The Maximum Minimum Parents and Children Algorithm

Mikael Petersson

LiTH - MAT - EX - - 2010 / 09 - - SE
The Maximum Minimum Parents and Children Algorithm

Matematiska institutionen, Linköpings Universitet

Mikael Petersson

LiTH - MAT - EX - - 2010 / 09 - - SE

Examensarbete: 15 hp

Level: C

Supervisor: John M. Noble,
Matematiska institutionen, Linköpings Universitet

Examiner: John M. Noble,
Matematiska institutionen, Linköpings Universitet

Linköping: June 2010
The Maximum Minimum Parents and Children Algorithm

Mikael Petersson

Given a random sample from a multivariate probability distribution $p$, the maximum minimum parents and children algorithm locates the skeleton of the directed acyclic graph of a Bayesian network for $p$ provided that there exists a faithful Bayesian network and that the dependence structure derived from data is the same as that of the underlying probability distribution.

The aim of this thesis is to examine the consequences when one of these conditions is not fulfilled. There are some circumstances where the algorithm works well even if there does not exist a faithful Bayesian network, but there are others where the algorithm fails.

The MMPC tests for conditional independence between the variables and assumes that if conditional independence is not rejected, then the conditional independence statement holds. There are situations where this procedure leads to conditional independence being accepted that contradict conditional dependence relations in the data. This leads to edges being removed from the skeleton that are necessary for representing the dependence structure of the data.

Bayesian networks, Structure learning, Faithfulness.
Abstract

Given a random sample from a multivariate probability distribution $p$, the maximum minimum parents and children algorithm locates the skeleton of the directed acyclic graph of a Bayesian network for $p$ provided that there exists a faithful Bayesian network and that the dependence structure derived from data is the same as that of the underlying probability distribution.

The aim of this thesis is to examine the consequences when one of these conditions is not fulfilled. There are some circumstances where the algorithm works well even if there does not exist a faithful Bayesian network, but there are others where the algorithm fails.

The MMPC tests for conditional independence between the variables and assumes that if conditional independence is not rejected, then the conditional independence statement holds. There are situations where this procedure leads to conditional independence being accepted that contradict conditional dependence relations in the data. This leads to edges being removed from the skeleton that are necessary for representing the dependence structure of the data.

**Keywords:** Bayesian networks, Structure learning, Faithfulness.
Acknowledgements

I would like to thank my supervisor John M. Noble for giving me this interesting project with both practical and theoretical aspects. I also want to thank my friends at the mathematics program at Linköpings Universitet for their support over the years. My thoughts also goes out to my family and other friends.

*Mikael Petersson*
Contents

1 Introduction

2 Mathematical Background
   2.1 Graph Theory .............................................. 3
   2.2 Bayesian Networks ........................................... 4
   2.3 Faithfulness .................................................. 5
   2.4 Testing for Conditional Independence ......................... 7

3 The MMPC Algorithm
   3.1 The Maximum Minimum Parents and Children Algorithm .... 9

4 Results and Discussion
   4.1 Distributions Without a Faithful Graph ...................... 13
      4.1.1 The Trek .............................................. 13
      4.1.2 Coin Tossing Example ................................ 16
   4.2 Learning the Graph From Observed Data ..................... 19
   4.3 Summary ..................................................... 24

A Implementation Details
   A.1 Matlab Programs .......................................... 27
Chapter 1

Introduction

Bayesian networks are used to model systems that involve complexity and uncertainty. The system is described by a directed acyclic graph (DAG), where the nodes represents random variables and the edges describes the relations between them. The graphical model suggests how to decompose the joint probability distribution of the system into smaller parts so that efficient calculations can be performed.

There are two basic problems when constructing a Bayesian network. The first is to decide which edges to use in the graph. The other is to specify the conditional probability tables used in the decomposition of the joint probability distribution when it is factorised along the directed acyclic graph. This thesis deals with the first of these two problems.

For a Bayesian network (definition 9), all conditional dependence relations between variables are represented by corresponding \( d \)-connection between the variables in the directed acyclic graph. A Bayesian network is said to be faithful if in addition to this, all the conditional independence statements between the random variables are represented by \( d \)-separation (definition 11) in the directed acyclic graph. There exist distributions where there does not exist a Bayesian network that is faithful to the distribution.

This thesis studies the maximum minimum parents and children (MMPC) algorithm, which is the first stage of the maximum minimum hill climbing (MMHC) algorithm, introduced in [3]. The aim of the MMHC algorithm is to return a Bayesian network corresponding to a probability distribution. The first stage, the MMPC algorithm, is a constraint based algorithm that determines the skeleton through testing for conditional independence. The construction of the skeleton is followed by the edge orientation phase, which is a search and score based greedy algorithm, orienting the edges that have been selected in the first stage of the algorithm.

If there is a faithful Bayesian network, then the MMPC algorithm locates its skeleton. The aim of the thesis is to explore what happens when the algorithm is applied when there does not exist a faithful Bayesian network. The MMPC algorithm constructs the skeleton by testing for conditional independence between variables. If there is a subset \( S \) such that \( X \perp Y|S \), then the skeleton does not include the edge \( (X, Y) \). If \( X \not\perp Y|S \) for any subset \( S \), the edge \( (X, Y) \) is included in the skeleton. This approach gives the correct skeleton if there is a faithful Bayesian network. The justification for this is given in theorem 3.
The tests for independence are each carried out using a nominal significance level of 5%. This means that with probability 0.05 the hypothesis of conditional independence will be rejected, even if it holds. More seriously, the approach taken by the MMPC algorithm is to accept an independence statement if the result of a hypothesis test is ‘do not reject conditional independence’. An example is given where this leads to accepting conditional independence statements that are incompatible with dependence statements that have been established. This is used to illustrate the problems that can arise with the resulting graphical model.

The outline of this report is as follows. In chapter 2, the mathematical background is presented with the necessary definitions and results and some of the key proofs. The necessary graph theory is discussed, together with some background and key results from Bayesian networks and the concept of faithfulness are discussed. The procedure for testing conditional independence is described.

In chapter 3, the MMPC algorithm is presented, with a proof that the algorithm returns the correct skeleton if there exists a faithful Bayesian network and the CI tests return the correct results. Chapter 4 is the core of the report, where the results are presented and discussed. The performance of the MMPC algorithm is described in various scenarios where there does not exist a faithful Bayesian network. A particular data set with six binary variables is considered that illustrates the problems that can arise with the method of determining CI statements. Finally, the matlab code together with a description of the programmes is given in the appendix.

The reader of this thesis is assumed to be familiar with the contents of undergraduate courses in probability theory and mathematical statistics. The theory is based on the book Bayesian Networks: An Introduction [2] by Timo Koski and John M. Noble, and tries to use the same notation as much as possible.
Chapter 2

Mathematical Background

In this chapter, the necessary mathematics for understanding the MMPC algorithm is presented.

2.1 Graph Theory

This section is basically a collection of definitions from graph theory which is needed for later purposes.

**Definition 1** (Graph). A graph \( G \) consists of a finite node set \( V = (\alpha_1, ..., \alpha_d) \) and an edge set \( E \) describing relations between the nodes in \( V \). If \( \langle \alpha_i, \alpha_j \rangle \in E \) there is an undirected edge between the nodes \( \alpha_i \) and \( \alpha_j \). If \( (\alpha_i, \alpha_j) \in E \) there is a directed edge from \( \alpha_i \) to \( \alpha_j \). If all edges in a graph are undirected, it is called an undirected graph, and if all edges are directed, it is called a directed graph.

**Example.** The graph \( G = (V, E) \) where \( V = \{X_1, X_2, X_3, X_4\} \) and \( E = \{(X_2, X_3), (X_2, X_4), (X_3, X_4)\} \) can be illustrated as follows.

![Figure 2.1: Example of a graph](image)

All graphs considered in this thesis will have no edges of form \( (\alpha_i, \alpha_i) \) (no loops) and any \( (\alpha_i, \alpha_j) \in E \) appears exactly once (no multiple edges). Such graphs are called simple graphs but we will refer to them simply as graphs.

**Definition 2** (Parent, Child). If for two nodes \( \alpha_i \) and \( \alpha_j \) there is a directed edge from \( \alpha_i \) to \( \alpha_j \), then \( \alpha_i \) is a parent of \( \alpha_j \) and \( \alpha_j \) is a child of \( \alpha_i \).

