbreidingen van ons raamwerk.


I would like to thank everyone who helped me finish this thesis, worked with me in my Ph.D. research, and played a role in my daily life over the past three years.

First of all, I want to express special thanks to my supervisor, Prof. Tom Heskes. In my view, Tom’s decision of admitting me as his student is a turning point in my career, after which I began to experience a significantly different yet colorful life. It is Tom who gave me the opportunity to reach the world-level research community, to broaden my horizon, and to get to know so many nice international friends. I really enjoy his working style, especially his way of communicating with students which often included surprises and jokes, making me feel at ease to discuss anything with him. His responsible supervision played a critical role in my Ph.D. study. Whenever I got stuck, he always came up with a solution. Even if the difficulties were beyond his expertise, he could also give me ideas that helped me towards the true direction. I would like to show my gratitude to my co-supervisor, Perry Groot, as well. With his help, I could make progress more quickly. For example, he taught me to try my best to pay attention to every detail, which is very important in scientific research.

Great thanks to the manuscript committee members of this thesis: Prof. dr. Bert Kappen, Prof. dr. Marloes Maathuis, and Dr. Joris Mooij for their efforts on reviewing and improving my work. I show my sincere appreciation to Prof. Jun Zhu for giving me the opportunity to conduct a visiting research at Tsinghua University, which enriches my knowledge in many aspects.

I want to thank all my colleagues. All of them are nice people, great friends. Gabriel, my officemate, helped me a lot in both research and life, in particular, during the early stage of my stay in the Netherlands. Jacopo, such a nice guy, has a lot of common interests with me, especially football that brought us so much fun. I also thank him for inviting me to visit his hometown, a great experience. Nicole and Kasper always showed much patience whenever I asked help from them. Tom C, a smart guy, often inspired me with his insightful ideas in our discussion. Elena M, Arjen, and Martha, you are not only professors, but also kind friends. I feel pretty happy to get to know you. Simone L, Luc, Henning, Zaheer, Konrad, Ridho, thank you for all the joys, meals, and parties we had together. Thanks to my collaborator, Moritz, for his help in my study especially the mathematical part. I also like to thank other colleagues: Twan, Maaike, Jesse, Nastaran, Feri, Johannes, Gido, Elena S, Bianca, Fabian, Simone M, Mike, Paulus, Joshua, etc. I enjoy the
time with you.

After work, an important part of my life was cooking and eating, possibly because it is an important part in Chinese culture. I have a deep appreciation for Zhuoran, Xiangrong, Wei, Zhengyu, and Xue, because they have almost-professional cooking skills and I benefited a lot from that. Particularly, Zhuoran invited me to have big meals made by himself many times when I was quite busy and could only feed myself with bread. Wei and Zhengyu, my housemates, allowed me to immediately enjoy dinner once I got home over a few months, which made me really feel at home. I also thank Rongbing, Manshu, and Liping for cooking nice meals together when they did their Master at Radboud, and for inviting me to their restaurant (or home) to enjoy real professional food. Another big part of my extracurricular life was going to the gym and playing badminton, where I met many good friends. Over here I need to again thank Xiangrong for encouraging and coaching me to strengthen my muscles, and Zhuoran for organizing the regular badminton events. Other playing mates include: Kai, Mo, Kaizheng, Teng, Yongjun, Baojun, Jiaying, Yidong, Anton, Mohsen, etc. It is quite nice to have you in my life.

I want to give another special thank to Sven and Manxia, who brought me from the train station to my living place on my first day arriving in the Netherlands. I could not imagine how I could arrive at my destination without their help, because everything at that moment was completely new to me. I also thank Jing for helping me look for the living place close to my office, which saved me a lot of time for writing this thesis. Great thanks to two special friends who ever lived in Hatert with me: Hongbo and Zijian. I got to know Hongbo in a English course for a special reason. Then we shared the same house, then we found we are from the same home university. Quite excited to meet you. Zijian is probably a person that I should know earlier, since we support the same football club for more than 15 years. I cannot forget the time of watching matches together with you in both TV live and real stadiums. You’ll never walk alone!

Finally, I want to show my deep-felt thank and love to my parents, wife, and daughter. Without the full support and encouragement of my wife, Qin Qin Dong, I could not finish this thesis. She definitely sacrificed a lot in order to support me to pursue my Ph.D. Ruoxi, my dear daughter, has been the biggest motivation for me to work hard since she was born. I love all of them!
Curriculum Vitae

Ruifei Cui was born in Zibo, China, on August 15th, 1989. He obtained his Bachelor’s Degree in Electronic Science and Technology at Hebei University of Technology in 2011. After that, he did his Master study in Communication and Information System at National Digital Switching System Engineering & Technological Research Center and obtained his Master Degree in 2014.

One year later, he went to Radboud University Nijmegen, the Netherlands, where he started his Ph.D. research on the subject of causal discovery at the Institute for Computing and Information Sciences under the supervision of Prof. dr. Tom Heskes and Dr. Perry Groot. During his Ph.D. career, he was in responsibility of organizing the weekly seminar “Coffee Talk” in Data Science group, served as a teaching assistant for the course “Machine Learning in Practice”, as well as attended several international conferences and presented his work over there. Also, he did a visiting research at Tsinghua University in the second-half of the third year of his Ph.D.