Moral Markov Blankets: An Investigation of Some Properties and Value for Machine Learning

Yang Li (Kelvin)

Supervisor: Dr. Kevin Korb
Dr. Lloyd Allison

Faculty of Information Technology
Monash University

This thesis is submitted for the degree of

Doctor of Philosophy

Clayton Campus May 2020

For my grandparents, my parents, my wife and my sons.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text.

Yang Li (Kelvin)

May 2020

Preface

The original research project was to scale up causal Bayesian network structure learning using the minimum message length principle and the Local-to-Global paradigm via Markov blankets. Due to my background in graph theory and enthusiasm in computational complexity theory, only parts of the original project were completed together with solutions to some interesting and challenging problems that I discovered on my own along the PhD journey.

The preprint that proposed polynomial time algorithms for checking morality for graphs with maximum degree at most 4 can be viewed at https://arxiv.org/pdf/1903.01707.pdf. The algorithms were implemented in an R package with some additional functions. It can be downloaded from https://github.com/kelvinyangli/wrsgraph. The Markov blanket discovery algorithms were implemented in another R package and can be downloaded from https://github.com/kelvinyangli/mbmml.

After completing a mathematics degree from the University of Melbourne in 2011, I struggled for almost two years between doing an ordinary nine to five office job and pursuing my dream career as a researcher in mathematics. I must confess that neither of these two options was easy to me at that time because I was, am and always will be an average person with relatively little interest in getting high GPA, but more interest in seeking knowledge and challenging myself. Until the mid of 2013, I met my current PhD supervisor Kevin Korb, who kindly offered to take me as a PhD student. After some difficulties of being officially admitted by Monash University, I started my PhD journey in September 2014 at the Faculty of Information Technology. A couple of weeks later, Kevin introduced me to Lloyd Allison who then became my second supervisor and continued offering me hours of discussions every week, especially during the tough times in life.

I admit that I had some dis-satisfactions with both of my supervisors because of their limited support in mathematical and programming tasks, unconcern for top conference/journal publications and short contact hours due to other businesses. Instead of proving or coding for me, they offered me suggestions and methodologies to guide me through the morass of research. For quite some time, I was upset about this supervising style. But today, I realised that it is because of the way they mentored me, I have become an independent researcher with academic integrity. Most importantly, I have become a better person, a person who understands the meaning of education. To quote a part of the commencement speech given by David Foster Wallace at Kenyon College on May 21, 2005.

The capital-T Truth is about life BEFORE death.

It is about the real value of a real education, which has almost nothing to do with knowledge, and everything to do with simple awareness; awareness of what is so real and essential, so hidden in plain sight all around us, all the time, that we have to keep reminding ourselves over and over:

"This is water."

"This is water."

Until now, I realised how lucky I am to have Kevin and Lloyd not only as my PhD supervisors, but as my life mentors!

Abstract

Causal discovery automates the learning of causal Bayesian networks from data and has been of active interest from their beginning. With the sourcing of large data sets off the internet, interest in scaling up to very large data sets has grown. One approach to this is to parallelize search using Markov blanket discovery as a first step, followed by a process of combining Markov blankets in a global causal model. This is also known as the Local-to-Global (LGL) paradigm.

Markov blanket was introduced in the 1980s as the smallest informative variable set for a target variable. Since then, it motivated a series of Markov blanket based feature selection as well as causal discovery methods. Most Markov blanket learning methods rely on the constraint or metric-based approaches. The majority of metric-based methods learn a regional structure around a target node then read off its Markov blanket. The problem of applying these regional structures to scale up structure learning is that they are likely to include mistakes of omission (false negatives) and commission (false positives), which then lead to discrepancies among neighbouring structures. Hence, it not only wastes computational resources on learning the regional structures, but resources on resolving these discrepancies in future steps.

To overcome this issue, the thesis proposes a framework to learn Markov blanket as the best variable subset that predicts a target. The framework employs the Minimum Message Length (MML) principle to select an optimal variable subset because MML, a Bayesian model selection metric that obeys *Occam's Razor*, proved its advantages in causal discovery in past literature. To calculate the message length, the framework assumes a regional structure to enable encoding the Markov blanket candidates. In particular, the regional structure can either be a conditional probability table, naive Bayes, or a Markov blanket polytree model. Experimental results suggest that the proposed framework is competitive with some state-of-the-art Markov blanket learners. In particular, the framework using the conditional probability table model shows superior accuracy under moderate sample size.

Next, this thesis addresses the importance of moral Markov blanket and explores its properties, some of which may have a potential effect on scaling up structure learning. Moralization is the process of connecting non-adjacent parents for each node in the Bayesian network and dropping all edge directions. The resulting moral graph can be alternatively obtained by connecting the Markov blanket candidates as each target node's neighbours. Hence, being moral is a necessary condition for a set of Markov blankets over all variables to be legitimate. We give rigorous definitions of moral graphs in order to explore the properties of these graphs and prove that morality can be decided in linear and quadratic time for graphs of restricted maximum degree 3 and 4.

The last part of the thesis studies the application of Markov blanket (with and without morality enforced) in causal discovery. Most of the existing LGL learners resolve discrepancies between overlapping regional structures in ad hoc ways which are hard to generalize to new methods. Through experiments, we claim that probabilistic Markov blanket priors, being encoded as *approximated moral priors*, can help the algorithm Causal MML (CaMML) to produce better results when speeding up CaMML by reducing its searching and sampling iterations in certain cases. To improve CaMML's overall performance, however, the learned Markov blankets need to be more accurate than a threshold, which as far as we know no current Markov blanket method can reach.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Causal discovery aims to learn causal Bayesian networks by using information about conditional dependencies between sets of variables gleaned from sample data. The research behind it is coterminous with research on Bayesian networks themselves, beginning at least with Glymour et al. [39], spurred on by the difficulty and cost of eliciting Bayesian networks from experts. The techniques behind causal discovery have since then become more varied and effective, expanding from the "constraint-based" learning of early efforts, which examine conditional dependencies in isolation, to "metric-based" learning, which apply Bayesian (or similar) metrics to causal models and data sets as a whole. Meanwhile, however, the challenge has itself increased manyfold, in particular through the additional challenge of "big data" driven by the expansion of the internet.

This challenge is also aggravated by the fact that causal discovery itself is NP-hard [15], forcing the use of heuristics when learning whole causal models. An alternative to learning causal models globally is the Local-to-Global (LGL) paradigm that first finds regional structures within subsets of variables then unifies them into a global structure.

A promising approach to a LGL technique is to parallelize search using Markov Blanket discovery as the first step, discovering Markov blankets centred around each variable, and then gluing them all together.

1.1 Motivation

Throughout this thesis, by *(global) structure* we mean a Bayesian network structure (i.e., Directed Acyclic Graph (DAG)). By a *regional structure* we mean a network structure over a subset of variables. (Note that some authors use local structure to mean regional structure.)[1] As Markov blankets are sets of variables, regional structures imply (include) Markov blankets, but include arcs and their orientations. A typical LGL structure learner consists of the following main steps.

1. Given a data set over n variables,

    (a) use a machine learning algorithm to split the set of n variables into variable subsets (that may or may not overlap),

    (b) use a structure learner to infer an optimal regional structure over each subset of variables.

2. Merge all n regional structures to get a partially directed global structure. If conflicts exist between neighbouring regional structures, resolve them in some way.

3. Apply a structure learner, starting from the regional structures, to obtain an optimal global structure that is either a fully directed Bayesian network or a DAG pattern (or pattern) that uniquely represents a Markov equivalence class of DAGs.

Sometimes Steps 1(a) and 1(b) are treated as one by learning an optimal regional structure around a target variable, but limit the variables to being within the neighbourhood, e.g., the Max-Min Hill-Climbing (MMHC) algorithm in [89] or Markov blanket of the target, e.g., the Score-based Local Learning (SLL) algorithm in [60].

There are, however, two potential issues with this standard approach. First, although regional structures are often over relatively small subsets of variables, there is no guarantee

[1] We use "regional structure" in view of the fact that "local structure" is commonly used to refer to the dependencies between parameters within the conditional probability distribution of each individual variable.

in general that these smaller regional structures can be learned with high accuracy. False discoveries could then lead to edge existence or direction conflicts among neighbouring regional structures, which eventually need to be resolved at some stage. In addition, learning regional structures could be a waste of computational resources if this step is not essential to subsequent steps. Second, the way these conflicts are resolved in Step 2 is not generalizable. Different learners often have specifically designed properties for resolving conflicts according to the nature of the learner, e.g., high precision but low recall or vice versa. Therefore, it is important to develop a LGL framework that skips learning regional structures (i.e., Step 1(b)) but still produces a global undirected or partially directed structure for subsequent steps. More importantly, the framework should encode Markov blankets as “prior” information in a probabilistic way that can be incorporated by a Bayesian structure learning algorithm in order to help the scalability of such a learner while still keep a reasonable accuracy.

1.2 Research Questions and Contributions

Although the underlying motivation of this thesis is to scale up Bayesian network structure learning using the LGL strategy, the research did not proceed far enough to demonstrate the effectiveness of such a technique. The research questions answered in this thesis are prerequisites to that goal, but valuable to answer in their own right:

Research Question 1: Can Markov blankets be learned effectively as purely variable subsets, without learning the regional structures within them?

Research Question 2: What is the connection between Markov blanket consistency and graph morality?

Research Question 3: How to efficiently check graph morality?

Research Question 4: How to use Markov blankets as “priors” in Bayesian structure learners?

The contributions of this thesis are stated below with respect to each of the above research questions:

1. Developed a Markov blanket discovery framework that uses the Minimum Message Length (MML) principle and a hypothetical regional structure, including the conditional probability table, Naive Bayes and Markov blanket polytree models. This work has been written in a journal paper that is ready for submission.

2. Proved the equivalence between Markov blanket consistency and graph morality (Research Question 2). Developed linear and quadratic time algorithms for checking morality for graphs with maximum degree 3 and 4 (Research Question 3). Demonstrated that enforcing morality (heuristically) has the potential to increase Markov blanket discovery accuracy. This work has been written in a paper that is ready for submission. The preprint is available at [51].

3. Used Markov blankets as approximate moral priors for the Bayesian structure learner Causal MML (CaMML) [61] and demonstrated their potential for increasing CaMML's scalibility (Research Question 4).

1.3 Structure of the Thesis

The main content of the thesis is structured in six chapters to answer the four research questions mentioned in the previous section.

Chapter 1 introduces the general background of Bayesian network structure learning, research questions and accomplishments and the structure of this thesis.

Chapter 2 starts by explaining some of the strategies in Markov blanket discovery, its connection with graph morality and its application in scaling up structure learning. The chapter draws attention to the specific problems that this thesis addresses and tries to solve.

Chapter 3 starts by giving the definitions (in the fields of graph theory, Bayesian network and computational complexity theory)[2] that are needed for the thesis. The concept of moral graph is formalized and its equivalence to related properties is proved. The motivations of this chapter are to prove the equivalence of Markov blanket consistency and graph morality, and to propose linear and quadratic time algorithms for checking morality of graphs with maximum degrees 3 and 4. The remainder of this chapter brings closure to the problem of deciding morality by proving the problem remains NP-complete for graphs with higher maximum degree, which then leads to the NP-hardness of the minimum and minimal moralization problems.

Chapter 4 focuses on developing a Markov blanket discovery algorithm using the MML principle. The focus here is to learn Markov blankets as variable subsets, without having to learn the regional structures within the Markov blankets. The work in this chapter assumes three different regional structures - the conditional probability table, Naive Bayes and Markov blanket polytree - to encode a Markov blanket without relying on any of these being strictly correct.

Chapter 5 covers the experimental work that has been done for this thesis. The first section includes the tests on both real and artificial data sets of the Markov blanket discovery algorithms proposed in the previous chapter. The second section experimentally justifies the potential of enforcing morality on learned Markov blankets in order to increase the accuracy of a Markov blanket learner. The last section studies how Markov blankets can be used as probabilistic priors and its potential of scaling up structure learners. In this thesis, the focus is on the CaMML algorithm.

Chapter 6 concludes the work that has been done in the main parts of the thesis and the theoretical and practical results achieved.

In addition to the main chapters described above, the thesis also addresses respectively the NP-hardness of the Markov blanket polytree problem and the optimal triangulation of

[2] See also Glossary and Acronyms at the end of the thesis.

moral graphs. Appendix A proves that the optimal Markov blanket polytree model cannot be approximated within a constant factor to the optimum. Appendix B addresses an incomplete proof of the NP-hardness of optimal triangulation.[3]

[3] The proofs state in this appendix, however, are overly complicated and be simplified to a short argument, which was pointed out by an anonymous reviewer.

Chapter 2

Literature Review

The purpose of this chapter is to review the work that has been done on learning Markov blankets and its applications in machine learning. The chapter starts by introducing the concept of Markov blankets, the motivations for using this notion in Bayesian network structure learning and feature selection, and the major approaches for learning Markov blankets from observational data. It follows by explaining what it means for a set of Markov blankets to be moral and why morality should be considered when learning Markov blankets, especially for structure learning. The chapter ends with a brief introduction to Bayesian network structure learning and how Markov blankets are brought into the stage for scaling up that problem.

2.1 Markov Blanket Discovery

The Markov blanket of a (target) variable is the smallest subset of variables conditioning on which renders the target independent of the remaining variables of a model.[1] This implies that other variables beyond the blanket carry no additional information about the target. In a faithful Bayesian network, a variable's Markov blanket contains its parents, its children and

[1] Originally, this is how Pearl [66] defined "Markov boundaries", but the literature has largely migrated to using "Markov blankets" to this minimalist sense.

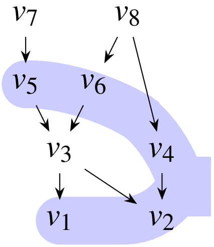


Fig. 2.1 The MB of v_3 in this faithful BN is $\{v_5, v_6, v_1, v_2, v_4\}$.

its spouses (i.e., the children's other parents, see Figure 2.1). However, knowing the variables of the Markov blanket does not tell us specifically *how* the target variable is connected to the Markov blanket variables. The above definition of Markov blankets entails that in principle Markov blankets are the optimal feature subsets for prediction. Hence, it motivated a series of Markov blanket based feature selection work such as [45, 20, 11, 83, 53], etc. Furthermore, since the Markov blanket "separates" the target from all other variables in the model, Markov blankets are natural choices for "breaking" a Bayesian network down to multiple pieces, so that structure learning tasks can be conducted on all pieces simultaneously. Some of the work in this direction includes [57, 4, 5, 60, 68, 34].

The approaches to learning Markov blankets from data can be divided into two major categories, i.e., constraint-based and metric-based.[2] A natural way of learning Markov blankets is suggested by their definition in terms of conditional independence: testing dependencies between a target and everything else given each possible subset of other variables, looking for the minimum subset yielding zero dependency. Conditional independencies are often tested by statistical hypothesis tests, such as the G-test or χ^2-test. An exhaustive search of this kind would, of course, be exponential in the size of the network. But we can try a heuristic search instead. This was first done by Margaritis and Thrun [55]. Their work, the Growth-Shrink (GS) algorithm, laid the foundation for a constraint-based Markov blanket discovery that typically consists of an addition and a deletion phase as shown in Algorithm 1. We use

[2] These two approaches are essentially the same as the two major approaches for Bayesian network structure learning that will be reviewed at the end of this chapter.

$X \not\perp\!\!\!\perp_P Y \mid S$ and $X \perp\!\!\!\perp_P Y \mid S$ to denote that when conditioning on the set of variables S, the random variables X and Y are dependent and independent, respectively, w.r.t. the underlying distribution P respectively. The subscript P can be dropped if the underlying distribution P is clear in the context.

Algorithm 1: The GS algorithm by Margaritis and Thrun [55]

```
Input  : A data set $D$ over a set of $n$ variables $V = (X_1, \ldots, X_n)$. A target variable $X_i$
Output : A candidate Markov blanket $S$ for $X_i$
$S = \emptyset$;
while exists $X_j \in V \setminus \{X_i\}$ such that $X_i \not\perp\!\!\!\perp X_j \mid S$ do
    $S = S \cup \{X_j\}$ ;                          // addition phase
end
while exists $X_j \in S$ such that $X_i \perp\!\!\!\perp X_j \mid S \setminus \{X_j\}$ do
    $S = S \setminus \{X_j\}$ ;                     // deletion phase
end
```

To reduce the chance of adding false positives into the potential Markov blanket with this heuristic approach, Tsamardinos et al. [88] added those variables to the candidate Markov blanket having the strongest dependencies with the target in advance. The strength of dependencies can be measured quickly by information-theoretical measures such as mutual information. The intuition behind the re-ordering in each of the addition steps is that false positive candidates are less likely to appear in variables who are strongly dependent to the target, and hence reducing the chance of adding more false positives in the subsequent steps. The proposed Incremental Association Markov Blanket (IAMB) algorithm has a worse computational complexity, $O(n^2)$, but it is usually much faster in practice.[3] Since then, the IAMB algorithm has attracted much attention in attempts at improving its speed and accuracy from the algorithmic perspective such as the works from Yaramakala and Margaritis [99] and Zhang et al. [101].

While employing the same statistical tests and heuristics, another slightly different strategy that learns the direct neighbours and spouses separately has proven superior, and

[3] The computational complexity for constraint-based methods are often measured in terms of the number of conditional independence tests that need to be conducted.

hence has been widely adopted in later constraint-based methods, including [3, 67, 31, 4, 5, 25, 52, 33], etc. The difference from the previous strategy is that the growing phase is done in two sub-steps, where the first sub-step learns the neighbours of a target variable and the following sub-step learns the neighbours of each variable that is in the neighbours of the target, i.e., the distance-two neighbours of the target variable. Algorithm 2 is one of the first algorithms of this kind, namely the Max-Min Markov Blanket (MMMB) algorithm developed by Tsamardinos et al. [87]. It relies on the Max-Min Parents and Children (MMPC) algorithm [87] to find the neighbours of a variable. The detail of MMPC is not presented here as it has similar ingredients to the GS algorithm but for neighbourhood discovery. It takes two inputs including a data set D and a target variable X_i and outputs a potential set of neighbours $MMPC(D, X_i)$ for X_i.

Algorithm 2: The MMMB algorithm by Tsamardinos et al. [87]

```
   Input  : A data set $D$ over a set of $n$ variables $V = (X_1, \dots, X_n)$. A target variable $X_i$
   Output : A potential set of Markov blanket $S$ for $X_i$
 1 $P_i = MMPC(D, X_i)$ ;                                    // addition phase
 2 $S = P_i \bigcup_{X_j \in P_i} MMPC(D, X_j)$;
 3 for each $X_j \in S \setminus P_i$ do
 4     Find $S'$ s.t. $X_j \perp\!\!\!\perp X_i \mid S'$ ;            // deletion phase
 5     for each $X_k \in P_i$ do
 6         if $X_j \not\perp\!\!\!\perp X_i \mid S' \cup X_k$ then
 7             Mark $X_j$ ;
 8         end
 9     end
10     Remove $X_j$ from $S$ unless it is marked ;
11 end
```

As can be seen in the pseudo-code, the MMMB algorithm looks at the dependencies with the target at a distance of one and two neighbours separately. Distance-two neighbours are then taken to a filtering process (lines 3-11) to remove false positives.

Note that some of the distance-two neighbour constraint-based methods utilize the symmetry condition to filter out false positives before the deletion phase, e.g., the Parents and

Children based Markov Blanket (PCMB) algorithm in [67]. The symmetry condition states that a random variable X is in the Markov blanket of a random variable Y if and only if Y is in the Markov blanket of X. There are, however, methods that relax this condition to improve the overall structural accuracy measured by a combined precision and recall such as [25] or the running time such as [33]. In general, the constraint-based approach has an advantage in scaling up to high-dimensional data sets. Recent work from Liu and Liu [53] generalized two existing constraint-based methods to Markov blanket discovery for multi-target variables. The constraint-based methods, however, suffer a lack of robustness under small samples and are sensitive to the exact settings used in the statistical tests.

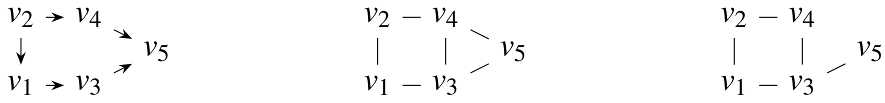
Metric-based learners, having proven themselves highly effective in general causal discovery, have subsequently been applied to Markov blanket discovery. There are several different features which distinguish between metric-based methods, such as the type of searches or metrics used, the search space, or whether the learning is exact or approximate. To illustrate an important motivation of this project, we group these methods into two categories by whether or not the learning process is target-variable oriented. The non-target-oriented methods search for the best regional structure around a target w.r.t. a metric function. Here, a pre-determined metric is calculated over the regional structure. A good example of this type is the Score-based Local Learning (SLL) algorithm proposed by Niinimaki and Parviainen [60]. In contrast to metric learning of full Bayesian networks, the search space of SLL is restricted to regional structures around a target variable without regard for unrelated adjacencies. A few methods of this type were published earlier than SLL, such as the works by Cooper et al. [20] and Madden [54] that are built upon the Bayesian network learner K2 [19], and the work from Acid et al. [1] that searches through a more restricted space than the space of all valid regional structures. It is worth mentioning that SLL is an exact algorithm that will find the optimal regional structure by using a dynamic programming exact Bayesian network learner developed by Silander and Myllymäki [79]. Given that most real models are sparse, Markov blankets tend to be small, allowing exact algorithms for learning small Bayesian networks to be applied to find optimal regional structures independently. Gao and Ji [33] recently reduced the computational complexity of SLL by a factor of n (i.e., the number of variables) by removing the enforcement of symmetry in SLL.

The target-oriented learners embedded potential Markov blanket candidates into fixed machine learning classification models. An early work from Frey et al. [29] built decision tree models to estimate Markov blanket candidates. The algorithm by Li et al. [49] learned Markov blanket candidates of a target variable by using linear models trained with the Least Absolute Shrinkage and Selection Operator (LASSO) method [85] for dimensionality reduction. More recently, Strobl and Visweswaran [83] proposed to use ridge regularized linear models to discover Markov blankets. This method has a memory limitation due to the necessity of storing large covariance matrices and is only valid for non-singular covariance matrices. Here, singularity is caused by colinear variables and hence is avoided in an improved method by Yan et al. [97], who added small random noise to entries of covariance matrices.

Some of the target-oriented metric-based methods mentioned above have limitations because of the restricted classification models used or the amount of calculation needed for high-dimensional data sets. There are other simple classification models that could potentially avoid these issues. In particular, what we are interested in is how simple models, such as the Conditional Probability Table (CPT), naive Bayes and polytree, perform in Markov blanket discovery when using the MML metric. This is studied thoroughly in Chapter 4.

2.2 Moral Graphs

One way to conduct an exact inference on a given Bayesian network is to transform it into a tree structure, called a *junction tree*, and conduct inference on that instead. The process of obtaining a junction tree from a Bayesian network consists of moralization, triangulation and tree decomposition. *Moralization* was introduced by Lauritzen and Spiegelhalter [47] as connecting non-adjacent parents for each node in the Bayesian network and dropping all edge directions. Any spouse of a vertex that is neither a parent nor a child of the vertex must be connected to the vertex during the moralization process. This implies that the Markov blanket of each vertex in a DAG becomes the vertex's neighbourhood in the moral graph. Figure 2.2 contains a DAG, its moral graph and a non-moral graph of this DAG.



(a) A DAG. (b) Its moral graph. (c) A non-moral graph.

Fig. 2.2 An example of a DAG, its moral graph and a non-moral graph.

Given a faithful Bayesian network, it follows that the Markov blankets must be symmetric and consistent. *Symmetry* states that a variable v_i is in the Markov blanket of another variable v_j if and only if v_j is in the Markov blanket of v_i. This is a consequence of the graphical interpretation of Markov blankets. *Consistency* is defined as: there exists at least one DAG that admits the set of Markov blankets over all variables as its set of Markov blankets. The definition of moral graph implies that each node's Markov blanket becomes its neighbourhood. Given a data set over n variables, the set of all learned Markov blankets together form an undirected graph by connecting the Markov blanket of each node as its neighbourhood. Hence, saying a set of Markov blankets is consistent with a DAG is equivalent to saying the undirected graph obtained is moral.

A family of learned Markov blankets, if not read off from a DAG, does not imply any specific relations among its variables. This does not stop symmetry being quickly checked and enforced. Previous work by Tsamardinos et al. [89], Peña et al. [67], Niinimaki and Parviainen [60], Aliferis et al. [4, 5] enforced symmetry arbitrarily by taking the union or intersection[4] of Markov blankets. It is, however, non-trivial to check the consistency

[4] If v_i is in the Markov blanket v_j but not vice versa, taking the union will add v_j into the Markov blanket of v_i whilst taking the intersection will delete v_i from the Markov blanket of v_j.

criterion. Lacking consistency with a DAG, one can be certain that the Markov blankets are incorrect, under the assumption that the generating model is a faithful Bayesian network with no hidden variables.[5] Hence, the global structure obtained from these Markov blankets, regardless of how this is done, can never be oriented to a DAG that admits the same Markov blankets.[6] Until now, the machine learning literature has not paid attention to Markov blanket consistency. Therefore, one of the motivations of this project is to address the importance of this property and its potential effect on structure learning and feature selection algorithms that use the Markov blanket approach.

2.3 Bayesian Network Structure Learning

We finish this chapter by giving a brief introduction to Bayesian network structure learning. A lot of the results for Markov blanket discovery mentioned in the first section come from general structure learning. Furthermore, an important underlying motivation of Markov blanket discovery is to scale up structure learning to high-dimension problems.

Bayesian network structure learning is the process of finding an optimal model structure for describing a data set. Here, optimality refers to a statistical metric or a set of statistical hypothesis tests calculated from the given data set. The time complexity of the learning problem is known to be at least super-polynomial from results published since the 1990s. Early work on the time complexity of general structure learning in Chickering et al. [15] proved that learning an optimal Bayesian network structure with maximum fan-in at least 2 is NP-hard, with respect to the Bayesian Dirichlet equivalent (BDe) scoring function (or

[5] We conducted tests of Markov blanket consistency for three learners on 10 random BNs with 30 binary variables and maximum fan-in 5. Each Bayesian network was used to generate 10 data sets of samples of sizes 100, 1000 and 5000. On average, SLL-MB [60] produced Markov blankets with $12, 0, 0$ percent consistency for the three sample sizes respectively. PCMB [67] had $92, 0, 0$ and IAMB [89] had $7, 0, 0$. The reason PCMB had high consistency for samples of size 100 is that it learned very few Markov blanket candidates, so the graphs were almost as sparse as trees, which always have consistent DAGs.

[6] Consistency does not apply to the skeleton (of a DAG) given by a neighbourhood learning method such as MMHC [89] and SLL+G [60]

anything equivalent) and a finite data set. This work was extended in Chickering, Heckerman, and Meek [16] to the conclusion that even with infinite data and the same scoring function, it is still NP-hard to learn the optimal structure with max fan-in at least 3.[7] In addition to the hardness of learning a general structure, some restricted structures are also NP-hard to learn. For instance, the work of Dasgupta [23] shows that the optimal polytree with max fan-in 2 cannot be approximated to within a constant factor of the optimum in polynomial time. This is in contrast with the polynomial time algorithm that learns an optimal polytree with max fan-in 1 (a.k.a. a Chow-Liu tree) as proved by Chow and Liu [17]. Inspired by the work of Dasgupta [23], Appendix A.1 looks into approximating a more restricted model family, namely Markov blanket polytree. By a similar construction, it is proved in the appendix that an optimal Markov blanket polytree with max fan-in 3 cannot be approximated in polynomial time to within a constant factor of the optimum Markov blanket polytree.

Regardless of the computational complexity of learning the optimal Bayesian network structure, an automated learning process is often preferred in practice due to the challenge of manually building a structure. For this reason, various heuristic techniques have been developed to learn good but possibly sub-optimal structures within a reasonable time limit. As in most of the structure learning literature, we divide the most popular methods into two categories, constraint-based and metric-based learning. As shown in Table 2.1, the structure learning problem can also be separated into learning a global structure at once or learning a batch of regional structures and then combining them into one global structure. The latter strategy has become popular since the 2000s due to the curse of dimensionality impeding the direct learning of global structures.

It follows that one approach to reducing the complexity of learning a full causal Bayesian network is to first learn n Markov blankets and then learn the causal structures within the

[7] Although there is a gap for the problem of max fan-in 2 structure learning with infinite data, the authors conjecture that the problem remains NP-hard [16].

Table 2.1 A brief summary of common structure learning approaches.

	Constraint-based	Metric-based
Global	Fast, piece by piece, lack of robustness, no uncertainty, e.g., PC	Slow, consistent metrics, structures can be ranked, e.g., CaMML, GES, K2
Local-to-global	Not necessary, as global learning usually is quite fast	Neighbourhood-based, Markov blanket-based, MMHC, SLL+C, SLL+G

n Markov blankets independently, which we call the *regional causal structure*, and finally stitching them together. A good review of this approach is presented by Aliferis et al. [4, 5].

2.3.1 Constraint-based Learning

The constraint-based approach relies on conditional independence tests to recover a structure piece by piece. The ideal of this approach, given unlimited time, would be to test whether or not a pair of variables is dependent, conditioning on all possible subsets of other variables. If they are, an edge is added to connect them and its parameters will reflect the direct dependency. Conditional independencies are tested by statistical hypothesis tests, such as the G-test or χ^2-test. And, of course, heuristics are used to by-pass conditioning on all possible subsets. This approach was developed in the early 1990s, utilizing the work of Verma and Pearl [90], in the development of the well-known PC algorithm by Spirtes, Glymour, and Scheines [82]. Since then several variations and improvements on the PC algorithm have been proposed to either reduce the number of conditional independence tests needed or to control the false positive rate when adding arcs. During the past few decades, more than a dozen new constraint-based algorithms have been developed for both general and restricted structure learning [18, 24, 82, 12]. One of the advantages of this type of learner is its efficiency in reconstructing a structure and, hence, scalability to large models. Although there are exponentially many subsets of variables to check when testing the conditional independency between two variables, reasonable constraints can be imposed to reduce the space of possible subsets, and so improve efficiency, e.g., the PC and PC* algorithms of Spirtes, Glymour, and Scheines [82]. A disadvantage of this approach is the lack of robustness of statistical hypothesis tests, in particular under small samples (see Dai et al. [22]). Another problem is the inability of quantifying structure uncertainty, which can be essential when multiple structures are statistically equivalent to each other. Despite the efficiency of constraint-based learners, their reconstruction accuracy is generally worse than metric-based methods. Although it was proved by Cowell [21] that under certain conditions, conditional independence testing is equivalent to local log scores, the latter still attracts much interest for reasons that will be briefly described in the next section.

2.3.2 Metric-based Learning

Compared to the constraint-based approach, a metric-based approach must choose from a large range of alternative ways of operating, including the search space, the search or sampling algorithm, and the statistical metric. Two of the most common search spaces are DAG space [19] and pattern space [13]; the former was proved by Robinson [70] to be super-exponential using a recursive formula and the latter was enumerated by Gillispie and Perlman [38] by a computer program up to 10 nodes (variables). Gillispie and Perlman [38] concluded that the ratio between the number of labelled Markov equivalence classes and DAGs approaches an asymptote of about 0.267, which suggests that even the equivalence class space is super-exponential. Because of this, it is unrealistic to search the entire space for an optimal structure, and hence quick but non-exhaustive search algorithms were proposed in early metric-based work, such as greedy search, Tabu search, genetic algorithms and simulated annealing. Given the increasing power of computers, some efforts have been made to search for the exact optimal structure within the entire search space using dynamic programming [63, 44, 80, 79]. These early works sacrifice scalability for finding the optimal structure, and are only feasible for models contain a few dozen variables. Later we examine the Local-to-Global strategy which helps exact methods scale to hundreds of variables. Note that a Bayesian approach of searching for the optimal structure is to get the posterior distribution over all possible structures. This can be estimated by samples obtained from Markov Chain Monte Carlo (MCMC), such as the Metropolis-Hasting (MH) algorithm. For instance, CaMML [61] uses the MH algorithm to sample through the space of Totally Ordered Models (TOMs) in order to estimate the posterior distribution over TOMs, which is then aggregated to the posterior distribution over DAGs, equivalence classes and MML equivalence classes.

Another important part of the metric-based approach is the statistical metric used by a structure learner. Two desirable properties of a metric are consistency and decomposability. A metric is consistent if it gives the optimal structure the best score in the limit of infinite data. Decomposability is a useful property that allows the score of a structure to be computed as the sum of the score for each node given its parent set. In consequence, local changes to a structure only require updating the affected variables' scores. Perhaps the most intuitive metric to be used is the likelihood function that returns the likelihood of the samples given a model structure. The risk of using maximum likelihood is that it tends to overfit, because the fully connected Bayesian network is capable of imitating every other model of the same size. For this reason, a more sensible solution is to penalize the model complexity by adding some kind of regularization term. Hence, most of the metrics used nowadays contain two parts, where one part is the likelihood of the observed data given the assumed model, and the other part is the model complexity that is used to avoid overfitting. Although these metrics have similar functional formats, they were developed from very different perspectives. Bayesian statisticians (and machine learners) aimed at calculating the posterior probability of a structure given data, which is equivalent to the joint distribution of the structure and data up to a normalizing constant. In this vein, some of the best known metrics are K2 [19], Bayesian

Dirichlet equivalent with uniform parameter prior (BDeu) [10][8] and the more general version BDe [41] that is expressed as

$$P(B_s,D) = P(B_s)\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij}+N_{ij})}\prod_{k=1}^{r_i}\frac{\Gamma(N'_{ijk}+N_{ijk})}{\Gamma(N'_{ijk})}, \tag{2.1}$$

where $P(B_s)$ is the user's prior belief in the structure B_s and the rest of the terms calculate the likelihood of the data with a Dirichlet prior N'_{ijk} on each of the state k of the variable i with parent instantiation j. The value N_{ijk} is the actual count of this combination in the give data set. Another way of finding the optimal structure for a given data set is to find the minimum encoding of the structure and data based on Shannon information theory. For example, the Minimum Message Length (MML) [93, 92] and Minimum Description Length (MDL) [69] are two minimum-coding methods that balance the complexity of a model H with the fit of the model to a given data set D by minimizing:

$$I(H,D) = I(H) + I(D|H). \tag{2.2}$$

The first part $I(H)$ measures the message length for stating a model. The second part $I(D|H)$ measures how well the specified model compresses the given data set; i.e., it is the message length for the data assuming the truth of the model. The aim in MML inference is to find the model having the shortest two part message length, and so maximizing the posterior probability of H given D.[9] Other well known information-theoretic metrics include Akaike Information Criteria (AIC) [2] and Bayesian Information Criteria (BIC) [76], which penalise model complexity proportional to the number of parameters.

Beyond the two main approaches of constraint-based and metric-based learning, another recently developed approach considers structure learning as an optimization problem and has

[8] A recent theoretical review of the BDeu metric by Suzuki [84] shows that the metric is biased against the Occam's Razor principle in certain conditions due to its uniform parameter prior.

[9] It is similar with MDL, however, assumptions it makes are generally not compatible with considering what is maximized to be probability.

also proved competitive e.g., Scanagatta, de Campos, Corani, and Zaffalon [73], Scanagatta, Corani, de Campos, and Zaffalon [74], Scanagatta, Corani, Zaffalon, Yoo, and Kang [75]. This approach learns structures with bounded treewidth (of the moral graph). It starts by learning a set of candidate parent sets for each variable and ranking them according to a scoring function, such as BDeu. An integer linear programming algorithm is then conducted on these lists of parent sets to find the optimal consistent structure that has the highest total score.

2.3.3 Local-to-Global Strategy

All of the methods described in the previous two sections tackle the structure learning problem as a whole. That is, given a data set with 100 variables, these methods will search through the space of structures containing 100 variables and try to find the best one. This section discusses the so called Local-to-Global (LGL) strategy that helps structure learners to scale up to hundreds of variables (or even millions, as claimed by some, e.g., Ramsey, Glymour, Sanchez-Romero, and Glymour [68]). Note that the constraint-based methods recover a structure piece by piece by running a set of conditional independence tests, so they are readily able to deal with structures with hundreds of variables. The problem, however, is more challenging for metric-based methods due to the super-exponential size of the DAG or pattern space and the relatively high computational costs of calculating a score for each structure. For these reasons, dividing the entire structure into pieces seems promising. The idea of applying the LGL strategy is to divide the entire *n* variables into "reasonable" subsets, then learn regional structures (or substructures) within these subsets. In the end, merge all the regional structures into a global structure as the final output.

There are two different foci on how to divide the variables into subsets. Most so far prefer to split variables into neighbourhoods, one for each variable. A benefit of doing so is that there is no need to learn the regional structure within a neighbourhood, since all neighbours are connected to the target variable directly, e.g., Tsamardinos, Brown, and Aliferis [89], Niinimaki and Parviainen [60]. The other focus is to divide variables into Markov blankets, one for each variable. The reason for using Markov blankets is that it is the smallest subset of variables, conditioning on which the target becomes independent of all other variables. A difficulty of this approach is that it is not clear how variables in the Markov blanket ought to be connected, unless the regional structure is also learned along with the Markov blanket. Although existing structure learners are able to learn regional structures, the outputs are very likely to contain conflicting edges, which again need to be dealt with carefully when moving to the global structure; see, e.g., Nägele, Dejori, and Stetter [57], Niinimaki and Parviainen [60], Ramsey, Glymour, Sanchez-Romero, and Glymour [68]. Based upon the exact learner from Niinimaki and Parviainen [60], recently Gao et al. [35] developed a LGL Bayesian network structure learning algorithm that saves both running time and memory by gradually expanding a regional structure to a global structure. Gao and Wei [34] further improved the running time of SLL-based learners by parallel learning of regional structures using multi agents.