**Definition 3** (Trail). A trail between two nodes in a graph is a collection of nodes \( \tau = (\tau_1, ..., \tau_m) \) such that there is an edge (directed or undirected) between \( \tau_i \) and \( \tau_{i+1} \) for all \( i = 1, ..., m - 1 \).
Definition 4 (Directed Path, Cycle). A trail $\tau = (\tau_1, ..., \tau_m)$ is called a directed path if there is a directed edge from $\tau_i$ to $\tau_{i+1}$ for all $i = 1, ..., m-1$. A directed path starting and ending at the same node is called a cycle.

Definition 5 (Descendant, Ancestor). A node $\gamma$ is a descendant of another node $\beta$ if there is a directed path from $\beta$ to $\gamma$. A node $\alpha$ is an ancestor of another node $\beta$ if there is a directed path from $\alpha$ to $\beta$.

Figure 2.2: Illustration of definition 5

Definition 6 (Immorality). An immorality in a graph is a triple of nodes $(\alpha, \beta, \gamma)$ such that $\beta$ is a child of both $\alpha$ and $\gamma$, and there is no edge (neither directed nor undirected) between $\alpha$ and $\gamma$.

Figure 2.3: An immorality

Definition 7 (Skeleton). The skeleton of a graph $G$ is the graph obtained by replacing every directed edge in $G$ with an undirected edge.

The final definition of this section defines the family of graphs which will serve as graphical illustrations of discrete probability distributions.

Definition 8 (DAG). A directed acyclic graph is a directed graph that contains no cycles.

2.2 Bayesian Networks

In this section it is described how a discrete probability distribution can be represented by a graphical model where the nodes in a DAG represents random variables. A formal definition of a Bayesian network can be fairly complicated. The following slightly simplified definition will be sufficient for the purposes of this thesis.

Definition 9 (Bayesian Network). Let $p$ be a probability distribution over the discrete random variables $X_1, ..., X_d$, each having a finite number of possible outcomes. Let $G = (V, E)$ be a directed acyclic graph where the node set $V$ represents the random variables and the edge set $E$ describes the relations between them. Let $X_{\Pi(i)}$ denote the set of parents of the node $X_i$. The graph is
constructed such that \( p \) factorizes along \( G \):

\[
p_{X_1,...,X_d}(x_1,...,x_d) = \prod_{i=1}^{d} p_{X_i|X_{\Pi(i)}}(x_i|x_{\Pi(i)}).
\]

A Bayesian network consists of \( G \) and a specification of the conditional probabilities in the factorization of \( p \).

Note that for any random variables \( X_1,...,X_d \) it always holds that

\[
p_{X_1,...,X_d}(x_1,...,x_d) = p_{X_1}(x_1) p_{X_2|X_1}(x_2|x_1) \cdots p_{X_d|X_1,...,X_{d-1}}(x_d|x_1,...x_{d-1}).
\]

A Bayesian network states the best way of decomposing the joint probability function.

The edge set \( E \) describes the conditional independence relations between the variables. If the distribution is faithful (Definition 12), every conditional independence statement can be derived from the graph. To do this, the concept of \( d \)-separation between nodes is needed.

Given three nodes \( X_1, X_2 \) and \( X_3 \) in a DAG such that the edges \( (X_1, X_2) \) and \( (X_2, X_3) \) are present, there can be three types of connections between the nodes, depending on the directions of the edges.

1. If \( X_1 \rightarrow X_2 \rightarrow X_3 \) or \( X_1 \leftarrow X_2 \leftarrow X_3 \) it is a chain connection and \( X_2 \) is called a chain node.

2. If \( X_1 \leftarrow X_2 \rightarrow X_3 \) it is a fork connection and \( X_2 \) is called a fork node.

3. If \( X_1 \rightarrow X_2 \leftarrow X_3 \) it is a collider connection and \( X_2 \) is called a collider node.

**Definition 10** (Blocked Trail). A trail \( \tau \) between two different nodes \( X \) and \( Y \) in a graph \( G = (V, E) \) is blocked by a set of nodes \( S \subseteq V \setminus \{X, Y\} \) if at least one of the two following conditions holds.

1. There is a node \( W \in S \) in \( \tau \) that is not a collider node.

2. \( W \) is a collider node in \( \tau \) and neither \( W \) nor any of its descendants belongs to \( S \).

**Definition 11** (\( d \)-separation). Two different nodes \( X \) and \( Y \) in a graph \( G = (V, E) \) are \( d \)-separated by a set of nodes \( S \subseteq V \setminus \{X, Y\} \), if every trail between \( X \) and \( Y \) is blocked by \( S \). This is denoted by \( X \perp Y | G S \). Otherwise, \( X \) and \( Y \) are \( d \)-connected given \( S \).

The nodes in \( S \) are called instantiated nodes.

### 2.3 Faithfulness

This section introduces the concept of a faithful Bayesian network. In a Bayesian network, all conditional dependence statements are represented by correspond-
ing $d$-connection statements. In a faithful Bayesian network, all conditional independence statements are represented by $d$-separation statements in the directed acyclic graph. Faithfulness is defined as follows.

**Definition 12** (Faithfulness). A probability distribution $p$ and a directed acyclic graph $G = (V, E)$ are faithful to each other if

$$X \perp Y|S \iff X \perp Y || S \quad \text{(2.1)}$$

for all $X \in V$, $Y \in V$ and $S \subseteq V$, where the variables $X$, $Y$ and those in $S$ are disjoint.

The faithful graph is not necessarily unique. There might exist more than one DAG faithful the the same distribution. Such two graphs are said to be Markov equivalent.

**Theorem 1.** Two directed acyclic graphs are Markov equivalent if and only if they have the same skeleton and the same immoralities.

**Corollary 2.** If $G_1$ and $G_2$ are two different directed acyclic graphs such that both are faithful to the same probability distribution $p$, then $G_1$ and $G_2$ have the same skeleton.

The proofs are omitted here. They can be found in [2]. The corollary will be needed to prove correctness of the MMPC algorithm. Theorem 1 gives a compact way of illustrating all DAGs that are Markov equivalent to a given DAG. The following definition shows how.

**Definition 13** (Essential Graph). Let $G = (V, E)$ be a DAG. Let $G^* = (V, E^*)$ be the graph obtained by making all the edges in $G$ undirected, except for those contained in an immorality. That is, if $(\alpha, \beta, \gamma)$ is an immorality in $G$, then $(\alpha, \beta) \in E^*$ and $(\gamma, \beta) \in E^*$. Then $G^*$ is called the essential graph associated with $G$.

The following theorem will also be needed to prove correctness of the MMPC algorithm.

**Theorem 3.** Let $p$ be a probability distribution such that there exist a DAG $G = (V, E)$ which is faithful to $p$. Then, in any such graph, there is an edge between $X \in V$ and $Y \in V$ iff $X \not\perp Y|S$ for all $S \subseteq V \setminus \{X, Y\}$.

**Proof.** Let $G$ be a graph faithful to $p$ and suppose that there is an edge between $X$ and $Y$. Then $X$ and $Y$ are $d$-connected given any subset $S$ of the other variables. Since the graph was faithful, this is equivalent to $X \not\perp Y|S$ for all $S \subseteq V \setminus \{X, Y\}$.

Conversely, suppose that $X \not\perp Y|S$ and hence $X \not\perp Y || S$ for all $S \subseteq V \setminus \{X, Y\}$. Define the set $S' \subseteq V \setminus \{X, Y\}$ by

$$S' = \{Z \in V \setminus \{X, Y\} : Z \text{ is an ancestor of } X \text{ or } Y \}.$$ 

By assumption, $X$ and $Y$ are $d$-connected given $S'$ so there must exist a trail $\tau$ between $X$ and $Y$ not blocked by $S'$. It follows that all collider nodes are either in $S'$ or has a descendant in $S'$ and the chain and fork nodes are not in $S'$. 
2.4 Testing for Conditional Independence

From the construction of $S'$ it follows that every node that has a descendant in $S'$ is also in $S'$ so all collider nodes in $\tau$ are in $S'$. Every other node (chain or fork) in $\tau$ is an ancestor of $X$, $Y$ or a collider node in $S'$. This implies that all nodes in $\tau$ except for $X$ and $Y$ are in $S'$ and hence all nodes in $\tau \setminus \{X, Y\}$ are collider nodes. So the only possible trails not blocked by $S'$ between $X$ and $Y$ are
\[ X \rightarrow Y \quad X \leftarrow Y \quad X \rightarrow Z \leftarrow Y \text{ for some } Z \in S' \]
But the third possibility leads to a contradiction because in that case $Z$ is a child of both $X$ and $Y$ and since $Z$ is in $S'$ it is either an ancestor of $X$ or $Y$ so then there is a cycle in $G$ which we assumed to be acyclic. From this it follows that there must be an edge between $X$ and $Y$.