While existing LGL methods are continuously being improved, their deterministic ways of resolving conflict edges between neighbouring regional structures makes them hard to be generalized when substituting different search or metric components. Therefore, this thesis proposes to encode learned Markov blankets as probabilistic structural priors. This method let a user specifies the confidence of the priors, according to the nature of the search and metric used in the pre-processing steps and allows these priors to be used by structure learners that adopt Bayes theorem.

Chapter 3

Moral graphs

A family of Markov blankets may not be consistent with any Bayesian network due, say, to imperfect data. Inconsistency may have harm Markov blanket-based structure learning. This chapter examines the checking of Markov blanket consistency using graph morality. An alternative concept of moral graph is defined in the next section - Weakly Recursively Simplicial (WRS) - without relying on Bayesian networks. In general, deciding whether or not an undirected graph is moral is NP-complete as proved by Verma and Pearl [91]. However, Sections 3.3.1 and 3.3.2 respectively present linear and quadratic time algorithms for deciding morality for maximum degree 3 and 4 graphs. Also, we will see that the problem remains NP-complete for graphs with maximum degree more than 4, hence leaving no gap between P and NP-complete for this problem.

3.1 Definitions and Notations

This section introduces some of the graph theory and Bayesian network definitions that are directly related to this project. They are adopted from Diestel [26], Neapolitan [58] and Sipser [81] respectively.

3.1.1 Graph Theory

In what follows, we refer to undirected graphs as graphs, and graphs with directions as directed graphs (or Bayesian network structures or Directed Acyclic Graph (DAG)). The graphs considered in this work are *simple*, which means there is at most one edge between a pair of vertices.

Definition 3.1.1. *A **graph** is a pair $G = (V, E)$ comprising a set V of vertices (or nodes) together with a set E of edges (or arcs).*

The vertex set of a graph G is referred to as $V(G)$, its edge set as $E(G)$. We use $u, v \in V(G)$ and $uv \in E(G)$ to respectively denote vertices and an edge of G. If the context is clear, the notations $V(G)$ and $E(G)$ can be simplified to V and G.

Definition 3.1.2. *Two vertices u and v of a graph G are **adjacent**, or **neighbours**, if uv is an edge of G.*

Definition 3.1.3. *If all n vertices of a graph $G = (V, E)$ are pairwise adjacent, then G is a **complete graph** (a.k.a., **clique**) over V denoted by $G = K^n$.*

Definition 3.1.4. *The **neighbourhood** (a.k.a., set of neighbours) of a vertex u in a graph G is the set of vertices adjacent to u in G, denoted by $N_G(u)$. The **closed neighbourhood** of the vertex u denoted by $N_G[u] = \{u\} \cup N_G(u)$ is the union of $\{u\}$ and its neighbourhood.*

Definition 3.1.5. *The **degree** $d_G(u)$ of a vertex u in a graph G is the number of vertices adjacent to u in G.*

Definition 3.1.6. *The **maximum degree** $\Delta(G)$ of a graph G is the maximum degree over all vertices in G.*

Throughout this thesis, we use $A \subseteq B$ to denote that the set A is a subset of the set B and $A \subsetneq B$ to denote that the set A is a *proper subset* of the set B.

Definition 3.1.7. *Let $G = (V,E)$ and $G' = (V',E')$ be two graphs. If $V' \subseteq V$ and $E' \subseteq E$, then G' is a **subgraph** of G (and G is a **supergraph** of G'), written as $G' \subseteq G$.*

Definition 3.1.8. *Let $G = (V,E)$ and $G' = (V',E')$ be two graphs. If G' is a subgraph of G and G' contains all the edges $uv \in E$ with $u,v \in V'$, then G' is an **induced subgraph** of G, written as $G' = G[V']$.*

Definition 3.1.9. *A **path** in a graph G is a non-empty graph $P = (V,E)$ of the form*

$$V = \{v_0, v_1, \dots, v_k\},\ E = \{v_0v_1, v_1v_2, \dots, v_{k-1}v_k\},$$

*where the v_is are all distinct. If $k \geq 3$ and $v_k = v_0$, then the graph $C = (V,E)$ is called a **cycle**.*

Definition 3.1.10. *The **length** of a path is the number of edges in a path.*

Definition 3.1.11. *A non-empty graph G is called **connected** if any two of its vertices are linked by a path in G.*

Definition 3.1.12. *A connected graph is called a **tree** if it contains no cycles.*

Definition 3.1.13. *A **component** of a graph G is a maximal connected subgraph of G.*

Definition 3.1.14. *A **forest** is a graph (not necessary connected), in which each component is a tree.*

Definition 3.1.15. *A **directed graph** $G = (V,E)$ is a graph, in which E is the set of ordered pairs $\overrightarrow{uv}$ of distinct vertices in V.*

Note that the following work involves vertex and edge removal, which could disconnect a graph. To be clear, if a graph is connected, we operate on it directly. Otherwise, operations take place on each of its components separately.

To distinguish from an undirected edge uv, we use $\overrightarrow{uv}$ to denote a directed edge from the vertex u to the vertex v.

Definition 3.1.16. *If there is a directed path from a node u to a node v in a directed graph G, then u is an **ancestor** of v and v is a **descendent** of u. When the directed path between u and v has length 1, we say u is a **parent** of v denoted by $u \in Pa_G(v)$ and v is a **child** of u denoted by $v \in Ch(u)$.*

We sometimes refer the number of parents and children of a vertex as its *fan-in* and *fan-out*. A *source* is a vertex with fan-in zero. A *sink* is a vertex with fan-out zero.

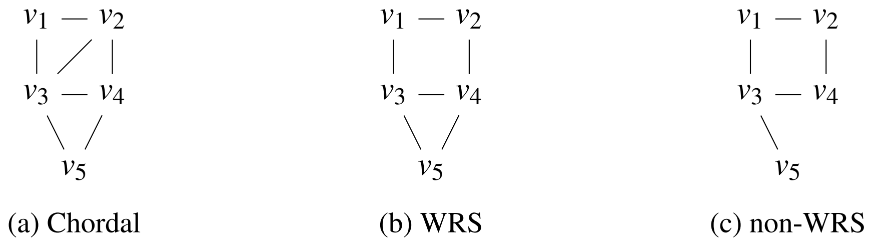
Definition 3.1.17. *A directed graph $G = (V, E)$ is called a **directed acyclic graph (DAG)** if it contains no directed cycles.*

Definition 3.1.18. *The **skeleton** of a directed graph G is the underlying undirected graph of G.*

Definition 3.1.19. *A directed acyclic graph G is a **polytree** if the skeleton of G is a tree.*

Definition 3.1.20. *A **hybrid graph** is a graph that contains both directed and undirected edges.*

Some recursive definitions are given below, but first some shorthand notations. Let V and V' be two sets of vertices of a graph G. For simplicity, if $V' \subseteq V$ then we use the arithmetic subtraction $G - V'$ to denote the induced subgraph $G[V \setminus V']$ over the vertices in V but not V'. If the set $V' = V(H)$ consists of vertices of a subgraph $H \subseteq G$, then we use $G - H$ to emphasize the deletion of the vertices of H. This can be further simplified to $G - u$ if the set $V' = \{u\}$ consists of a single vertex. Similarly, we simplify the notations for edge set. Let E and E' be two sets of edges of a graph G. If $E' \subseteq E$ then we use the arithmetic subtraction $G - E'$ to denote the subgraph $(V, E \setminus E')$ obtained by deleting the edge set E' from the graph G. If $E' \cap E = \emptyset$ then we use arithmetic addition $G + E'$ to denote the supergraph graph $(V, E \cup E')$ obtained by adding the edge set E' to the graph G. The subtraction and addition notations can be further simplified to $G + uv$ or $G - uv$ if the set $E' = \{uv\}$ is made of a single edge.



(a) Chordal (b) WRS (c) non-WRS

Fig. 3.1 Examples of chordal, WRS and non-WRS graphs.

Definition 3.1.21. *A **simplicial vertex** in a graph is a vertex whose neighbours form a complete subgraph.*

Definition 3.1.22. *Let $G = (V,E)$ be a graph. The **deficiency** of a node u in G is $D_G(u) = \{xy \notin E \mid x,y \in N_G(u)\}$ the set of edges that makes the neighbours of u complete.*

By the above two definitions, a vertex u of a graph G is simplicial if and only if $D_G(u) = \emptyset$. That is, the neighbourhood of a simplicial vertex forms a complete subgraph of G.

Example 3.1.1. *The vertex v_5 in each of the three graphs of Fig. 3.1 is simplicial, because its neighbourhood forms a clique. This is equivalent to say that its deficiency $D_G(v_5) = \emptyset$ is the empty set.*

Definition 3.1.23. *A graph G is **recursively simplicial** if it has a simplicial vertex x such that the subgraph $G' = G - x$ is recursively simplicial.*

Definition 3.1.24. *An **ordering** of a graph $G = (V,E)$ with n vertices is a bijection $\alpha : \{1,\dots,n\} \leftrightarrow V$ from the natural numbers to the vertex set of G.*

Without loss of generality, assume the orders are from 1 to the number of vertices of G. For simplicity, we use $\alpha = \{v_1,\dots,v_n\}$ to denote the ordering α such that $\alpha(i) = v_i$ for $i \in [1,n]$. Given an ordering $\alpha = \{v_1,\dots,v_n\}$ of a graph G, the graph can be eliminated recursively by removing one vertex at a time starting from v_1 until the graph is empty. By convention, define $\alpha(0) = \emptyset$. The ordering α is called a *perfect elimination ordering (PEO)* of G if each vertex v_i is simplicial in the subgraph $G - \{\alpha(0),\dots,\alpha(i-1)\}$ for all $i \in [1,n]$.

Next, we introduce chordal graphs. They play an important role in computer science and machine learning because of their unique characteristics and recursive properties stated in Theorem 3.1.1 [32, 72].

Definition 3.1.25. *A graph G is* ***chordal*** *(a.k.a.,* ***triangulated****) if it contains no induced cycles of length* 4 *or more.*

Theorem 3.1.1. *Let G be a graph. The following are equivalent:*

1. *G is chordal.*

2. *G is recursively simplicial.*

3. *G has a perfect elimination ordering.*

Example 3.1.2. *The graph shown in Fig. 3.1a is chordal that is equivalent to having a perfect elimination ordering $\alpha = \{v_5, v_4, v_3, v_2, v_1\}$ as well as being recursively simplicial when removing vertices in the order of α.*

The following definitions generalize the concepts of recursively simplicial and perfect elimination ordering and are proved to be equivalent to moral graphs in the next section.

Definition 3.1.26. *A graph $G = (V,E)$ is* ***weakly recursively simplicial*** *if it has a simplicial vertex x and a subset of edges $E' \subseteq E(G[N(x)])$ between the neighbours of x such that the subgraph $G' = G - x - E'$ is weakly recursively simplicial.*

Note that the subset E' may be empty. Also, the empty graph is weakly recursively simplicial.

Proposition 3.1.1. *If a graph is recursively simplicial then it is weakly recursively simplicial.*

Proof. Recursively simplicial is a special case of weakly recursively simplicial when the subset $E' = \emptyset$ for each simplicial vertex x during the elimination. □

The converse of the proposition is not necessarily true as can be seen from the graph shown in Fig. 3.1b.

Example 3.1.3. *The graph shown in Fig. 3.1b is Weakly Recursively Simplicial (WRS). It can be recursively eliminated by removing vertices and edges in the following order:* $\{\{v_5, v_3v_4\}, v_4, v_3, v_2, v_1\}$. *Each vertex is simplicial after removing the vertices and edges (if there are any) prior to it. Each removed edge (e.g.,* v_3v_4*) is between the neighbours of the corresponding simplicial vertex (e.g.,* v_5*). The graph shown in Fig. 3.1c, however, is not weakly recursively simplicial because none of the remaining vertices can become simplicial after removing the simplicial vertex* v_5.

The next definition generalizes the concept of perfect elimination ordering to the composition the ordering and some excessive neighbouring edges that stop a vertex being simplicial.

Definition 3.1.27. *A set of* ***excesses*** *of a graph* $G = (V,E)$ *w.r.t. an ordering* α *is a bijection* $\varepsilon_\alpha : \{\alpha(1),\dots,\alpha(n)\} \leftrightarrow \{\varepsilon_\alpha(\alpha(1)),\dots,\varepsilon_\alpha(\alpha(n))\}$ *from the ordered vertices of G to a set of subsets of edges, in which each* $\varepsilon_\alpha(\alpha(i)) \subseteq E(G[N(\alpha(i))])$ *consists of some edges between the neighbours of the vertex* $\alpha(i)$.

Let $\alpha = \{v_1,\dots,v_n\}$ be an ordering of a graph G and ε_α be a set of excesses of G w.r.t. α. The composition $\kappa = (\alpha, \varepsilon_\alpha)$ of the ordering and excesses is called a *kit* of G. For convenience, denote by $\kappa(i)$ the i^{th} order and excess in the kit κ of a graph G and let $\kappa(0) = \emptyset$ by convention. Same as before, given a kit κ of a graph G, the graph can be recursively eliminated by removing a vertex and some edges staring from the first to the last item in κ until the graph becomes empty.

Example 3.1.4. *The ordering* $\beta = \{v_1,\dots,v_{13}\}$ *and the set of empty excess* $\varepsilon_\beta = \{\emptyset,\dots,\emptyset\}$ *form an elimination kit of the graph shown in Fig. 3.2, because the graph can be eliminated completely by following* $(\beta, \varepsilon_\beta)$. *But the elimination kit is not perfect that will be defined next.*

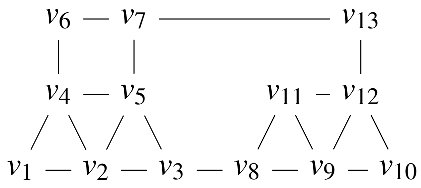


Fig. 3.2 An example of a WRS graph.

Definition 3.1.28. *Let $G = (V,E)$ be a graph. A kit $\kappa = (\alpha, \varepsilon_\alpha)$ of the graph G is a* ***perfect elimination kit*** *if each vertex $\alpha(i)$ is simplicial in the subgraph $G - \{\kappa(0), \dots, \kappa(i-1)\}$.*

Given an elimination kit κ of a graph G, we can denote by $G^i = G - \{\kappa(0), \dots, \kappa(i)\}$ the subgraph obtained from G by removing the first $i-1$ terms in κ. We call the resulting subgraph G^i an *elimination graph.*

Example 3.1.5. *The elimination kit $(\beta, \varepsilon_\beta)$ in Example 3.1.4 is not perfect for the graph G shown in Fig. 3.2, because the vertex v_2 is not simplicial in the elimination graph G^1. A perfect elimination kit (PEK) for G is the combination of the ordering and excesses stated in the following:*

$$\alpha = \{v_{10}, v_9, v_8, v_{11}, v_{12}, v_{13}, v_1, v_3, v_2, v_4, v_5, v_6, v_7\},$$

$$\varepsilon_\alpha = \{\{v_9 v_{12}\}, \{v_8 v_{11}\}, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \{v_4 v_5\}, \emptyset, \emptyset, \emptyset, \emptyset\}.$$

3.1.2 Graphical Model

In graphical model, a vertex (or node) in a graph represents a random variable with its probability distribution.

Definition 3.1.29. *Let P be a joint probability distribution of a set V of random variables and $G = (V,E)$ be a directed acyclic graph. We say $< G,P >$ satisfies the* ***Markov condition*** *if each variable $x \in V$ is conditionally independent of its non-descendants given its parents.*

Definition 3.1.30. *Let P be a joint probability distribution of a set V of random variables and $G = (V,E)$ be a directed acyclic graph. We call $< G,P >$ a* ***Bayesian network*** *if it satisfies the Markov condition.*

Definition 3.1.31. *Let $< G,P >$ be a Bayesian network. We say $< G,P >$ satisfies the* ***faithfulness condition*** *if G entails all and only the conditional independencies in P.*

If $< G,P >$ satisfies the faithfulness condition, we say the DAG G and the joint distribution P are faithful to each other. Throughout this thesis, we focus on Bayesian networks with discrete random variables. It was proved by Meek [56] that the set of unfaithful discrete Bayesian networks has Lebesgue measure zero, so all the Bayesian networks considered in this thesis are assumed to satisfy the faithfulness assumption unless it is mentioned otherwise.

Definition 3.1.32. *Let $< G = (V,E),P >$ be a Bayesian network. The* ***Markov blanket*** *of a variable $x \in V$ in the Bayesian network, denoted by $MB(x)$, is the minimum subset of variables satisfying $x \perp\!\!\!\perp_P v \mid MB(x)$ for each $v \in V \setminus MB[x]$, where $MB[x] = MB(x) \cup \{x\}$.*

Introduced by Pearl [66] as the smallest subset of variables in a Bayesian network, given which the target variable is conditionally independent from the rest of the variables, the Markov blanket [1] has become popular for feature selection [45] and scaling up learning causal models [68]. In a faithful Bayesian network, the Markov blanket of a target variable consists of its parents, children and children's other parents (a.k.a., spouses). For example, in the Bayesian network shown in Fig. 3.3, the variable v_3's Markov blanket consists of v_3's parents v_5 and v_6, its children v_1 and v_2, its spouse v_4.

Definition 3.1.33. *The* ***moral graph*** *of a directed acyclic graph $G = (V,E)$ is the skeleton of the hybrid graph $H = (V,E \cup F)$, where the set of additional edges $F = \{uv \mid u,v \in P_G(x), \{\overrightarrow{uv}, \overrightarrow{vu}\} \cap E = \emptyset, \forall x \in V\}$ connect non-adjacent parents of each vertex x in G.*

[1] Originally, this is how Pearl [66] defined "Markov boundaries", but the literature has migrated "Markov blankets" to this minimalist sense.

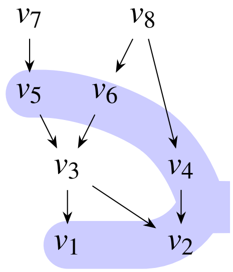


Fig. 3.3 The MB of v_3 in this BN is $\{v_5, v_6, v_1, v_2, v_4\}$.

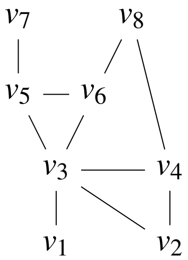


Fig. 3.4 The moral graph of the BN in Fig. 3.3.

The above definition implicitly states a trivial *moralization* process that turns a DAG into a moral graph. That is, by joining all pairs of non-adjacent parents for each vertex in the DAG, then dropping all the directions. The graph shown in Fig. 3.4 is the moral graph of the Bayesian network in Fig. 3.3. It is obtained by joining the vertices v_3, v_4 that are common parents of v_2 and v_5, v_6 that are common parents of v_3, then dropping all the directions. Appendix B gives more on the background of moral graphs and its application in belief propagation via the junction tree algorithm.

Remark 3.1.1. *Let $G = (V, E)$ be the directed acyclic graph of a Bayesian network and H be the moral graph of G. For each vertex $x \in V$, its Markov blanket in G is identical to its neighbourhood in H. That is, $MB_G(x) = N_H(x)$.*

Any spouse v of x that is neither a parent nor child of x must be connected to x during the moralization process. Therefore, the Markov blanket of each vertex x in G becomes its neighbourhood in the moral graph H. For example, the Markov blanket $MB_G(v_3) =$

$\{v_1, v_2, v_4, v_5, v_6\}$ in Fig. 3.3 is identical to the neighbourhood $N_G(v_3) = \{v_1, v_2, v_4, v_5, v_6\}$ in Fig. 3.4.

The next definition is an alternative way, without using d-separation, to define the Markov equivalence between two DAGs.

Definition 3.1.34. *The directed acyclic graphs of two Bayesian networks are **Markov equivalent** if and only if they entail the same conditional independencies.*

The Bayesian networks that are (Markov) equivalent to each other are in the same *equivalence class*. The class can be represented by a single structure, called a *(DAG) pattern*. It is a hybrid graph that has the same skeleton as all the DAGs in the equivalent class and the directed edges that are common to all the DAGs in the class.

3.1.3 Computational Complexity Theory

In this section, we briefly introduce some fundamental concepts in computational complexity theory that will be used throughout this thesis. In particular, we only focus on the time complexity of solving our problems in the worst case scenario that is measured by the asymptotic notation (a.k.a., big-O-notation). We assume the readers are familiar with the basic concepts in computational complexity theory and the complexity classes *P, NP* and *NP-complete*.

The big-O-notation is useful when analyzing the running time of an algorithm for different input size, because it suppresses the lower order terms in an equation that typically do not increase as fast as the highest order term for large input size.

Definition 3.1.35. *Let f and g be functions that measure the running time of an algorithm on input of size n. We say $f(n) = O(g(n))$ if there exist positive integers c and n_0 such that for every $n \geq n_0$, they satisfy $f(n) \leq cg(n)$.*

Next, we introduce 3SAT that is one of the fundamental problems that is known to be NP-complete to decide [81]. In this context, we call a Boolean variable x or its negation $\bar{x}$ a *literal*. A *clause* is a disjunction of literals written in the form $c_i = (x_1 \vee \overline{x_2} \vee \cdots \vee x_k)$. Hence, a clause is TRUE if it contains at least one TRUE literal. A *CNF-formula* is a conjunction of clauses written in the form of $c_1 \wedge \cdots \wedge c_m$. Hence, a CNF-formula is TRUE if all its clauses are TRUE simultaneously. In such case, we say the formula is *satisfiable*.

Definition 3.1.36. *The language* 3SAT *is the set of satisfiable 3CNF formulas.*

3SAT is a decision problem which is to decide whether or not a 3CNF-formula is satisfiable. For example, the formula $(x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$ is satisfiable by assigning $x_1 = TRUE, x_2 = FALSE$ and $x_3 = FALSE$ respectively.

A common way of proving the NP-completeness of a problem is by *reducing* a known NP-complete problem to the problem at hand; the reduction must take polynomial time and be such that an instance of the known NP-complete problem is TRUE if and only if the corresponding transformed instance is TRUE.

3.2 Morality, Weak Recursive Simpliciality and Perfect Elimination Kit

This section proves the equivalence of some properties to morality, initially showing that there is a one-to-one correspondence between being weakly recursively simplicial and having a perfect elimination kit.

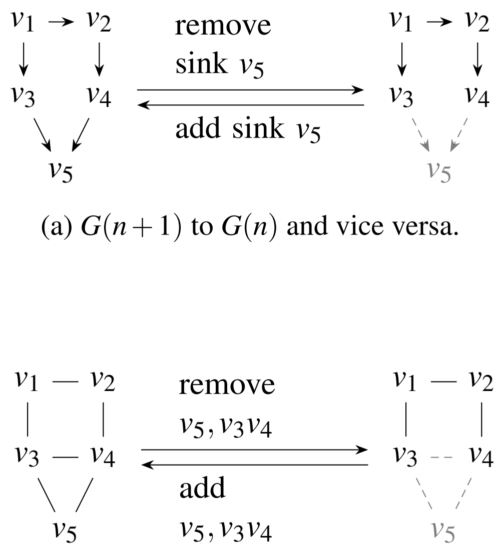
Theorem 3.2.1. *A graph is weakly recursively simplicial if and only if it has a perfect elimination kit.*

Proof. If a graph $G = (V, E)$ is weakly recursively simplicial, the simplicial vertex u and the set of edges $E' \subseteq E(G[N_G(u)])$ between the neighbours of u that are removed at each step of the recursion form an ordering α of G and a set of excesses ε_α. Since each vertex u in α is simplicial in the corresponding elimination graph, the elimination kit $(\alpha, \varepsilon_\alpha)$ of the graph G is perfect. The converse is also true because if G has a perfect elimination kit $\kappa = (\alpha, \varepsilon_\alpha)$, then each vertex $\alpha(i)$ is simplicial in the elimination graph G^{i-1} obtained by removing $\kappa(i-1)$ from the previous elimination graph. Therefore, the graph G is weakly recursively simplicial by definition. □

Next, we show the equivalence between moral graphs and weakly recursively simplicial graphs. This is proved by the following two lemmas.

Lemma 3.2.1. *Let $G = (V, E)$ be a DAG and H be the moral graph of G. Then H is weakly recursively simplicial.*

Proof. The lemma is proved by induction on the number of vertices. Let $G(n)$ and $H(n)$ denote, respectively, a DAG and its moral graph over n vertices. The lemma is true for $n \leq 3$, because all graphs containing three vertices or less are chordal, so weakly recursively simplicial. Assuming for $n \geq 3$ that the moral graph $H(n)$ of any n vertices DAG is weakly recursively simplicial. We want to show that the moral graph $H(n+1)$ of any DAG $G(n+1)$ is also weakly recursively simplicial. Each DAG contains a sink u. After the moralization process that transforms a DAG $G(n+1)$ to its moral graph $H(n+1)$, the sink u becomes a simplicial vertex in $H(n+1)$, because its parents form a clique. By removing the sink u from the DAG $G(n+1)$ we obtain a subgraph $G(n)$ that is also a DAG. In addition, the moral graph $H(n)$ of $G(n)$ is a subgraph of the moral graph $H(n+1)$. The difference between the two moral graphs is the sink u and some edges between its parents (that do not appear in $G(n+1)$). For example, Fig. 3.5. The inductive hypothesis assumes that each moral graph $H(n)$ is weakly recursively simplicial. Therefore, the supergraph $H(n+1)$ is also weakly recursively simplicial. □



(a) $G(n+1)$ to $G(n)$ and vice versa.

(b) $H(n+1)$ to $H(n)$ and vice versa.

Fig. 3.5 Vertex addition and removal between DAGs and their corresponding moral graphs.

Lemma 3.2.2. *Let $H = (V,E)$ be a weakly recursively simplicial graph. Then H is the moral graph of a DAG.*

Proof. Let $H(n)$ denote a weakly recursively simplicial graph over n vertices and $G(n)$ as defined above. The statement is true for $n = 1$, because a single node graph $H(1)$ is a weakly recursively simplicial graph as well as the moral graph of the DAG $G(1)$. Assume that for $n \geq 1$ a weakly recursively simplicial graph $H(n)$ is the moral graph of a DAG $G(n)$, we want to show that a weakly recursively simplicial graph $H(n+1)$ is the moral graph of a DAG $G(n+1)$. By the definition, if a graph $H(n+1)$ is weakly simplicial, it has a simplicial vertex u and an excess $\varepsilon(u)$ such that $H(n+1) - u - \varepsilon(u)$ is again weakly recursively simplicial. By the inductive assumption, any $H(n)$ is the moral graph of a DAG $G(n)$. Hence, connecting the vertex u to the vertices $N_{H(n)}(u)$ in $G(n)$ as a sink, we obtain a DAG $G(n+1)$, whose moral graph is $H(n+1)$. □

Theorem 3.2.2. *Let G be a graph. The following are equivalent:*

1. *G is moral.*

2. *G is weakly recursively simplicial.*

3. *G has a perfect elimination kit.*

Proof. The equivalence between 1 and 2 follows from Lemma 3.2.1 and 3.2.2. The equivalence between 2 and 3 follows from Theorem 3.2.1. □

The above theorem provides two different ways of checking morality when it is not known whether a graph is resulted from moralizing a DAG. In general, a graph can have more than one simplicial vertex. Hence, we prove next lemma to demonstrate that a recursive process can start eliminating any simplicial vertex.

Lemma 3.2.3. *Let H be a weakly recursively simplicial graph. For any simplicial vertex u of H, the graph H has a perfect elimination kit $\kappa = (\alpha, \varepsilon_\alpha)$ with $\alpha(1) = u$.*

Proof. Theorem 3.2.2 implies that there is a DAG $G = (V, F)$, whose moral graph is H. For any sink u in G, the subgraph $G' = G - u$ is also a DAG. Let H' be the moral graph of G'. Hence, we have $H' = H - u - S$ for a subset $S \subseteq E(H[N_H(u)])$ of edges between the neighbours of u. By Theorem 3.2.2, the graph H' has a perfect elimination kit $\kappa' = (\beta, \varepsilon_\beta)$. Hence, adding u and S to the front of β and ε_β gives a perfect elimination kit $\kappa = (\alpha, \varepsilon_\alpha)$ of H such that the ordering starts with the sink u i.e., $\alpha(1) = u$. □

Based on Theorem 3.2.2 and Lemma 3.2.3, it is possible to use a backtracking algorithm to decide whether or not a given graph G is moral. If it is, the algorithm will return TRUE and orient G into a hybrid graph, of which there always exist a consistent DAG extension as proved by Dor and Tarsi [27].

The following two remarks are also made by Verma and Pearl [91]. Being necessary conditions for a graph to be moral, they can be used to reduce the running time of the backtracking Algorithm. In particular, Remark 3.2.1 is plays an important role in the optimal moral graph triangulation that will be discussed in Appendix B.

Remark 3.2.1. *If a graph is moral, it has at least one simplicial vertex.*

Remark 3.2.2. *If a graph is moral, each cycle of length at least* 4 *in it shares an edge with a k-clique for $k \geq 3$.*

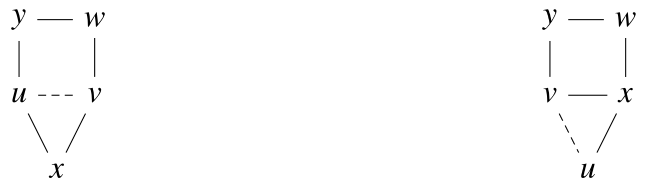
3.3 Complexity of Checking Morality

The WRS property gives a way to check morality. However, it was proved by Verma and Pearl [91] that deciding morality is NP-complete. This is not so surprising because first, the number of choices of what edges to remove between a simplicial vertex's neighbours can grow exponentially with the degree of the vertex; second, the deletion of some edges can stop a vertex being simplicial in a later recursive step, which cannot be anticipated at the time of that deletion. Noting that high vertex degree can be a cause of the NP-completeness of deciding morality, we next consider restricted maximum degrees starting with the easy case of 3.

3.3.1 Maximum Degree Three Graphs

It is trivial to check morality for graphs with the maximum degree at most 2, because all graphs in this case are chordal and can be recognized in polynomial time [72]. To prove that there exists a polynomial time algorithm for maximum degree 3 graphs, we prove the following lemmas first. We use E_x to denote the set $E(G[N(x)])$ of edges between the neighbours of the vertex x. The next lemma states that if a graph is not moral, adding an edge between non-adjacent vertices who have no common neighbours will not make it moral.

Lemma 3.3.1. *Let $G = (V,E)$ be a non-moral graph. If two vertices $u, v \in V$ have no common neighbours in G, then the supergraph $H = G + uv$ is not moral.*



(a) F is over $\{u,v,w,x,y\}$ (b) F is over $\{v,w,x,y\}$

Fig. 3.6 Two cases when adding the edge uv makes the vertex x simplicial in the corresponding elimination graph.

Proof. The graph G is not moral implies two possibilities. First, G has no simplicial vertex at all. Second, G has some simplicial vertices but G cannot be fully eliminated to the empty graph. The two cases are dealt separately.

Case 1: $D(G) \neq \emptyset$. The only chance to turn a non-simplicial vertex x in the graph G into a simplicial vertex in the supergraph H is to add x's deficiency to G so that its deficiency $D_H(x) = \emptyset$. This requires x to be the common neighbour of the two end vertices u, v of each added edge. Hence, the premise u, v have no common neighbour in G implies that the supergraph H remains non-moral.

Case 2: $D(G) = \emptyset$. The assumption G is non-moral but has simplicial vertices implies that for all elimination process, there always exists a proper subgraph $F \subsetneq G$ that is non-empty. If the supergraph H is moral, then the additional edge uv must enable a simplicial vertex x by emptying its deficiency in the corresponding elimination graph. This is true only in the cases where the deficiency of x in the subgraph F is $D_F(x) = uv$ or $D_{F-xv}(x) = \emptyset$ as shown in Fig. 3.6. Both cases, however, contradict with the assumption that u, v have no common neighbours in G. Therefore, the supergraph H remains non-moral. □

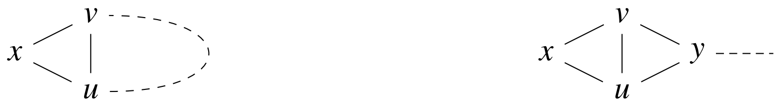
The benefit of a graph having a low vertex degree is that each vertex is connected to the rest of the graph in limited ways. Based on this, we show in the next lemma that under a certain case, morality is preserved after removing a simplicial vertex and all of the edges between its neighbours. Define the *closed neighbourhood* of a vertex x to be $N[x] = N(x) \cup \{x\}$ the union of x's neighbourhood and itself.

Lemma 3.3.2. *Let $G = (V, E)$ be a moral graph that contains a simplicial vertex x. If each pair of x's neighbours have common neighbours only in the closed neighbourhood $N_G[x]$ of x, then the subgraph $G' = G - x - E_x$ is moral.*

Proof. Assume the subgraph G' is not moral. The deletion of the edges in E_x implies that every pair of x's neighbours u, v are non-adjacent in G'. In addition, the fact that their common neighbours $N_G(u) \cap N_G(v) \subsetneq N_G[x]$ is in the closed neighbourhood of x implies $N_{G'}(u) \cap N_{G'}(v) = \emptyset$. By Lemma 3.3.1, no non-empty proper subset $S \subsetneq E_x$ of edges between the neighbours of x can make the subgraph $G'' = G - x - S$ moral. Moreover, since u and v have no common neighbours in G', the graph $G - x = G' + E_x$ has no additional simplicial vertices than G', hence is not moral either. Hence, for any subset $\varepsilon(x) \subseteq E_x$ of edges between the neighbours of x, the subgraph $G - x - \varepsilon(x)$ is non-moral. By lemma 3.2.3, the morality of G can be checked by starting with any simplicial vertex. Therefore, $G - x - \varepsilon(x)$ is not moral contradicts to G being moral, so G' must be moral. □

Based on Lemma 3.3.2, we conclude that the morality of maximum degree 3 graphs can be checked by recursively removing a simplicial vertex and all the edges between its neighbours. This is proved in the next lemma.

Lemma 3.3.3. *Let $G = (V, E)$ be a moral graph with the maximum degree $\Delta(G) = 3$. If G has a simplicial vertex x, then the subgraph $G' = G - x - E_x$ is moral.*



(a) u, v have one common neighbour

(b) u, v have two common neighbours

Fig. 3.7 Two maximum degree 3 graphs. The vertex x is simplicial with two neighbours u, v.

Proof. When the vertex x has degree $d_G(x) = 1$, it is a leaf in G and can be removed without causing any issue in the following steps. When $d_G(x) = 3$, the closed neighbourhood $N_G[x]$

of x forms a clique and each vertex in $N_G[x]$ has degree 3. The assumption that G is connected then implies that the graph $G = K_4$ is a complete graph over 4 vertices. When x has degree $d_G(x) = 2$, the neighbourhood $N_G(x) = \{u, v\}$ contains two vertices. If u, v have no common neighbours besides the vertex x as shown in Fig. 3.7a, then $G - x - E_x$ is moral by Lemma 3.3.2. If u, v have another other common neighbour y in G as shown in Fig. 3.7b, then their degrees $d_G(u) = d_G(v) = \Delta(G)$ reach the maximum. Hence, the rest of the graph is connected to the induced subgraph over $\{u, v, x, y\}$ via the vertex y only. Therefore, the subgraph $G' = G - x - E_x$ is moral too. □

Algorithm 3: Checking morality for maximum degree 3 graphs

```
Input  : graph $G = (V, E)$ with maximum degree 3
Output : TRUE or FALSE
while exists simplicial vertex $x$ do
    $G = G - x - E_x$
end
if $G = \emptyset$ then
    return TRUE;
else
    return FALSE;
end
```

Algorithm 3 presents pseudocode for checking morality of graphs with maximum degree 3. At each step of the *While* loop the algorithm eliminates from the input graph a simplicial vertex and all the edges between its neighbours. Once finished, the algorithm returns TRUE if the graph is fully eliminated, otherwise it returns FALSE. A theorem completes this section:

Theorem 3.3.1. *The morality of maximum degree* 3 *graphs can be decided in linear time.*

Proof. The correctness of Algorithm 3 is proved by Lemma 3.3.3 and Theorem 3.2.2 which proves the equivalence between graph morality and weak recursive simpliciality. Assume a graph with n vertices is represented by an adjacency list. Since each vertex's degree is bounded above by 3, it takes $O(1)$ time to check the simplicity of a vertex x. Initially, the algorithm spends $O(n)$ time to check simpliciality for all n vertices and store the simplicial vertices in a queue. Due to the bounded maximum degree, not only the vertex and edge removals take constant time, but the deletions affect the simpliciality at most a constant number of neighbours. So the queue can be updated in constant time. Therefore, Algorithm 3 runs in $O(n)$ time. □

3.3.2 Maximum Degree Four Graphs

This section focuses on developing a polynomial time algorithm to check morality for graphs with the maximum degree 4. Higher maximum degree allows a simplicial vertex to have more freedom when connecting to the rest of the graph, therefore different choices of edge removal need to be considered when removing a simplicial vertex, depending on its degree in the corresponding elimination graph. Later in the section, we prove that in some cases, simplicial vertices must be removed in a fixed order in terms of their degrees. Again, we use E_u to denote the set $E(G[N_G(u)])$ of all edges between the vertex u's neighbours in a graph G. First, we get rid of simplicial vertices with degrees $1,3$ and 4.