2.4 Testing for Conditional Independence

This section describes the procedure used to determine whether or not a statement $X \perp Y | S$ is to be included in the set of conditional independence (CI) statements. All random variables are assumed to be discrete and have a finite sample space.

Definition 14 (Independence). Two random variables $X$ and $Y$ are independent if
\[ p_{X,Y}(x, y) = p_X(x)p_Y(y), \]
where $p$ denotes the probability function. This is denoted $X \perp Y$.

Definition 15 (Conditional Independence). Let $X, Y$ and $Z$ be random variables. $X$ and $Y$ are conditionally independent given $Z$ if
\[ p_{X,Y|Z}(x, y|z) = p_{X|Z}(x|z)p_{Y|Z}(y|z). \]
This is denoted $X \perp Y | Z$.

The variables in the definitions may be multivariate. In this report $X$ and $Y$ will be one-dimensional random variables and $Z$ will be considered as a set of one-dimensional random variables. If $X$ and $Y$ are (unconditionally) independent, this will sometimes be denoted $X \perp Y | \emptyset$, where $\emptyset$ is the empty set.

Let $\hat{p}$ denote the empirical probability distribution and $n$ the number of observations. To test if $X \perp Y | \emptyset$ the following test statistic will be used.
\[ G_\emptyset^2 = 2n \sum_{x,y} \hat{p}_{X,Y}(x,y) \log \frac{\hat{p}_{X,Y}(x,y)}{\hat{p}_X(x)\hat{p}_Y(y)} \]
The test is
\[ H_0 : X \perp Y \quad \text{vs.} \quad H_1 : X \not\perp Y. \]
From the definition of independence, it follows that the test statistic should be small if $X$ and $Y$ are independent. Therefore, the hypothesis test is rejected for large values of the statistic.

The statistic used to test
\[ H_0 : X \perp Y | Z \quad \text{vs.} \quad H_1 : X \not\perp Y | Z \]
where $Z \neq \phi$, is

$$G^2 = 2n \sum_{x, y, z} \hat{p}_{X,Y,Z}(x, y, z) \log \frac{\hat{p}_{X,Y,Z}(x, y, z) \hat{p}_Z(z)}{\hat{p}_{X,Z}(x, z) \hat{p}_{Y,Z}(y, z)}.$$  

Large values of the statistic support the alternative hypothesis. The following lemma shows this.

**Lemma 4.** $X \perp Y \mid Z$ if and only if

$$p_{X,Y,Z}(x, y, z) = p_{X,Z}(x, z)p_{Y,Z}(y, z)p_Z(z). \quad (2.2)$$

**Proof.** If $X \perp Y \mid Z$ then

$$p_{X,Y,Z}(x, y, z) = p_{X,Y|Z}(x, y|z)p_Z(z) = p_{X|Z}(x|z)p_{Y|Z}(y|z)p_Z(z)$$

$$= p_{X,Z}(x, z)p_{Y,Z}(y, z)p_Z(z) = \frac{p_{X,Z}(x, z)p_{Y,Z}(y, z)}{p_Z(z)} = \frac{p_{X,Y,Z}(x, y, z)}{p_Z(z)}$$

Conversely, if (2.2) holds, then

$$p_{X,Y|Z}(x, y|z) = \frac{p_{X,Y,Z}(x, y, z)}{p_Z(z)} = \frac{p_{X,Z}(x, z)p_{Y,Z}(y, z)}{p_Z(z)}$$

$$= p_{X|Z}(x|z)p_{Y|Z}(y|z)$$

Let $j_x$ and $j_y$ be the number of possible outcomes for $X$ and $Y$ respectively. By the central limit theorem, the distribution of $G^2_{\phi}$ is approximately chi squared with $(j_x-1)(j_y-1)$ degrees of freedom. When considering the test of conditional independence, $G^2(X,Y|Z = z)$ is approximately chi squared with $(j_x-1)(j_y-1)$ degrees of freedom for each instantiation $z$ of $Z$ and these random variables are independent of each other. The sum of independent chi squared variables is again chi squared, where the number of degrees of freedom is obtained by summing.

The central limit theorem approximation is inaccurate unless each cell count is greater than or equal to 5. Therefore, only instantiations of $Z$ where the cell count is greater than or equal to 5 for each pair $(x, y)$ are considered. If there is insufficient data to perform the test, then the relation $X \perp Y \mid Z$ is not added to the set of CI statements.
Chapter 3

The MMPC Algorithm

The maximum minimum parents and children (MMPC) algorithm is presented in [3] as the first stage of the maximum minimum hill climbing (MMHC) algorithm. The purpose of the MMHC algorithm is to learn the directed acyclic graph of a Bayesian network given observed data. The first stage is the MMPC algorithm, which locates the skeleton using a constraint-based technique of inserting an edge $\langle X, Y \rangle$ if and only if $X \not\perp Y | S$ for any subset $S$. The second stage is the edge orientation stage, which uses a search and score technique. Using only edges in the skeleton, at each stage the algorithm chooses the single operation add edge / delete edge / change orientation of an existing edge that does not produce a graph structure that was previously visited, which gives the highest score. The directed acyclic graph returned is the graph with the highest score visited.

3.1 The Maximum Minimum Parents and Children Algorithm

Suppose that $p$ is a probability distribution such that there exist a graph $G$ which is faithful to $p$. Then the MMPC algorithm locates the skeleton of any DAG faithful to $p$. Recall that corollary 2 states that if $G_1$ and $G_2$ are two different DAGs which are both faithful to $p$, then they have the same skeleton. The algorithm works in three stages. In stage 1 and 2 a superset of the parents / children set is located for each variable and stage 3 is a symmetry correction so that the correct parents / children sets of each variable is returned.

The algorithm

Stage 1. Let $T$ be one of the variables in the distribution. Let $(X_i)_{i=1}^{d}$ be an ordering of the other variables and set $Z_0 = \phi$, the empty set. For $i = 1,...,d$ do the following.

$$Z_i = \begin{cases} Z_{i-1} & \text{if } X_i \perp T | Z_{i-1} \\ Z_{i-1} \cup \{X_i\} & \text{otherwise} \end{cases}$$

Stage 2. Set $Z_0 = Z_d$ and let $(X_i)_{i=1}^{k}$ be an ordering of the variables in $Z_0$. For $i = 1,...,k$ do the following.
Stage 3. First run stage 1 and 2 on all the variables in the distribution. Then the sets $Z_{X_i}$ are known for all variables $X_i$ in the distribution. Let $T$ be one of these variables and let $(X_i)_{i=1}^j$ be an ordering of the variables in $Z_T$. Set $Y_0 = Z_T$. For $i = 1, ..., j$ do the following.

$$Y_i = \begin{cases} Y_{i-1} & \text{if } T \in Z_{X_i} \\ Y_{i-1} \setminus \{X_i\} & \text{otherwise} \end{cases}$$

Set $Y_T = Y_j$. This is the parents / children set of $T$.

**Theorem 5.** Suppose that $p$ is a probability distribution that satisfies the following two conditions.

1. There exists a DAG $G$ which is faithful to $p$.
2. All the conditional independence statements derived from the data are present in the distribution and all conditional independence statements that were rejected at the 5% significance level are not present in the distribution.

Then the MMPC algorithm will return the skeleton of any DAG faithful to $p$.

**Proof.** Let $\mathcal{PC}_T$ denote the correct parents / children set of $T$ and let $Z_T$ be the set of nodes returned by the MMPC algorithm after stage 2. Assume $X \in \mathcal{PC}_T$. By Theorem 3, $X \not\perp T|S$ for any $S \subseteq V \setminus \{X, T\}$. This implies that $X$ will be selected in stage 1 and will not be removed in stage 2, so that $X \in Z_T$. This proves that $\mathcal{PC}_T \subseteq Z_T$.

Next it will be proved that if $X \in Z_T$ but $X \not\in \mathcal{PC}_T$, then $X$ is a descendant of $T$ in any DAG $G$ faithful to $p$. If $X \in Z_T$, then $X \not\perp T|S$ for any $S \subseteq Z_T$. In particular, $X \not\perp T|S$ for any $S \subseteq \mathcal{PC}_T$. This implies that at least one of the nodes in the parents / children set is both a collider node on one trail between $X$ and $T$ and a fork or chain node on another. Such a node is therefore a child of $T$ and is a collider node on one trail and a chain node on the other trail. Without loss of generality, no descendants of this node are children of $T$ (otherwise choose a node on the trail for which this is not the case). This trail is only open when the node is uninstantiated if $X$ is a descendant of $T$. This is illustrated in figure 3.1.

![Figure 3.1: X is a descendant of T](image-url)
Finally it will be proved that the MMPC algorithm returns $\mathcal{PC}_T$. Suppose $X \in \mathcal{PC}_T$ and hence also $T \in \mathcal{PC}_X$. It follows from above that after stage 2, $X \in \mathcal{Z}_T$ and $T \in \mathcal{Z}_X$ so $X$ will not be removed from the parents / children set of $T$ in stage 3 so the algorithm returns all members of $\mathcal{PC}_T$.

Suppose that $X$ is returned and $X \notin \mathcal{PC}_T$. Then we also have $T \notin \mathcal{PC}_X$ and since $X$ was not removed in stage 3, $X \in \mathcal{Z}_T$ and $T \in \mathcal{Z}_X$. This implies that $X$ is a descendant of $T$ and $T$ is a descendant of $X$ in any DAG $\mathcal{G}$ faithful $p$. But this is a contradiction since $\mathcal{G}$ is acyclic so only members of the correct parents / children sets for each variable will be returned.

So the MMPC algorithm returns the skeleton of a DAG $\mathcal{G}$ faithful to $p$ and because of corollary 2 this is the skeleton of any DAG faithful to $p$.

Remark. In [3] a different version of stage 1 is presented. It uses a heuristic to decide an order in which the nodes enters $\mathcal{Z}$ and this ordering is then used in stage 2. However, this version is only for the purposes of computational efficiency. The result will be the same as in the version presented here. Since this report mainly concerns the graph that the algorithm returns, this version without the modification will be sufficient.

The Kullback Leibler Measure of Divergence

Given two probability distributions $p$ and $q$ over a set of variables (for example $q$ may be a fitted distribution, derived from data that factorises according to a Bayesian network, while $p$ may be a target distribution), it is important to have an idea of the extent to which the distributions differ. One common measure is the Kullback Leibler measure of divergence, which is defined as follows.

**Definition 16** (Kullback-Leibler Divergence). Let $p$ and $q$ be two discrete probability distributions with the same sample space $\Omega = \{\omega_1, \omega_2, ..., \omega_n\}$. Let $p_i$ be the probability of the outcome $\omega_i$ for the distribution $p$, and $q_i$ corresponding for $q$. Then the Kullback-Leibler divergence is defined as

$$D(p|q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}. $$

Here it is defined that $0 \cdot \log 0 = 0$. This satisfies $D(p|q) \geq 0$, with $D(p|q) = 0$ if and only if $p = q$. The proof of this uses Jensen’s inequality which states that for any random variable $X$ and convex function $f$, it holds that

$$E(f(X)) \geq f(E(X)),$$

where $E$ denotes the expected value. Moreover, if $f$ is strictly convex and $E(f(X)) = f(E(X))$ then $X$ is a constant. The proof of this can be found in [1].

**Lemma 6.** The Kullback-Leibler divergence satisfies $D(p|q) \geq 0$, with $D(p|q) = 0$ if and only if $p = q$.

**Proof.** Let $X$ be a random variable defined by

$$P(X = q_i/p_i) = \begin{cases} p_i & \text{for } i = 1, ..., n \\ 0 & \text{otherwise} \end{cases}$$ (3.1)
Then

\[ D(p||q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i} = \sum_{i=1}^{n} p_i \left( -\log \frac{q_i}{p_i} \right) = E(-\log(X)) \]

\[ \geq -\log(E(X)) = -\log \left( \sum_{i=1}^{n} p_i \frac{q_i}{p_i} \right) = -\log(1) = 0, \tag{3.2} \]

where the inequality follows from Jensen’s inequality since \(-\log(\cdot)\) is a convex function. If \(p = q\) then it is clear that \(D(p||q) = 0\). Conversely if \(D(p||q) = 0\), then the inequality in (3.2) must be an equality so it follows from Jensen’s inequality that \(X\) is a constant since \(-\log(\cdot)\) is a strictly convex function. Then by (3.1) we must have \(q_i = kp_i\) for all \(i\), where \(k\) is a constant and since \(p\) and \(q\) are probability distributions it follows that

\[ 1 = \sum_{i=1}^{n} q_i = k \sum_{i=1}^{n} p_i = k, \]

so \(p_i = q_i\) for all \(i = 1, ..., n\). \(\square\)
Chapter 4

Results and Discussion

This chapter considers the graph returned by the MMPC algorithm in several situations when there is no faithful DAG for the distribution, and discusses the performance of the MMPC algorithm when used on a data set with six binary variables.

4.1 Distributions Without a Faithful Graph

This section examines the performance of the MMPC algorithm when used on a distribution \( p \), where there exists no DAG \( G \) faithful to \( p \).

4.1.1 The Trek

The first example of a distribution without a faithful representation factorizes along a graph known as a trek.

**Definition 17** (Trek). Let \( G \) be a directed acyclic graph. A trek is subgraph of \( G \) over four variables \( X_1, \ldots, X_4 \) which only contains the following directed edges:

\[
X_1 \to X_2 \quad X_1 \to X_3 \quad X_2 \to X_4 \quad X_3 \to X_4
\]

It is illustrated in figure 4.1.

![Figure 4.1: A trek](image-url)
A construction of a distribution which factorizes along the trek and does not have a faithful graphical representation can be done in the following way. The distribution must satisfy

\[ p_{X_1, X_2, X_3, X_4} = p_{X_1} p_{X_2|X_1} p_{X_3|X_1} p_{X_4|X_2, X_3}. \]

Assume that all the variables are binary, each taking values 1 or 0. Let the probabilities \( p_{X_1}(1) \) and \( p_{X_1}(0) \) be arbitrary and let the other probabilities be given by the following relations.

\[
\begin{align*}
p_{X_2|X_1}(1|0) &= 1 - p_{X_3|X_1}(1|1) = a \\
p_{X_2|X_1}(1|1) &= 1 - p_{X_3|X_1}(1|0) = b \\
p_{X_4|X_2, X_3}(1|1, 1) &= p_{X_4|X_2, X_3}(1|0, 0) = c \\
p_{X_4|X_2, X_3}(1|0, 1) &= p_{X_4|X_2, X_3}(1|1, 0) = d
\end{align*}
\]

Using this the following shows that \( X_4 \perp X_1 \).

\[
\begin{align*}
p_{X_4|X_1}(1|1) &= \frac{p_{X_1, X_4}(1, 1)}{p_{X_1}(1)} \\
&= \sum_{x_2, x_3} p_{X_2|X_1}(x_2|1)p_{X_3|X_1}(x_3|1)p_{X_4|X_2, X_3}(1|x_2, x_3) \\
&= b(1 - a)c + bad + (1 - b)(1 - a)d + (1 - b)ac
\end{align*}
\]

\[
\begin{align*}
p_{X_4|X_1}(1|0) &= \frac{p_{X_1, X_4}(0, 1)}{p_{X_1}(0)} \\
&= \sum_{x_2, x_3} p_{X_2|X_1}(x_2|0)p_{X_3|X_1}(x_3|0)p_{X_4|X_2, X_3}(1|x_2, x_3) \\
&= a(1 - b)c + abd + (1 - a)(1 - b)d + (1 - a)bc
\end{align*}
\]

Choose \( a \neq \frac{1}{2}, b \neq \frac{1}{2}, c \neq \frac{1}{2}, d \neq \frac{1}{2}, a \neq b \) and \( c \neq d \). Then it can be shown that the entire list of conditional independence statements that hold for \( p \) is

\[ X_1 \perp X_4 \quad X_2 \perp X_3|X_1 \quad X_1 \perp X_4|\{X_2, X_3\} \]

By theorem 3, a faithful DAG for this distribution does not contain an edge between two variables if any conditional independence relation holds between the two variables, given a subset of the remaining variables. Since \( X_1 \perp X_4 \) and \( X_2 \perp X_3|X_1 \), the skeleton does not have an edge \( \langle X_1, X_4 \rangle \) or an edge \( \langle X_2, X_3 \rangle \).