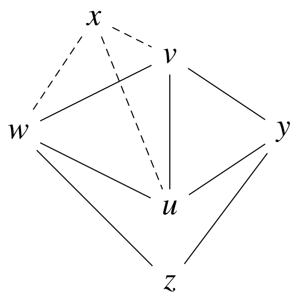


Fig. 3.8 A maximum degree 4 graph. The vertex x is simplicial with three neighbours u,v,w. The vertices u,v have a common neighbour that is not in the closed neighbourhood $N_G[x]$.

Lemma 3.3.4. *Let $G=(V,E)$ be a moral graph with the maximum degree $\Delta(G)=4$. If a vertex x is simplicial in G and its degree $d_G(x)\in\{1,3,4\}$, then the subgraph $G'=G-x-E_x$ is moral.*

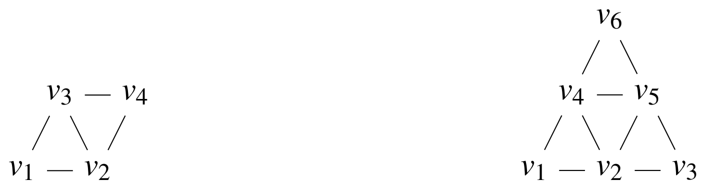
Proof. If the vertex x has degree $d_G(x) = 1$, then it is a leaf in G and can be removed without causing any issues in the following steps. If $d_G(x) = 4$, the closed neighbourhood $N[x]$ of x forms a clique and each vertex in $N[x]$ has degree 4. Hence, the graph $G = K_5$ is a complete graph over 5 vertices because G is assumed to be a connected graph. It remains to prove the lemma when x is a degree 3 simplicial vertex.

Suppose the neighbourhood of x is $N_G(x) = \{u, v, w\}$. The fact that x is simplicial implies that each neighbour of x has degree at least 3. The maximum degree 4 assumption of G ensures their degree is at most 4. If each pair of x's neighbours have common neighbours only in $N_G[x]$, then the subgraph $G' = G - x - E_x$ is moral by Lemma 3.3.2. If a pair of x's neighbours u, v have a common neighbour y that is not in $N_G[x]$, then both u and v have degree 4. The only chance that the subgraph G' that is obtained by removing from G the vertex x and the edge set E_x is not moral is when the 3-clique over $\{u, v, y\}$ shares an edge with a cycle C_m for $m \geq 4$, but is broken after removing E_x. Given x is set to have degree 3 and u, v reach the maximum degree 4, such a cycle (if exists) is only adjacent to the vertices w and y. If this is the case, the deletion of E_x will break this cycle and hence the 3-clique over $\{u, v, y\}$ can be broken too without causing any issues in the following steps. Therefore, the subgraph G' is moral. □

Before looking into the where a simplicial vertex is of degree 2, we define another concept that is useful for separating the case into subcases.

Definition 3.3.1. *Let $P = x_1, \ldots, x_n$, $Q_1 = y_1, \ldots, y_{n-1}$ and $Q_2 = z_1, \ldots, z_n$ be three non-intersecting paths of length n, $n-1$ and n respectively. A* ***3-clique path*** *is a graph that is obtained either by*

- *adding the edges $y_i x_i$ and $y_i x_{i+1}$ between P and Q_1 for each $i \in [1, n-1]$, or by*
- *adding the edges $y_i x_i$, $y_i x_{i+1}$ and $y_n x_n$ between P and Q_2 for each $i \in [1, n-1]$.*



(a) 3-clique path of length 2.

(b) 3-clique path of length 3.

Fig. 3.9 Examples of 3-clique paths of length 2 and 3.

By definition, the number of vertices must satisfy $n \geq 2$ for otherwise the path Q_1 is an empty graph. The length of a 3-clique path is equal to the number of distinct 3-cliques it contains. When connecting vertices between the paths P and Q_1, the 3-clique path contains an odd number of 3-cliques and hence its length is equal to $2(n-1)-1$. When connecting vertices between the paths P and Q_2, the 3-clique path contains an even number of 3-cliques and hence its length is equal to $2(n-1)$. We use K_3^m to denote a 3-clique path of length m. That is, a 3-clique path that contains m distinct 3-cliques. Since this section focuses on maximum degree 4 graphs, the only vertices that do not reach the maximum degree are the two vertices (with degree 2 and 3 respectively) at each end of a 3-clique path. For example, Fig. 3.9a is a 3-clique path of length 2, where the two paths are $P = v_1v_2$ and $Q = v_3v_4$. The vertices v_1 and v_3 are at the left end of the 3-clique path K_3^2 and their degrees are 2 and 3 respectively. The vertices v_2 and v_4 are at the right end of K_3^2 and their degrees are also 2 and 3 respectively. The graph in Fig. 3.9b contains three 3-clique paths of length 3, over the sets of vertices $\{v_1, v_2, v_3, v_4, v_5\}$, $\{v_1, v_2, v_4, v_5, v_6\}$ and $\{v_2, v_3, v_4, v_5, v_6\}$ respectively. In the induced subgraph K_3^3 over $\{v_1, v_2, v_3, v_4, v_5\}$, only the vertex v_2 reaches the maximum degree 4.

Definition 3.3.2. *Let K_3^m be a 3-clique path of length m that is contained in a graph G as a subgraph. It is a **maximal** K_3^m if it is not a proper subgraph of another 3-clique path.*

For example, the 3-clique path over $\{v_1, v_2, v_4, v_5\}$ in the graph shown in Fig. 3.9b is not maximal, because it is contained as a proper subgraph of the 3-clique path over

$\{v_1, v_2, v_3, v_4, v_5\}$. The 3-clique path over $\{v_1, v_2, v_3, v_4, v_5\}$, however, is maximal because it is not a proper subgraph of any other 3-clique path. Since in most situations we only deal with maximal cases, all the 3-clique paths considered in this section are assumed to be maximal unless stated otherwise.

Corollary 3.3.1. *Let $G = (V,E)$ be a moral graph with the maximum degree $\Delta(G) = 4$. Assume G contains a 3-clique path K_3^1 of length 1 as a subgraph. If a degree 2 vertex $v_1 \in K_3^1$ is simplicial, then the subgraph $G' = G - v_1 - E_{v_1}$ is moral.*

Proof. This follows from Lemma 3.3.2, because the two neighbours of v_1 have no common neighbour other than v_1. □

The next lemma demonstrates how morality can be preserved when deleting a degree 2 simplicial vertex that is in a 3-clique of length 2.

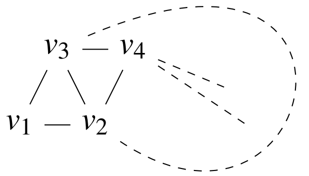


Fig. 3.10 A maximum degree 4 graph that contains a 3-clique path of length 2.

Lemma 3.3.5. *Let $G = (V,E)$ be a moral graph with the maximum degree $\Delta(G) = 4$. Assume G contains a 3-clique path K_3^2 of length 2 as a subgraph. If a degree 2 vertex $v_1 \in K_3^2$ is simplicial, then the subgraph $G' = G - v_1$ is moral.*

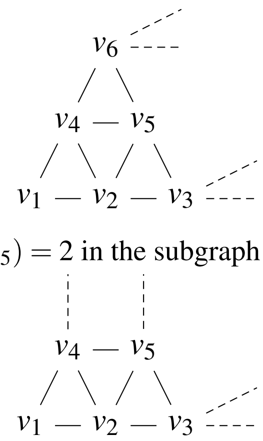
Proof. The graph G is shown in Figure 3.10. Assume the subgraph G' is not moral. This implies that G' has no perfect elimination kit. Let $G'' = G - v_1 - v_2v_3$ be another subgraph of G that is different from G' by the edge v_2v_3. By Lemma 3.2.3, the graph G is moral whilst the subgraph G' is not imply that G'' must be moral. The fact that missing edge v_3v_4 is in

G'' and the assumption that the 3-clique path K_3^2 is maximal ensure that the vertex v_4 is not eliminated before v_2 or v_3 when eliminating G''. Therefore, either v_2 or v_3 will appear as a degree 1 simplicial vertex (i.e., leaf) in the corresponding elimination graph. Let $\kappa = (\alpha, \varepsilon_\alpha)$ be a perfect elimination kit of the subgraph G''. Assume without loss of generality that $\alpha^{-1}(v_2) < \alpha^{-1}(v_3)$. Then the same elimination kit κ is also perfect for the subgraph G', because the vertex v_2 is also simplicial in G' and the edge v_2v_3 is deleted when deleting v_2. Hence, the subgraph G' is also moral, which contradicts with the assumption that it is not. Therefore, $G' = G - v_1$ is a moral graph. □

The following three lemmas consider the case when a degree 2 simplicial vertex is in a 3-clique path of length 3. As discussed before the only vertices that do not reach the maximum degree 4 in a 3-clique path are the two vertices on each end of the path, for instance, the vertices v_4 and v_5 in the induced subgraph over $\{v_1, \ldots, v_5\}$ in Fig. 3.11. Due to the flexibility of these two vertices, the middle 3-clique they are in may be critical to the following elimination steps, depending on whether the edge v_4v_5 is in a cycle of length at least 4 that is not in the 3-clique path K_3^3. Without loss of generality, assume a 3-clique path of length 3 is over the set of vertices $\{v_1, \ldots, v_5\}$ and the vertices have degrees $d_{K_3^3}(v_1) = d_{K_3^3}(v_3) = 2$, $d_{K_3^3}(v_4) = d_{K_3^3}(v_5) = 3$ and $d_{K_3^3}(v_2) = 4$.

Lemma 3.3.6. *Let $G = (V, E)$ be a moral graph with maximum degree $\Delta(G) = 4$. Assume G contains a 3-clique path K_3^3 of length 3 as a subgraph. If a degree 2 vertex $v_1 \in K_3^3$ is simplicial and the distance $d(v_4, v_5) \in \{2, \infty\}$ between the two degree 3 vertices in the subgraph $G - \{v_1, v_2, v_3\} - v_4v_5$, then the subgraph $G' = G - v_1$ is moral.*

Proof. The graph G is shown in Fig. 3.11. Assume the subgraph G' is not moral. Let $G'' = G' + v_2v_4$ be another subgraph of G that is obtained by adding to G' the edge v_2v_4. The addition of the edge guarantees a 3-clique over $\{v_2, v_4, v_5\}$ in G''. But this clique is only useful for producing a moral graph if it can break unbreakable cycles in G'. The degree $d_G(v_2) = \Delta(G)$ implies that the edges v_2v_4 and v_2v_5 do not appear in any cycles other than

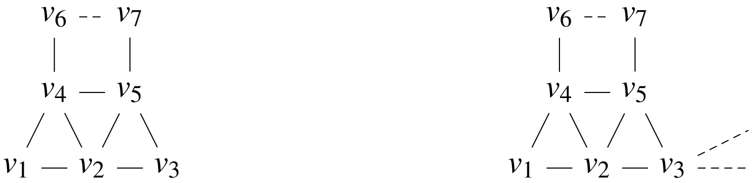


(a) The distance $d(v_4, v_5) = 2$ in the subgraph $G - \{v_1, v_2, v_3\} - v_4v_5$.

(b) The distance $d(v_4, v_5) = \infty$ in the subgraph $G - \{v_1, v_2, v_3\} - v_4v_5$.

Fig. 3.11 Two maximum degree 4 graphs. Each contains a 3-clique path of length 3 as a subgraph.

those in K_3^3. Furthermore, the distance $d(v_4, v_5)$ in the subgraph $G - \{v_1, v_2, v_3\} - v_4v_5$ is either 2 or infinity implies that the edge v_4v_5 is not in any cycle of length at least 4. Hence, if the subgraph G' is not moral the subgraph G'' is not moral either. This contradicts to premise that G is moral. Therefore, the subgraph G' must be moral. □



(a) Both v_1 and v_3 are simplicial.

(b) v_1 is simplicial while v_3 is not.

Fig. 3.12 Two maximum degree 4 graphs. Each contains a K_3^3 as a subgraph. The distance $d(v_4, v_5) \in [3, \infty)$ in the subgraph $G - \{v_1, v_2, v_3\} - v_4v_5$.

Lemma 3.3.7. *Let $G = (V, E)$ be a moral graph with the maximum degree $\Delta(G) = 4$. Assume G contains a 3-clique path K_3^3 of length 3 as a subgraph. If both degree 2 vertices v_1 and v_3 are simplicial and the distance $d(v_4, v_5) \in [3, \infty)$ between the two degree 3 vertices in the subgraph $G - \{v_1, v_2, v_3\} - v_4v_5$, then the subgraph $G' = G - \{v_1, v_3\}$ is moral.*

Proof. The graph G is shown in Fig. 3.12a. Since the vertices v_2, v_4 and v_5 reach the maximum degree 4, the two 3-cliques over $\{v_1, v_2, v_4\}$ and $\{v_2, v_3, v_5\}$ do not share edges with any cycle in G that does not contain the 3-clique $\{v_2, v_4, v_5\}$. Assume without loss of generality that the there is a four cycle over $\{v_4, v_5, v_6, v_7\}$. Removing the two simplicial vertices v_1, v_3 renders the vertex v_2 simplicial in the elimination graph, which can break the four cycle. Therefore, if G is moral then the subgraph G' remains moral. □

Lemma 3.3.8. *Let $G = (V,E)$ be a moral graph with the maximum degree $\Delta(G) = 4$. Assume G contains a 3-clique path K_3^3 of length 3 as a subgraph. If the vertex v_1 is simplicial while the vertex v_3 is not and the distance $d(v_4, v_5) \in [3, \infty)$ between the two degree 3 vertices in the subgraph $G - \{v_1, v_2, v_3\} - v_4v_5$, then the subgraph $G' = G - v_1 - E_{v_1}$ is moral.*

Proof. Assume without loss of generality that there is only one simplicial vertex in the graph G as shown in Fig. 3.12b. Removing the simplicial vertex v_1 does not introduce new simplicial vertices in the elimination graph. Hence, if G is moral, the subgraph $G' = G - v_1 - v_2v_4$ must be moral too. □

So far, we have considered the case when a degree 2 simplicial vertex is in a 3-clique path of length at most 3. The next lemma shows how a long path of length at least 4 can be shortened while morality is still preserved.

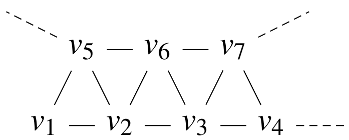
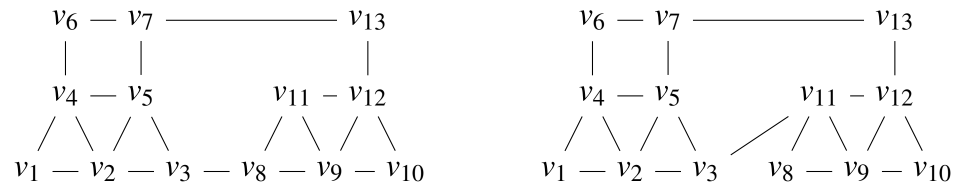


Fig. 3.13 An example of a 3-clique path of length 5.

Lemma 3.3.9. *Let $G = (V,E)$ be a moral graph with the maximum degree $\Delta(G) = 4$. Assume G contains a 3-clique path K_3^m of length $m \geq 4$ as a subgraph. If a degree 2 vertex $v_1 \in K_3^m$ is simplicial, then the subgraph $G' = G - v_1 - E_{v_1}$ is moral.*

Proof. Assume the graph G contains a 3-clique path of length 4 as shown in Fig. 3.13. Since the vertex v_2 reaches the maximum degree 4 and v_1 is simplicial, the subgraph G' remains moral after removing v_1 from G. In addition, the length of the 3-clique path reduced by 2. □

It can be seen from this lemma that after deleting a simplicial vertex, the number of 3-cliques that a path contains is reduced by 2 and consequently the length of the 3-clique path is decreased by 2. Therefore, we always deal with a simplicial vertex that is in a long 3-clique path first. The aim is to reduce the length of a long path to either $1,2$ or 3, so that it can then be dealt with by using the actions proved previously. Algorithm 4 is the pseudocode for checking morality for graphs with maximum degree 4. The algorithm deals with simplicial vertices that are in different conditions in a fixed order. Fig. 3.14 shows two examples of moral graphs that cannot be completely eliminated if simplicial vertices are removed in a different order.



(a) Remove $\{v_{10}, v_9v_{12}\}$ before $\{v_1, v_2v_4\}$.

(b) Remove $\{v_8, v_{10}\}$ before $\{v_1, v_2v_4\}$.

Fig. 3.14 Two maximum degree 4 moral graphs, whose simplicial vertices are in K_3^3s. The simplicial vertices must be removed in the order stated in Algorithm 4, otherwise these graphs will not be recognized as moral by taking the actions proved in the above lemmas.

Theorem 3.3.2. *The morality of maximum degree* 4 *graphs can be checked in quadratic time.*

Proof. The correctness of Algorithm 4 can be proved by the corollary and lemmas proved in this section.

Although this algorithm is more complicated than Algorithm 3, both algorithms run in the same magnitude in the worst case scenario. Same as Algorithm 3, it takes an initial $O(n)$ time

Algorithm 4: Checking morality for maximum degree 4 graphs

```
Input  : graph $G = (V, E)$ with maximum degree 4
Output : TRUE or FALSE
while exists simplicial vertex $v_1$ do
    if $d_G(v_1) = 4$ then
        return TRUE;
    else if $d_G(v_1) = 1$ then
        $G = G - v_1$;
    else if $d_G(v_1) = 3$ then
        $G = G - v_1 - E_{v_1}$;
    else
        if $v_1 \in K_3^4$ then
            $G = G - v_1 - E_{v_1}$;
        else if $v_1 \in K_3^1$ then
            $G = G - v_1 - E_{v_1}$;
        else if $v_1 \in K_3^2$ then
            $G = G - v_1$;
        else
            if $d(v_4, v_5) \in \{2, \infty\}$ in $G - \{v_1, v_2, v_3\} - v_4v_5$ then
                $G = G - v_1 - E_{v_1}$;
            else
                if $v_3 \in K_3^3$ is also simplicial then
                    $G = G - \{v_1, v_3\}$;
                else
                    $G = G - v_1 - E_{v_1}$;
                end
            end
        end
    end
end
if $G = \emptyset$ then
    return TRUE;
else
    return FALSE;
end
```

to get a queue of simplicial vertices in the current graph. Because of the limited maximum degree, it takes $O(1)$ time to find a simplicial vertex's degree, remove it and its neighbouring edges. The worst case of this algorithm appears in the case when a simplicial vertex has degree 2, because the algorithm needs to know the length of a 3-clique path. This process, however, can be done in $O(n)$ time because it is sufficient to identify the length of a path up to 4. Once a path of length 3 is discovered, it takes $O(n+e)$ time to run the breadth-first-search to calculate the distance $d(v_4, v_5)$ in the corresponding subgraph, where e is the number of edges in the graph. Because of the maximum degree 4 constraint, the number of edges is much less than the number of edges, so the running time of the breadth-first-search is $O(n)$, dominated by n. In the worst case, the algorithm always produces a new degree 2 simplicial vertex that will be added into the queue. Therefore, the algorithm runs in $O(n^2)$ time in the worst case. □

It can be seen that Algorithm 4 also works for maximum degree 3 graphs. Because degree 1 and 3 simplicial vertices can be dealt with line 4 and 6 of Algorithm 4. As vertex degree is bounded by 3, a degree 2 simplicial vertex is in a 3-clique path of length at most 2. Line 11 takes the same action as Algorithm 3. Although line 13 removes only the simplicial vertex, it has the same effect as removing both v_1 and E_{v_1} because both neighbours of v_1 have degree 3 and consequently are not incident to the rest of the graph.

To this point, we have proved that for graphs with maximum degree 3 and 4, their morality can be checked in polynomial time. In the next section, we focus on proving that the problem of checking morality for graphs with higher maximum degrees remains NP-complete.

3.3.3 Graphs with Larger Maximum Degrees

It has been proved by Verma and Pearl [91] that deciding morality is an NP-complete problem. The hardness for general graphs arises from the freedom of deleting different subset of edges between the neighbours of a simplicial vertex, which could lead to a dead end in the following steps that is impossible to anticipate at the time of deletion. This is not an issue for lower maximum degree graphs as has been demonstrated in the preceding sections. However, as it will be shown in the next theorem that the problem remains NP-complete for maximum degree 5 graphs, hence leaving no gap between P and NP-complete for this problem. [2]

Theorem 3.3.3. *The problem of deciding morality for maximum degree* 5 *graphs is NP-complete.*

The theorem can be straightforwardly proved by modifying the reduction in [91] to build graphs with max degree 5 based on an input 3CNF formula ϕ. Denote each variable in ϕ by v_i and each clause by c_j. The reduction starts by building a variable gadget for each variable v_i and a clause gadget for each clause c_j to simulate the behaviour of a variable and clause in the formula ϕ. It then builds two auxiliary gadgets to connect all the variable and clause gadgets together to form a desired graph. We need to prove that such a reduction always produces graphs with maximum degree 5 and the produced graph is moral if and only if the input formula is satisfiable. Most of the work in [91] is sufficient to prove Theorem 3.3.3, except that the auxiliary gadget and how the gadgets are connected to each other must be modified to avoid violating the maximum degree 5 constraint. As we have introduced the concept of weakly recursively simplicial, we will assess a graph's morality by testing its weakly recursively simpliciality rather than directing it to obtain a DAG (as Verma and Pearl [91] did).

Proof. A variable in a 3CNF formula takes either TRUE or FALSE but not both values simultaneously. Hence, a variable gadget must be a graph that looks different when taking different values. Fig. 3.15 is the variable gadget used in [91]. It contains two isomorphic subgraphs (but with different labels), one for the variable v_i and one for its negation $\bar{v_i}$. Later in the proof, this gadget is connected to other gadgets via the vertices v_i^0 and $\bar{v_i}^0$, in which

[2] The theorem also implies that deciding morality is not *fixed-parameter tractable* with respect to the maximum degree.

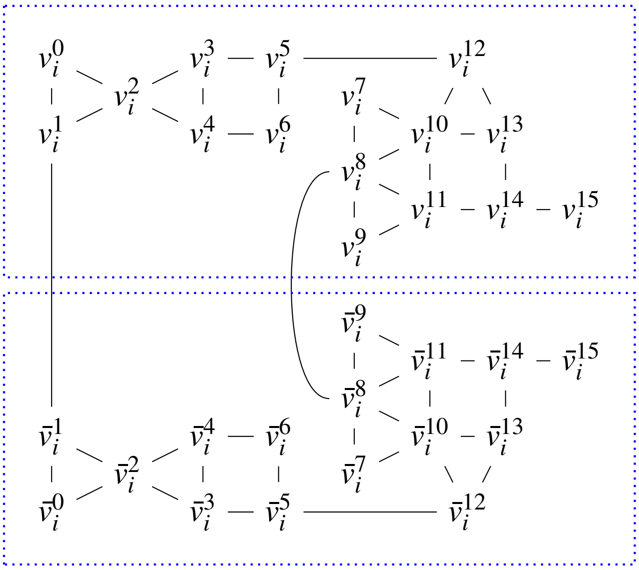


Fig. 3.15 A variable gadget for the variable v_i in a formula ϕ. It contains two subgraphs, where the top corresponds to the literal v_i and the bottom corresponds to the literal $\bar{v}_i$. The edge $v_i^8\bar{v}_i^8$ ensures that when the vertices v_i^0 and $\bar{v_i}^0$ are not simplicial and the elimination process starts from neither the vertex v_i^{15} nor $\bar{v_i}^{15}$, only one of v_i^{15} and $\bar{v_i}^{15}$ can be eliminated.

case v_i^7, v_i^9 and $\bar{v_i}^7, \bar{v_i}^9$ are the only simplicial vertices in this gadget. The purpose of the edge $v_i^8\bar{v_i}^8$ is to avoid the upper and lower subgraphs having the same perfect elimination kits that result in isomorphic sub-DAGs.

A clause in a formula is either TRUE or FALSE, depending on the values of its literals. Each clause of a 3CNF formula contains at most 3 literals. For simplicity, we only explain here the case when each clause takes exactly 3 literals. Since a clause is a disjunction of literals, it is TRUE when one literal is TRUE. Fig. 3.16 is the clause gadget used in [91]. Each of the three subgraphs on the left corresponds to a literal. The isomorphism (up to labelling) of these literals indicates all of them can take the same value simultaneously. All three subgraphs are connected to a 4-clique that simulates the disjunction operation.

The auxiliary gadgets help obtain the desired graph. The auxiliary gadget 1 shown in Fig. 3.17a is the smallest moral graph that has exactly one simplicial vertex. The auxiliary gadget 2 is a path of length $m+2$ that is used to connect all the clause gadgets together, where m is

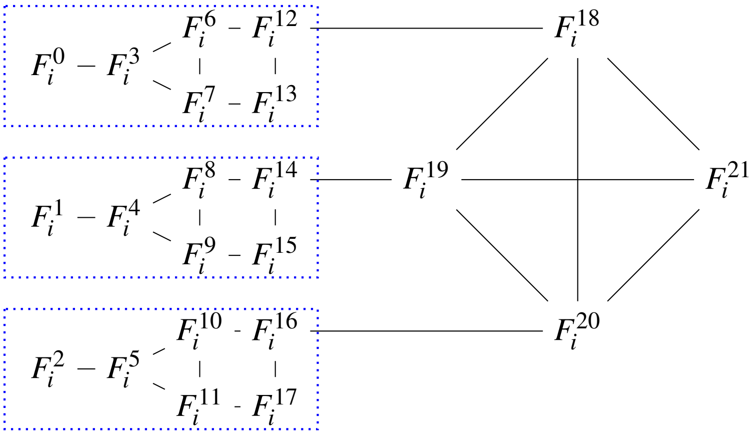
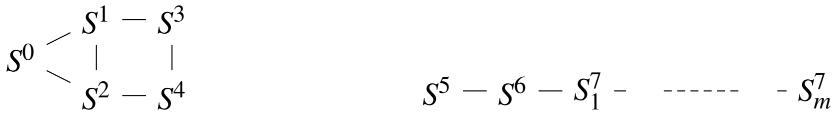


Fig. 3.16 A clause gadget for a clause c_i in a formula ϕ. It consists of three isomorphic subgraphs (up to labelling), one for each literal in c_i and a 4-clique to simulate the disjunctive operation.

the number of clauses in the formula ϕ. This is different from what is used in [91] so as to avoid a vertex with degree m in their work.



(a) Auxiliary gadget 1.

(b) Auxiliary gadget 2.

Fig. 3.17 Two auxiliary gadgets assist to connect all the gadgets together to obtain a desired graph. Auxiliary gadget 2 contains $m+2$ vertices, where m is the number of clauses in a formula ϕ for the 3CNF problem.

Given a 3CNF formula ϕ with n variables and m clauses, our construction will build a graph with $32n+23m+7$ vertices. These vertices are made of 32 vertices from each of the n variable gadgets, 22 vertices from each of the m clause gadgets and $7+m$ vertices from the two auxiliary gadgets. The gadgets are connected together to build a connected graph in the following ways:

1. all the variable gadgets are connected together by the edges $\bar{v}_i^0 v_{i+1}^0$ for $i \in [1, n-1]$,
2. the auxiliary gadget 1 is connected to the first variable gadget by the edge $S^0 v_1^0$ and the auxiliary gadget 2 is connected to the last variable gadget by the edge $S^5 \bar{v}_n^0$,

3. the clause gadgets are connected together via the auxiliary gadget 2 by the edge $S_i^7F_i^{21}$ for each $i \in [1,m]$,

4. if a literal v_i appears in only one clause c_j in the formula ϕ, then connect the corresponding variable and clause gadgets by the edge $v_i^{15}F_j^k$ when v_i is the $(k+1)^{th}$ element in the clause for $k \in [0,2]$,

5. if v_i appears in more than one clauses, say in clauses c_r, c_s, c_t as their $(k_r+1)^{th}, (k_s+1)^{th}, (k_t+1)^{th}$ element respectively for $k_r, k_s, k_t \in [0,2]$, then connect the corresponding clause gadgets by the edges $F_r^{k_r}F_s^{k_s}, F_s^{k_s}F_t^{k_t}, F_s^{k_s}F_r^{k_r+3}, F_t^{k_t}F_s^{k_s+3}$ and connect them to v_i's gadget by the edge $v_i^{15}F_r^{k_r}$,

Figure 3.18 is an example of a graph that is constructed from a satisfiable 3CNF formula. The first two steps are as in [91]. Step 3 connects all the clause gadgets by a path of length equal to the number of clauses in ϕ in order to avoid high degree vertex in auxiliary gadget 2. Steps 4 and 5 first connect all clause gadgets that contain the same literal together as a 'path' and then connect one end of this 'path' to the corresponding variable gadget in order to avoid the vertices v_i^{15} and $\bar{v}_1^{15}$ having arbitrary high degrees. According to Verma and Pearl [91], the vertices that have degrees more than 5 are $d(S^7) = m$, $d(v_i^{15})$ and $d(\bar{v_i}^{15})$ equals to the number of times v_i and $\bar{v}_i$ respectively appears in the formula ϕ. By the construction described above, these vertices are guaranteed to have degrees no more than 5.

The reduction from a 3CNF formula ϕ to the desired graph G is clearly in polynomial time. It remains to show that ϕ is satisfiable if and only if G is moral. The way the gadgets are connected implies that the graph G can only be eliminated perfectly starting from the variable gadgets. Due to the edge $v_i^8\bar{v_i}^8$, the elimination will go through either the vertex v_i^{15} or $\bar{v_i}^{15}$ but not both. Suppose it goes through v_i^{15}, then the subgraph that corresponds to the variable v_i in each clause gadget that is connected to the vertex v_i^{15} will be eliminated. In such a case, we say that the clause is satisfied by the literal v_i, otherwise it is satisfied by the literal $\bar{v}_i$. If a formula ϕ is satisfiable, then every clause is satisfied by a literal. This implies that the 4-clique in every clause gadget in the graph G can be eliminated and consequently G can be perfectly eliminated completely. If a formula ϕ is not satisfiable, then there is a clause that cannot be satisfied by any assignments to the variables. Hence, the corresponding clause gadget cannot be eliminated at all and consequently stops the elimination process to go back to the variable gadgets from the vertex $\bar{v_n}^0$. Therefore, the graph G is not moral. □

Corollary 3.3.2. *The problem of deciding morality for graphs with maximum degree at least* 5 *is NP-complete.*

Proof. The corollary follows from Theorem 3.3.3. □

3.4 Minimum and Minimal Moralizations

Despite the potential value of moral graph in Markov blanket discovery and structure learning that will be discussed in Chapter 5, if one insists on obtaining a moral graph from the current graph, what is an optimal way of doing it? Here optimal means making the minimum number of changes to the edge set in the original graph. The two extremes are filling in edges until the graph is moral or removing edges until the graph is moral. In the context of chordal graphs, extensive work has been done on filling in edges to make a graph chordal. This is also known as *triangulation* (or *chordal completion*) [42]. An obvious reason for focusing on edge filling is because of chordal graph's application in efficient Gaussian elimination in solving a system of linear equations [71]. A zero entry in a matrix could become non-zero during the elimination process, which corresponds to a filled-in edge between two vertices in the corresponding graph. In the context of a Bayesian network, correctly representing a distribution's conditional independencies is crucial in structure learning. A false positive dependency between two variables is often preferred over a false negative. Because when a model is set, the structure usually will not be changed, a false positive connection may be

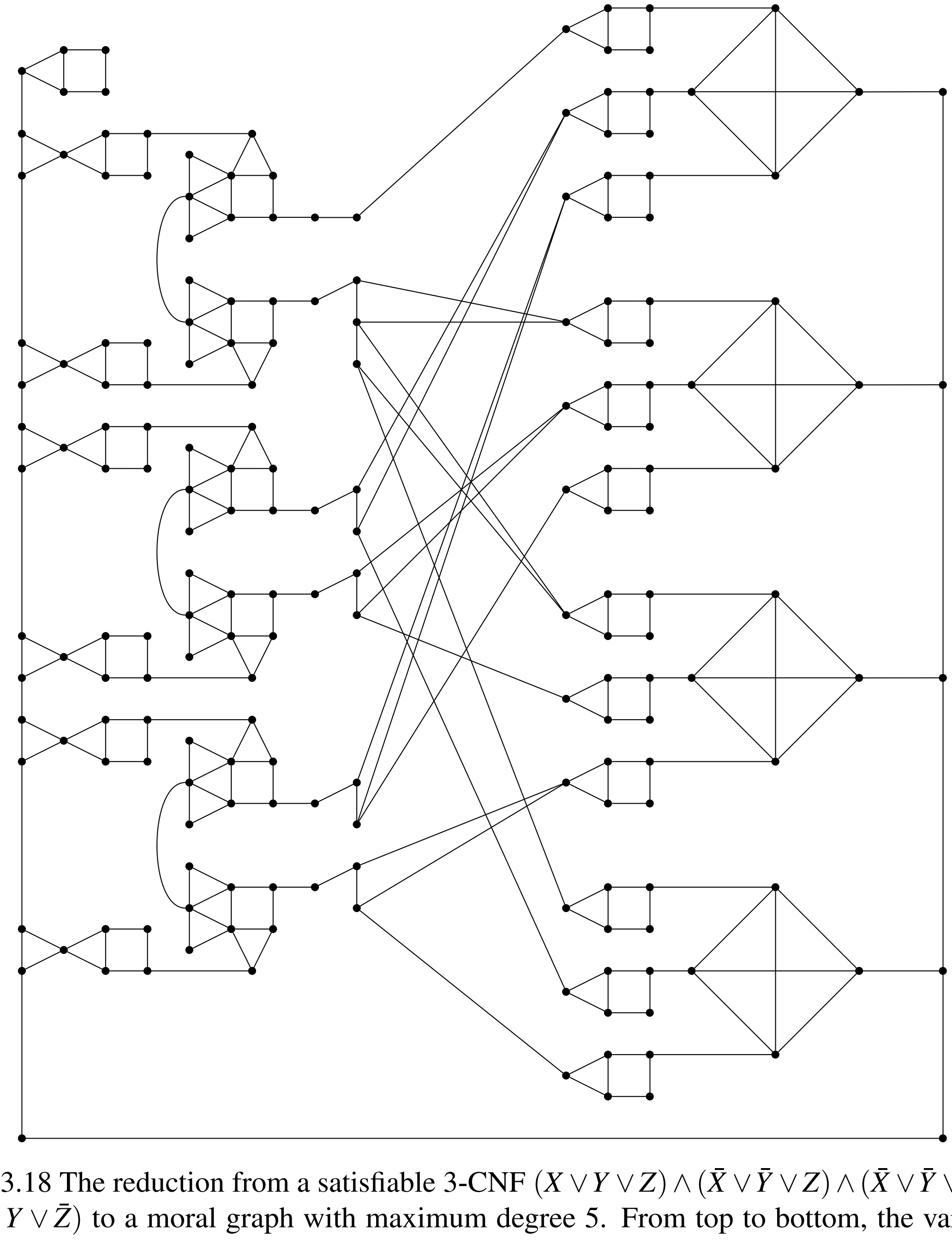


Fig. 3.18 The reduction from a satisfiable 3-CNF $(X \vee Y \vee Z) \wedge (\bar{X} \vee \bar{Y} \vee Z) \wedge (\bar{X} \vee \bar{Y} \vee \bar{Z}) \wedge (\bar{X} \vee Y \vee \bar{Z})$ to a moral graph with maximum degree 5. From top to bottom, the variable gadgets are for X, Y, Z and the clause gadgets are for F_1, F_2, F_3, F_4.

assumed extremely weak via its parameters to omit the dependency whilst a false negative cannot be put back.

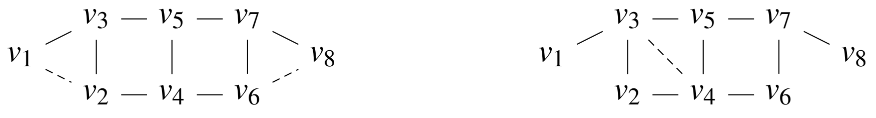
For the above reason, this section focuses on the computational complexity of turning a graph moral by filling in the minimum number of edges.

Definition 3.4.1. *A graph $H = (V, E \cup F)$ is called a* ***moralization*** *of a graph $G = (V, E)$ if H is moral.*

Assume without loss of generality that $E \cap F = \emptyset$, so every edge $e \in E \cup F$ is either an edge in the original graph G or a filled-in edge in the moral graph H.

Definition 3.4.2. *Let $G = (V, E)$ be a graph, and $H = (V, E \cup F)$ be a moral graph such that $E \cap F = \emptyset$. The graph H is a* ***minimal moralization*** *of G if there exists no proper subset $F' \subsetneq F$ such that the graph $(V, E \cup F')$ is moral. It is a* ***minimum moralization*** *if there exists no edge set E' satisfying $|E'| < |F|$ such that the graph $(V, E \cup E')$ is moral.*

Fig. 3.19a is an example of a minimal moralization. The edge set of the original graph is represented by the solid lines and the filled-in edges are represented by the dashed lines. Removing either edge will result in a non-moral graph, so the graph in Fig. 3.19a is a minimal moralization. It is, however, not the minimum, because the graph in Fig. 3.19b is the minimum moralization of the original graph.



(a) Minimal moralization.

(b) Minimum moralization.

Fig. 3.19 Examples of minimal and minimum moralizations.

It follows from the NP-completeness of deciding morality proved by Verma and Pearl [91] that both minimal and minimum moralizations are NP-hard optimization problems as stated in the following theorem.

Theorem 3.4.1. *Let G be a graph. It is NP-hard to find the minimum and minimal moralization of G.*

Proof. Assuming there is an efficient algorithm to find the minimum (or minimal) moralization of the G. Then such an algorithm can be used to decide morality in polynomial time by testing if the minimum set of filled-in edges is empty or not. This contradicts with the NP-completeness of deciding morality. □

Due to the NP-hardness of minimal and minimum moralizations, we will look into heuristics that efficiently produce a moral graph from a given undirected graph in Chapter 5. The rest of this section proves the connection between minimal moralization and minimal elimination kit that is useful for the work in Chapter 5.