The remaining edges must be included in a faithful graph. To see this, assume that the edge \( \langle X_1, X_2 \rangle \) is removed. Then the only trail between these two variables is \( X_1 - X_3 - X_4 - X_2 \). Since \( X_1 \not\perp X_2|\{X_3, X_4\} \) both \( X_1 \) and \( X_4 \) must be collider nodes for the corresponding d-connection statement to hold. But this is a contradiction so the edge \( \langle X_1, X_2 \rangle \) can not be removed. The same argument holds for the other edges as well.

To see that there is no DAG faithful to \( p \) the following lemma which shows that the MMPC algorithm may be extended to detect immoralities is needed.
Lemma 7. Let $G$ be a DAG faithful to a distribution $p$. Suppose that the skeleton of $G$ has edges $(X, Y)$ and $(Y, Z)$ but no edge $(X, Z)$. Then there is a set $S$ such that $X \perp Z | S$ and $(X, Y, Z)$ is an immorality if $Y \notin S$ and is not an immorality otherwise.

Proof. The existence of the set $S$ follows from theorem 3 since there is no edge $(X, Z)$ in the graph. Since the graph is faithful, $X$ and $Z$ are $d$-separated given $S$ so the trail $X \rightarrow Y \rightarrow Z$ must be blocked. From this it follows that if $Y \notin S$ then $Y$ must be a collider node and if $Y \in S$ then $Y$ must be a chain or fork node.

Assuming existence of a faithful graph for the trek distribution this lemma implies that $X_2$ and $X_3$ are collider nodes. This is because $X_1 \perp X_4 | \phi$ and $X_2, X_3 \notin \phi$ so that $(X_1, X_2, X_4)$ and $(X_1, X_3, X_4)$ are immoralities. But this is a contradiction since it also holds that $X_2 \perp X_3 | X_1$ and $X_4 \notin \{X_1\}$ so $(X_2, X_4, X_3)$ is also an immorality and hence contradictory directions for the edges $(X_2, X_1)$ and $(X_2, X_4)$ are obtained. From this it can be concluded that there exists no faithful DAG for this distribution.

The result when the MMPC algorithm is run on this distribution is presented in the following tables.

| $T$ | $X$ | $Z$ | $T \perp X | Z$ |
|-----|-----|-----|----------------|
| $X_1$ | $X_2$ | $\phi$ | No |
| $X_1$ | $X_3$ | $\{X_2\}$ | No |
| $X_1$ | $X_4$ | $\{X_2, X_3\}$ | Yes |
| $X_2$ | $X_1$ | $\phi$ | No |
| $X_2$ | $X_3$ | $\{X_1\}$ | Yes |
| $X_2$ | $X_4$ | $\{X_1\}$ | No |
| $X_3$ | $X_1$ | $\phi$ | No |
| $X_3$ | $X_2$ | $\{X_1\}$ | Yes |
| $X_3$ | $X_4$ | $\{X_1\}$ | No |
| $X_4$ | $X_1$ | $\phi$ | Yes |
| $X_4$ | $X_2$ | $\phi$ | No |
| $X_4$ | $X_3$ | $\{X_2\}$ | No |

Table 4.1: Stage 1 of the MMPC on the trek example

| $T$ | $X$ | $Z \setminus \{X\}$ | Set $S \subseteq Z \setminus \{X\}$ such that $T \perp X | S$ |
|-----|-----|---------------------|----------------------------------------------------------------|
| $X_1$ | $X_2$ | $\{X_3\}$ | No set |
| $X_1$ | $X_3$ | $\{X_2\}$ | No set |
| $X_2$ | $X_1$ | $\{X_4\}$ | No set |
| $X_2$ | $X_4$ | $\{X_1\}$ | No set |
| $X_3$ | $X_1$ | $\{X_4\}$ | No set |
| $X_3$ | $X_4$ | $\{X_1\}$ | No set |
| $X_4$ | $X_2$ | $\{X_3\}$ | No set |
| $X_4$ | $X_3$ | $\{X_2\}$ | No set |

Table 4.2: Stage 2 of the MMPC on the trek example
No edges are removed in stage 3 so the following skeleton is located.

\[
\begin{array}{c}
X_2 \\
\downarrow \\
X_1 \\
\downarrow \\
X_3 \\
\downarrow \\
X_4
\end{array}
\]

Figure 4.2: Graph obtained by MMPC on the trek distribution

That is, it produces the skeleton of the trek. If the direction of the edges is chosen as in the trek the following holds.

\[
X_1 \not\perp X_4 | G \phi \quad X_2 \perp X_3 | G X_1 \quad X_1 \perp X_4 | G \{X_2, X_3\}
\]

That is, in this DAG two out of three of the CI statements correspond to a \(d\)-separation statement in the graph. Furthermore, all conditional dependence statements in the distribution correspond to \(d\)-connection statements in the graph and it is the smallest graph that achieves this; a graph with this property requires all four edges present (recall the discussion above of what happened when removing one of these four edges). So the graph returned can be considered optimal and hence the MMPC returns the correct skeleton. The following example describes a situation where this is not the case.

### 4.1.2 Coin Tossing Example

Consider the following random variables; toss three different coins and for \(i = 1, 2, 3\) define

\[
X_i = \begin{cases} 
1 & \text{if the outcome of coin } i \text{ is heads} \\
0 & \text{if the outcome of coin } i \text{ is tails}
\end{cases}
\]

Then define three new random variables by:

\[
\begin{align*}
Y_1 &= 1 \quad \text{if } X_2 = X_3 \quad \text{and } 0 \quad \text{otherwise} \\
Y_2 &= 1 \quad \text{if } X_1 = X_3 \quad \text{and } 0 \quad \text{otherwise} \\
Y_3 &= 1 \quad \text{if } X_1 = X_2 \quad \text{and } 0 \quad \text{otherwise}
\end{align*}
\]

Then \(Y_1, Y_2, \text{ and } Y_3\) will be pairwise independent but not jointly independent. To see this, first note that the sample space of \((X_1, X_2, X_3)\) consists of eight equally likely outcomes. The corresponding values of \((Y_1, Y_2, Y_3)\) are shown in the following table.
4.1. Distributions Without a Faithful Graph

From this it follows that the joint distribution of \((Y_1, Y_2, Y_3)\) is

\[
\begin{array}{cccc}
Y_1 & Y_2 & Y_3 & p_{Y_1, Y_2, Y_3}(y_1, y_2, y_3) \\
1 & 1 & 1 & 1/4 \\
1 & 0 & 0 & 1/4 \\
0 & 1 & 0 & 1/4 \\
0 & 0 & 1 & 1/4 \\
\end{array}
\]

and \(p_{Y_1, Y_2, Y_3}(y_1, y_2, y_3) = 0\) for other \((y_1, y_2, y_3)\). It follows that

\[
p_{Y_1, Y_2}(y_1, y_2) = \begin{cases} 
1/4 & (y_1, y_2) \in \{(1, 1), (1, 0), (0, 1), (0, 0)\} \\
0 & \text{for other } (y_1, y_2) 
\end{cases}
\]

\[
p_{Y_1}(y_1) = 1/2 \quad \text{for } y_1 \in \{0, 1\} \quad \text{and } 0 \quad \text{otherwise} \\
p_{Y_2}(y_2) = 1/2 \quad \text{for } y_2 \in \{0, 1\} \quad \text{and } 0 \quad \text{otherwise}
\]

This gives that \(p_{Y_1, Y_2}(y_1, y_2) = p_{Y_1}(y_1)p_{Y_2}(y_2)\) for all \((y_1, y_2)\) and hence \(Y_1 \perp Y_2\). Similar calculations shows that \(Y_1 \perp Y_3\) and \(Y_2 \perp Y_3\). On the other hand

\[
p_{Y_1, Y_2, Y_3}(y_1, y_2, y_3) = \frac{1}{4} \neq \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2} = p_{Y_1}(y_1)p_{Y_2}(y_2)p_{Y_3}(y_3),
\]

for \((y_1, y_2, y_3) \in \{(1, 1, 1), (1, 0, 0), (0, 1, 0), (0, 0, 1)\}\) so the variables are not jointly independent. The entire list of conditional independence statements that hold for this distribution is

\[
Y_1 \perp Y_2 \quad Y_1 \perp Y_3 \quad Y_2 \perp Y_3
\]

It follows that, if a graph is constructed by applying the principles behind the MMPC algorithm, the graph is the empty graph, since there is a conditional independence statement between each pair of variables. But then \(Y_1 \perp Y_2|Y_3\) (a \(d\)-separation statement in the graph) while \(Y_1 \not\perp Y_2|Y_3\) (the corresponding conditional independence statement does not hold). The graph is therefore not faithful. If there were a faithful graphical model, the MMPC procedure would construct it. It follows that there does not exist a faithful DAG for this distribution.