Section 3.1 introduced elimination kit and proved that a graph is moral if and only if it has a perfect elimination kit (Theorem 3.2.2). Before defining when an elimination kit is minimal, we introduce a revised version of the *Elimination Game (EG)* algorithm in Algorithm 5. The EG algorithm was originally developed in [65] for triangulating graphs. For a given graph and a node order, the procedure follows the order and makes the node at each step simplicial. It then removes this simplicial node and the edges incident to it. The amended version here takes a set of excesses (i.e., a set of edges to remove when deleting a simplicial node) as an additional input. So after making a node simplicial in the given order, the algorithm removes this simplicial node, the edges incident to it and the excess for this node. The output of this algorithm w.r.t. an elimination kit κ is denoted by G_{κ}^{+}.

Definition 3.4.3. *Let $\kappa = (\alpha, \varepsilon_{\alpha})$ be an elimination kit of a graph G. It is* ***minimal*** *if there is no other elimination kit κ' such that $G_{\kappa'}^{+}$ is a proper subgraph of G_{κ}^{+}.*

The consequence of Theorem 3.4.2 is that finding a minimal moralization is equivalent to finding a minimal elimination kit. It is proved by the following two lemmas that are similar to the work in [62] for graph triangulation.

Algorithm 5: Elimination game algorithm for graph moralization

```
Input  : A graph $G = (V,E)$ and an elimination kit $\kappa = (\alpha, \varepsilon_\alpha)$, where
         $\alpha = (v_1, \dots, v_n)$ of $V$ and $\varepsilon_\alpha = (\varepsilon_1, \dots, \varepsilon_n)$
Output : The filled graph $G^+_\kappa$
$G^0 = G$ ;
for $i = 1$ to $n$ do
    $F^i = D_{G^{i-1}}(v_i)$ ;                                  // Deficiency of $v_i$
    $G^i = G^{i-1} + F^i - v_i - \varepsilon_i$ ;               // Adding $F^i$ and removing $v_i$
end
$G^+_\alpha = (V, E \cup \bigcup_{i=1}^n F^i)$;
```

Lemma 3.4.1. *Let $G = (V,E)$ be a graph and $H = (V, E \cup F)$ be a minimal moralization of G. Then there exists a minimal elimination kit κ such that $G^+_\kappa = H$.*

Proof. The graph H is moral implies it has a perfect elimination kit κ. Applying Algorithm 5 on the graph G with the kit κ will produce another moral graph G^+_κ that is identical to the graph H. Since H is minimal, by definition it has no proper subgraph that is both a supergraph of G and moral. Therefore, the elimination kit κ is a minimal elimination kit for G. □

Lemma 3.4.2. *Let κ be a minimal elimination kit of a graph $G = (V,E)$. Then G^+_κ is a minimal moralization of G.*

Proof. Assuming $G^+_\kappa = (V, E \cup E')$ is not minimal. That is, there exists another set of fill edges $F \subsetneq E'$ satisfying $H = (V, E \cup F)$ being moral. Without loss of generality, assuming H is a minimal moralization of the graph G. Lemma 3.4.1 implies that there exists a minimal elimination kit κ' for the graph H such that $H = G^+_{\kappa'} \subsetneq G^+_\kappa$. It contradicts with the premise that κ is a minimal elimination kit. □

Theorem 3.4.2. *A graph H is a minimal moralization of G if and only if there exists a minimal elimination kit κ such that $H = G^+_\kappa$.*

Proof. The theorem follow from Lemma 3.4.1 and Lemma 3.4.2. □

3.5 Summary

In this chapter, we emphasized the importance of Markov blanket consistency for structure learning and feature selection that use the Markov blanket approach. We drew a connection between checking Markov blanket consistency and graph morality. Despite the NP-completeness of deciding morality, we proposed a linear and quadratic time algorithms for deciding morality for graphs with maximum degrees at most 3 and 4 respectively. The algorithms were developed based on the notion of weakly recursively simplicial.[3] We also closed the gap between P and NP-complete for this problem by a minor modification of the proof in [91] to always produces graphs with maximum degree 5.

Here we address two practical concerns in regards to morality. They will be experimentally justified in Chapter 5. First, if the learned Markov blankets do not form a moral graph (i.e., they are not consistent with any DAG), what is the "easiest" way of enforcing morality? The NP-completeness of deciding morality implies that *minimum (or minimal) moralization* is NP-hard. There are, however, fast ways of enforcing morality that may produce close approximations, such as the *minimum degree (or deficiency)* algorithm (for graph triangulation). Second, can a set of learned Markov blankets (with or without morality enforced) be given as structure priors to a Bayesian structure learner and hence either improve the reconstruction accuracy or running time?

[3] The way a moral graph is decomposed by the notion of weakly recursively simplicial is similar to the steps of getting a component tree for a given Bayesian network in [100].

Chapter 4

Markov Blanket Discovery

Causal discovery automates the learning of causal Bayesian networks from data and has been of active interest from their beginning. With the sourcing of large data sets off the internet, interest in scaling up to very large data sets has grown. One approach to this is to parallelize search using Markov blanket discovery as a first step, followed by a process of combining Markov blankets in a global causal model. We develop and explore three new methods of Markov blanket discovery using Minimum Message Length (MML) and compare them empirically to the best existing methods, whether developed specifically as Markov blanket discovery or as feature selection. While we did this with the ultimate goal of learning global causal models, in this chapter we limit ourselves to the first step only, Markov blanket discovery. Our best MML method is consistently competitive and has some advantageous features as shown in Chapter 5.

Section 4.1 introduces the relevant concepts for MML learning and follow by introducing three models for representing Markov blankets using MML and algorithms for learning them in Section 4.2. The experimental work on comparing our methods against some of the state-of-the-art methods is covered in Chapter 5.

4.1 Minimum Message Length

Our approach to Markov blanket discovery is metric-based. In particular, we apply the Bayesian inferential technique of Minimum Message Length (MML) coding [92]. Here is provided a brief overview of MML and how it is used in this research.

Minimum message length (MML) was devised by Wallace and Boulton [93] as a way of balancing the complexity of a statistical model H with the fit of the model to a given data set D. It implements Bayes' theorem

$$p(H|D) = \frac{p(H,D)}{p(D)} = \frac{p(H) \times p(D|H)}{p(D)},$$

where $p(H)$ is the prior probability distribution of a model, $p(D|H)$ is the likelihood of a data set given this model. In addition, it conforms to Shannon's concept of an efficient code, satisfying

$$I(E) = -log(p(E))$$

to measure the cost or information content for stating an event of probability $p(E)$.[1] Putting these together, the information cost of stating a model and a data set in a two-part message is

$$I(H,D) = I(H) + I(D|H). \tag{4.1}$$

The first part $I(H)$ measures the message length for stating a model (i.e., its structure and parameters for a certain precision). The second part $I(D|H)$ measures how well the specified model compresses the given data set. The aim in MML inference is to find the model having the shortest two-part message length, and so maximizing the posterior probability of H.

[1] Throughout this thesis, we use the natural log to calculate the MML score unless stated otherwise. Information is then measured in "nits", rather than bits.

A feasible approximate method for calculating the total message length is known as *MML87*, from Wallace and Freeman [94]. It approximates the two parts as follows:

$$I(H) = -ln(p(\vec{\theta})) + \frac{1}{2}ln(F(\vec{\theta})) + \frac{|\vec{\theta}|}{2}ln(\kappa_{|\vec{\theta}|}), \tag{4.2}$$

$$I(D|H) = -ln(p(D|H)) + \frac{|\vec{\theta}|}{2}. \tag{4.3}$$

For a given model with a parameter set $\vec{\theta}$ of size $|\vec{\theta}|$, $p(\vec{\theta})$ specifies the parameter prior. The other terms in $I(H)$ give the precision of $\vec{\theta}$, where $F(\vec{\theta})$ is the determinant of the expected Fisher information matrix and $\kappa_{|\vec{\theta}|}$ are lattice constants [92]. The $\frac{|\vec{\theta}|}{2}$ term in $I(D|H)$ is the extra cost of using an estimate with optimal limited precision. (Note that a continuous datum, d, can only ever be measured to limited accuracy, $\pm\frac{\varepsilon}{2}$, so it has not just a probability density, $f(d)$, but a proper probability, $f(d)\cdot\varepsilon$, assuming that the pdf varies slowly around d.)

From equations (4.2) and (4.3), one is able to calculate the total message length if the determinant of the expected Fisher information matrix is calculable, and, in particular, one is interested in knowing the MML estimates of the parameters. Assuming that a data set D of N *i.i.d.* samples of a random variable comes from a multi-state distribution, the total message length to state the hypothesis and data set can be calculated efficiently by

$$I(H,D) = -\ln\left(\frac{(N+r-1)!}{(r-1)!\times\prod_{i=1}^{r} n_i!}\right). \tag{4.4}$$

This was presented by Boulton and Wallace [9] as the factorial form of multistate MML, where the random variable takes r states and each state appears n_i times in D. Equation (4.4) will be shorter than the *MML87* message length by a constant difference of $\ln\frac{\pi e}{6}$ for each parameter, because it does not state the MML estimated parameters.

In this chapter, we no longer deal with graphs only. Instead, we deal with Bayesian networks that consist of a DAG and a joint probability distribution. For this reason, we use upper case letters (in particular X) to represent a random variable. To distinguish, we use the calligraphic font of upper case letters to denote the set of models or data sets. For example, we use $\mathscr{G}$ to denote the set of all DAGs (over the same set of variables) and $\mathscr{D}$ to denote the set of all data sets (over the same set of variables).

Definition 4.1.1. *Let D be a data set of N i.i.d. records sampled from a Bayesian network $< G = (\mathbf{X}, E), P >$. A metric $I : \mathscr{G} \times \mathscr{D} \to \mathbb{R}^+$ from the set of all Bayesian networks and data sets is **decomposable** if it can be written as a sum of scores for each variable X_i given its parent set π_i. That is,*

$$I(G,D) = \sum_{X_i \in X} I(X_i | \pi_{\mathbf{i}}, D).$$

Decompability simplifies the calculation of a metric. For example, the second part of MML corresponds to message length of sending the data set given a hypothesis. This can be factorized into a sum of message length for each node, i.e., negative log probability for each node given its parent set. Alternative metrics used in causal discovery, such as BDe, MDL, K2, are also decomposable.

Here we make a few more assumptions. Besides faithfulness, we only consider Bayesian networks with discrete variables and no hidden variables. We assume the parameters are independent and obey a uniform distribution $U(0,1)$ (which is generalized to the symmetric Dirichlet distribution in the next section), so the parameter prior can be dealt with individually for each variable. The next two definitions and propositions are adapted from Chickering [14].

Definition 4.1.2. *Let P be a joint probability distribution of the random variables in X, and $G = (X, E)$ be a directed acyclic graph. We say G **entails** the conditional independency $X_i \perp\!\!\!\perp_P X_j \mid X_k$ for some variables $X_i, X_j, X_k \in X$, if the conditional independency holds for every joint probability distribution P such that $< G, P >$ satisfies the Markov condition.*

Definition 4.1.3. *A directed acyclic graph G is called an* ***independence-map (or I-map)*** *of a joint probability distribution P, if G entails all the conditional independencies in P.*

I-mapness is defined structurally, without regard to parameters. Although Markov blankets, regional structures and Bayesian networks all focus on (hypothetical) structures, their scores are calculated based on an agreed parameter estimation method. Therefore, we introduce *parameterized I-map*, which refers to a DAG that is an I-map of a distribution and its parameters are obtained by maximum likelihood estimation. With this concept, we can define consistency and local consistency.

Definition 4.1.4. *Let D be a data set of N i.i.d. records sampled from a joint probability distribution P over a variable set X. Assume $G_1 = (X, E_1)$ and $G_2 = (X, E_2)$ are distinct directed acyclic graphs. A metric $I : \mathscr{G} \times \mathscr{D} \to \mathbb{R}^+$ from the set of all directed acyclic graphs and data sets over the variable set X that measures the information content for stating a model and the given data set is* ***consistent*** *if the following hold:*

1. *if G_1 is an I-map of P and G_2 is not, then $\lim_{n\to\infty} I(G_1, D) < \lim_{n\to\infty} I(G_2, D)$,*
2. *if G_1 and G_2 are both parameterized I-maps of P and G_1 has fewer parameters than G_2, then $\lim_{n\to\infty} I(G_1, D) < \lim_{n\to\infty} I(G_2, D)$.*

Definition 4.1.5. *Let D be a data set of N i.i.d. records sampled from a probability distribution P over a variable set X. Assume $G_1 = (X, E_1)$ and $G_2 = (X, E_2)$ are two directed acyclic graphs such that $E_2 = E_1 \cup \{X_i \to X_j\}$. A consistent metric $I : \mathscr{G} \times \mathscr{D} \to \mathbb{R}^+$ from the set of all directed acyclic graphs and data sets over the variable set X that measures the information content for stating a model and the given data set is* ***locally consistent*** *if the following hold:*

1. *if $X_i \not\perp\!\!\!\perp_P X_j \mid \pi_j^{G_1}$, then $\lim_{n\to\infty} I(G_2, D) < \lim_{n\to\infty} I(G_1, D)$,*
2. *if $X_i \perp\!\!\!\perp_P X_j \mid \pi_j^{G_1}$, then $\lim_{n\to\infty} I(G_1, D) < \lim_{n\to\infty} I(G_2, D)$,*

where $\pi_j^{G_1}$ is the parent set of X_j in G_1.

Proposition 4.1.1. *Under the assumptions made above, MML is a consistent scoring function.*

Proof. Since the models considered in this paper are discrete and have no hidden variables, they belong to the curved exponential family [37]. According to equations (4.2) and (4.3), the total message length can be expressed as

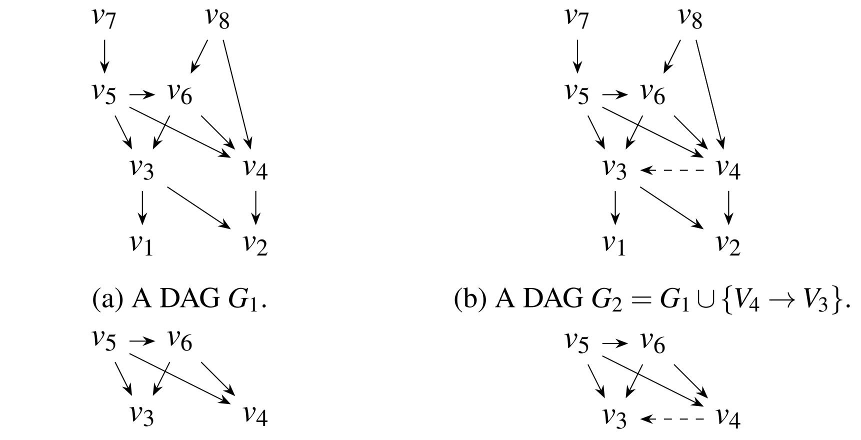
$$I(H,D) = -\left(ln(p(D|H)) - \frac{|\vec{\theta}|}{2} a_N \right), \text{ where}$$
$$a_N = 1 - \frac{2ln(p(\vec{\theta}))}{|\vec{\theta}|} + \frac{1}{|\vec{\theta}|} ln(F(\vec{\theta})) + ln(\kappa_{|\vec{\theta}|})$$

The only term in a_N that is a function of sample size N is the determinant of the expected Fisher information matrix. The likelihood grows linearly with N, so the determinant of the expected Fisher information matrix grows as $N^{|\theta|}$. Hence, the log of the determinant of FIM grows as $|\theta|\log N$. Consequently, as $N \to \infty$, the term $a_N \to \infty$ slower than N, so $a_N/N \to 0$. From Haughton [40], it follows that MML must be a consistent scoring function.[2] □

Using consistency and decomposability, one can prove that MML is a locally consistent scoring function. This allows MML to find the optimal Markov blanket in the limit of infinite data.

Proposition 4.1.2. *Under the assumptions made above, MML is a locally consistent scoring function.*

[2] Haughton's [1988] result for consistent scoring functions applies to both the linear and curved exponential families. The linear exponential family contains undirected graphical models that have no hidden variables [37]. The curved exponential family contains directed acyclic graphs, chain graphs without hidden variables and several families of models (e.g., decision trees) that can approximate a full CPT. Geiger et al. [37] treated graphical acyclic models with hidden variables in the stratified exponential family and emphasized that Haughton's [1988] argument does not extend to them because some of his assumptions are violated in this family.



(a) A DAG G_1.

(b) A DAG $G_2 = G_1 \cup \{V_4 \to V_3\}$.

(c) A DAG H_1 s.t. $\pi_3^{H_1} = \pi_3^{G_1}$.

(d) A DAG $H_2 = H_1 \cup \{V_4 \to V_3\}$.

Fig. 4.1 The score change between H_1 and H_2 is identical to the score change between G_1 and G_2 because of the decomposibility of MML.

Proof. Let D be a set of *i.i.d.* samples generated from a distribution P over a variable set X. Let $G_1 = (X, E_1)$ and $G_2 = (X, E_1 \cup \{X_i \to X_j\})$ be two DAGs different by exactly one edge, e.g., as shown in Figure 4.1a and 4.1b. Then there is a pair of DAGs $H_1 = (Y, F_1)$ and $H_2 = (Y, F_1 \cup \{X_i \to X_j\})$ over a subset $Y \subseteq X$ of variables such that in H_1 the parent set for the variable X_j satisfies $\pi_j^{H_1} = \pi_j^{G_1}$ and H_2 is a complete DAG, such as shown in Figure 4.1c and 4.1d. If $X_i \not\!\perp\!\!\!\perp_P X_j \mid \pi_j^{G_1}$ then $X_i \not\!\perp\!\!\!\perp_{P|_Y} X_j \mid \pi_j^{H_1}$, so H_2 is an I-map of the joint distribution $P|_Y$ restricted to the variable subset Y whilst H_1 is not. By decomposability and consistency of MML we have $\lim_{n\to\infty} I(G_1, D) - \lim_{n\to\infty} I(G_2, D) = \lim_{n\to\infty} I(H_1, D|_{\mathbf{Y}}) - \lim_{n\to\infty} I(H_2, D|_{\mathbf{Y}}) = d > 0$.

On the other hand, if $X_i \perp\!\!\!\perp_P X_j \mid \pi_j^{G_1}$, then both H_1 and H_2 are parameterized I-maps of $P|_Y$ but the former has fewer parameters than the latter. Hence, $\lim_{n\to\infty} I(G_1, D) - \lim_{n\to\infty} I(G_2, D) = \lim_{n\to\infty} I(H_1, D|_{\mathbf{Y}}) - \lim_{n\to\infty} I(H_2, D|_{\mathbf{Y}}) = d < 0$. □

It is worth noting that, all other things being equal, MML differentiates between DAGs in the same Markov equivalence class, according to a prior inductive bias favouring simpler models, where simplicity refers to fewer model parameters. Such a prior is consistent with Occam's Razor, which avoids models with unnecessary complexity.

4.2 Learning Markov Blanket using MML

The problem we set ourselves was to search the space of Markov blankets for each variable in a data set to find a complete set of Markov blankets that minimizes an MML score (equivalently, maximizes the corresponding posterior Bayesian score). Of course, in principle this involves searching the exponential space of all possible subsets of variables, so we used a heuristic greedy search rather than exhaustive search. For MML to operate, we also had to define a model space for representing the probability distribution of each target variable given its Markov blanket. The ideal model structure would be the subgraph of the true DAG induced by the Markov blanket, on the general principle that you can't outdo the truth. But since we don't know the true causal DAG, we tried a variety of models which can plausibly do a good job of representing that conditional probability distribution: a Conditional Probability Table (CPT) reflecting all Markov blanket variables as parents of the target, which maximizes the number of parameters, meaning it has maximal representational power at the expense of requiring the most data to parameterize accurately; a Naive Bayes (NB) model that assumes independence between all Markov blanket variables given the value of the target variable, which minimizes the number of parameters at the expense of misrepresenting dependencies between them; and Markov Blanket Polytrees (MBPs), which compromise between these two extremes by representing Markov blanket variables as related to the target variable and other Markov blanket variables via a singly connected DAG. There are many other alternative local models discussed in the broader literature (e.g., Neil et al. [59]), but here we limit ourselves to these three.

We now explain each of these models and their MML scores in detail.

4.2.1 MML for Conditional Probability Table Models

For any discrete variable $X_i \in X$, its probability density function conditioning on the full joint distribution of its parents set π_i can be expressed by a $r_i \times r_{\pi_i}$ conditional probability table, where r_i and r_{π_i} are the number of states of X_i and π_i respectively (while the densities of continuous variables can be approximated by such a table). We use a CPT model to describe the relationship between a target and its Markov blanket variables by treating those variables as if they are all parents, without claiming that they actually are all parents, much as in a multiple regression model. A full CPT can capture any interactions between the Markov blanket variables (e.g., an XOR) as long as there are enough data to support effective parameterization; this is a requirement that grows exponentially in r_{π_i}. We use $\phi_i(S)$ to denote the CPT model of X_i with a subset $S \subseteq X$ being the hypothetical parent set of X_i.

The parent instantiations partition X_i into r_s multi-state distributions. By the parameter independence assumption, the message length of a CPT model is a sum of the message length of each multi-state distribution over all r_s partitions [93]. Assuming the parameters follow Dirichlet distributions, multi-state MML can be applied. Hence, the total message length for stating a CPT model $\phi_i(S)$ with the hypothetical parent set S and the data D_{X_i} for the target X_i is

$$I(\phi_i(S), D_{X_i}) = \sum_{j=1}^{r_s} \ln\left(\frac{(n_j+\alpha_0-1)!\prod_{k=1}^{r_i}(\alpha_k-1)!}{(\alpha_0-1)!\prod_{k=1}^{r_i}(n_{jk}+\alpha_k-1)!}\right) + \frac{r_s(r_i-1)}{2}\ln\frac{\pi e}{6}, \tag{4.5}$$

where $\vec{\alpha} = (\alpha_1, \dots, \alpha_{r_i})$ is the vector of Dirichlet's concentration parameters for the variable X_i. The parameter value of each of X_i's state is controlled by the corresponding $\alpha_k \in \vec{\alpha}$ and define $\alpha_0 = \sum \vec{\alpha}$. The term n_{jk} is the count of matching data points for π_i being in state j and X_i in state k, and $n_j = \sum_{k=1}^{r_i} n_{jk}$. In the tested models we have no prior knowledge favoring one state over another, so we used a symmetric Dirichlet distribution with the concentration parameter $\alpha_{r_i} = 1$ for all states r_i of a variable X_i. This is the same as saying that each variable's state parameter follows a uniform distribution, so the above equation generalizes equation 4.4. Note since the symmetric Dirichlet distribution with concentration parameter 1 is equivalent to a uniform distribution, the rest of this thesis use the term "uniform distribution" for simplicity, unless the concentration parameter takes a different value.

Suppose only a learned CPT model is used to encode a data set, the next proposition shows that the shortest MML code length in the limit is achieved when the hypothetical parent set of X_i equals MB_i.

Proposition 4.2.1. *Let D be a data set with N i.i.d. records sampled from a joint probability distribution P over variables $X = \{X_1, \dots, X_n\}$. The MML score for stating a CPT model of X_i and the given data set satisfies the following:*

$$\lim_{n\to\infty} I(\phi_i(MB_i), D_{X_i}) < \lim_{n\to\infty} I(\phi_i(S), D_{X_i}), \forall S \subseteq X \text{ s.t. } S \neq MB_i.$$

Proof. Suppose the subset $S = MB_i \cup \{X_j\}$ such that the variable $X_j \notin MB_i$. Let G_1 be a DAG over $\{X_i, X_j\} \cup MB_i$ such that the parent sets $\pi_i^{G_1} = MB_i$ and $\pi_k^{G_1} = \emptyset$ for all $X_k \in \{X_j\} \cup MB_i$. In addition, let G_2 be the same as G_1 but with an additional edge $X_i - X_j$. Since $X_j \notin MB_i$, we have $X_i \perp\!\!\!\perp_P X_j | \pi_i^{G_1}$. By the local consistency of MML, the scores of the two models satisfy $\lim_{n\to\infty} I(G_1, D_{G_1}) < \lim_{n\to\infty} I(G_2, D_{G_2})$. Since all variables in G_1, G_2 have the same parent sets except for X_i and MML is a decomposable scoring function, we have $\lim_{n\to\infty} I(\phi_i(MB_i), D_{X_i}) = \lim_{n\to\infty} I(X_i | MB_i, D_{X_i}) < \lim_{n\to\infty} I(X_i | S, D_{X_i}) = \lim_{n\to\infty} I(\phi_i(S), D_{X_i})$.

Suppose the subset $S = MB_i \setminus \{X_j\}$. Similarly, define G_1, G_2 as above but $\pi_i^{G_1} = S$. Since $X_j \in MB_i$, it implies that $X_i \not\perp\!\!\!\perp_P X_j | \pi_i^{G_1}$. Then local consistency implies $\lim_{n\to\infty} I(G_2, D_{G_2}) < \lim_{n\to\infty} I(G_1, D_{G_1})$. For the same reasons, we have $\lim_{n\to\infty} I(\phi_i(MB_i), D_{X_i}) < \lim_{n\to\infty} I(\phi_i(S), D_{X_i})$. □

4.2.2 MML for Naive Bayes Models

Naive Bayes (NB) models invert the structure of regressions: a central (target) variable is treated as the parent of the other attributes, inducing a marginal dependency between every pair (if faithful), while inducing a conditional independency between them. It is very popular in machine learning for two reasons: it minimizes the number of parameters, making it useful even in data poor environments and it works reasonably well on many problems, even many that violate the independence assumption so long as the dependencies omitted are not overly strong. With the conditional independence assumption, naive Bayes parameters increase linearly in the number of variables, which makes it useful for dealing with large problems. The posterior probability of the target variable X_i given a hypothetical child set S is

$$p(X_i|S) = \frac{p(X_i)\prod_{X_j \in S} p(X_j|X_i)}{\sum_{x_i=1}^{r_i} p(x_i)\prod_{X_j \in S} p(X_j|x_i)}, \tag{4.6}$$

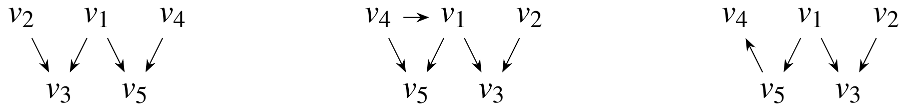
where $p(x_i)$ is a short for $p(X_i = x_i)$. Each term $p(X_i)$ and $p(X_j|X_i)$ is a single or a set of multi-state distributions, so can be calculated using adaptive coding MML by Equation 4.5 or Equation 4.4 if priors are assumed to come from a symmetric Dirichlet distribution. Hence, the total message length for stating a naive Bayes model and the data for the target is

$$I(\phi_i(S), D_{X_i}) = -\ln p(X_i) - \sum_{X_j \in S} \ln p(X_j|X_i) + \ln \sum_{x_i=1}^{r_i} p(x_i) \prod_{X_j \in S} p(X_j|x_i). \tag{4.7}$$

Notice that this is the message length omitting the MML estimate of the parameters. (Note that in this thesis the main motivation of studying Markov blanket discovery is to help scale up structure learning. Hence, the learned Markov blankets will not be used for predicting target variables. This makes learning the parameters of the underlying models for Markov blanket discovery unnecessary. Hence, the message length calculated in this thesis do not contain the MML estimate of the parameters. This, however, can be changed by taking into account the additional message length for stating the parameters which was mentioned in Equation 4.5.)

4.2.3 MML for Markov Blanket Polytree Model

Between the extremes of a regression structure, with all attributes as independent parents of the target, and naive Bayes models, with all attributes as isolated children, come almost every other possible DAG structure relating Markov blanket variables with their target. The true model is likely to be amongst them, but as with many learning problems where the truth is unknown, some ensembling approach suggests itself as a way of approximating the truth. Here we use an ensembling method that samples as many local polytrees as possible, then outputs a weighted average message length over all the samples. This way the Markov blanket variables are encoded by a good variety of network structures, allowing many interactions to be modelled, but is nevertheless limited, so that the number of model parameters on average is less than that of local DAG's.



(a) A MBP for v_1. (b) A non-MBP for v_1. (c) A non-MBP for v_1.

Fig. 4.2 Examples of MBP and non-MBPs for the variable v_1. The MB of v_1 is $\{v_2,\dots,v_5\}$.

We call the restricted regional structures being sampled Markov Blanket Polytrees (MBPs). A polytree is a DAG such that its underlying undirected graph is a tree, such as Fig. 4.2b and 4.2c.

Definition 4.2.1. *Let $< G = (X,E),P >$ be a Bayesian network. A* ***Markov blanket polytree*** *T_i of a target variable X_i is a polytree over the variables $\{X_i\} \cup MB_i$ such that*

$$MB^{T_i}(X_i) = MB^G(X_i).$$

Fig. 4.2a is an example of a MBP for the variable v_1, given the Markov blanket of v_1 is $\{v_2,\dots,v_5\}$. The other two are not MBPs, because Fig. 4.2b is not a polytree, Fig. 4.2c entails a different Markov blanket for v_1.

The next proposition presents a recursive formula for counting the number of labeled MBPs over a set of n variables.

Proposition 4.2.2. *Let Y be a variable whose Markov blanket contains $n \in [1,\infty)$ variables. The number $f(n)$ of labeled Markov blanket polytrees for Y can be computed by the following* *recursive equation*

$$f(n) = \sum_{i=0}^{n} \binom{n}{i} + \sum_{m=1}^{\lfloor \frac{n}{2} \rfloor} \sum_{k=1}^{n-2m+1} g(n,m,k), \tag{4.8}$$
$$g(n,m,k) = \binom{n}{k+1}(k+1) \sum_{k'=1}^{\min\{k,n-k-2(m-1)\}} \frac{q}{m} \cdot g(n-k-1,m-1,k'),$$

where $q = 1$ *if* $k = k'$ *and* m *otherwise.*

Proof. It is trivial to bound the number of colliders $m \in \left[0, \lfloor \frac{n}{2} \rfloor\right]$.

Case 1: When $m = 0$

MB_i contains only parents and/or children. There are $\binom{n}{i}$ ways of selecting $i \in [0,n]$ children from n labeled nodes. The order of these parents or children does not matter in a polytree. Therefore, the number of labeled MBPs when $m = 0$ is

$$\sum_{i=0}^{n} \binom{n}{i}. \tag{4.9}$$

Case 2: When $m > 0$

Each of Y's children and its spouses (if there are any) forms a branch. The largest branch with k spouses can be enumerated in

$$\binom{n}{k+1}(k+1) \tag{4.10}$$

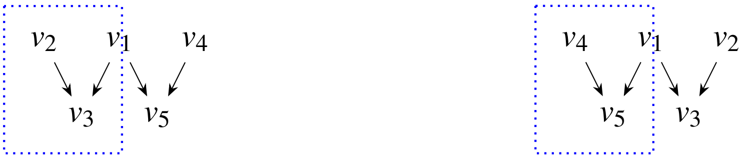
ways, where $k \in [1, n-2m+1]$. There are $\binom{n}{k+1}$ ways selecting $k+1$ nodes to form the largest branch. And each one of the $k+1$ nodes needs to be a common child once in order to fully enumerate all cases. k's upper bound is obtained if each of the other $m-1$ branches contains only a collider and a spouse, in which case $n-2(m-1)-1 = n-2m+1$. Hence, when $m > 0$ the number of MBPs can be obtained by multiplying equation (4.10) with the total enumeration of the remaining $n-k-1$ nodes. The subgraph over the remaining nodes can be counted by the same approach. By doing this recursively, we will end up with a subgraph in which Y has no spouse. It can then be enumerated by equation (4.9). Therefore, the total enumeration of MBPs when $m > 0$ is

$$\sum_{m=1}^{\lfloor \frac{n}{2} \rfloor} \sum_{k=1}^{n-2m+1} g(n,m,k), \tag{4.11}$$

for

$$g(n,m,k) = \binom{n}{k+1}(k+1) \sum_{k'=1}^{\min\{k,n-k-2(m-1)\}} \frac{q}{m} \cdot g(n-k-1,m-1,k'), \tag{4.12}$$

with $q = 1$ if $k = k'$ and m otherwise. The maximum number of spouses k' in a subgraph is bounded above by the minimum between the maximum number of available nodes $n-k-2(m-1)$ and k from its supergraph.



(a) Start counting with labels V_2, V_3.

(b) Start counting with labels V_4, V_5.

Fig. 4.3 An example when double counting the same MBP occurs for two colliders of the same size.

As the largest branch is enumerated independently from the remaining nodes, some of the graphs are counted multiple times. For example, to enumerate the MBPs for the variable V_1 with $n = 4$ Markov blanket candidates, when going through the case where there are two colliders (i.e., $m = 2$), we obtain Figure 4.3a when labelling the largest branch (i.e., left/right) with $\{V2, V3\}$, and Figure 4.3b when labelling the largest branch (i.e., left/right) with $\{V4, V5\}$. The resulting two labelled graphs, however, are identical, and hence we divide the total number by $\frac{1}{2}$. In general, the total number needs to be divided by $\frac{1}{m}$; hence $\frac{q}{m}$ appears in equation (4.12). □

The total number of MBPs is dramatically reduced compared with DAGs, as shown in Table 4.1.

Table 4.1 The number of labeled DAGs and MBPs over different number of nodes $n \in [1,7]$.

# nodes	# DAGs	# MBPTs
1	1	1
2	3	2
3	25	6
4	543	23
5	29281	104
6	3781503	537
7	1138779265	3100

The message length for transmitting data using an MBP model is calculated as the log of the conditional probability

$$p(X_i|S) = \frac{p(X_i \mid \pi_i^{T_i}) \prod_{X_j \in S} p(X_j|\pi_j^{T_i})}{\sum_{x_i=1}^{r_i} p(x_i \mid \pi_i^{T_i}) \prod_{X_j \in S} p(X_j|\pi_j^{T_i})}$$

which is factorized into a product of each variable's probability conditioned on its parent set in a MBP T_i as estimated using the adaptive code method. Hence, the total message length has the form

$$I(\phi_i(S), D_{X_i}) = -\ln p(X_i \mid \pi_i^{T_i}) - \sum_{X_j \in S} \ln p(X_j|\pi_j^{T_i}) + \ln \sum_{x_i=1}^{r_i} p(x_i \mid \pi_i^{T_i}) \prod_{X_j \in S} p(X_j|\pi_j^{T_i}). \tag{4.13}$$

To calculate a weighted average score, we uniformly average the conditional probabilities $p(X_i \mid S)$ over all sampled MBPs in the set $\mathscr{T}_i$ containing the variables $\{X_i\} \cup MB_i$, then take the negative log. The uniform prior can be replaced by any reasonable prior over the possible MBPs.

4.2.4 The MBMML Algorithm

This section presents two MBMML algorithms in pseudocode for learning Markov blankets using either a fixed regional structure (i.e., CPT or NB) or an ensemble of random regional structures (i.e., MBPs).[3] Both algorithms use a greedy search starting with an empty Markov blanket and iteratively adding the highest ranked candidate for reducing the total message length (equation 4.5 or 4.7, or 4.13). Both algorithms stop and output a learned Markov blanket if no scores can be improved by adding a single variable to the current Markov blanket.

Algorithm 6: MB discovery using MBMML+CPT/NB

```
Input  : a data set $D$ over a variable set $X$, a target variable $X_i$ and a predetermined
         model $\phi_i$ that is either CPT or NB model
Output : a set of learned MBs
$S = X \setminus X_i$                                              // unchecked variables
$Z = \emptyset$                                                    // learned MB
$L = I(\phi_i(\emptyset), D_{X_i})$                                // empty model score
while $S \neq \emptyset$ do
    $X_k = \arg\min_{X_j} I(\phi_i(Z \cup \{X_j\}), D_{X_i}), \forall X_j \in S$   // best candidate
    $L' = I(\phi_i(Z \cup \{X_k\}), D_{X_i})$                      // current best score
    if $L' < L$ then
        $Z = Z \cup \{X_k\}$                                       // admit when score reduces
        $S = S \setminus \{X_k\}$
        $L = L'$                                                   // update best score
    else
        Stop
    end
end
Output $Z$
```

To ensure there is no conflict between the learned Markov blankets, we pass the outputs from both MBMML+CPT/NB and MBMML+MBP algorithms to a symmetry enforcement

[3] In this work, when we refer to randomly generated DAGs these refer to networks generated by: uniformly selecting a total ordering of variables; then for each node in order, uniformly select the number of its parents up to min(# of predecessors, the maximum fan-in for the experiment); uniformly select its parents from amongst its predecessors. In the case of MBPs in particular, any arc addition is suppressed if it would introduce an undirected loop.

Algorithm 7: MB discovery using MBMML+MBP

```
Input  : a data set $D$ over a variable set $X$, a target variable $X_i$ and a MBP model $\phi_i$,
         the number $K$ of randomly sampled MBPs
Output : a set of learned MBs
$S = X \setminus X_i$                                          // unchecked variables
$Z = \emptyset$                                                // learned MB
$L = I(\phi_i(\emptyset), D_{X_i})$                            // empty model score
while $S \neq \emptyset$ do
    if $f(|Z|+1) \leq K$                    // number of MBPs by equation 4.8
     then
        $\mathscr{T}_i := \{$all MBPs over $Z \cup \{X_j\}\}$                  // all MBPs
    else
        $\mathscr{T}_i = \{K$ random MBPs over $Z \cup \{X_j\}\}$   // randomly sampled MBPs
    end
    $X_k = \arg\min_{X_j} E_{\mathscr{T}_i}(I(\phi_i(Z \cup \{X_j\}), D_{X_i})), \forall X_j \in S$   // best candidate
    $L' = E_{\mathscr{T}_i}(I(\phi_i(Z \cup \{X_k\}), D_{X_i}))$   // current best expected score
    if $L' < L$ then
        $Z = Z \cup \{X_k\}$                          // admit when score reduces
        $S = S \setminus \{X_k\}$
        $L = L'$                                      // update best score
    else
        Stop
    end
end
Output $Z$
```

algorithm, per §4.2. There are two simple ways of enforcing symmetry, taking either the union or the intersection of two neighbouring Markov blankets. We used Union enforcement for MBMML+CPT, because a CPT model's precision converges to 1 as sample size increases. So, its exponential increase in parameters is likely to result in more false negatives than false positives. We used Intersection enforcement for MBMML+NB, because a naive Bayes model will produce more false positives than a CPT model due to its lack of representational power, but fewer false negatives. It is unclear which enforcement is a better option for MBMML+MBP, but we took Union enforcement. [4]

The process of the symmetry enforcement is shown in Algorithm 8.

Algorithm 8: Symmetry enforcement

```
Input  : a set of learned Markov blankets $\{MB_i\}, \forall X_i \in X$
Output : a set of symmetric Markov blankets
for each $MB_i$ do
    for each $X_j \in MB_i$ do
        if $X_i \notin MB_j$ then
            if Union then
                $MB_j = MB_j \cup \{X_i\}$ ;
            else
                $MB_i = MB_i \setminus \{X_j\}$ ;
            end
        else
            do nothing;
        end
    end
end
Output $\{MB_i\}$
```

[4] A recent publication by Zhao and Ho [102] proposed score-based and constraint-based methods to enforce symmetry. That work appears to have the potential to increase the accuracy of the MML methods presented here but is not in the scope of the current work.

4.2.5 A running example

This subsection provides a running example to illustrate the three algorithms described in the previous subsection. The generating model structure is shown in Figure 4.4. The model contains 8 binary variables, whose parameters were sampled uniformly between 0 and 1.

To illustrate how the proposed algorithms in the previous section finds Markov blanket candidates, we pick V_4 as the target. All three algorithms are based on greedy search that adds the most promising candidate into the learned Markov blanket. The scoring process runs through all remaining nodes to calculate the message length then pick the one with the lowest score. For example, when the MML+CPT algorithm is testing for V_1 being the Markov blanket of V_4, it assumes V_1 is the parent of V_4. The algorithm then runs the given data set to calculates the message length according to Equation 4.5. Assuming the algorithm is given a data set with 1000 samples, in which the number of cases where (V_4, V_1) equals $(0,0),(1,0),(0,1),(1,1)$ is 17,1,64,918 respectively. Then the message length is

$$\begin{aligned} I_{CPT}(V4 \mid V_1) =& \ln\left[\frac{((17+1)+2-1)!}{(2-1)!}\frac{(2-1)!}{(17+1-1)!}\frac{(2-1)!}{(1+1-1)!}\right]+ \\ & \ln\left[\frac{((64+918)+2-1)!}{(2-1)!}\frac{(2-1)!}{(64+2-1)!}\frac{(2-1)!}{(918+2-1)!}\right] \\ \approx& 246.4. \end{aligned}$$

Log factorial is approximated by the Lanczos approximation for fast computation.[5]. The message length of every other remaining node can be calculated in the same way. This yields $I_{CPT}(V_4 \mid V_2) = 287.6, I_{CPT}(V_4 \mid V_3) = 287.5, I_{CPT}(V_4 \mid V_5) = 283.9, I_{CPT}(V_4 \mid V_6) = 281.2, I_{CPT}(V_4 \mid V_7) = 286.3, I_{CPT}(V_4 \mid V_8) = 286.8$. Hence, the best candidate at this step is V_1. Next, the MBMML+CPT algorithm will check each of the remaining nodes $\{V_2, V_3, V_5, V_6, V_7, V_8\}$ and selects the best node $V_i = \arg\min I_{CPT}(V_4 \mid V_1, V_i)$ from them that gives the smallest mes-

[5] https://en.wikipedia.org/wiki/Lanczos_approximation

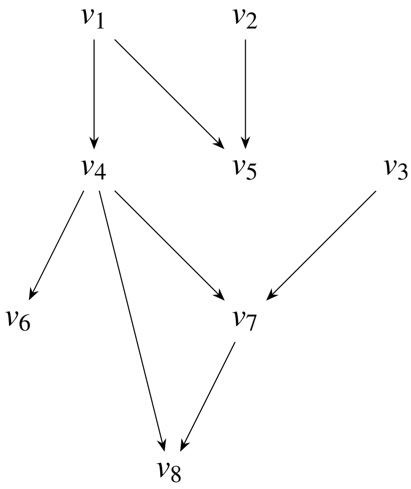


Fig. 4.4 The DAG of the generating model.

sage length. The algorithm is terminated when adding nodes cannot reduce the MML score. In this case, the MBMML+CPT algorithm learns the Markov blanket of V_4 is $\{V_1, V_6\}$.

The MBMML+NB and MBMML+MBP algorithms are similar to the MBMML+CPT algorithm, except that the assumed underlying model structures are not the CPT model. The former assumes each possible Markov blanket candidate is a child of the target V_4 whilst the latter assumes each possible candidate is either a parent, child or spouse of V_4, depending on the specific Markov blanket polytree. The message length of V_4 given a possible candidate for each of these two models can be calculated according to Equation 4.7 and 4.13 respectively.

4.3 Summary

We have proposed three alternative MML methods for learning Markov blankets. The three methods all use the multi-state MML measure, but apply them with different models, namely a conditional probability table model, naive Bayes and an ensemble of random Markov blanket polytrees. We proved that the MBMML+CPT algorithm will find the correct Markov blankets given perfect data (i.e., infinite samples), although it will not be data efficient for large Markov blankets due to the exponential number of parameters required. We looked at one of the more common answers to data inefficiency in naive Bayes models, which sacrifice the modelling of conditional dependencies for speed and simplicity. As a compromise between the correctness but inefficiency of CPT and the efficiency but strong assumptions of naive Bayes, we also explored an ensemble technique in MBMML+MBP, using random polytrees within Markov blankets.

The next chapter covers tests of the three MML algorithms against three of the best alternatives reported in recent literature, with both real data and artificial Bayesian networks at a range of sample sizes.

Chapter 5

Experiment

This chapter includes the experimental work that has been done on testing the three Markov blanket learners presented in Chapter 4, and the potential of graph morality to improve Markov blanket discovery and structure learning accuracy. Section 5.1 compares the accuracy of our Markov blanket learners with three alternatives on both real and synthetic data sets. Section 5.2 tries to improve the quality of learned Markov blankets by enforcing morality with node orders obtained from several methods. The last section explores a probabilistic way of using Markov blankets in a Bayesian structure learner, namely Causal MML (CaMML). We study the potential effect on CaMML's accuracy and scalability by encoding learned Markov blanket in prior probabilities for CaMML.

5.1 Markov Blanket Discovery

This section presents experimental results from testing three Markov blanket discovery methods, MBMML+CPT, MBMML+NB and MBMML+MBP against three leading alternatives:

- Incremental Association Markov Blanket (IAMB) is a constraint-based algorithm that uses a mutual information (MI) test for candidate admission [88]. In phase 1 it adds variables that are interdependent with the target variable when conditioning on the current candidate Markov Blanket. That is followed by a pruning phase, deleting any variables found to be false positives.

- Parents and Children based Markov Blanket (PCMB) is another two-phase constraint-based algorithm [67]. First, it finds the direct neighbours of the target using a conditional independence test (G^2 test). It then finds the neighbours of each neighbour of the target, and prunes away false positives.
- Score-based Local Learning (SLL), by contrast with earlier Markov blanket learners, is metric-based, using the BDeu score in a dynamic programming algorithm [60]. It is an exact algorithm that searches through the entire space of equivalence classes of local DAGs around the target variable, then reads off the Markov Blanket from the optimal DAG. Niinimaki and Parviainen [60] used SLL also to scale up general Bayesian network structure learning.

The implementation of PCMB was provided by Peña et al. [67] and IAMB came from R's *bnlearn* package [77]. The significance level $\alpha = 0.01$ was set for both algorithms for conditional independence tests, as used in [67]. We used SLL's source code from Niinimaki and Parviainen [60] and its default equivalent sample size 1 for BDeu. SLL reverts to the GES algorithm [13] for learning the Markov blanket's regional structure if it finds more than 20 variables in the Markov blanket. The three MML methods assumed the uniform parameter prior (i.e., symmetric Dirichlet with a single concentration parameter $\alpha = 1$). The MBMML+MBP algorithm was set to randomly sample 100 Markov Blanket polytrees from the total polytree space. It then calculates the unweighted average probability of the data given each polytree. This averaged probability is used to calculate an average message of the data given each polytree.

Section 5.1.1 focuses on testing with real models (Table 5.1) and test data sets provided by Tsamardinos et al. [89], which we identify by name. [1] These models and data sets have

[1] http://pages.mtu.edu/~lebrown/supplements/mmhc_paper/mmhc_index.html

Table 5.1 A summary of the tested real and artificial BNs. The artificial BNs contain 30, 50 or 80 variables with maximum fan-in 5, maximum number of states 4 per variable. The artificial BNs' parameters are sampled from uniform distributions.

Network	# variables	Max fan-in	Mean MB size
CHILD	20	2	3
INSURANCE	27	3	5.19
ALARM	37	4	3.51
BARLEY	48	4	5.25
HAILFINDER	56	4	3.54
30-5-4-1	30	5	8
50-5-4-1	50	5	9.73
80-5-4-1	80	5	10.08

often been used for testing Markov Blanket and causal discovery learners. [2] Section 5.1.2 then extends the experiments to artificial Bayesian networks containing 30 or 50 variables, with fixed maximum fan-ins and arities. For each class of model, we randomly generated 5 different Bayesian networks, each of which was then used to generate 5 different data sets for each of the sample sizes 100, 500, 2000, 5000.

Precision, recall, and edit distance were used to evaluate the performance of the different algorithms in finding the true Markov blankets, that is, the percentage of true positives amongst those asserted to be in the Markov blanket (true plus false positives), the percentage of Markov blanket variables that were found (i.e., the ratio true positives: true positives + false negatives), and the sum of false positives and false negatives, respectively. These are all accuracy-oriented measures, which in a strictly methodological (non-applied) study may be about the best we can expect to do. That is, utilities for different kinds of errors or successes cannot be stated in the abstract, so Bayesian evaluation measures are hard to identify. Nevertheless, we might assert some preference for precision over recall, on the grounds that our intended purpose is to improve causal discovery, for example by feeding the results of Markov blanket discovery into a causal discovery process. In that domain, it

[2] These are presumptively the true models, but were in fact built by the researchers, and hence are not guaranteed to be the true generative models.

seems at least plausible that false positives (reducing precision) are more damaging than false negatives (reducing recall), since falsely asserting membership in a Markov blanket would positively mislead subsequent causal discovery, whereas an error of omission leaves causal discovery no worse off than not doing any Markov blanket discovery, with respect to that variable and Markov blanket, at any rate. We do not have the temerity to try to quantify that intuition, however, we shall consider it in interpreting our results. Edit distance is a kind of compromise between these two accuracy measures.

Results are reported using 95% confidence intervals, in line with current APA guidelines [7], either in the main text or appendices.

5.1.1 Accuracy on Real Models

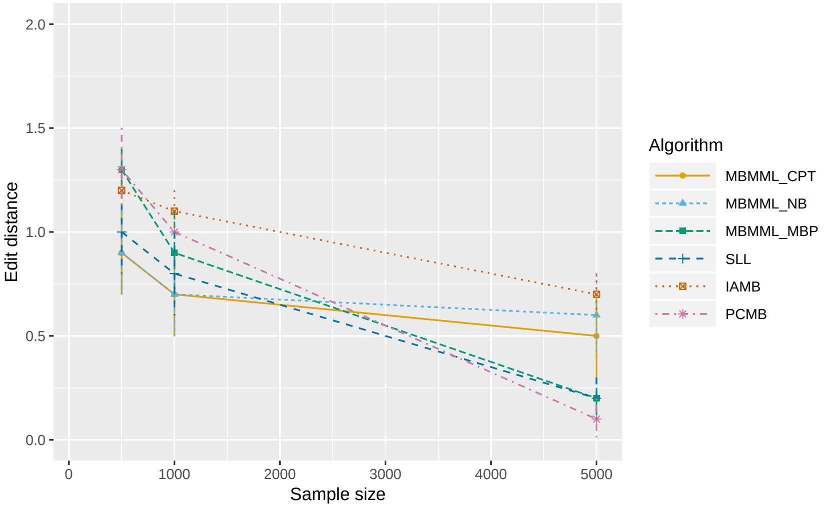


Fig. 5.1 MB learners' edit distances (with 95% confidence intervals) vs. sample size on the CHILD network.

Figures 5.1 and 5.2 report the mean edit distance (with confidence intervals) of all algorithms on the CHILD and BARLEY networks. [3] In most cases, the algorithms show no statistically significant difference from each other, except that PCMB is less robust under small samples. Both PCMB and SLL appear to converge slightly faster than the

[3] Note that PCMB failed to run on the BARLEY network for 1000 and 5000 sample sizes.

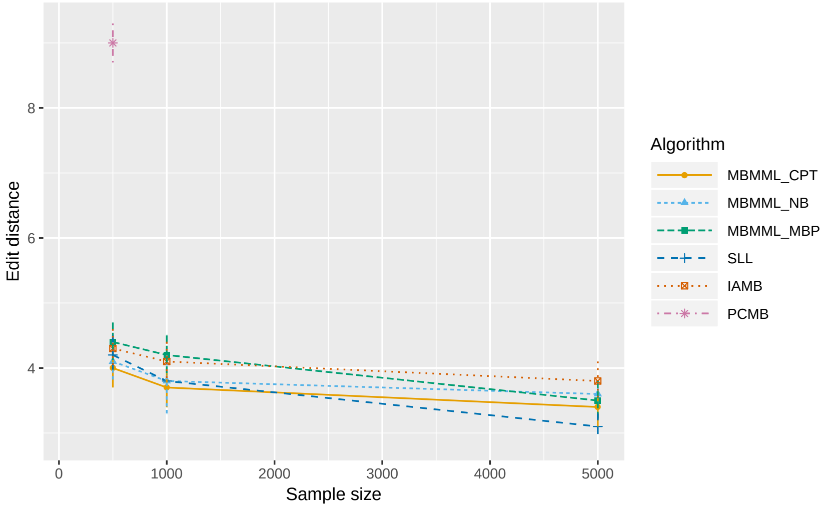


Fig. 5.2 MB learners' edit distances (with 95% confidence intervals) vs. sample size on the BARLEY network. PCMB failed under 1000 and 5000 samples due to an unknown reason.

MML methods and IAMB. The edit distance, precision and recall on all six models are summarized in Tables 5.2 and 5.3 in the next section. Although MBMML+MBP does not show competitive precision, it has the highest recall in all cases with clear margins, especially under small samples. On these real networks, SLL tends to outperform the alternative techniques, with MML+CPT being its equal or near equal.

5.1.2 Accuracy on Artificial Models

It is also useful to test machine learning algorithms on artificial models using generated data, where the ground truth is known. The DAGs we used tended to be more complex than the real models above. While having similar numbers of variables, the fan-in and arity of variables were somewhat higher (cf. Table 5.1). Their parameters were independently sampled from a uniform distribution (i.e., symmetric Dirichlet distribution with concentration parameter $\alpha = 1$) which matched the parameter prior used in the multi-state MML metric. This could provide an advantage to the MML methods, since they assume as much. However, as we discuss more below, we also tested the MBMML+CPT method on models whose parameters are sampled from symmetric Dirichlet distribution with the concentration parameter taking different values. The experimental results show very similar performance when the MML method using the uniform and non-uniform priors, suggesting that MML performance does not vary much when the parameter assumptions are approximately correct.

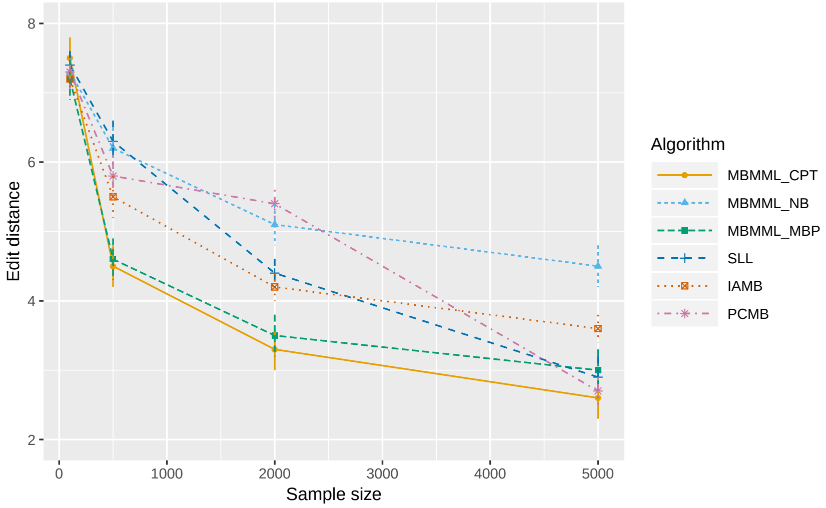


Fig. 5.3 MB learners' edit distances (with 95% confidence intervals) vs. sample size on artificial BNs (30-5-4-1) containing 30 variables, maximum 5 parents and maximum 4 states for each variable.

Figure 5.3 shows how the different algorithms perform with different sample sizes from the artificial networks. All algorithms perform similarly with very small samples (i.e., 100), while MBMML+NB and IAMB fall away from the pack at large samples (i.e., 5000). Ignoring MBMML+NB and IAMB, there is nothing to choose between the algorithms in terms of edit distance at 5000 samples. For moderate sample sizes (i.e., 500 and 2000), MBMML+CPT and MBMML+MBP show significantly lower edit distances than the others. This suggests that the explanatory power of the CPT model is significantly improved when increasing the sample size from small to medium. Looking at precision and recall (Table 5.3), MBMML+CPT and MBMML+MBP have the highest recall in all cases, with MBMML+CPT's precision being the highest under medium and large samples. Figures 5.4 and 5.5 show similar trends as Figure 5.3, except for the case of 100 samples where MBMML+CPT and MBMML+MBP appear to have higher edit distance than the others. Looking at their precision and recall

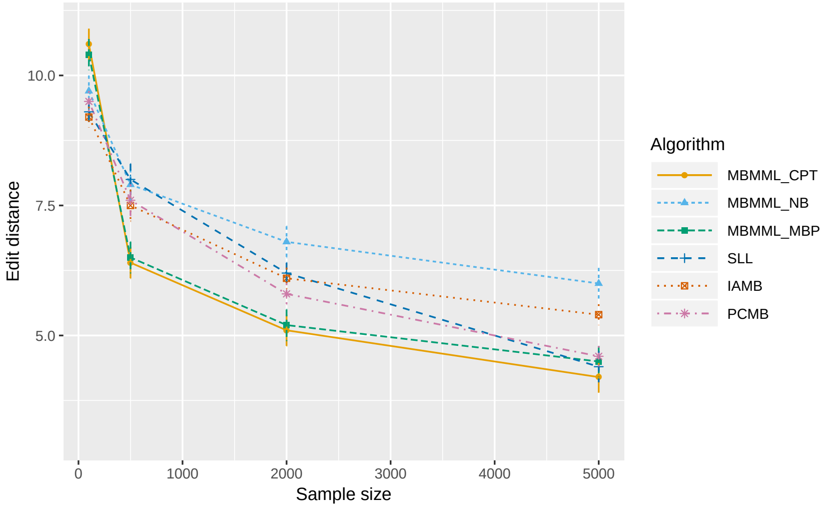


Fig. 5.4 MB learners' edit distances (with 95% confidence intervals) vs. sample size on artificial BNs (50-5-4-1) containing 50 variables, maximum 5 parents and maximum 4 states for each variable.

(Table 5.3) suggests that both models tend to underfit with 100 samples, but the problem is fixed by feeding in more data. Note that we did not run MBMML+MBP on 80-5-4-1 due to its high computational cost. Also, SLL's edit distance does not drop much for 5000 samples, suggesting difficulties with larger 80-variable Bayesian networks even given large samples. It is worth pointing out that SLL did poorly for all 25 data sets of 5000 samples in this case, although the program falls back to an approximate algorithm for large Markov blankets.

Figure 5.6 shows results for 50 variable networks and a fixed sample size of 500. Average Markov blanket sizes are shown on the X-axis; these were not controlled for, so the results simply reflect a correlation between larger Markov Blankets and poorer accuracy, which is to be expected of course. While on networks with small Markov blankets all the algorithms perform similarly, the MBMML+CPT and MBMML+MBP algorithms are clearly outperforming the rest when Markov blankets contain 20 or more variables.[4] That is, these MML algorithms clearly recommend themselves for dealing with more complex discovery

[4] It is worth noting that for these larger Markov blankets, SLL reverts to the GES algorithm.

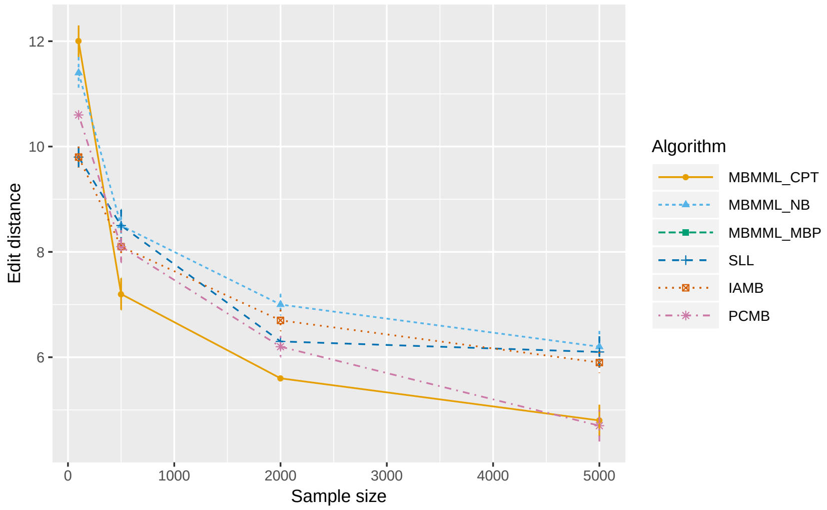


Fig. 5.5 MB learners' edit distances (with 95% confidence intervals) vs. sample size on artificial BNs (80-5-4-1) containing 80 variables, maximum 5 parents and maximum 4 states for each variable.

problems with moderate sample sizes. Overall, MBMML+CPT and MBMML+MBP have the lowest edit distances, which is consistent with the ranking in Figure 5.4 for 500 samples.

Upping the sample size to 5000 (Figure 5.7), we find a somewhat different story. While MBMML+CPT/MBP are always competitive, only occasionally being significantly worse than the best performer per Markov blanket size (as shown by non-overlapping CIs), PCMB and SLL (or GES) also are performing well across the board with the larger samples. IAMB and MBMML+NB are pretty clearly underperforming at larger samples and larger Markov blankets, while MBMML+NB does well with smaller Markov blankets and large samples.

To sum up, IAMB has clearly been superceded by subsequently developed algorithms. PCMB and SLL show some weaknesses in small and moderate sized samples, but perform as well or better than alternatives given larger samples. MBMML+NB does well with small samples, while MBMML+CPT/MBP perform well across the board.

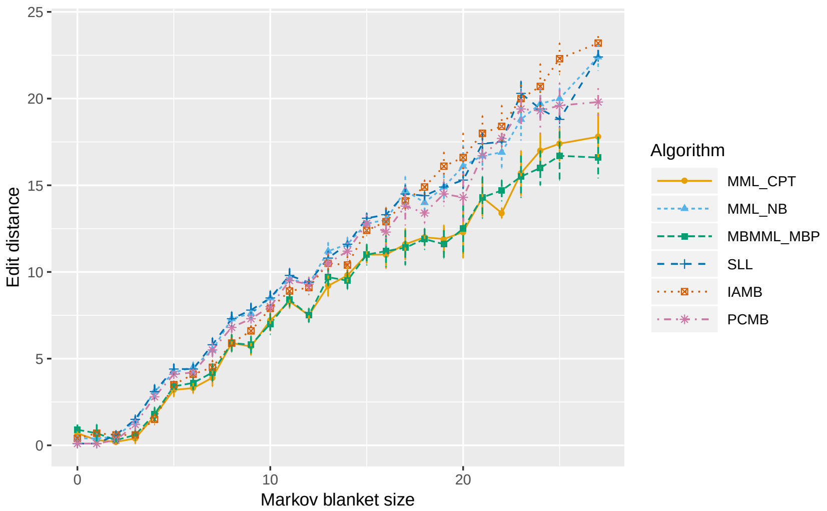


Fig. 5.6 MB learners' edit distances (with 95% confidence intervals) vs. average MB size on artificial BNs (50-5-4-1) with 500 samples.

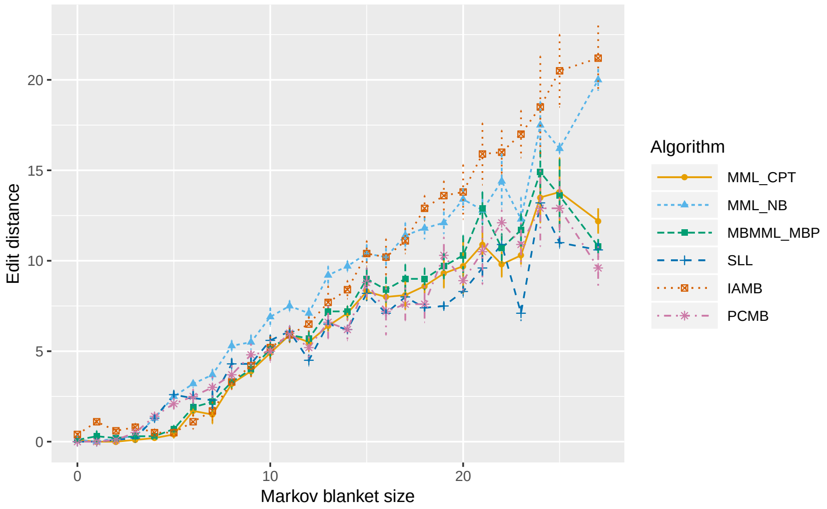


Fig. 5.7 MB learners' edit distances (with 95% confidence intervals) vs. MB size on artificial BNs (50-5-4-1) with 5000 samples.

Table 5.2 A summary of MB learners' edit distances (with 95% confidence intervals) on both real and artificial BNs. The best results are highlighted in pink. In real networks, SLL wins in all cases followed by MBMML+CPT, PCMB, MBMML+NB, IAMB and MMLL+MBP. Noted PCMB failed to run on BARLEY networks with 1000 and 5000 samples due to unknown reasons. In artificial networks, MBMML+CPT and MBMML+MBP win most of the times followed by SLL/PCMB and MBMML+NB/IAMB. We did not run MBMML+MBP on 80-5-4-1 due to its high running time. Note that SLL's edit distance is much worse on 80-5-4-1 models under 5000 samples.

Network	SAMPLES	*MBMML +CPT*	*MBMML +NB*	*MBMML +MBP*	IAMB	PCMB	SLL
CHILD	500	0.9+-0.2	0.9+-0.2	1.3+-0.2	1.2+-0.2	1.3+-0.2	1+-0.2
	1000	0.7+-0.1	0.7+-0.2	0.9+-0.1	1.1+-0.2	1+-0.2	0.8+-0.1
	5000	0.5+-0.1	0.6+-0.1	0.2+-0.1	0.7+-0.2	0.1+-0	0.2+-0.1
INSURANCE	500	3.3+-0.2	3.5+-0.2	4+-0.3	3.4+-0.3	3.2+-0.2	3.1+-0.2
	1000	2.9+-0.2	3.3+-0.2	3.5+-0.3	3.1+-0.3	2.9+-0.2	2.7+-0.2
	5000	2.1+-0.2	2.8+-0.2	2.4+-0.2	2.7+-0.2	1.8+-0.2	2+-0.2
ALARM	500	1.4+-0.1	2.1+-0.2	3.4+-0.2	1.9+-0.2	1.5+-0.1	0.8+-0.1
	1000	1+-0.1	1.8+-0.2	2.8+-0.2	1.6+-0.2	1.1+-0.1	0.6+-0.1
	5000	0.5+-0.1	1.5+-0.2	1.7+-0.1	1.3+-0.2	0.3+-0.1	0.2+-0
BARLEY	500	4+-0.3	4.1+-0.3	4.4+-0.3	4.3+-0.3	9+-0.5	4.2+-0.2
	1000	3.7+-0.3	3.8+-0.3	4.2+-0.3	4.1+-0.3	NA	3.8+-0.2
	5000	3.4+-0.3	3.6+-0.3	3.5+-0.3	3.8+-0.3	NA	3.1+-0.2
HAILFINDER	500	4.4+-0.3	4.3+-0.2	5.2+-0.3	4.1+-0.2	7.1+-0.5	4.3+-0.3
	1000	4.4+-0.3	4.3+-0.2	5+-0.3	4.1+-0.2	6.2+-0.4	4.1+-0.3
	5000	4.3+-0.3	4.3+-0.2	5.1+-0.3	4.2+-0.2	3.8+-0.2	4+-0.3
30-5-4-1	100	7.5+-0.3	7.3+-0.3	7.2+-0.3	7.2+-0.3	7.3+-0.3	7.4+-0.3
	500	4.5+-0.3	6.2+-0.3	4.6+-0.3	5.5+-0.3	5.8+-0.3	6.3+-0.3
	2000	3.3+-0.2	5.1+-0.3	3.5+-0.2	4.2+-0.3	5.4+-0.2	4.4+-0.3
	5000	2.6+-0.2	4.5+-0.2	3+-0.2	3.6+-0.3	2.7+-0.2	2.9+-0.2
50-5-4-1	100	10.66+-0.3	9.7+-0.3	10.4+-0.3	9.2+-0.3	9.5+-0.3	9.3+-0.3
	500	6.4+-0.3	7.9+-0.3	6.5+-0.3	7.5+-0.3	7.6+-0.3	8+-0.3
	2000	5.1+-0.2	6.8+-0.3	5.2+-0.2	6.1+-0.3	5.8+-0.2	6.2+-0.3
	5000	4.2+-0.2	6+-0.2	4.5+-0.2	5.4+-0.3	4.6+-0.2	4.4+-0.2
80-5-4-1	100	12+-0.3	11.4+-0.3	NA	9.8+-0.3	10.6+-0.3	9.8+-0.3
	500	7.2+-0.2	8.5+-0.3	NA	8.1+-0.3	8.1+-0.3	8.5+-0.3
	2000	5.6+-0.2	7+-0.3	NA	6.7+-0.3	6.2+-0.2	6.3+-0.3
	5000	4.8+-0.2	6.2+-0.2	NA	5.9+-0.3	4.7+-0.2	6.1+-0.3

Table 5.3 A summary of MB learners' precisions and recalls (with 95% confidence intervals) on both real and artificial BNs. The best precisions and recalls are respectively highlighted in pink and blue. In real networks, IAMB almost always has the highest precision, followed by SLL, MBMML+CPT/NB, PCMB and MBMML+MBP. However, MBMML+MBP has the highest recall across all cases by significant margins. In artificial networks, MBMML+CPT has the highest precision in all medium and large sample cases, followed by IAMB who has the highest precision in 100 samples. In terms of recall, MBMML+CPT and MBMML+MBP both win in all cases. We did not run MBMML+MBP on 80-5-4-1 due to its high running time. SLL again shows no significant winnings in 50 and 80-variable models.

Network	SAMPLES	MBMML+CPT		MBMML+NB		MBMML+MBP		IAMB		PCMB		SLL	
		Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
CHILD	500	0.94+-0.03	0.8+-0.04	0.94+-0.03	0.82+-0.04	0.78+-0.04	0.89+-0.03	0.95+-0.03	0.77+-0.04	0.87+-0.04	0.77+-0.04	0.94+-0.03	0.78+-0.04
	1000	0.98+-0.02	0.88+-0.03	0.97+-0.02	0.86+-0.03	0.82+-0.03	0.96+-0.01	0.95+-0.02	0.83+-0.03	0.94+-0.03	0.8+-0.04	0.97+-0.02	0.84+-0.03
	5000	1+-0.01	0.91+-0.02	1+-0.01	0.89+-0.02	0.95+-0.02	1+-0	0.98+-0.01	0.9+-0.02	1+-0	0.99+-0.01	1+-0	0.97+-0.01
INSURANCE	500	0.82+-0.04	0.48+-0.03	0.8+-0.04	0.42+-0.03	0.64+-0.04	0.67+-0.03	0.9+-0.03	0.43+-0.03	0.77+-0.04	0.5+-0.03	0.83+-0.04	0.51+-0.04
	1000	0.86+-0.03	0.54+-0.03	0.85+-0.04	0.45+-0.03	0.65+-0.03	0.73+-0.03	0.93+-0.03	0.5+-0.03	0.81+-0.04	0.53+-0.03	0.88+-0.03	0.58+-0.03
	5000	0.95+-0.01	0.68+-0.03	0.93+-0.02	0.57+-0.03	0.78+-0.03	0.8+-0.03	0.97+-0.02	0.59+-0.03	0.94+-0.03	0.68+-0.03	0.98+-0.01	0.69+-0.03
ALARM	500	0.85+-0.03	0.77+-0.03	0.79+-0.03	0.67+-0.03	0.53+-0.03	0.92+-0.02	0.9+-0.02	0.62+-0.03	0.85+-0.03	0.7+-0.03	0.92+-0.02	0.89+-0.02
	1000	0.9+-0.02	0.82+-0.03	0.86+-0.02	0.68+-0.03	0.59+-0.03	0.94+-0.02	0.96+-0.01	0.71+-0.03	0.9+-0.03	0.79+-0.03	0.94+-0.01	0.94+-0.01
	5000	0.97+-0.01	0.93+-0.02	0.95+-0.02	0.7+-0.03	0.7+-0.02	0.97+-0.01	0.94+-0.02	0.79+-0.03	1+-0.01	0.95+-0.01	0.98+-0.01	0.98+-0.01
BARLEY	500	0.74+-0.03	0.37+-0.03	0.74+-0.03	0.35+-0.02	0.66+-0.03	0.51+-0.03	0.79+-0.04	0.29+-0.02	0.32+-0.02	0.56+-0.03	0.63+-0.04	0.25+-0.02
	1000	0.79+-0.03	0.42+-0.03	0.79+-0.03	0.37+-0.02	0.68+-0.03	0.57+-0.03	0.8+-0.03	0.33+-0.03	NA	NA	0.72+-0.04	0.35+-0.03
	5000	0.8+-0.03	0.52+-0.03	0.81+-0.03	0.47+-0.02	0.72+-0.03	0.7+-0.03	0.84+-0.03	0.38+-0.03	NA	NA	0.85+-0.03	0.5+-0.03
HAILFINDER	500	0.3+-0.03	0.18+-0.02	0.25+-0.03	0.12+-0.01	0.27+-0.03	0.2+-0.02	0.32+-0.03	0.18+-0.02	0.3+-0.03	0.18+-0.02	0.28+-0.03	0.14+-0.02
	1000	0.31+-0.03	0.22+-0.02	0.26+-0.03	0.12+-0.01	0.29+-0.03	0.24+-0.02	0.34+-0.03	0.2+-0.02	0.34+-0.03	0.2+-0.02	0.3+-0.03	0.18+-0.02
	5000	0.34+-0.03	0.26+-0.02	0.26+-0.03	0.14+-0.02	0.3+-0.03	0.27+-0.03	0.33+-0.02	0.22+-0.02	0.32+-0.03	0.21+-0.02	0.34+-0.03	0.22+-0.02
30-5-4-1	100	0.56+-0.02	0.36+-0.02	0.6+-0.03	0.23+-0.02	0.58+-0.02	0.36+-0.02	0.65+-0.03	0.21+-0.02	0.59+-0.03	0.25+-0.02	0.5+-0.03	0.17+-0.02
	500	0.91+-0.02	0.56+-0.02	0.86+-0.02	0.35+-0.02	0.86+-0.02	0.56+-0.02	0.9+-0.02	0.48+-0.02	0.89+-0.02	0.38+-0.02	0.79+-0.03	0.3+-0.02
	2000	0.97+-0.01	0.68+-0.02	0.94+-0.01	0.48+-0.02	0.94+-0.01	0.68+-0.02	0.95+-0.01	0.64+-0.02	0.6+-0.03	0.38+-0.03	0.94+-0.02	0.54+-0.02
	5000	0.99+-0	0.76+-0.02	0.96+-0.01	0.57+-0.02	0.96+-0.01	0.73+-0.02	0.96+-0.01	0.71+-0.02	0.94+-0.01	0.77+-0.02	0.98+-0.01	0.7+-0.02
50-5-4-1	100	0.44+-0.02	0.28+-0.01	0.47+-0.02	0.19+-0.01	0.42+-0.02	0.27+-0.01	0.56+-0.03	0.17+-0.01	0.48+-0.02	0.18+-0.01	0.45+-0.03	0.12+-0.01
	500	0.85+-0.02	0.46+-0.02	0.77+-0.02	0.29+-0.02	0.8+-0.02	0.46+-0.02	0.83+-0.02	0.38+-0.02	0.81+-0.02	0.3+-0.02	0.74+-0.02	0.26+-0.02
	2000	0.97+-0.01	0.59+-0.02	0.91+-0.01	0.4+-0.02	0.92+-0.01	0.6+-0.02	0.93+-0.01	0.54+-0.02	0.92+-0.01	0.49+-0.02	0.9+-0.02	0.44+-0.02
	5000	0.99+-0	0.68+-0.01	0.97+-0.01	0.49+-0.02	0.97+-0.01	0.67+-0.01	0.95+-0.01	0.62+-0.02	0.94+-0.01	0.63+-0.01	0.96+-0.01	0.6+-0.02
80-5-4-1	100	0.36+-0.01	0.24+-0.01	0.35+-0.01	0.2+-0.01	NA	NA	0.52+-0.02	0.16+-0.01	0.39+-0.01	0.18+-0.01	0.41+-0.02	0.12+-0.01
	500	0.83+-0.01	0.44+-0.01	0.74+-0.02	0.3+-0.01	NA	NA	0.8+-0.01	0.38+-0.01	0.82+-0.02	0.28+-0.01	0.72+-0.02	0.24+-0.01
	2000	0.97+-0.01	0.58+-0.01	0.92+-0.01	0.43+-0.02	NA	NA	0.89+-0.01	0.54+-0.01	0.93+-0.01	0.49+-0.01	0.93+-0.01	0.46+-0.01
	5000	0.99+-0	0.65+-0.01	0.96+-0.01	0.51+-0.01	NA	NA	0.91+-0.01	0.62+-0.01	0.94+-0.01	0.64+-0.01	0.79+-0.02	0.52+-0.02

There may be a concern that if the generating models had non-uniform parameter priors, the MML methods would perform differently. We hypothesized that if the true priors are not uniform, then using uniform priors would give noticeably worse results than using the true prior. To be sure, performance will also depend on the quality and size of the samples. To check the impact of using an uninformative, uniform prior, MBMML+CPT was given both the true priors and uniform priors when tested on models (30-5-4-1) whose parameters were sampled from a symmetric Dirichlet distribution with a non-uniform concentration parameter α taking values from $\{0.1, 0.4, 0.7, 1, 10, 40, 70, 100\}$. The experiments were done for 500 and 5000 samples.