One possible DAG for this example is the DAG given in figure 4.3. All conditional dependence statements between the variables are represented in this
graph. Furthermore, there does not exist a graph with fewer edges that represents all the associations between the variables. But the graph does not represent all the CI statements; only one out of three are represented by the graph. Those missing are $Y_1 \perp Y_2$ and $Y_1 \perp Y_3$.

![Figure 4.3: Suggestion of a DAG for the coin distribution](image)

An important point is that the DAG returned by the MMHC algorithm (which is the skeleton returned by the MMPC algorithm, since there are no edges to orient) provides a very poor fit to the true distribution. The fitted distribution is $\hat{p}_{Y_1, Y_2, Y_3}(y_1, y_2, y_3) = 1/8$ for each possible $(y_1, y_2, y_3)$. For example, in the fitted distribution, $\hat{p}_{Y_1, Y_2, Y_3}(1, 1, 0) = 1/8$ even though the outcome $(1, 1, 0)$ is impossible in the actual distribution. This implies that the Kullback-Leibler divergence between these distributions will be $+\infty$.

In the example given above, there are three hidden variables that have not been considered: $X_1, X_2, X_3$. Since $Y_1$ is a function of $X_2$ and $X_3$, $Y_2$ is a function of $X_1$ and $X_3$ and $Y_3$ is a function of $X_1$ and $X_2$, this suggests the graphical model shown in figure 4.4, where the directed arrows have a causal interpretation.

![Figure 4.4: Graph including the hidden variables](image)

This could be considered the ‘correct’ directed acyclic graph for the distribution, since all the dependence relations are represented by $d$-connection statements and it is the smallest graph for which this property holds. But, again, the MMPC algorithm applied to these six variables would return the empty graph. This is because all six variables are pairwise independent so that no edges will be chosen in the first stage of the algorithm.
4.2 Learning the Graph From Observed Data

This section illustrates the performance of the algorithm on a small example with 1190 observations on 6 binary variables. It shows that failure to reject a conditional independence statement can lead to CI statements that contradict dependence relations in the data that have been established. The data used to illustrate this is taken from a survey regarding attitudes of New Jersey high-school students towards mathematics. The example may be found in for example [2]. The main goal of the study was to evaluate the influence of WAM lectures. These were lectures in mathematical science, all given by women, designed to encourage more interest in mathematics from female students.

A total of 1190 students from eight high-schools (four urban and four suburban) took part in the survey. The result of each student is represented by six binary variables as follows.

A | attendance at WAM lecture | yes/no
B | gender | female/male
C | school type | suburban/urban
D | ‘need mathematics in future work’ | agree/disagree
E | subject preference | mathematical/arts
F | future plans | higher education/immediate job

The result of this survey is given in the following table.

<table>
<thead>
<tr>
<th>school</th>
<th>suburban</th>
<th>urban</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>female</td>
<td>male</td>
</tr>
<tr>
<td>lecture</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td>college</td>
<td>mathematical</td>
<td>y</td>
</tr>
<tr>
<td></td>
<td>n</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>arts</td>
<td>y</td>
</tr>
<tr>
<td></td>
<td>n</td>
<td>12</td>
</tr>
<tr>
<td>job</td>
<td>mathematical</td>
<td>y</td>
</tr>
<tr>
<td></td>
<td>n</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>arts</td>
<td>y</td>
</tr>
<tr>
<td></td>
<td>n</td>
<td>8</td>
</tr>
</tbody>
</table>

The result when the MMPC algorithm is run on this data set is given in the following tables.
## Table 4.3: Stage 1 of the MMPC on the WAM data

| $T$ | $X$ | $Z$ | $G^2$ | df | p-value | $H_0: T \perp X|Z$ |
|-----|-----|-----|-------|----|---------|-------------------|
| $A$ | $B$ | $\phi$ | 0.01  | 1  | 0.9318  | Accept            |
| $A$ | $C$ | $\phi$ | 0.03  | 1  | 0.8633  | Accept            |
| $A$ | $D$ | $\phi$ | 0.19  | 1  | 0.6607  | Accept            |
| $A$ | $E$ | $\phi$ | 0.05  | 1  | 0.8257  | Accept            |
| $A$ | $F$ | $\phi$ | 0.08  | 1  | 0.7771  | Accept            |
| $B$ | $A$ | $\phi$ | 0.01  | 1  | 0.9318  | Accept            |
| $B$ | $C$ | $\phi$ | 0.03  | 1  | 0.8723  | Accept            |
| $B$ | $D$ | $\phi$ | 32.23 | 1  | 0.0000  | Reject            |
| $B$ | $E$ | $\{D\}$ | 37.28 | 2  | 0.0000  | Reject            |
| $B$ | $F$ | $\{D, E\}$ | 1.18  | 4  | 0.8812  | Accept            |
| $C$ | $A$ | $\phi$ | 0.03  | 1  | 0.8633  | Accept            |
| $C$ | $B$ | $\phi$ | 0.03  | 1  | 0.8723  | Accept            |
| $C$ | $D$ | $\phi$ | 0.44  | 1  | 0.5062  | Accept            |
| $C$ | $E$ | $\phi$ | 6.15  | 1  | 0.0132  | Reject            |
| $C$ | $F$ | $\{E\}$ | 56.56 | 2  | 0.0000  | Reject            |
| $D$ | $A$ | $\phi$ | 0.19  | 1  | 0.6607  | Accept            |
| $D$ | $B$ | $\phi$ | 32.23 | 1  | 0.0000  | Reject            |
| $D$ | $C$ | $\{B\}$ | 6.99  | 2  | 0.0304  | Reject            |
| $D$ | $E$ | $\{B, C\}$ | 63.18 | 4  | 0.0000  | Reject            |
| $D$ | $F$ | $\{B, C, E\}$ | 13.87 | 6  | 0.0311  | Reject            |
| $E$ | $A$ | $\phi$ | 0.05  | 1  | 0.8257  | Accept            |
| $E$ | $B$ | $\phi$ | 51.63 | 1  | 0.0000  | Reject            |
| $E$ | $C$ | $\{B\}$ | 6.74  | 2  | 0.0344  | Reject            |
| $E$ | $D$ | $\{B, C\}$ | 63.18 | 4  | 0.0000  | Reject            |
| $E$ | $F$ | $\{B, C, D\}$ | 9.67  | 6  | 0.1394  | Accept            |
| $F$ | $A$ | $\phi$ | 0.08  | 1  | 0.7771  | Accept            |
| $F$ | $B$ | $\phi$ | 0.65  | 1  | 0.4210  | Accept            |
| $F$ | $C$ | $\phi$ | 54.40 | 1  | 0.0000  | Reject            |
| $F$ | $D$ | $\{C\}$ | 29.05 | 2  | 0.0000  | Reject            |
| $F$ | $E$ | $\{C, D\}$ | 9.83  | 4  | 0.0434  | Reject            |
4.2. Learning the Graph From Observed Data

| $T$ | $X$ | $\mathcal{Z} \setminus \{X\}$ | $\text{Set } S \subseteq \mathcal{Z} \setminus \{X\}$ such that $T \perp X|S$ | $G^2$ | df | p-value |
|-----|-----|-----------------|---------------------------------|------|-----|---------|
| $B$ | $D$  | $\{E\}$       | No set                           | —    | —   | —       |
| $B$ | $E$  | $\{D\}$       | No set                           | —    | —   | —       |
| $C$ | $E$  | $\{F\}$       | No set                           | —    | —   | —       |
| $C$ | $F$  | $\{E\}$       | No set                           | —    | —   | —       |
| $D$ | $B$  | $\{C, E, F\}$ | No set                           | —    | —   | —       |
| $D$ | $C$  | $\{B, E, F\}$ | No set                           | —    | —   | —       |
| $D$ | $E$  | $\{B, F\}$    | $\phi$                           | 0.44 | 1   | 0.5062  |
| $D$ | $F$  | $\{B, E\}$    | No set                           | —    | —   | —       |
| $E$ | $B$  | $\{C, D\}$    | No set                           | —    | —   | —       |
| $E$ | $C$  | $\{B, D\}$    | $\{B, D\}$                      | 7.73 | 4   | 0.1021  |
| $E$ | $D$  | $\{B\}$       | No set                           | —    | —   | —       |
| $F$ | $C$  | $\{D, E\}$    | No set                           | —    | —   | —       |
| $F$ | $D$  | $\{C, E\}$    | No set                           | —    | —   | —       |
| $F$ | $E$  | $\{C, D\}$    | $\phi$                           | 2.18 | 1   | 0.1399  |

Table 4.4: Stage 2 of the MMPC on the WAM data

Note that $E$ is in the neighbour set of $C$ but $C$ is not in the neighbour set of $E$ so the edge $\langle C, E \rangle$ is removed in stage 3. These results correspond to the following skeleton.