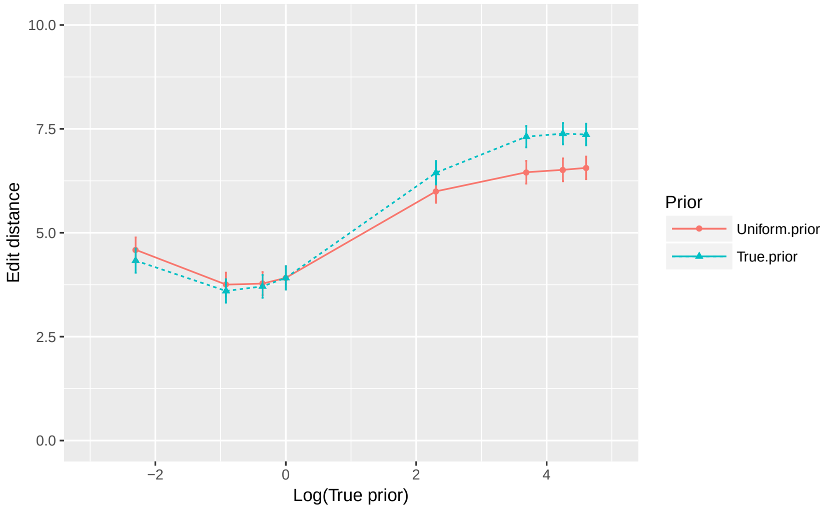


Fig. 5.8 MBMML+CPT's edit distances (with 95% confidence intervals) using the true prior and uniform prior on artificial BNs (30-5-4-1) with 500 samples. The X-axis is the natural log scale of the true symmetric Dirichlet concentration parameter α taken from $\{0.1, 0.4, 0.7, 1, 10, 40, 70, 100\}$.

Figures 5.8 and 5.9 report no significant differences between the use of true priors and uniform priors when the concentration parameters $\alpha \leq 1$. This is because adding small α values to parameter estimates will have a small effect, swamped even by modest data. When $\alpha > 1$, uniform priors track the true priors until the non-uniformity begins to get extreme ($\alpha \geq 10$, i.e., $\ln \alpha > 2$).

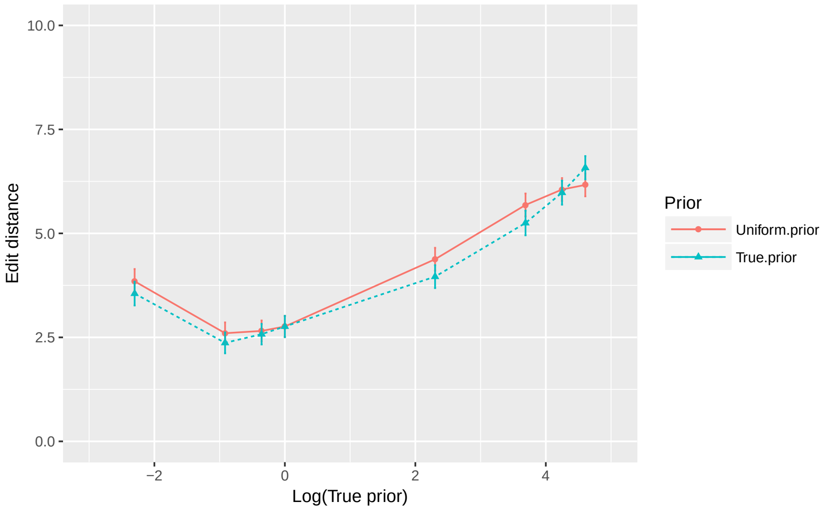


Fig. 5.9 MBMML+CPT's edit distances (with 95% confidence intervals) using the true prior and uniform prior on artificial BNs (30-5-4-1) with 5000 samples. The X-axis is the natural log scale of the true symmetric Dirichlet concentration parameter α taken from $\{0.1, 0.4, 0.7, 1, 10, 40, 70, 100\}$.

As α increases, the learned Markov Blankets increase in complexity. This is analogous to the increased complexity of models learned using the BDeu metric when increasing its equivalent sample size [78]. The MML metric for the CPT model with symmetric Dirichlet priors is similar to the BDeu metric, although MML includes costs for the precision of parameter estimates. But both metrics penalize model complexity using a function of α, which decreases as α increases. Hence, given larger α, MML methods more easily discover larger Markov Blankets. These may contain a larger proportion of false positives, especially with small samples. At larger samples, these differences between MML with uniform and true priors, however, appear to be erased.

In general, we didn't find important differences between MML-CPT with uniform and with true priors, supporting our use of the practical, and non-informative, uniform priors.

Table 5.4 The computational complexity of the tested MB discovery algorithms.

Algorithm	Big O notation
IAMB	$O(n^2)$
MBMML+NB	$O(n^2)$
MBMML+CPT	$O(n2^{n-1})$
MBMML+MBP	$O(n2^{n-1})$
PCMB	$O(n^2 2^{n-1})$
SLL	$O(n^4 2^n)$

5.1.3 Case Study

This section provides an illustrative case study to show the performance of the three MML based Markov blanket learners on a specific Bayesian network that contains 30 variables as shown in Figure 5.10. This model is one of the artificial models used to compare MB learners accuracy in the previous section. For this example, we look at how each of the MBMML learners does on finding candidates for nodes with different structural features.

Figure 5.11 plots the proportion of the edit distance (divided by the Markov blanket size) for each node for all three methods. The x-axis are the nodes sorted in a top-down order, so nodes near the end tend to have smaller Markov blankets due to fewer children and spouses. It can be seen from the figure that CPT and MBP methods agree on many nodes, especially for nodes with smaller Markov blankets. For nodes with larger Markov blankets, the CPT method outperformed the MBP method more often than the other way around. Both methods are superior than the NB method for most cases.

The nodes at which each of the algorithms wins are $\{V1, V5, V8, V10, V19, V23, V30\}, \{V2, V7, V20, V24\}, \{V3, V4, V29\}$ for CPT, NB and MBP respectively. Although the pattern is not entirely clear, the winning cases rather suggest that each of the CPT, NB and MBP methods tends to do slightly better than the other two when target nodes have more fan-in, fan-out and spouses respectively. This is consistent with the what each of the models assumes, i.e., CPT models fan-in exactly, NB models fan-out exactly and MBP models spouses fairly well (nearer than the other two). To further understand in what circumstances

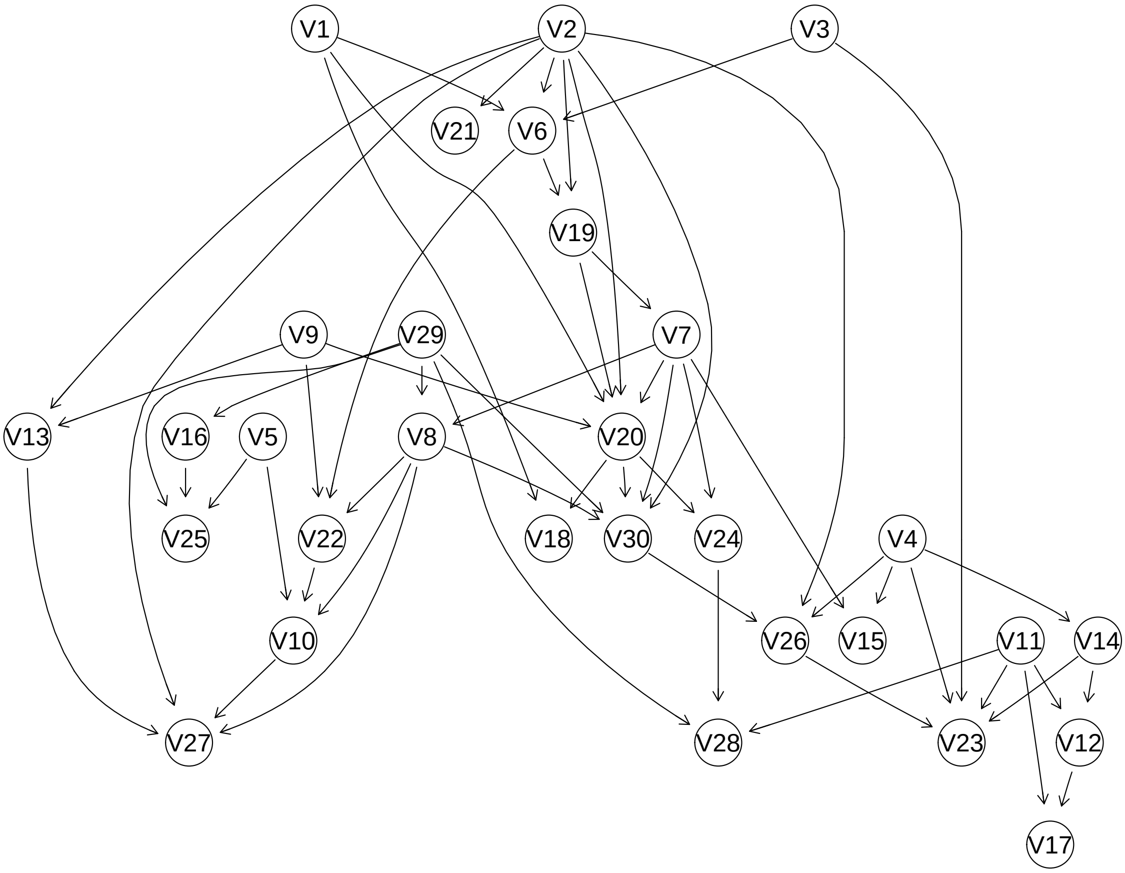


Fig. 5.10 A case study on an artificial Bayesian network that contains 30 variables.

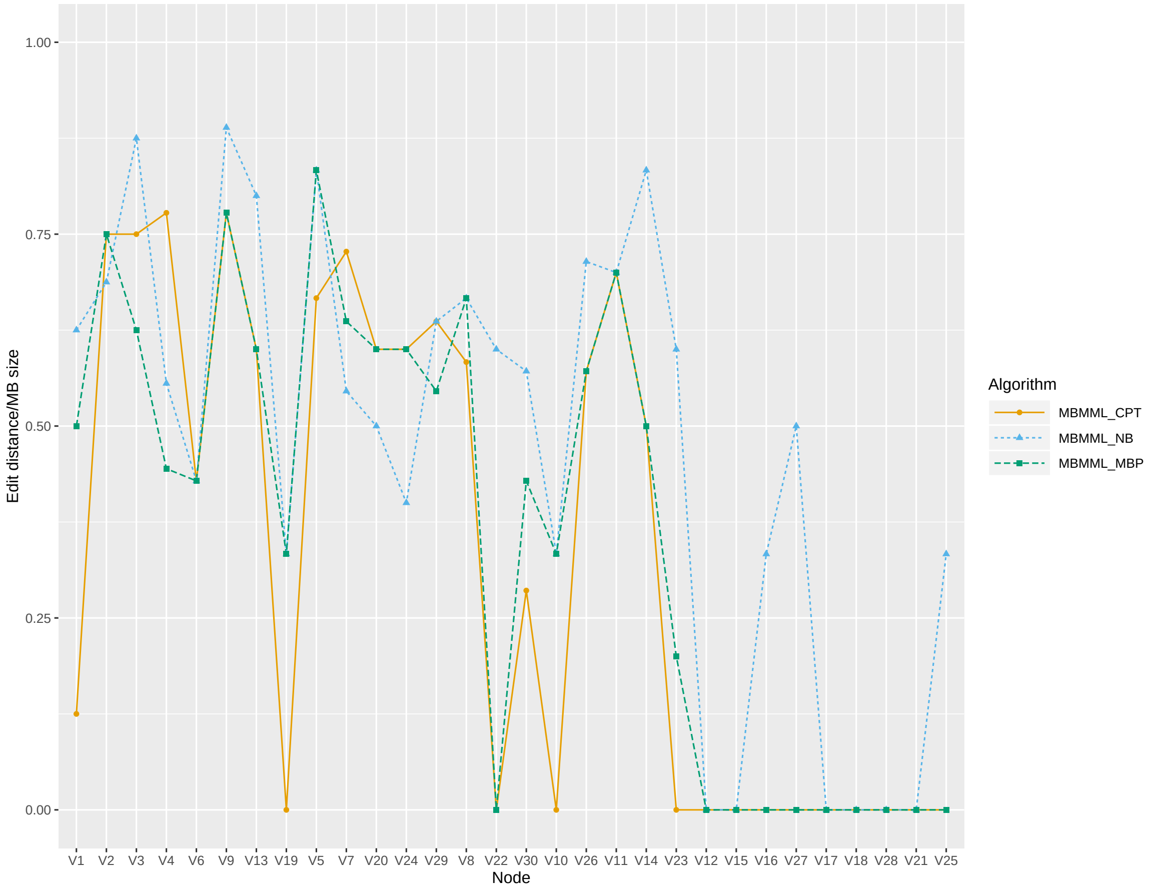


Fig. 5.11 Compare MB discovery accuracy for all three MML based methods. The x-axis are the nodes. The y-axis is the edit distance divided by the MB size for a node.

each of the method does better than the others, more experiments need to be done as a future research work.

5.1.4 Algorithmic Complexity

Table 5.4 orders all algorithms by ascending computational complexity. The main loop in Algorithm 6 for MBMML+CPT/NB runs at most $n-1$ times. Each time it runs through all unchecked nodes to find the best candidate to add to the Markov blanket using the MML metric. For a CPT model, there can be at most $n-1$ parents, with the multi-state MML summed over all 2^{n-1} parent instantiations. So the computational complexity of the MBMML+CPT algorithm is $O(n2^{n-1})$. For an NB model, the worst case is when all

$n-1$ nodes are children of the target, which is linear in n within the *WHILE* loop, and so gives a complexity of $O(n^2)$. The worst case for MBMML+MBP is when one of the sampled polytrees is just the full CPT model, with all Markov blanket variables parents of the target. In general, a random polytree model is slower than a CPT by a constant factor, which is determined by the number of sampled regional structures. But this doesn't affect the O-notation complexity.

For PCMB, the total time required is dominated by the process of finding the direct neighbours of the target. This process tries to find a subset of the neighbour set, conditioning on which the target is independent from a candidate. And such a process runs through all variables to ensure the symmetry property holds. Hence, its complexity in the worst case is $O(n^2 2^{n-1})$. The total time required by IAMB and SLL were published in the associated papers, by Tsamardinos et al. [88] and Niinimaki and Parviainen [60] respectively.[5]

5.2 Improving Markov Blanket Discovery by Morality

This section is concerned with the potential of morality enforcement for improving the quality of learned Markov blankets. As before, we produce an undirected graph by connecting the Markov blanket to its target node.[6] Hence, the edit distance of these Markov blankets to the true Markov blankets is proportional to the edit distance of this graph to the moral graph of the true model.

As discussed in Section 3.4, finding a minimal moralization of an undirected graph is NP-hard, so heuristics are the best practical choice if one wants to add the fewest possible edges to the given graph. By Theorem 3.4.2, optimal moralization is equivalent to finding an optimal elimination kit that includes a node ordering and a set of edges to remove for each

[5] Gao and Ji [33] made significant practical gains in SLL's performance by relaxing symmetry enforcement, but we did not explore that here.

[6] The Markov blankets need to be symmetric to produce a graph. Symmetry is enforced by taking the Union rule unless mentioned otherwise.

node during elimination. Note that a triangulation process does not remove any edge. For simplicity and in order to distinguish from triangulation, the moralization process stated in this section removes the edges between each pair of a simplicial node's neighbours.[7] Hence, the problem of optimal moralization is restricted to finding an optimal node ordering for a given undirected graph. Before looking for good node ordering heuristics, one concerns whether or not enforcing morality with good node orders can reduce the edit distance between learned graphs and the truth, which is equivalent to improve the quality of learned Markov blankets (produced by some learners). To answer this question, the following two sections examine the effectiveness of various node orders on improving Markov blanket quality.

5.2.1 Moralization with Synthetic Node Orders

The examination of node order's effectiveness on improving Markov blanket quality via moralization starts with random orders that have different levels of correctness. The random orders are given to the revised Elimination Game (EG) algorithm (Algorithm 5) to produce a moral graph for the input graph. The excess for each node in a given order is the set of edges between each pair of the node's neighbours. For simplicity, we call this the full excess for each node. The input graphs come from learned Markov blankets from data that are sampled from randomly generated Bayesian networks with 50 variables. Each of these networks contains a structure with a maximum of five parents per node and a set of parameters taken from uniform distributions. To minimize random effects within a reasonable time bound, this experiment used 40 random Bayesian networks, each of which was used to sample 10 different data sets with $400, 800, 1600, 3200, 6400$ and 12800 samples. The Markov blankets learners used in the experiments are the IAMB[8], PCMB[9], SLL[10] and MMLCPT[11] algorithms

[7] The problem of finding an optimal set of edges to remove for each simplicial node is marked as a future research question.

[8] The *bnlearn* implementation in R with the significance level $\alpha = 0.01$ for conditional independence test.

[9] The original c++ implementation used in [67].

[10] The original c++ implementation used in [60].

[11] The R implementation used in Chapter 4 with a uniform prior for the MMLCPT model.

(details of these algorithm are explained in Chapter 4). The average performance, measured by edit distance from a learned graph to the moral graph of the true underlying structure, is calculated over all 400 cases for each learner.

To obtain random node orders that vary in correctness, a true node order from each generating model[12] was used and a fraction of it was held fixed while the rest was permuted randomly. Using these orders to enforce morality on the learned Markov blankets from the algorithms mentioned above, we get the results presented in Fig. 5.12. The plot is separated into sections for different sample sizes. The x-axis is the percentage of nodes hold fixed when randomly shuffling a given true order. The label "w/o" on the x-axis refers to Markov blankets without moralization. That is, the graph produced directly from the learned Markov blankets was used.[13] The y-axis is the edit distance between the enforced moral graphs and the true moral graph.

To get a more intuitive understanding of how edit distance changes according to the quality of a node order, Fig. 5.12 is re-plotted in Fig. 5.13 but the x-axis is changed to one minus the normalized Kendall Tau (KT) distance that measures the percentage of pairwise agreements between a random order and the truth. For each pair of nodes (i,j) in a node order, the KT distance measures the number of pairwise disagreements between a random node order r and the true order t by

$$K(t,r) = |\{(i,j) : i < j, (t(i) < t(j) \wedge r(i) > r(j)) \vee (t(i) > t(j) \wedge r(i) < r(j))\}|. \quad (5.1)$$

This distance can be normalized by dividing the actual KT distance by the maximum KT distance $\frac{n(n-1)}{2}$.

[12] A DAG contains either exactly one total order or multiple total orders consistent with it (linear extensions). The order used in this section and next can be any of these; we chose one linear extension arbitrarily for each experiment. Note that its reverse is fed to Algorithm 5 as a total order; that is because each step of moralization removes a simplicial node that corresponds to a child of at least one parent.

[13] Note that a set of learned Markov blankets may be moral. In practice, however, a large portion of them are likely immoral.

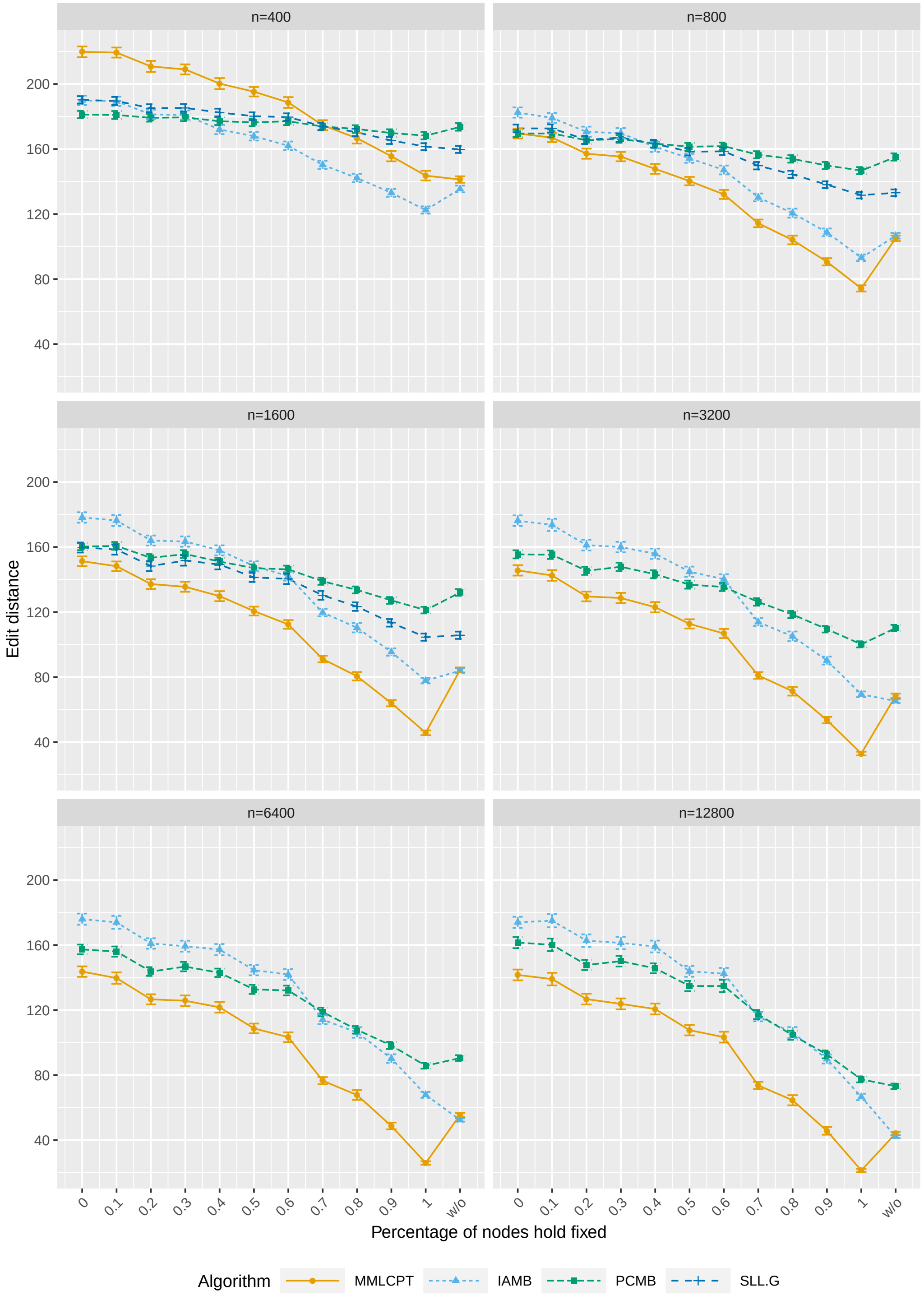


Fig. 5.12 Enforcing morality by the revised EG Algorithm on learned MBs with random node orders and full excess for each node. The x-axis is the percentage of nodes hold fixed when randomly shuffling a given true order. The x-axis label “w/o” (i.e., without) refers to the immoral graph directly learned by each of the three Markov blanket learners.

It can be seen from Fig. 5.13 that the quality of the learned Markov blankets can be improved under some scenarios. Note that no results of SLL was shown for sample size $n \in \{3200, 6400, 12800\}$ due to its long running time and the program's occasional failure when dealing with complicated structures. The improvements are particularly clear for MMLCPT's outputs. For instance, when $n \in \{800, 1600, 3200, 6400\}$, moralizing the outputs of MMLCPT using a node order with 94% pairwise agreement (to a true order) reduces the edit distance to lower than the immoral graph. Moralizing outputs of the same algorithm using the same orders, however, could not produce smaller edit distance than the immoral graphs when $n \in \{400, 12800\}$. Despite the challenge of getting a near optimal node order (that will be discussed in the next section), the outputs of MMLCPT among all four learners have the greatest potential to be further improved by enforcing morality using a high quality node order. It is worth mentioning that Markov blanket evaluation metrics (e.g., edit distance, precision, recall, etc.) are based on set membership. That is, a node is either in or not in the Markov blanket of a target variable. The outputs of moralizations are sensitive to structural features such as graph density, maximum degree, number of cliques and so on, which cannot be reflected by Markov blanket evaluation metrics. Hence, it is not entirely unexpected to see different moralization accuracy, although applied to Markov blanket learners who have similar Markov blanket discovery accuracy. To better understand why MMLCPT shows the greatest potential of improving its outputs by moralizing its learned Markov blankets, more theoretical and experimental analyses need to be done to further understand what node (structural) features are preferred by one algorithm but not another, hence leading to different moralization outcomes.

To further consider moralization's potential for improving Markov blanket quality, we applied triangulation on the same learned Markov blankets with the same node orders used for the above moralization experiments and plotted the results in Fig. 5.14. It can be seen from this plot that in most cases triangulation fails to reduce the edit distance of the given

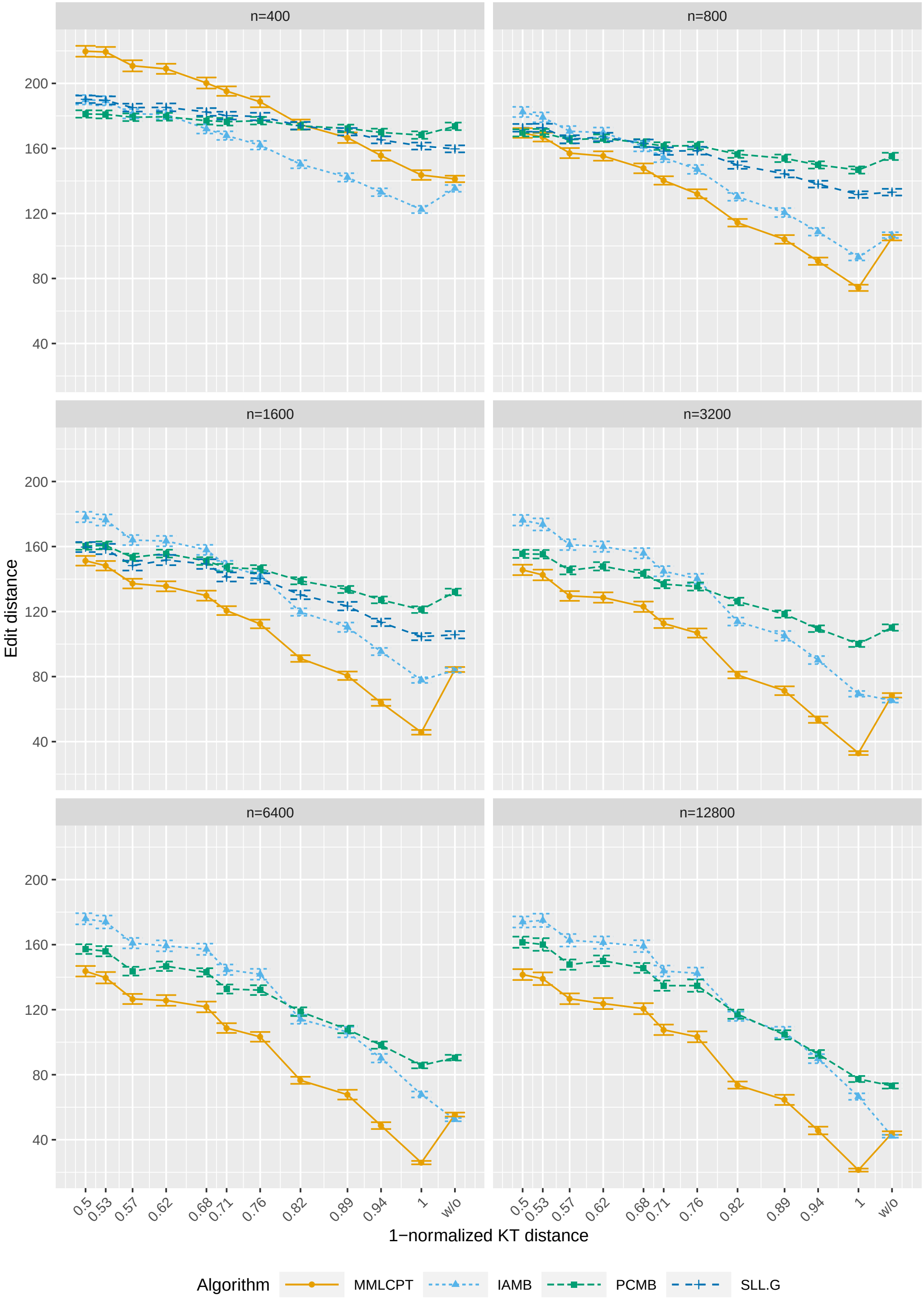


Fig. 5.13 Enforcing morality by the revised EG Algorithm on learned MBs with random node orders and full excess for each node. The x-axis is one minus normalized KT distance percentage. It measures the percentage of pairwise agreements between two orders. The x-axis label “w/o” (i.e., without) refers to the immoral graph directly learned by each of the three Markov blanket learners.

Markov blankets even with true node orders. The contrast is more obvious for MMLCPT given medium to large samples. This suggests that moralization does not reduce edit distance by just randomly throwing in some edges.

5.2.2 Moralization with Approximate Node Orders

This section examines the impact on given Markov blankets of node orders that are inferred from heuristics and machine learning methods. The previous section demonstrated that when using objectively good node orders, the outputs of Markov blanket learners, in particular the MMLCPT algorithm, can be dramatically improved for medium and large samples. It is, however, not trivial to find accurate node orders in practice.

We first look at two node ordering heuristics, namely the *minimum degree* and *minimum deficiency* algorithms. These algorithms are known for their simplicity and reasonably good performance for triangulation, but provide no guarantee of accuracy. In triangulation, the minimum degree algorithm selects a node with the minimum degree at each step, then makes it simplicial (if it is not already) and removes it from the original graph. If there is a tie, the algorithm randomly selects a node from all suitable ones. For simplicity and to distinguish from triangulation, we also let the minimum degree algorithm remove all edges between the neighbours of the removed simplicial node. The minimum deficiency algorithm is similar to the minimum degree algorithm except it selects a node that has the minimum deficiency at each step.

In addition to the heuristics, we also learn node orders from data using the *maximum weight spanning tree (MWST)* [17], MMHC [89], SLL (with greedy search) and CaMML [61] algorithms. The output of the MWST algorithm is an undirected tree, so it requires a root node to be determined by users to order the nodes. To satisfy this, a source node from the true DAG was given to the algorithm to produce a node order. SLL was used with its default settings in the original implementation. Refer to Section 5.1 for more details. Due

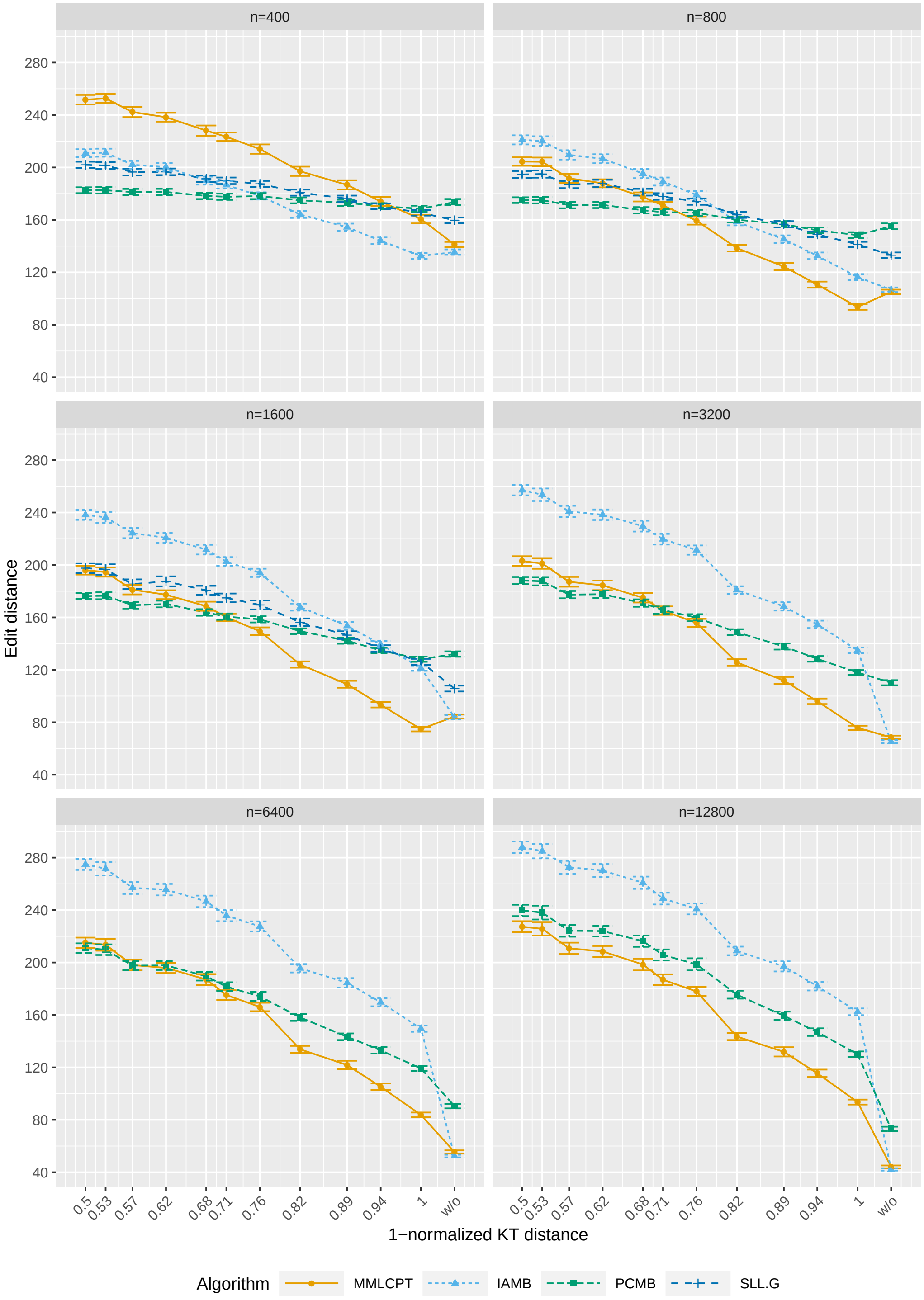


Fig. 5.14 Enforcing triangulation by the original EG Algorithm on learned MBs with random node orders. The x-axis is one minus KT distance. It measures the percentage of pairwise agreements between two orders. The x-axis label "w/o" (i.e., without) refers to the immoral graph learned by each of the three Markov blanket learners.

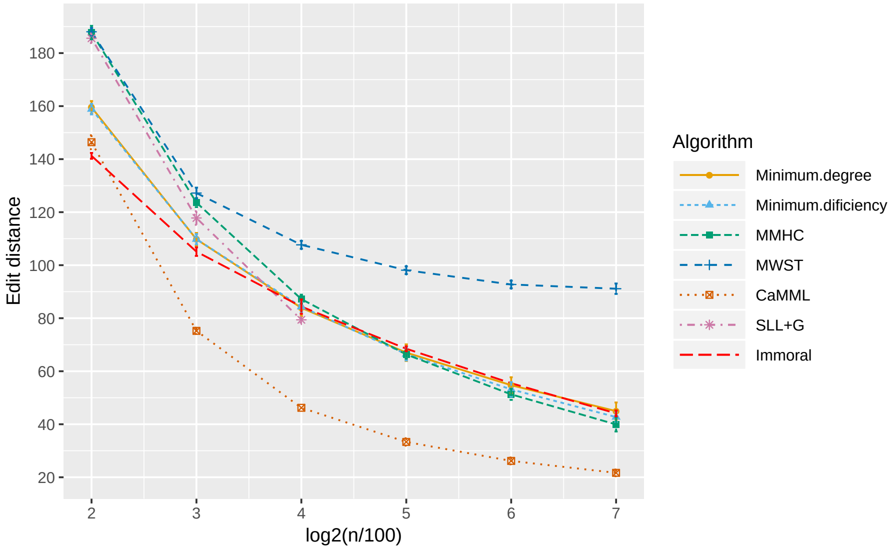


Fig. 5.15 Enforcing morality on MBs learned by MMLCPT by the revised EG Algorithm with learned orders and full excess for each node. The x-axis is the re-scaled sample size by $\log_2(n/100)$. The y-axis is the mean edit distance. The red line marked as “Immoral” corresponds to the immoral MBs (or undirected graph) produced by MMLCPT.

to time constraints, we restricted CaMML to sample $200 * n$ *totally ordered models (TOMs)* instead of the default $200 * n^3$. More details of the CaMML algorithm will be discussed in the following section.

As MMLCPT has been demonstrated to have the most potential to be further improved by moralization, in the current experiment we illustrate how node orders that are inferred from practical methods could make a difference to MMLCPT’s outputs. Fig. 5.15 shows how edit distance of the MMLCPT’s outputs change when moralizing with the node orders inferred by each of the methods described above. The red dashed line ("Immoral") represents the edit distance of the given immoral Markov blankets. It is included in the plot as a baseline. Note the confidence intervals are so tiny to be seen on the current scale.

In general, the MWST algorithm performs the worst, except at small samples $n = 400$ when its edit distance overlaps with MMHC and SLL. This could partially be due to the under-representation of the true model by a tree structure and the straightforward way of ordering nodes by directing edges away from a given root node. The minimum degree and minimum deficiency algorithms perform almost except for 6400 and 12800 samples where the minimum degree algorithm is slightly (significantly) better than the minimum deficiency algorithm. SLL and MMHC do not start well but quickly overtake the two non-machine learning methods at $n = 1600$ and 3200 respectively. This under-performance of SLL and MMHC shows their lack of robustness when learning with small smaples. [14] Notice again that there is no result for SLL when $n > 1600$ due to its long running time. Comparing these five methods with the baseline, it suggests that if one is concerned about Markov blanket morality, then the two heuristics are better alternatives than the three machine learning methods both in speed and accuracy. Importantly, one should keep in mind that the quality of given Markov blankets learned under small samples may be reduced after moralization. The CaMML algorithm, however, is way ahead of the others in terms of generating quality orders. The only case where CaMML has no positive effect on given Markov blankets is under 400 samples. This is not unexpected, because Fig. 5.13 suggests that the 400 case has almost no improvement at all for MMLCPT regardless of the sample size. For the other cases, the edit distance reduction in Fig. 5.15 indicates that CaMML can get node orders with at least 90% pairwise agreements with a true order.

To sum up, the revised Elimination Game algorithm (Algorithm 5) with the full excess is a simple and deterministic way of moralizing a given graph. It has been empirically justified to have the potential of improving the quality of given Markov blankets produced by a few learners, in particular, Markov blankets produced by the MMLCPT algorithm. Nevertheless, the improvements may only happen when moralizing with "good" node orders. Although efficiently obtaining an accurate node order is again a challenging task, if time is not a concern, CaMML has demonstrated its ability to produce node orders that are almost as good as real orders when given sufficient samples. Clearly, using CaMML may not be a

[14] The superior of the minimum degree and minimum deficiency algorithms at small sampes may suggest that they can be used to incorporate with the MMLCPT algorithm (or another Markov blanket learner) to efficiently produce "good" node orders which can then be used by some structure learners, such as the K2 algorithm. This is out of the scope of this project, but maybe worth pursuing in future.

feasible preprocessing step for scaling up structure learning; but it does show the potential improvement possible in such preprocessing. More practically, where time is a concern, and one still wants to obtain moral graphs that are at least as good as the immoral ones, the minimum degree and minimum deficiency algorithms are faster alternatives to the other machine learning algorithms and result in performance about as good.

5.3 Structure Learning via Markov Blankets

One of the motivations for Markov blanket discovery is to scale up structure learning to high dimensional data sets. Ideally, this can be done by first efficiently learning appropriate variable subsets, then give those learned subsets (with or without regional structures) to a subsequent structure learner as a starting point that is better than knowing nothing. The potential scalability of this strategy is down to the possible parallel learning of variable subsets and advantageous use of an initial structure that could reduce the search space of a subsequent global structure learner. The previous chapter has discussed some strategies of learning variable subsets, i.e., Markov blankets. This section explores the possibilities of producing probabilistic Markov blanket priors to boost the performance of the Bayesian structure learner CaMML. It is worth pointing out that Markov blanket priors are not genuine prior information, because they are derived from data rather than being gathered from one's belief about the joint probability distribution that underlies the generating Bayesian network.

5.3.1 Learning Bayesian Networks with CaMML

To begin with, this section briefly summarizes the CaMML algorithm, which will be used as the structure learner to test the effectiveness of probabilistic Markov blanket priors. More theoretical details and experimental work can be found in [61].

The CaMML algorithm samples through the space of Totally Ordered Models (TOMs) using the Metropolis-Hasting algorithm and MML with the CPT (as well as decision tree and first order logit) model. A TOM consists of a total node ordering and a set of edges between pairs of nodes. When a TOM is used to express a DAG, an edge ab in the TOM (where a comes before b in the total order) represents a directed arc $a \rightarrow b$ in the DAG. For instance, the DAG in Fig. 5.16 has a TOM $(a,b,c) \cup \{ac, bc\}$. But this is not the only TOM it has, because a different total order $(b,a,c) \cup \{bc, ac\}$ is also consistent with the DAG.

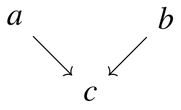


Fig. 5.16 A DAG that has two consistent TOMs.

While sampling, CaMML records the number of times each TOM is visited and aggregates the count to get an estimate of the posterior probability of MML equivalent DAG patterns. Aggregate counts are collected for TOMs, DAGs, clean DAGs, DAG patterns and MML equivalent DAG patterns and accumulated in that order, since every structure in this order has several equivalent corresponding representatives in the previous class. For example, a DAG may have multiple consistent TOMs. To satisfy the practical need of producing fully directed Bayesian networks, CaMML will always output a consistent extension of the best DAG pattern.

The centre of CaMML is the MCMC sampling process, in particular, the Metropolis-Hasting algorithm. Before going into the sampling process, however, CaMML starts with several epochs of greedy and probabilistic searches to find the best starting TOM for the sampling process. For a data set with n variables, the pre-sampling steps start with the probability of an arc existing $p(arc) = 0.5$, then updates it by $p(arc) = \frac{0.5+|E|}{1+\binom{n}{2}}$ using the best TOM found at the end of each epoch, where $|E|$ is the number of edges in the best TOM. The pre-sampling steps are stated in the following:

1. Start with an empty TOM (i.e., no edges) with $p(arc) = 0.5$, do $7 * n^3$ mutations, store the best TOM and its cost, re-estimate $p(arc)$, repeat for 1 epoch.

2. Start with a highly connected TOM (not necessarily fully connected, because CaMML has a default maximum number of parents per node) with the updated $p(arc)$, do $7 * n^3$ mutations, store the best TOM and its cost, re-estimate $p(arc)$, repeat for 10 epochs.

3. Start with an empty TOM with the updated $p(arc)$, do $7 * n^3$ mutations, store the best TOM and its cost, re-estimate $p(arc)$, repeat for 10 epochs.

4. Start with the best TOM found in the previous 21 epochs with the updated $p(arc)$, do $10 * n^3$ mutations, if a better TOM is found, update the best TOM and its cost, re-estimate $p(arc)$.

These pre-sampling steps are here to make sure the sampling process starts with a good candidate TOM, so it converges to the stable distribution relatively quicker.

CaMML has been demonstrated to be superior to some structure learners by O'Donnell [61] (at that time). To test CaMML's performance among recent results, we set up experiments using both synthetic and real models. To test on synthetic models, the experiments are carried out on 10 randomly sampled Bayesian networks with 50 binary variables with maximum fan-in 5. The CPT parameters are sampled from a symmetric Dirichlet distribution with concentration parameter $\alpha \in \{0.2, 1, 5\}$. As the default MML setting in CaMML takes uniform prior (i.e., symmetric Dirichlet concentration parameter $\alpha = 1$), sampling CPT parameters from various α values tests CaMML's robustness on models with different parameter priors. Each of these 10 models is then used to generate 10 different data sets with size $\{300, 5000\}$ to represent small and large sample sizes. To test on real models, we use the medium size benchmark Bayesian networks listed in Table 5.5. In particular, we focus on testing accuracy under small samples. First, these models are used to generate 20 data sets with 300 samples. Then we keep the model structures but vary the model parameters by sampling 10 sets of parameters from symmetric Dirichlet with $\alpha \in \{0.2, 1, 5\}$. Each of these parameter sets is then used to generate 2 data sets of size 300. The average edit distance from the learned to the true DAGs are calculated with 95% confidence intervals. The experiments are by no means exhaustive, but to obtain an update on CaMML's accuracy among more recent structure learners.

Table 5.5 A summary of the real models used for testing structure learners accuracy.

Network	Number of nodes	Max fan-in	Number of arcs	Number of parameters
Child	20	2	25	230
Insurance	27	3	52	984
Water	32	5	66	10083
Mildew	35	3	46	540150
Alarm	37	4	46	509
Barley	48	4	84	114005

The competitors in the experiments include the constraint-based PC algorithm [82], the metric-based CaMML algorithm [61], the hybrid MMHC algorithm [89], the local-to-global exact SLL algorithm (two versions) [60] and the exact KMAX algorithm [73]. The PC and MMHC algorithms are available in the *bnlearn* package in R with hypothesis test significant level sets to $\alpha = 0.05$. The CaMML algorithm uses its default settings but $200 * n$ TOMs in the sampling phase to reduce running time. Both the constraint and metric-based versions of SLL are used with their default settings in the original C++ implementation. The KMAX algorithm [73] is another exact algorithm but takes a different approach from SLL. It starts with learning a few possible parent sets for each node and ranked them by a metric (e.g., BDeu). KMAX then uses integer programming to find the optimal parent sets for all nodes. The program has a constraint on the learned DAG's treewidth (of its moral graph). In these experiments, we set it to 20. Since KMAX is a real time algorithm, the waiting time is set to the average running time of the other algorithms for each case.

The experimental results for the artificial and real models are shown in Fig. 5.17 and Fig. 5.18 respectively. The parameter prior $\alpha = 0.2$ and 1 often produce dependencies that are relatively stronger than $\alpha = 5$, so the learning tasks are relatively easier or require less training data.

For the artificial models, CaMML almost always performed the best, except for the case when $n = 300$ and $\alpha = 5$, where it has a higher mean edit distance than MMHC but not to a statistically significant extent. Although the tasks became more difficult as α increasing, CaMML still outperformed the others under the same 5000 samples.

In Fig. 5.18, the value 0 on the x-axis is the case with real structures and their default parameters and the other three cases are for real structures with artificial parameters. As can be seen, CaMML's performance is consistent with the results in Fig. 5.17 under 300 samples. It clearly outperformed the others for $\alpha = 0.2$ and 1, but its advantage is not so clear for the difficult case $\alpha = 5$. The story, however, is different when using the default model parameters. Although some of the learners are indistinguishable from each other, CaMML has the highest edit distance in many cases. We suspect this is due to the MML prior of model parameters. By default, CaMML takes uniform parameter prior. But some real world models' parameters may be better represented by asymmetric Dirichlet distributions with parameters larger or smaller than 1.[15]

By now, we have given a high level description of the CaMML algorithm and some non-exhaustive experiments to compare CaMML with some state-of-the-art/popular structure learning algorithms. In the next section, CaMML will be used as a default structure learner to test the potential of probabilistic Markov blanket prior on structure learning.

5.3.2 CaMML's Accuracy with Approximated Moral Prior

One of the key features of CaMML is the ability to take different kinds of probabilistic priors including directed and undirected arc priors, node order prior, ancestor prior, etc, with a user specified confidence level for each piece of prior information. Unfortunately, the

[15] An interesting future research area is to let CaMML use estimated Dirichlet priors from data.

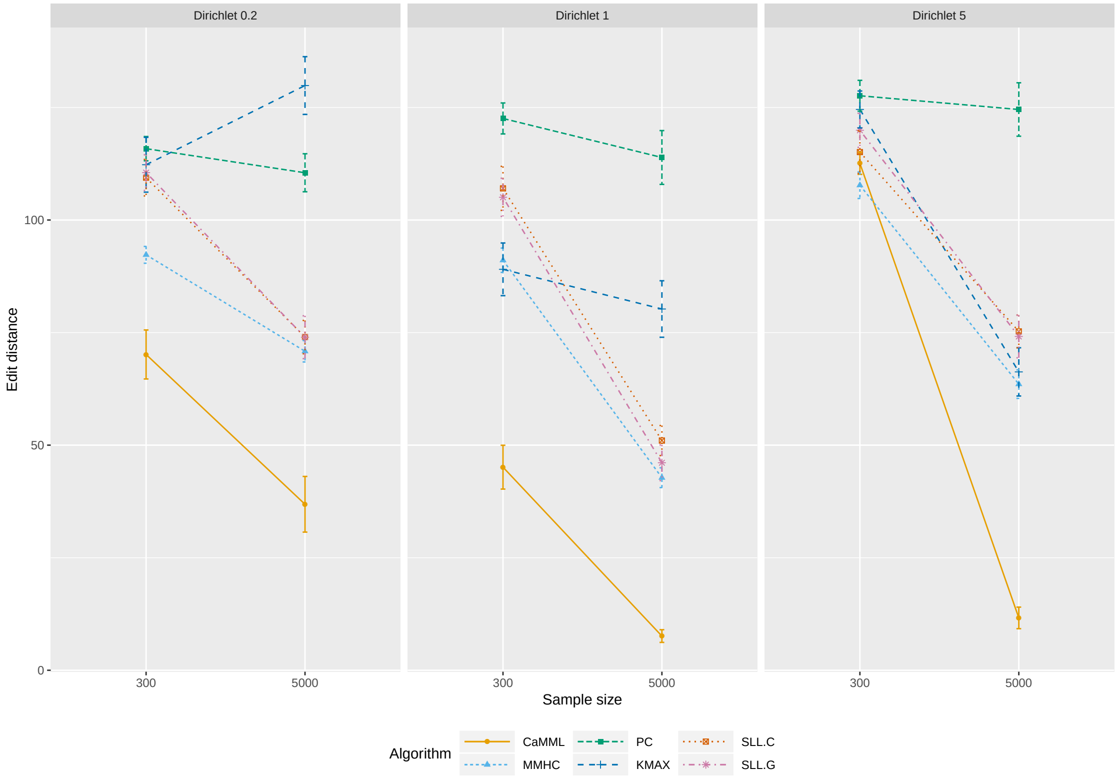


Fig. 5.17 Artificial models with 50 binary nodes and max fan-in 5. The model parameters are sampled from symmetric Dirichlet distribution with different concentration parameters. For each case, there are 10 data sets of size 300 and 5000 for learning. An average edit distance is reported with a 95% confidence interval.

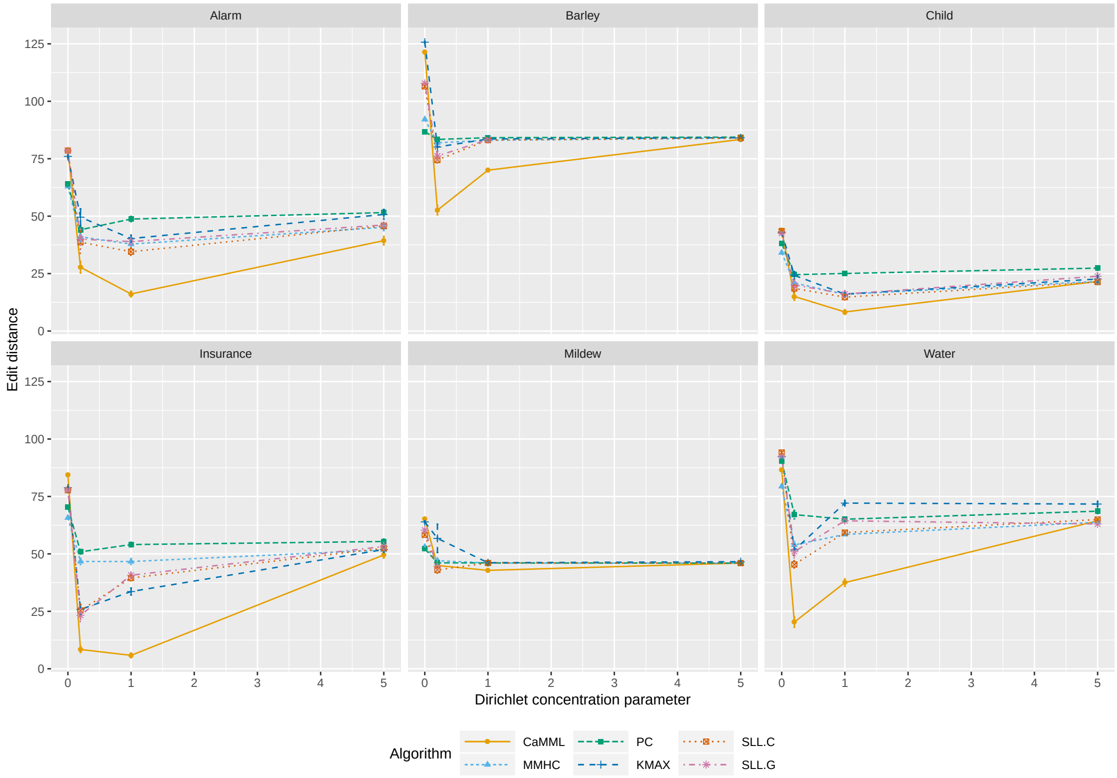


Fig. 5.18 Medium size real models with real and artificial parameters. The Dirichlet concentration parameter 0 is the case when real parameters are taken. The sample size for learning is 300. For each case, there are 20 data sets for learning. An average edit distance is reported with a 95% confidence interval.

program does not directly take a Markov blanket prior.[16] There is, however, an indirect (weak) way of encoding Markov blanket prior into CaMML via the undirected arc prior. For example, if the learned Markov blankets from a data set over three variables $\{a,b,c\}$ are $MB(a) = \{b,c\}, MB(b) = \{a,c\}, MB(c) = \{a,b\}$, then an undirected graph can be obtained by connecting each Markov blanket to its target node as shown in Fig. 5.19. This graph can then be encoded into CaMML as undirected arc priors with a real confidence taking from $[0,1]$ for each arc. The extreme values 0 and 1 correspond to deterministic confidence that totally ignores and believes a given piece of information.

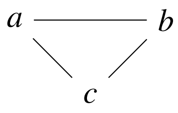


Fig. 5.19 The undirected graph formed by the learned Markov blankets $MB(a) = \{b,c\}, MB(b) = \{a,c\}, MB(c) = \{a,b\}$.

In what follows, this weak way of encoding Markov blanket prior is referred to as *approximated moral prior*.Noticing that an approximated moral prior may or may not be moral, depending on whether or not the used Markov blankets are moral.

Before testing CaMML with approximated moral prior, we should point out an issue that this encoding method may cause. If an arc *ab* does not exist in the true DAG (Fig. 5.19), but appears in the moral graph to connect the common parents of a child, then given a high confidence of the arc *ab* will misdirect CaMML when sampling through the TOM space, unless there are enough data to correct this mistake. Hence, even if the learned Markov blankets are perfect, unless the parent set of each node forms a clique in the true DAG, there always exist false positive arcs with respect to the true skeleton. In that sense, one should not expect CaMML to output perfect DAGs when taking perfect Markov blanket priors with absolute confidence 1, even learning under infinite samples.

[16] A direct way of incorporating Markov blanket prior is to get the Markov blankets from a sampled TOM and compare these Markov blankets with those given by a user. If the sampled Markov blankets do not consistent with the given ones, a penalty is applied to the sampled TOM. Otherwise, a reward is applied.

We set up experiments to run CaMML with no prior information given and with approximated moral priors for different confidence levels. To test the effectiveness of this weak way of encoding Markov blankets, we randomly sample approximated moral priors with a variety of qualities that are measured by edit distance to the true moral graphs. The way CaMML parses a given arc probability, say an undirected edge ab with confidence 0.7 is converting this connection into four distinct relationships $a \rightarrow b$, $b \rightarrow a$, $a \not\rightarrow b$ and $b \not\rightarrow a$. The expression $a \not\rightarrow b$ represents a is ahead of b in the TOM but there is no direct connection between them. As no other information about nodes a, b is given, CaMML then treats the first two relationships equal likely and the last two equal likely. This implies that each of the first two relationships has probability 0.35 and each of the last two relationships has probability 0.15 each. Although these values look small, a practical model is likely to be sparse, hence it is not unrealistic to have small arc probability $p(arc) \approx 0.1$, in which case even a prior probability of 0.15 is relatively strong.

The experiments are conducted on 5 random Bayesian networks with 50 binary variables and maximum fan-in 5. The parameters are sampled from uniform distributions. Each of these BNs is then used to generate 5 data sets of 300 samples. To obtain approximated moral priors with edit distances taking from $\{0, 10, 30, 50, 70, 90\}$, we randomly select pairs of vertices from a true moral graph and toggle the existence of an edge between each pair. To test CaMML's reaction on taking different confidence levels, all edges in the approximated moral priors are assigned each of the three values from $\{0.5, 0.7, 0.9\}$ as their confidence.

Fig. 5.20 shows the edit distance of the learned structures by CaMML with and without approximated moral priors. The black dash line in the middle of the figure corresponds to CaMML's mean edit distance without any prior. It is included in the results as a baseline. The two orange dash lines represent the 95% confidence interval. The priors are taken by CaMML with three confidence levels 0.9, 0.7, 0.5 as indicated by the three coloured lines red, green, blue in the plot. The values on the x-axis indicate the quality of the randomly constructed priors from the true prior that is at the 0 value. The last three labels on the x-axis correspond to priors inferred from MMLCPT's outputs. The label "pri1" corresponds to approximated moral priors learned by MMLCPT which are mostly immoral. The label "pri2" corresponds to the outputs from MMLCPT with minimum degree moralization to ensure morality. Both pri1 and pri2 take the pre-determined probability 0.5 for all arcs, because 0.5 confidence performed relatively better than the other two. The last label "pri3" corresponds to the moralized priors used by the second type, but with different confidence for different arcs. In particular, arcs correspond to the first learned Markov blanket candidate for each node take 0.95 confidence indicating the certainty of these arcs. The rest of the arcs take 0.3 confidence indicating relatively less certainty.[17]

As mentioned earlier, a perfect approximated moral prior may still contain false positive edges that do not exist in the true DAG. Hence, when taking the true moral graphs as priors, CaMML does not produce absolute correct structures. This is consistent with the non-zero edit distance as shown in Fig. 5.20 when x-axis is equal to 0. Moreover, the higher the arc confidence, the larger the edit distance, because some correct information inferred from data is largely disregarded due to high confidence false priors. The trend that 0.9 confidence almost always has the largest edit distance across the figure suggests that the confidence is too high to produce any useful information in practice.

The positive side of Fig. 5.20 is that lower arc confidence levels 0.5 and 0.7 greatly increased CaMML's reconstruction accuracy, provided approximated moral priors that are no more than 30 edit distance away from the true moral graphs. Once the prior quality is between 50 to 90, there is no clear distinction between CaMML's accuracy with and without this type of prior.

The negative side of the story is that MMLCPT (which has been proved to be superior over others under a similar experimental setting) does not seem to have the potential to

[17] Our observations strongly suggest that the first learned Markov blanket variable by the MMLCPT algorithm is a true positive. The probability of a newly added node being a true positive degrades quickly as the number of existing nodes increasing.

significantly reduce CaMML's edit distance when taking no priors at all as shown by the average edit distance at pri1. After enforcing moralization (i.e., pri2), the average edit distance reduced to a similar level as the baseline. This again proves the potential of moralization on improving Markov blanket quality. We observed that the first found Markov blanket candidate by MMLCPT is highly likely to be a true positive, with a likelihood approximately over 90%. As more candidates are included, the current added node being a true positive decreases quickly. For this reason, we test moralized approximated moral prior with different arc confidence. For simplicity, we start at 0.95 for the first candidate and 0.3 for the rest. The accuracy is shown in the figure by the purple point (i.e., pri3). Although the average edit distance is further reduced (insignificantly) and the error bar may shrink more by running more experiments to produce statistically higher accuracy, it is unlikely that pri3 will help CaMML to perform significantly better than the baseline.

The use of the priors pri1, pri2 and pri3 together indicate that almost no existing Markov blanket discovery algorithm can produce good enough approximated moral priors for CaMML right now; however, it also shows that there is a potential for improving general causal discovery results through better Markov blanket learning.

5.3.3 CaMML's Speed with Approximated Moral Prior

This section studies the feasibility of using approximated moral priors to scale up CaMML to learn structures from high-dimension data sets. In particular, the focus here is on reducing the running time of the algorithm. As discussed previously, CaMML can be made faster by reducing the number of searching or sampling steps. This, however, could also reduce the chance of finding an optimal structure. Hence, this section explores the possibility of using approximated moral priors to increase CaMML's accuracy to compensate for the loss of accuracy due to less searching or sampling iterations.

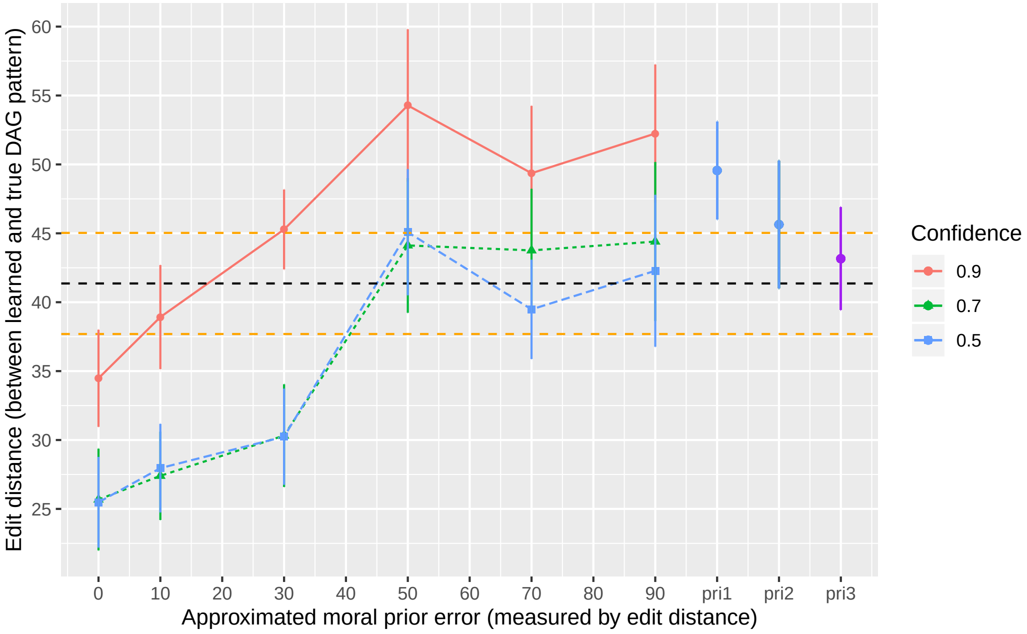


Fig. 5.20 CaMML's edit distance vs. the quality of approximated moral priors. The black and orange dash lines correspond to CaMML's average edit distance and a 95% confidence interval when not taking priors. X-axis measures the quality of the used approximated moral priors with 0 being the true moral priors. The last three labels on the x-axis correspond to approximated moral priors learned by MMLCPT (that are mostly immoral) with 0.5 confidence, pri1 with enforced moralization, and pri2 with 0.95 confidence for the first learned Markov blanket candidate and 0.3 for the rest.

The experiments are conducted on the same artificial models and data sets used for testing CaMML's reconstruction accuracy in Section 5.3.1. That is, 10 random Bayesian networks with 50 binary variables and maximum fan-in 5. The model parameters are sampled from uniform distributions. Each model is then used to generate 10 different data sets of size 300.

CaMML is first set to take dn^3 mutations in each of the pre-sampling phases, where d is equal to 7 and 10 corresponds to greedy and annealing searches respectively. At this setting, the Metropolis-Hasting (MH) process is then set to take $200n^3, 200n^2, 200n$ and $200\log n$ samples respectively. To test the effectiveness of approximated moral prior, we run CaMML with no prior, artificially constructed approximated moral priors (whose edit distance to the true moral graph is 30) and MMLCPT produced approximated moral priors. All priors are used at 0.3 confidence level. The experiments are then repeated with the same settings but dn^2 and $d * 1000$ mutations respectively.

First, we notice in Table 5.6 that the number of mutations in the pre-sampling phases is essential for getting small edit distance. Given that CaMML takes a large number of mutations, there is no significant difference between the edit distance produced by the MH algorithm for any sample size. However, when reducing the number of mutations, the advantage of sampling more TOMS in the MH phase is quite obvious.

It is expected that artificial approximated moral priors with 30 edit distance would produce the best results, because the previous section has shown that high quality Markov blankets encoded in this weak way could increase CaMML's reconstruction accuracy.

We are more interested in the case when CaMML takes MMLCPT produced Markov blanket priors. As it can be seen in this figure, most cases when CaMML takes MMLCPT priors the edit distance are higher than not take any priors. There are, however, two cases (i.e., $200n^2$ MH samples) in less mutation situations (i.e., $dn^2, d * 1000$), where the confidence intervals have small overlaps. To prove that the difference in average edit distances can be significant in these two cases, we increase the number of experiments by another 100 to shrink the error bars. These further results are shown in the brackets next to the original results in the table. They have shown that when using MMLCPT produced priors, CaMML can outputs statistically significant better structures with fewer mutations and less MH samples.

Further looked into the results, in the 2-dimensional space spanned by the number of mutations and number of MH samples, we conjecture that there is a region near less mutations and moderate MH samples (in this case dn^2 or $1000d$ mutations and $200n^2$ MH samples), in which MMLCPT produced approximated moral prior can help CaMML to find better structures. Although not thoroughly proved, this conjecture makes sense. Because a large number of mutations could obtain a reasonably good starting point for the subsequent MH sampling phase, so leaves very little chance for it to go wrong, with or without priors, unless the priors are mistaken with high confidence. Fewer mutations gives some room for the sampling phase to improve the current starting TOM, the final TOM, of course, depends on the number of sampled TOMs and the priors used to guide the sampling process. Although we conjecture such a region exists, it may be at a different location for different quality of priors with different confidence level.

In summary, the experiments demonstrated that there are cases that approximated moral prior could improve CaMML's reconstruction accuracy, even though the priors do not come from high quality learned Markov blankets. Since both improved cases appear for smaller number of mutations and MH samples, which is essential to scale up CaMML to larger models, it indicates that there is a possibility that CaMML could be scaled up without losing much of its accuracy.

Table 5.6 A summary of CaMML's average edit distances (with 95% confidence intervals) when running with different number of mutations in the pre-sampling phase and sampling different number of TOMs in the Metropolis-Hasting phase. For each setting, CaMML is tested with and without synthetics and MMLCPT approximated moral priors. The constant d in the greedy and annealing search takes value 7 and 10 respectively. CaMML with artificial priors always have the best edit distance because these priors are generated with high quality. The MMLCPT learned priors can reduce the edit distance of CaMML (with no prior) in some cases that are highlighted. The confidence intervals are further reduced by increasing the number of experiments to justify the significance of the differences.

Pre-sampling mutations	MH samples	No prior	Artificial priors	MMLCPT priors
dn^3	$200n^3$	$43 + -2.3$	$32.8 + -2.1$	$44 + -2.5$
	$200n^2$	$41.9 + -2.3$	$32.7 + -2.1$	$45.2 + -2.4$
	$200n$	$42.1 + -2.2$	$32.5 + -2.1$	$44.2 + -2.5$
	$200\log n$	$43 + -2.3$	$32.8 + -2.1$	$44.9 + -2.6$
dn^2	$200n^3$	$50.2 + -2.6$	$35.5 + -2.3$	$47.5 + -2.6$
	$200n^2$	$66.3 + -3$ $(66 + -1.9)$	$48.1 + -3.3$	$60.6 + -2.9$ $(61.6 + -2)$
	$200n$	$72.9 + -2.7$	$61.8 + -3.1$	$71.8 + -3$
	$200\log n$	$72.6 + -2.7$	$62 + -2.8$	$71.3 + -2.4$
$d * 1000$	$200n^3$	$47.6 + -2.4$	$32.7 + -2.1$	$47 + -2.4$
	$200n^2$	$65.2 + -2.9$ $(64.5 + -1.8)$	$42.5 + -2.6$	$56.6 + -2.8$ $(56.2 + -1.9)$
	$200n$	$83.8 + -2.7$	$77.8 + -2.6$	$85.4 + -2.7$
	$200\log n$	$82.4 + -2.6$	$78 + -2.8$	$85.6 + -2.5$

Chapter 6

Conclusions and Future Work

Learning Bayesian network structures from observational data remains a challenging problem in machine learning. A satisfactory solution to this problem has a great potential of revealing complex causal relations among variables that are beyond human beings' capability of handling. Hence, the problem is likely to keep attracting attention from researchers in a variety of scientific fields. By now, there have been a few methods claiming to be able to push the limit of this automated learning process to data sets containing thousands or even millions of variables such as [75, 68]. Scalability is an attractive feature, but alone, it is an unconvincing measure of the performance of a structure learner. Like many machine learning tasks, the objective of learning is to find an optimal balance between metrics, e.g., the balance between model fit and complexity or a data analysis algorithm's utility and privacy in the perspective of differential privacy. In scaling up structure learning, this thesis emphasized that the balance is between an algorithm's scalability and its reconstruction accuracy. Most of the work that has been done in this thesis focus on the preliminary steps of the Local-to-Global strategy of structure learning, rather than pushing the limit of the current state-of-the-art algorithms to the next level.

6.1 Main Contributions

Research Question 1 (Section 1.2) concerns how to learn Markov blankets without having to learn the regional structures within them. As an answer to this question, a general framework using the minimum message length principle to learn Markov blankets was proposed in Chapter 4. It predicts the target variable using either the conditional probability table, naive Bayes or Markov blanket polytree models, with the potential Markov blanket candidates being the predictors. The one with the shortest minimum message length is selected as the best model, which in turns gives the best Markov blanket candidates. Although each of the three predictive models has a fixed structure, none of these structures has to be true for such a framework to work. The experimental work presented in Chapter 5 implies that the framework with the conditional probability table and Markov blanket polytree models have superior performance when comparing with other Markov blanket learners. In particular, the conditional probability table model shows superior performance in almost all tested cases except when the sample size is very small relative to the model complexity.

As explained in Chapter 3, a set of learned Markov blankets may not have a consistent DAG at all. If so, one is certain that there are false findings in these Markov blankets. A natural question to ask is how to efficiently tell whether or not a set of learned Markov blankets is consistent with at least one DAG. This is exactly what has been asked by Research Question 3 (Section 1.2). To answer this question, Chapter 3 stated the equivalent relation between Markov blanket consistency and graph morality. It introduced the concept of weakly recursively simplicial and proved that it is equivalent to being moral. Although deciding morality is NP-complete in general, the problem can be efficiently decided for graphs with low maximum degrees. To be precise, the morality of graphs with maximum degree 3 and 4 can be decided in linear and quadratic time respectively. The algorithms were given in Chapter 3 together with proofs. Moreover, it was also experimentally justified in Chapter 5 that enforcing morality using the minimum degree and minimum deficiency algorithms can improve the Markov blanket discovery accuracy of the MMLCPT method.

One of the motivations of this project is to set up a framework that allows a Bayesian learner to utilize the information learned in preceding steps, i.e., Markov blankets, to scale up structure learning to high dimensional data sets (Research Question 4, Section 1.2). To achieve this goal, this thesis studied the possibility of encoding Markov blankets as probabilistic priors for a Bayesian learner. In particular, the proposed encoding method is called approximated moral prior, which is a weak way of encoding the learned Markov blankets by connecting all candidates in a Markov blanket as neighbours of the target node. The framework was tested on the CaMML algorithm under a variety of experimental settings. The results demonstrated that this weak way of encoding the learned Markov blankets has the potential to scale up CaMML to larger models while retaining a similar accuracy.

6.2 Future work

Due to time constraints, there is only so much that one can do in the scope of this thesis. Below are some of the interesting problems that arose during the investigation of the main objectives of this project.

1. Markov blanket prior from counting the number of consistent sets of Markov blankets. Given that moral graphs are weakly recursively simplicial, they are generalizations of chordal graphs. Generating functions have been used by Wormald [96] to count the number of labelled chordal graphs. Moral graphs do not have the desirable properties that chordal graphs have, so one is not sure whether or not there could be an analytical solution to counting them. If not, an alternative is to get a useful approximate (asymptotic) formulae for counting.

2. A stronger encoding of Markov blankets for CaMML. The current version of CaMML samples from the space of all Totally Ordered Models (TOMs). A stronger Markov blanket encoding is to let CaMML take into account the learned Markov blankets, then either penalize or reward a sampled TOM, depending on its consistency with the given Markov blankets.

3. The experimental work in Chapter 5 mostly used a fixed confidence level for all arc priors. In reality, this is hardly the case. A sophisticated alternative is to estimate a confidence level for each arc, e.g., using bootstrapping [30].

References

[1] S. Acid, L. M. de Campos, and M. Fernández. Score-based methods for learning Markov boundaries by searching in constrained spaces. *Data Mining and Knowledge Discovery*, 26(1):174–212, 2013.

[2] H. Akaike. A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*, pages 215–222. Springer, 1974.