Figure 4.5: Graph obtained by MMPC on the WAM data

To construct the essential graph (definition 13) from this, check for possible immoralities ($(B, D, F), (C, F, D), (E, D, F)$). Using lemma 7 and the result tables from the algorithm the following is obtained.

$\langle B, F \rangle$ is removed since $B \perp F|\{D, E\}$ so $(B, D, F)$ is not an immorality.

$\langle C, D \rangle$ is removed since $C \perp D|\phi$ so $(C, F, D)$ is an immorality.

$\langle E, F \rangle$ is removed since $E \perp F|\{B, C, D\}$ so $(E, D, F)$ is not an immorality.

These results give the following essential graph.

Figure 4.6: Essential graph for the WAM data
Everything seems fine so far but if looking more closely at the test results there is a serious problem with this graph. In a Bayesian network all conditional dependence statements between the variables are represented by corresponding $d$-connection between the variables in the directed acyclic graph. In this graph $C$ is $d$-separated from $E$ even though the independence between these variables is rejected at the 5% significance level so the MMPC algorithm has removed an edge from the skeleton necessary for representing the dependence structure of the data.

The edge $\langle C, E \rangle$ is removed in stage 2 when $E$ is target variable. In table 4.4 it can be seen that the reason for this is that the statement $C \perp E |\{B, D\}$ is accepted. Since the statement $C \perp E |D$ is rejected, this result together with the graph structure seems weird. The problem is that the conditional independence statements obtained may contradict dependence relations obtained earlier. To show this the following result is needed.

**Theorem 8.** For any discrete random variables $X, Y, Z, W$ the following two statements hold.

1. If $X \perp Y |\{Z, W\}$ and $Y \perp Z |W$, then $X \perp Y |W$.
2. If $X \perp Y |\{Z, W\}$ and $Y \perp \{Z, W\}$, then $X \perp Y$.

**Proof.** If $X \perp Y |\{Z, W\}$ and $Y \perp \{Z, W\}$, then

$$p_{X,Y|W}(x, y|w) = \frac{p_{X,Y,W}(x, y, w)}{p_W(w)}$$

$$= \frac{1}{p_W(w)} \sum_z p_{X,Y|Z,W}(x, y|z, w)p_{Z,W}(z, w)$$

$$= \frac{1}{p_W(w)} \sum_z p_{X|Z,W}(x|z, w)p_{Y|Z,W}(y|z, w)p_{Z,W}(z, w)$$

$$= \frac{p_{Y|W}(y|w)}{p_W(w)} \sum_z p_{X,Z,W}(x, z, w) = p_{X|W}(x|w)p_{Y|W}(y|w).$$

If $X \perp Y |\{Z, W\}$ and $Y \perp \{Z, W\}$, then

$$p_{X,Y}(x, y) = \sum_{z,w} p_{X,Y|Z,W}(x, y|z, w)p_{Z,W}(z, w)$$

$$= \sum_{z,w} p_{X|Z,W}(x|z, w)p_{Y|Z,W}(y|z, w)p_{Z,W}(z, w)$$

$$= p_Y(y) \sum_{z,w} p_{X,Z,W}(x, z, w) = p_X(x)p_Y(y).$$

This theorem can be used to show that the CI statements obtained are contradictory. When testing if $E \perp C |\{B, F\}$ and $C \perp B |F$ the tests say that both these statements should be accepted. But according to statement 1 of theorem 8 this implies that $E \perp C |F$ and this statement is rejected so accepting the first two statements is a contradiction to a conditional dependence statement. This means that to avoid logical inconsistencies either $E \perp C |\{B, F\}$
or $C \perp B \mid F$ must be rejected. Since tests with smaller conditioning sets are more accurate it is most reasonable to reject $E \perp C \mid \{B, F\}$.

This is the reason that the algorithm misses the edge $(C, E)$; accepting CI statements that contradict conditional dependence relations. When $E \perp C \mid \{B, D\}$ is accepted the algorithm makes a decision that is a contradiction to a dependence relation in the data. It is accepted that $C \perp B$ and $C \perp D$ so if the underlying distribution of the data has a faithful graph this implies that $C \perp \{B, D\}$. This statement and $E \perp C \mid \{B, D\}$ implies that $E \perp C$ by statement 2 of theorem 8. So $E \perp C \mid \{B, D\}$ should be rejected but the algorithm accepts it and thereby misses an edge needed for representing the dependence structure of the data.

Suppose that the algorithm is modified so that a CI statement is not accepted if it contradicts dependence relations in the data. Then everything will be the same except for when the algorithm is applied on variable $E$. In the modified version of the algorithm also $F$ will enter the neighbour set of $E$ since $E \perp F \mid \{B, C, D\}$ is not accepted because this would contradict the dependence relation $E \not\perp F \mid \{C, D\}$. Then in stage 2 the edge $(C, E)$ will not be removed since $C \perp E \mid S$ is rejected for any $S \subseteq \{B, D, F\}$. In four of these tests the chi squared test can not reject the CI statement, but in all these cases accepting the statement would contradict dependence relations. This can be shown in a similar way as above. So the modified algorithm returns the following skeleton.

![Graph obtained by modified MMPC on the WAM data](image)

Figure 4.7: Graph obtained by modified MMPC on the WAM data

To construct the essential graph, check for possible immoralities ( $(B, E, C)$, $(B, D, F)$, $(C, E, D)$, $(C, F, D)$, $(E, C, F)$, $(E, D, F)$ ).

$(B, C)$ is removed since $B \perp C \mid \emptyset$ so $(B, E, C)$ is an immorality.

$(B, F)$ is removed since $B \perp F \mid \{D, E\}$ so $(B, D, F)$ is not an immorality.

$(C, D)$ is removed since $C \perp D \mid \emptyset$ so $(C, E, D)$ and $(C, F, D)$ are immoralities.

$(E, F)$ is removed since $E \perp F \mid \emptyset$ so $(E, C, F)$ and $(E, D, F)$ are immoralities.

This shows that the set of CI-statements derived is not faithful since these immoralities are contradictory. If there exists a faithful graph the ordering of the variables does not affect the result. In this case the immoralities in the obtained graph depends on the ordering of the variables. In this project the ordering used was $A, B, C, D, E, F$ so the following essential graph is obtained.
This graph is not faithful since $B \perp F$ and $E \perp F$ but corresponding $d$-separation statements in the graph does not hold. More serious, there is still conditional dependence relations not captured by the graph. Both $C \perp D | B$ and $E \perp F | \{C, D\}$ are rejected at the 5% significance level in the first stage of the algorithm but the corresponding $d$-connection in the graph does not hold so as in the coin tossing example, the lack of faithfulness causing the algorithm to miss dependence relations in the located graph.

4.3 Summary

The aim of the project was firstly to investigate the MMPC algorithm and study its performance when the assumption that there existed a faithful Bayesian network for a distribution failed and secondly to investigate problems that could arise with the method for determining conditional independence relations.

For the first question, the example indicate that while there are situations where lack of faithfulness does not cause serious difficulties, there are situations where the MMPC algorithm performs spectacularly badly when the assumption of faithfulness does not hold. This indicates that, in situations where there are large number of variables, the algorithm should only be used when it is clear a priori that the faithfulness assumption holds.

For the second question, the possible weakness of the testing procedure is a consequence of theorem 8 and the data set under consideration, which was a randomly chosen data set, illustrated that this weakness arises in practice. Furthermore, the faithfulness assumption was not satisfied for the CI statements derived for that data set, indicating that such an assumption may be inappropriate without further information about the variables.
Bibliography


Appendix A

Implementation Details

For this project Matlab was used to implement the MMPC algorithm. This chapter presents and explains the matlab programs used on the women and mathematics example.

A.1 Matlab Programs

The inputs $p$, $j$ and $n$ reoccurs in several of the programs. The matrix $p$ contains observed data from the variables considered. Let $d$ be the number of variables and $N$ the number of possible outcomes for one observation. Then $p$ is a $N \times (d + 1)$ matrix with entries as follows.

- $p_{r,k}$ = The value of variable $k$ in outcome $r$ for $r = 1, ..., N$ and $k = 1, ..., d$.
- $p_{r,d+1}$ = The number of occurrences of outcome $r$ for $r = 1, ..., N$

The input $j$ is a row vector with $d$ entries with the number of possible outcomes for each variable in the distribution and $n$ is the number of observations in the data set. For the cases where variables are inputs in programs, the numbers of the ordering in the $p$ matrix is used. In the following pages the programs and description of them are presented.
count.m

This program first calculates the marginal frequencies for a subset $X$ of the variables. In the resulting table, defined in the same manner as the matrix $p$, the outcomes of the variables in $domain$ are present.

**Input** Two row vectors $X$ and $domain$, where $X$ must be a subset of $domain$, each containing a subset of the variables. The third input is $p$.

**Output** The program calculates a table with the marginal frequencies expanded to specified domain. First output $q$ is a row vector containing only the frequencies and the second output $qfull$ is the full table represented by a matrix.

function[q,qfull] = count(X,domain,p)
q = sortrows(p,X);
[u,last] = unique(q(:,X),’rows’);
k = 1;
for i = last’
    q(k:i,end) = sum(q(k:i,end))*ones(i-k+1,1);
k = i+1;
end
q = unique(q(:,[domain end]),’rows’);
qfull = q;
q = q(:,end);
modify.m

This program is used to modify the test $X \perp Y \mid Z$ when $Z \neq \emptyset$ and the number of observations is less than 5 for at least one outcome $(x, y, z)$.

**Input** The variables $X$ and $Y$. The set of variables $Z$ as a row vector. $p$ is also needed.

**Output** A vector `terms` containing the indices of the terms that is considered in the sum of the $G^2$-statistic. A number $j2$ which is the number of states of $Z$ for which we have 5 or more observations of $(x, y, z)$ for all $(x, y)$.

```matlab
function [terms, j2] = modify(X, Y, Z, p)

[q2, q] = count([X Y Z], [X Y Z], p);
q = sortrows(q, 3:(2+length(Z)));
q = [q ones(length(q), 1)];
[u, last] = unique(q(:, 3:(2+length(Z))), 'rows');
k = 1;
j2 = 0;
for i = last'
    if any(q(k:i, (end-1)) < 5)
        q(k:i, end) = 0;
    else
        j2 = j2+1;
        k = i+1;
    end
end
q = sortrows(q, 1:length([X Y Z]));
terms = find(q(:, end));
```

isci.m

This program tests if two variables $X$ and $Y$ are conditionally independent given another set of variables $Z$.

**Input** The variables $X$ and $Y$. The set of variables $Z$ as a row vector. If $Z$ is the empty set, then let $Z$ be an empty vector. $p$, $j$ and $n$.

**Output** $CI$ is 1 if the statement is true, 0 otherwise. $G2$ is the value of the $G^2$-statistic. The number of degrees of freedom corresponding to the $\chi^2$-distribution of $G^2$ under $H_0$ is given by $df$. The final output is the p-value of the test.

```matlab
function [CI, G2, df, pvalue] = isci(X, Y, Z, p, j, n)

if isempty(Z)
    k1 = count([X Y], [X Y], p);
    if any(k1 < 5)
        CI = 0;
        return
    end
    k2 = count(X, [X Y], p);
    k3 = count(Y, [X Y], p);
    G2 = 2*sum( k1.*log(n*k1 ./ (k2.*k3)) );
    df = (j(X) - 1)*(j(Y) - 1);
else
    k1 = count([X Y Z], [X Y Z], p);
    k2 = count(Z, [X Y Z], p);
    k3 = count([X Z], [X Y Z], p);
    k4 = count([Y Z], [X Y Z], p);
    if any(k1 < 5)
        [terms, j2] = modify(X, Y, Z, p);
        if j2 == 0
            CI = 0;
            return
        end
        k1 = k1(terms);
        k2 = k2(terms);
        k3 = k3(terms);
        k4 = k4(terms);
        df = (j(X) - 1)*(j(Y) - 1)*j2;
    else
        df = (j(X) - 1)*(j(Y) - 1)*prod(j(Z));
    end
    G2 = 2*sum( k1.*log((k1.*k2) ./ (k3.*k4)) );
end

pvalue = 1-chi2cdf(G2, df);
if pvalue < 0.05
    CI = 0;
else
    CI = 1;
end
```
existset.m

Checks if $X$ and $Y$ are conditionally independent given some subset of $S$.

**Input** The variables $X$ and $Y$. The set of variables $S$ as a row vector, where $S$ is an empty vector if $S$ is the empty set. $p$, $j$ and $n$.

**Output** The output `exist` is 1 if such set exists, otherwise 0. The second output `set` is a row vector with the corresponding set if such exist, otherwise `set` is the text string 'No set'. $G2$, $df$ and $pvalue$ are information from isci.m for the test where the desired set was found. If no such set is found, all these outputs are the text string '—'.

```matlab
function [exist, set, G2, df, pvalue] = existset(X, Y, S, p, j, n)
exist = 0;

[CI, G2, df, pvalue] = isci(X, Y, [], p, j, n);
if CI
    exist = 1;
    set = [];
    return
elseif isempty(S)
    set = 'No set';
    G2 = '---';
    df = '---';
    pvalue = '---';
    return
end
for i = 1:length(S)
    subsets = nchoosek(S, i)';
    for s = subsets
        [CI, G2, df, pvalue] = isci(X, Y, s', p, j, n);
        if CI
            exist = 1;
            set = s';
            return
        end
    end
end
set = 'No set';
G2 = '---';
df = '---';
pvalue = '---';
```


mmpc.m

This is the main program. It locates the skeleton of a Bayesian network for \( p \) using the MMPC algorithm.

**Input** \( p, j \) and \( n \) as described in the beginning of this section.

**Output** An undirected graph represented as a sparse matrix \( E \). The matrix has entry 1 at row \( r \) and column \( c \) if \( r < c \) and there is an edge between the nodes \( r \) and \( c \). All other entries are zeros. *Stage 1* is a cell array where information from each step of the algorithm in stage 1 is stored; the test considered, value of test statistic, degrees of freedom and p-value. *Stage 2* is corresponding for stage 2, but here each row corresponds to an attempt of finding a conditioning set that makes two variables independent. *Stage 3* is a sparse matrix containing the edges removed in stage 3.

```matlab
function [E, Stage1, Stage2, Stage3] = mmpc(p, j, n)

d = length(j);
E = sparse(1,1,d,d);
Stage2 = {};
k = 1;

% Stage 1
for T = 1:d
    for i = setdiff(1:d,T)
        Z = find(E(T,:));
        [CI,G2,df,pvalue] = isci(T,i,Z,p,j,n);
        if CI
            Stage1(k,:) = {T i Z G2 df pvalue 'Accept'};
        else
            E(T,i) = 1;
            Stage1(k,:) = {T i Z G2 df pvalue 'Reject'};
        end
        k = k+1;
    end
end

% Stage 2
for T = 1:d
    for i = find(E(T,:))
        S = setdiff(find(E(T,:)),i);
        [exist,set,G2,df,pvalue] = existset(T,i,S,p,j,n);
        if exist
            E(T,i) = 0;
            Stage2(k,:) = {T i S set G2 df pvalue};
            k = k+1;
        end
    end
end

% Stage 3
[r1,c1] = find( (tril(E)'+triu(E)) == 1);
[r2,c2] = find( (tril(E)'+triu(E)) == 2);
Stage3 = sparse(r1,c1,1,d,d);
E = sparse(r2,c2,1,d,d);
```
Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a period of 25 years from the date of publication barring exceptional circumstances. The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

Upphovsrätt

Detta dokument hålls tillgängligt på Internet - eller dess framtidiga ersättare - under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår. Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art. Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådan sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart. För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

© 2010, Mikael Petersson