[3] C. F. Aliferis, I. Tsamardinos, and A. Statnikov. HITON: a novel Markov blanket algorithm for optimal variable selection. In *AMIA Annual Symposium Proceedings*, pages 21–25. American Medical Informatics Association, 2003.

[4] C. F. Aliferis, A. R. Statnikov, I. Tsamardinos, S. Mani, and X. Koutsoukos. Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11:171–234, 2010.

[5] C. F. Aliferis, A. R. Statnikov, I. Tsamardinos, S. Mani, and X. Koutsoukos. Local causal and Markov blanket induction for causal discovery and feature selection for classification part II: Analysis and extensions. *Journal of Machine Learning Research*, 11:235–284, 2010.

[6] E. Amir. Efficient approximation for triangulation of minimum treewidth. In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence*, pages 7–15. Morgan Kaufmann Publishers Inc., 2001.

[7] APA. *APA Publication Manual*. American Psychological Association, 6th edition, 2013.

[8] S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987.

[9] D. M. Boulton and C. S. Wallace. The information content of a multistate distribution. *Journal of Theoretical Biology*, 23(2):269–278, 1969.

[10] W. Buntine. Theory refinement on Bayesian networks. In *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann Publishers Inc., 1991.

[11] J. Cheng, C. Hatzis, H. Hayashi, M. A. Krogel, S. Morishita, D. Page, and J. Sese. Kdd cup 2001 report. *ACM SIGKDD Explorations Newsletter*, 3(2):47–64, 2001.

[12] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning Bayesian networks from data: an information-theory based approach. *Artificial intelligence*, 137(1-2):43–90, 2002.

[13] D. M. Chickering. Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, 2(Feb):445–498, 2002.

[14] D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

[15] D. M. Chickering, D. Geiger, and D. Heckerman. Learning Bayesian networks is NP-hard. Technical report, Citeseer, 1994.

[16] D. M. Chickering, D. Heckerman, and C. Meek. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research*, 5(Oct):1287–1330, 2004.

[17] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[18] G. F. Cooper. A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Mining and Knowledge Discovery*, 1(2): 203–224, 1997.

[19] G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347, 1992.

[20] G. F. Cooper, C. F. Aliferis, R. Ambrosino, J. Aronis, B. G. Buchanan, R. Caruana, M. J. Fine, C. Glymour, G. Gordon, B. H. Hanusa, J. E. Janosky, C. Meek, T. Mitchell, T. Richardson, and P. Spirtes. An evaluation of machine-learning methods for predicting pneumonia mortality. *Artificial Intelligence in Medicine*, 9(2):107–138, 1997.

[21] R. G. Cowell. Conditions under which conditional independence and scoring methods lead to identical selection of Bayesian network models. *arXiv preprint arXiv:1301.2262*, 2013.

[22] H. Dai, K. B. Korb, C. S. Wallace, and X. Wu. A study of causal discovery with weak links and small samples. In *Proceedings of the 6th International Joint Conference on Artificial Intelligence*, pages 1304–1309. Citeseer, 1997.

[23] S. Dasgupta. Learning polytrees. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 134–141. Morgan Kaufmann Publishers Inc., 1999.

[24] L. M. de Campos. Independency relationships and learning algorithms for singly connected networks. *Journal of Experimental & Theoretical Artificial Intelligence*, 10 (4):511–549, 1998.

[25] S. R. de Morais and A. Aussem. A novel Markov boundary based feature subset selection algorithm. *Neurocomputing*, 73(4):578–584, 2010.

[26] R. Diestel. *Graph Theory (Graduate Texts in Mathematics)*. Springer, August 2005.

[27] D. Dor and M. Tarsi. A simple algorithm to construct a consistent extension of a partially oriented graph. *Technical Report R-185, Cognitive Systems Laboratory, UCLA*, 1992.

[28] M. J. Flores and J. A. Gámez. A review on distinct methods and approaches to perform triangulation for Bayesian networks. In *Advances in Probabilistic Graphical Models*, pages 127–152. Springer, 2007.

[29] L. Frey, D. Fisher, I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Identifying Markov blankets with decision tree induction. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, 2003.

[30] N. Friedman, M. Goldszmidt, and A. Wyner. Data analysis with Bayesian networks: A bootstrap approach. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 196–205. Morgan Kaufmann Publishers Inc., 1999.

[31] S. Fu and M. C. Desmarais. Fast Markov blanket discovery algorithm via local learning within single pass. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 96–107. Springer, 2008.

[32] D. Fulkerson and O. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15(3):835–855, 1965.

[33] T. Gao and Q. Ji. Efficient score-based Markov Blanket discovery. *International Journal of Approximate Reasoning*, 80:277–293, 2017.

[34] T. Gao and D. Wei. Parallel Bayesian network structure learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1671–1680, 2018.

[35] T. Gao, K. Fadnis, and M. Campbell. Local-to-global Bayesian network structure learning. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1193–1202, 2017.

[36] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W.H. Freeman, San Francisco, CA, 1979.

[37] D. Geiger, D. Heckerman, H. King, and C. Meek. Stratified exponential families: graphical models and model selection. *Annals of Statistics*, pages 505–529, 2001.

[38] S. B. Gillispie and M. D. Perlman. Enumerating Markov equivalence classes of acyclic digraph models. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pages 171–177. Morgan Kaufmann Publishers Inc., 2001.

[39] C. N. Glymour, R. Scheines, P. Spirtes, and K. Kelly. *Discovering Causal Structure: Artificial Intelligence, Philosophy of Science, and Statistical Modeling*. Academic Press, 1987.

[40] D. M. Haughton. On the choice of a model to fit data from an exponential family. *Annals of Statistics*, 16(1):342–355, 1988.

[41] D. Heckerman and D. Geiger. Likelihoods and parameter priors for Bayesian networks. *Tech. MSRTR-95-54. Microsoft Research*, 1995.

[42] P. Heggernes. Minimal triangulations of graphs: A survey. *Discrete Mathematics*, 306 (3):297–317, 2006.

[43] U. Kjærulff. Triangulation of graphs-algorithms giving small total state space. Technical report, Aalborg University, Denmark, 1990.

[44] M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, 5(May):549–573, 2004.

[45] D. Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of the 13th International Conference on Machine Learning*, pages 284–292, 1996.

[46] P. Larrañaga, C. M. H. Kuijpers, M. Poza, and R. H. Murga. Decomposing Bayesian networks: triangulation of the moral graph with genetic algorithms. *Statistics and Computing*, 7(1):19–34, 1997.

[47] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 157–224, 1988.

[48] C. Li and M. Ueno. An extended depth-first search algorithm for optimal triangulation of Bayesian networks. *International Journal of Approximate Reasoning*, 80:294–312, 2017.

[49] G. Li, H. Dai, and Y. Tu. Identifying Markov blankets using lasso estimation. In *Advances in Knowledge Discovery and Data Mining*, number 2004, pages 308–318. Springer, 2004.

[50] Y. Li, L. Allison, and K. Korb. Proving the NP-completeness of optimal moral graph triangulation. *arXiv preprint arXiv:1903.02201*, 2019.

[51] Y. Li, K. Korb, and L. Allison. The Complexity of Morality: Checking Markov Blanket Consistency with DAGs via Morality. *arXiv preprint arXiv:1903.01707*, 2019.

[52] X. Liu and X. Liu. Swamping and masking in Markov boundary discovery. *Machine Learning*, 104(1):25–54, 2016.

[53] X. Liu and X. Liu. Markov blanket and Markov boundary of multiple variables. *Journal of Machine Learning Research*, 19(1):1658–1707, 2018.

[54] M. G. Madden. A new Bayesian network structure for classification tasks. In *Irish Conference on Artificial Intelligence and Cognitive Science*, pages 203–208. Springer, 2002.

[55] D. Margaritis and S. Thrun. Bayesian network induction via local neighborhoods. In *Proceedings of the 12th Advances in Neural Information Processing Systems*, pages 505–511, 1999.

[56] C. Meek. Strong completeness and faithfulness in Bayesian networks. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, 1995.

[57] A. Nägele, M. Dejori, and M. Stetter. Bayesian substructure learning-approximate learning of very large network structures. In *Proceedings of the 18th European Conference on Machine Learning*, pages 238–249. Springer, 2007.

[58] R. E. Neapolitan. *Learning Bayesian Networks*, volume 38. Pearson Prentice Hall Upper Saddle River, NJ, 2004.

[59] J. R. Neil, C. S. Wallace, and K. B. Korb. Learning Bayesian networks with restricted causal interactions. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 486–493. Morgan Kaufmann Publishers Inc., 1999.

[60] T. Niinimaki and P. Parviainen. Local structure discovery in Bayesian networks. 2012.

[61] R. T. O'Donnell. *Flexible Causal Discovery with MML*. Monash University, 2010.

[62] T. Ohtsuki, L. K. Cheung, and T. Fujisawa. Minimal triangulation of a graph and optimal pivoting order in a sparse matrix. *Journal of Mathematical Analysis and Applications*, 54(3):622–633, 1976.

[63] S. Ott, S. Imoto, and S. Miyano. Finding optimal models for small gene networks. In *Biocomputing 2004*, pages 557–567. World Scientific, 2003.

[64] T. J. Ottosen and J. Vomlel. All roads lead to Rome-New search methods for the optimal triangulation problem. *International Journal of Approximate Reasoning*, 53 (9):1350–1366, 2012.

[65] S. Parter. The use of linear graphs in Gauss elimination. *SIAM review*, 3(2):119–130, 1961.

[66] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann San Mateo, CA, 1988.

[67] J. M. Peña, R. Nilsson, J. Björkegren, and J. Tegnér. Towards scalable and data efficient learning of Markov boundaries. *International Journal of Approximate Reasoning*, 45 (2):211–232, 2007.

[68] J. Ramsey, M. Glymour, R. Sanchez-Romero, and C. Glymour. A million variables and more: the Fast Greedy Equivalence Search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics*, pages 1–9, 2016.

[69] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[70] R. W. Robinson. Counting labeled acyclic digraphs. In *New Directions in the Theory of Graphs*, pages 239–273. Academic Press, 1973.

[71] D. J. Rose. A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. In *Graph Theory and Computing*, pages 183–217. Elsevier, 1972.

[72] D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, 5(2):266–283, 1976.

[73] M. Scanagatta, C. P. de Campos, G. Corani, and M. Zaffalon. Learning Bayesian networks with thousands of variables. In *Proceedings of the 28th Advances in Neural Information Processing Systems*, pages 1864–1872, 2015.

[74] M. Scanagatta, G. Corani, C. P. de Campos, and M. Zaffalon. Learning treewidth-bounded Bayesian networks with thousands of variables. In *Proceedings of the 29th Advances in Neural Information Processing Systems*, pages 1462–1470, 2016.

[75] M. Scanagatta, G. Corani, M. Zaffalon, J. Yoo, and U. Kang. Efficient learning of bounded-treewidth Bayesian networks from complete and incomplete data sets. *International Journal of Approximate Reasoning*, 95:152–166, 2018.

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

[77] M. Scutari. Learning Bayesian networks with the bnlearn R package. *arXiv preprint arXiv:0908.3817*, 2009.

[78] T. Silander. On sensitivity of the MAP Bayesian network structure to the equipment sample size parameter. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, 2007.

[79] T. Silander and P. Myllymäki. A simple approach for finding the globally optimal Bayesian network structure. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, pages 445–452. AUAI Press, 2006.

[80] A. P. Singh and A. W. Moore. Finding optimal Bayesian networks by dynamic programming. 2004.

[81] M. Sipser. *Introduction to the Theory of Computation*, volume 2. Thomson Course Technology Boston, 2006.

[82] P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, Prediction, and Search.* MIT Press, 2000.

[83] E. V. Strobl and S. Visweswaran. Markov boundary discovery with ridge regularized linear models. *Journal of Causal Inference*, 4(1):31–48, 2016.

[84] J. Suzuki. A theoretical analysis of the BDeu scores in Bayesian network structure learning. *Behaviormetrika*, 44(1):97–116, 2017.

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

[86] L. Trevisan. Non-approximability results for optimization problems on bounded degree instances. In *Proceedings of the 33th Annual ACM Symposium on Theory of Computing*, pages 453–461. ACM, 2001.

[87] I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Time and sample efficient discovery of Markov blankets and direct causal relations. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 673–678. ACM, 2003.

[88] I. Tsamardinos, C. F. Aliferis, A. R. Statnikov, and E. Statnikov. Algorithms for large scale Markov blanket discovery. In *FLAIRS Conference*, pages 376–381, 2003.

[89] I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

[90] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the 6th Conference on Uncertainty in Artificial Intelligence*, pages 255–270. Elsevier Science Inc., 1990.

[91] T. S. Verma and J. Pearl. Deciding morality of graphs is NP-complete. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 391–399. Elsevier, 1993.

[92] C. S. Wallace. *Statistical and inductive inference by minimum message length*. Springer, 2005.

[93] C. S. Wallace and D. M. Boulton. An information measure for classification. *The Computer Journal*, 11(2):185–194, 1968.

[94] C. S. Wallace and P. R. Freeman. Estimation and inference by compact coding. *Journal of the Royal Statistical Society*, 49(3):240–265, 1987.

[95] W. X. Wen. Optimal decomposition of belief networks. In *Proceedings of the 6th Annual Conference on Uncertainty in Artificial Intelligence*, pages 209–224. Elsevier Science Inc., 1990.

[96] N. C. Wormald. Counting labelled chordal graphs. *Graphs and Combinatorics*, 1(1): 193–200, 1985.

[97] S. Yan, C. Cui, B. Sun, and R. Wang. Markov Boundary Discovery Based on Variant Ridge Regularized Linear Models. *IEEE Access*, 7:113206–113215, 2019.

[98] M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM Journal on Algebraic Discrete Methods*, 2(1):77–79, 1981.

[99] S. Yaramakala and D. Margaritis. Speculative Markov blanket discovery for optimal feature selection. In *Proceedings of the 5th IEEE International Conference on Data Mining*, pages 4–7. IEEE, 2005.

[100] L. Zhang and D. Poole. Sidestepping the triangulation problem in Bayesian net computations. In *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*, pages 360–367. Elsevier, 1992.

[101] Y. Zhang, Z. Zhang, K. Liu, and G. Qian. An Improved IAMB Algorithm for Markov Blanket Discovery. *Journal of Computers*, 5(11):1755–1761, 2010.

[102] J. Zhao and S. Ho. Improving Bayesian network local structure learning via data-driven symmetry correction methods. *International Journal of Approximate Reasoning*, 107: 101–121, 2019.

Appendix A

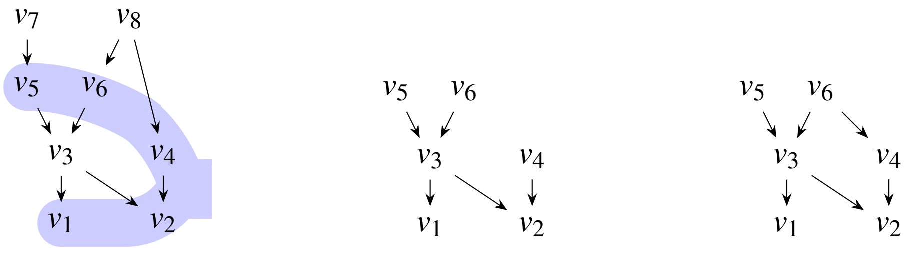
Inapproximability of the MBPT Problem

A.1 Introduction

Polytrees are relatively simpler graphical models (comparing to DAGs) to represent the relationships between random variables. They are often used when one seeks fast exact belief propagation on graphical models. Recall that a polytree is a directed acyclic graph whose skeleton is a tree (Definition 3.1.19). Although polytrees are simpler than general DAGs, learning the optimal maximum likelihood polytree from data is just as difficult as learning the optimal Bayesian network, unless the polytree is a Chow-Liu tree (i.e., a polytree with max fan-in 1) [17]. This is the consequence of the inapproximability of the optimal polytree problem that was proved in [23]. In that work, the author drew a polynomial time gap-preserving reduction from the MAX 3SAT-E3 problem to prove that the optimal max fan-in 2 polytree problem is a c-gap problem, so it cannot be approximated to a better constant factor than the size of the gap.[1]

This appendix studies the inapproximability of optimizing the learning of a Markov blanket polytree that is from a more restricted model family. The following is a formal definition of this model.

[1] In general, the gap c in a c-gap problem does not have to be constant. It can be a function of the input size.



(a) DAG (b) MBPT (c) non-MBPT

Fig. A.1 A DAG, a MBP of the variable v_3 and a non-MBP of the variable v_3.

Definition A.1.1. *Let $< G = (V,E), P >$ be a Bayesian network. A **Markov Blanket polytree** T_i of a target variable X_i is a polytree over the target variable and its Markov blanket such that X_i's Markov blanket is the same in the polytree T_i and the DAG G.*

In a Markov Blanket Polytree (MBP), each of the other variables is either a parent, child or spouse of the target variable, provided they have a common child.

Example A.1.1. *Let G be the DAG as shown in Fig. A.1a. For the variable v_3, the graph in Fig. A.1b is a Markov blanket polytree of v_3 whilst the graph in Fig. A.1c is not.*

In practice, when the true model is unknown, the true Markov blanket is unknown. Given a data set over the set $V = (X_1, \ldots, X_n)$ of variables, for the target variable X_i, a Markov blanket polytree is a polytree over a possible candidate subset $B_i \subseteq V$ that satisfies the above definition. Assuming X_i and B_i are fixed, the aim is to find the best Markov blanket polytree model over $B_i \cup \{X_i\}$ with respect to an objective function, in this case the maximum likelihood score.

Comparing with general polytree structures, a Markov blanket polytree is more restricted because it has to satisfy the Markov blanket condition in addition to being a polytree. Proposition 4.2.2 states a recursive formula to compute the number of labelled Markov blanket polytrees. The number is much smaller when compared with the number of labelled DAGs and polytrees. Hence, one would like to know whether or not there is a polynomial time algorithm that finds the optimal maximum likelihood Markov blanket polytree model.

According to the Markov condition (Definition 3.1.29) and the equivalence between log likelihood and entropy (as stated in [23]), the log likelihood of a Markov blanket polytree G can be written as

$$L(G) = -\sum_{i=1}^{n} H(X_i \mid \Pi_i). \tag{A.1}$$

That is the negative sum of the entropy for each random variable conditioning on its parent set in the graph G. Therefore, finding the maximum likelihood Markov blanket polytree model is equivalent to finding the model with the minimum total conditional entropy. The following statement defines the problem as an optimization problem.

Maximum likelihood Markov blanket polytree (ML MBPT)

INSTANCE: A data set D that contains m observations for n random variables $(X_1, \ldots, X_n)$, where $0 < m < \infty$. A target variable X_i and its Markov blanket B_i.

QUESTION: What is the optimal maximum likelihood score in all possible Markov blanket polytrees over $B_i \cup \{X_i\}$?

A.2 Gap-preserving reduction

The rest of this appendix proves that it is NP-hard to approximate the ML MBPT problem within a certain constant factor. The proof is similar to the work in [23], by producing a polynomial time gap-preserving reduction from the c-gap version of the MAX 3SAT-E3 problem to the ML MBPT problem. In addition to the proof in [23], this reduction must guarantee the transformed graph satisfies both the Markov blanket and polytree constraints. The MAX 3SAT-E3 problem is an optimization problem that is stated below.

MAX 3SAT-E3

INSTANCE: A CNF formula ϕ such that each clause contains at most 3 literals and each variable appears exactly 3 times.

QUESTION: What is the maximum number of clauses that can be simultaneously satisfied?

The problem is known to be NP-hard to approximate to within a constant factor $\frac{7}{8}+\varepsilon$ [86]. The c-gap version of this problem is a promise problem[2] that is stated below.

c-gap MAX 3SAT-E3

INSTANCE: A 3SAT-E3 formula ϕ with m clauses such that it is

- either a YES instance if all m clauses are satisfied
- or a NO instance if no more than $\frac{m}{c}$ clauses are satisfied, for $c > 1$.

QUESTION: Is the formula ϕ a YES or NO instance?

It can be proved by gap-producing reduction from the MAX 3SAT problem to show that the c-gap version of this problem is also NP-hard. If we can construct a polynomial time reduction from this problem to the ML MBPT problem while still preserves the gap (not necessarily the same magnitude), then it is equivalent to showing that our problem is also hard to approximate to within a constant factor, because the NP-hardness of a gap problem implies that the original problem is NP-hard to approximate to within a factor. The following theorem is based on Markov blanket polytrees with maximum fan-in 3. The inapproximability of larger max fan-in models follows from this result.

[2] A promise problem is a generalization of a decision problem, where only a subset of the entire input space is considered. For example, instead of taking the set of all graphs as inputs, a promise problem may only consider connected graphs. Hence, it is the union of two disjoint subsets, one contains all the YES instance and one contains all the NO instance. If a given input does not belong to either of these two, the algorithm for solving the promise problem can return any output or may not even halt.

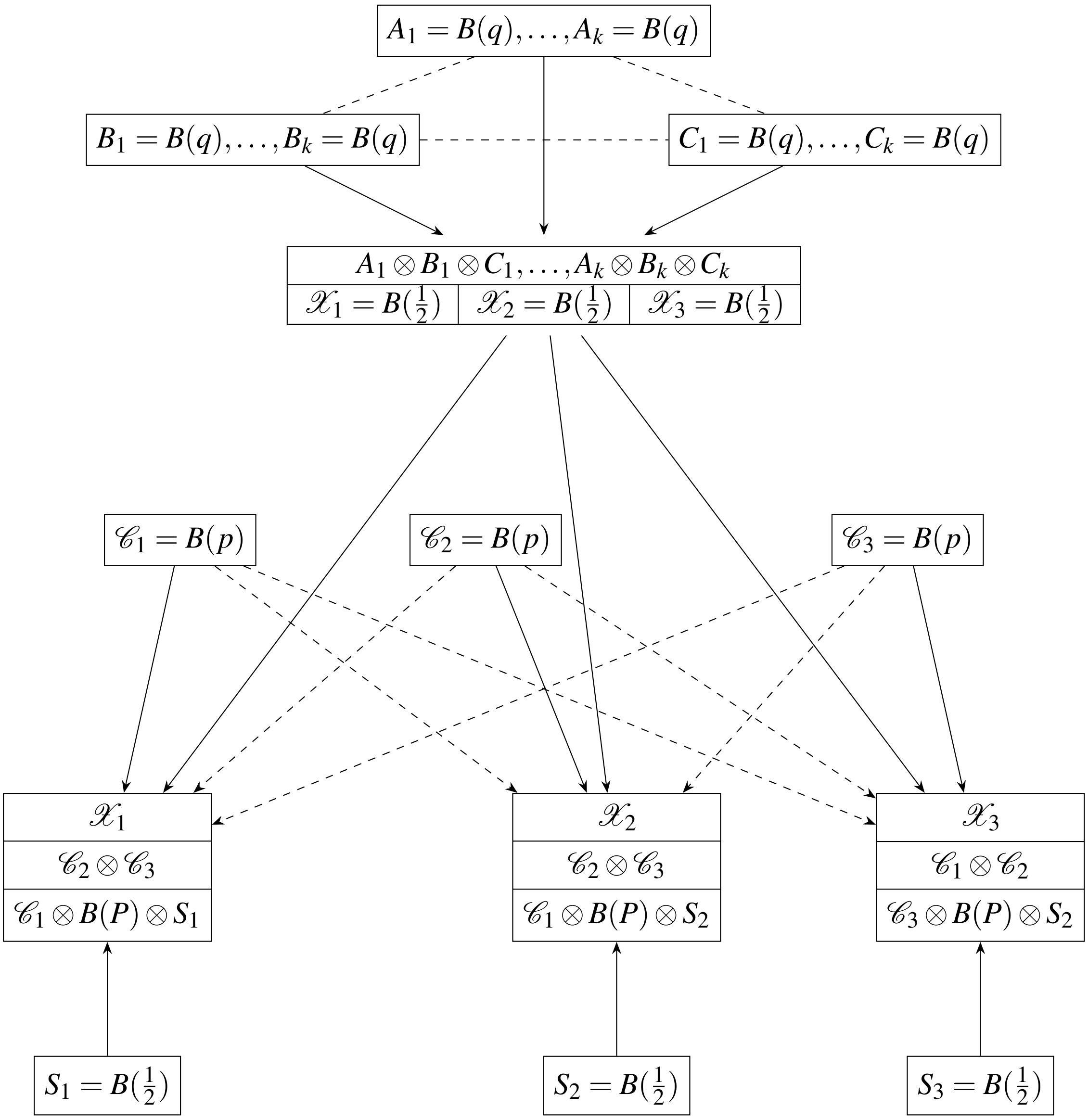

$A_1 = B(q), \ldots, A_k = B(q)$
$B_1 = B(q), \ldots, B_k = B(q)$
$C_1 = B(q), \ldots, C_k = B(q)$
$A_1 \otimes B_1 \otimes C_1, \ldots, A_k \otimes B_k \otimes C_k$
$\mathscr{X}_1 = B(\frac{1}{2})$
$\mathscr{X}_2 = B(\frac{1}{2})$
$\mathscr{X}_3 = B(\frac{1}{2})$
$\mathscr{C}_1 = B(p)$
$\mathscr{C}_2 = B(p)$
$\mathscr{C}_3 = B(p)$
$\mathscr{X}_1$
$\mathscr{C}_2 \otimes \mathscr{C}_3$
$\mathscr{C}_1 \otimes B(P) \otimes S_1$
$\mathscr{X}_2$
$\mathscr{C}_2 \otimes \mathscr{C}_3$
$\mathscr{C}_1 \otimes B(P) \otimes S_2$
$\mathscr{X}_3$
$\mathscr{C}_1 \otimes \mathscr{C}_2$
$\mathscr{C}_3 \otimes B(P) \otimes S_2$
$S_1 = B(\frac{1}{2})$
$S_2 = B(\frac{1}{2})$
$S_3 = B(\frac{1}{2})$

Fig. A.2 A MBP transformed from a 3SAT-E3 formula $\phi = (X_1 \vee X_2 \vee X_3) \wedge (\overline{X_1} \vee \overline{X_2} \vee X_3) \wedge (\overline{X_1} \vee \overline{X_2} \vee \overline{X_3})$. The formula is satisfiable with an assignment $X_1 = 1, X_2 = 0, X_3 = 0$.

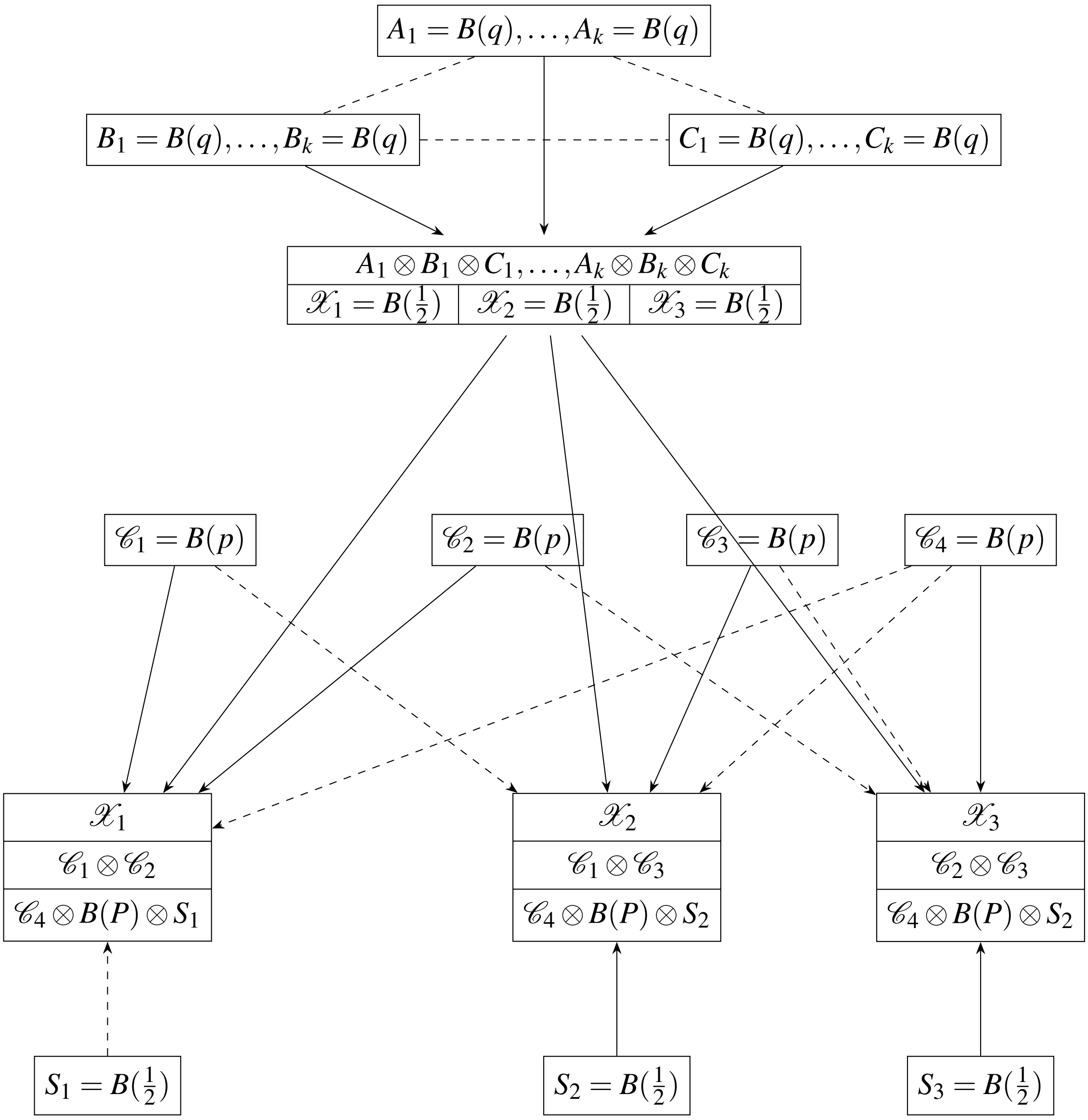

$A_1 = B(q), \ldots, A_k = B(q)$
$B_1 = B(q), \ldots, B_k = B(q)$
$C_1 = B(q), \ldots, C_k = B(q)$
$A_1 \otimes B_1 \otimes C_1, \ldots, A_k \otimes B_k \otimes C_k$
$\mathscr{X}_1 = B(\frac{1}{2})$
$\mathscr{X}_2 = B(\frac{1}{2})$
$\mathscr{X}_3 = B(\frac{1}{2})$
$\mathscr{C}_1 = B(p)$
$\mathscr{C}_2 = B(p)$
$\mathscr{C}_3 = B(p)$
$\mathscr{C}_4 = B(p)$
$\mathscr{X}_1$
$\mathscr{C}_1 \otimes \mathscr{C}_2$
$\mathscr{C}_4 \otimes B(P) \otimes S_1$
$\mathscr{X}_2$
$\mathscr{C}_1 \otimes \mathscr{C}_3$
$\mathscr{C}_4 \otimes B(P) \otimes S_2$
$\mathscr{X}_3$
$\mathscr{C}_2 \otimes \mathscr{C}_3$
$\mathscr{C}_4 \otimes B(P) \otimes S_2$
$S_1 = B(\frac{1}{2})$
$S_2 = B(\frac{1}{2})$
$S_3 = B(\frac{1}{2})$

Fig. A.3 A MBP transformed from a 3SAT-E3 formula $\phi = (\overline{X_1} \vee X_2) \wedge (\overline{X_1} \vee \overline{X_3}) \wedge (X_2 \vee \overline{X_3}) \wedge (X_1 \vee \overline{X_2} \vee X_3)$. The formula is satisfiable with an assignment $X_1 = 0, X_2 = 1, X_3 = 1$.

For simplicity, we use $B(p)$ to denote a binary random variable with probability $p \in (0,1)$ being TRUE, and $H(p)$ to denote the entropy of the binary random variable $B(p)$.

Theorem A.2.1. *For a constant $c > 1$, the problem of finding a max fan-in 3 Markov blanket polytree whose maximum likelihood is within $1/c$ times the optimum is NP-hard.*

Proof. The proof is based on polytrees with the minimum total conditional entropy, because of the equivalence between the maximum likelihood and the minimum total conditional entropy. Given a 3SAT-E3 formula ϕ, we first build a graph by using the following steps:

1. **Clause vertex** - for every clause C_j, build a clause vertex V_c^j that is a binary random variable $\mathscr{C}_j = B(p)$.

2. **Variable vertex** - for every variable X_i in the formula, build a variable vertex V_x^i that is a 3-tuple. The first element in the 3-tuple is a binary random variable $\mathscr{X}_i = B(\frac{1}{2})$. The second element is the XOR of two binary random variables $\mathscr{C}_j \otimes \mathscr{C}_k$, both of which contain the variable X_i with the same polarity (i.e., $\mathscr{C}_j$ and $\mathscr{C}_k$ contain either X_i or $\overline{X_i}$). The third element is the XOR of three binary random variables $\mathscr{C}_l \otimes B(p) \otimes S_i$, where C_l contains the variable X_i with the opposite polarity of C_j and C_k, the variable $S_i = B(\frac{1}{2})$.

3. **Auxiliary vertices** V_A, V_B, V_C - for constants $k > 0$ and $0 < q < \frac{1}{2}$, make three auxiliary vertices, each of which is a joint probability distribution of k *i.i.d.* binary random variables $B(q)$.

4. **Auxiliary target vertex** - build an auxiliary vertex V_t that is a $(n+1)$-tuple, where n is the number of variables in the formula ϕ. The first element in V_t is the joint probability distribution of k *i.i.d.* XOR $A_i \otimes B_i \otimes C_i$ for $i \in [1,k]$. Each of the remaining n elements is a uniform binary random variable $\mathscr{X}_i = B(\frac{1}{2})$ that is the same as the first element in the variable vertex V_x^i for $i \in [1,n]$.

5. **Auxiliary vertices** S_i - for each variable vertex V_x^i, create an auxiliary vertex $S_i = B(\frac{1}{2})$.

6. **Edges** - fully connect the four auxiliary vertices V_A, V_B, V_C and V_t; connect V_t to all the variable vertices; connect a variable vertex to the three clause vertices, in which it appears; connect each auxiliary vertex S_i to its corresponding variable vertex V_x^i.

It is clear that the graph G constructed from a given formula ϕ by using the above steps contains cycles, so is not a polytree. In addition, some vertices may violate the max fan-in 3 constraint if all edges incident to it are directed towards it. To obtain a polytree, in particular, a max fan-in 3 Markov blanket polytree, some edges need to be removed to get rid of the cycles and reduce some vertices' fan-in. We argue in the following that under certain parameter settings, if the learning objective is to find the optimal ML MBPT model, then the choices of edge removal and orientations are deterministic.

Due to the way this graph is constructed, an edge removal can only affect the entropy of vertices incident to it, so it is sufficient to calculate the entropy changes locally. We first determine the edge removals and orientations among V_t, V_A, V_B, V_C. It is clear that the edges among the three auxiliary vertices do not contribute to reducing the total entropy, so they should be removed. The entropy of these vertices are as the following:

$$H(V_A) = H(V_B) = H(V_C) = kH(q),$$
$$H(V_t) = kH(q^3 + 3q(1-q)^2) + n.$$

The XOR implies that knowing any three from $\{A_i, B_i, C_i, A_i \otimes B_i \otimes C_i\}$ can determine the fourth value. Hence, the vertex that is chosen as the common child of the other three vertices must reduce most of the total entropy. If V_t is the child, then the total entropy is reduced by $kH(q^3 + 3q(1-q)^2)$. If V_A (or V_B or V_C) is the child, then the total entropy is reduced by $kH(q)$. In addition, since the fan-in of V_t does not reach the upper limit, it allows an extra incoming edge from any other vertices. This can reduce its entropy by at most $H(\mathscr{X}_i) = 1$ if the edge is from a variable vertex $\mathscr{X}_i$. Hence, the total entropy reduction is $kH(q)+1$. The setting of $0<q<\frac{1}{2}$ entails that $q<q^3+3q(1-q)^2<\frac{1}{2}$. So for sufficiently large k, we have $k(H(q^3+3q(1-q)^2)-H(q))>1$, which implies that the optimal choice is to set V_t as the common child of V_A, V_B, V_C. Consequently, the rest of the edges that incident to V_t must be directed away from V_t.

Each of the variable vertex V_x^i can allow three incoming edges from the three clause vertices, the target vertex and the auxiliary vertex S_i. The choice again depends on which three vertices give the most entropy reduction. Knowing the values of the elements in the target vertex V_t, the entropy of each variable vertex V_x^i is reduced by 1, which is the maximum possible reduction. In addition, the element $\mathscr{X}_i$ is independent of the other elements in V_x^i, so the edge from V_t to each of V_x^i is always kept in the graph. Take V_x^1 as an example. Let $H(p)=\frac{1}{2}$ and $H(2p(1-p))=1-\delta$. The other vertices reduce the entropy of V_x^1 as the following:

- V_c^1: $H(\frac{1}{2})-H(\frac{1}{2})=0$.
- V_c^2 or V_c^3: $H(2p(1-p))-H(p)=\frac{1}{2}-\delta$.
- S_1: $H(\frac{1}{2})-H(2p(1-p))=\delta$.
- S_1, V_c^1: $H(\frac{1}{2})-H(p)=1-\frac{1}{2}=\frac{1}{2}$.
- S_1, V_c^2 or V_c^3: $\delta+\frac{1}{2}-\delta=\frac{1}{2}$.
- V_c^1, V_c^2 or V_c^3: $\frac{1}{2}-\delta$.
- V_c^2, V_c^3: $H(2p(1-p))=1-\delta$.

The most entropy reduction is among $1-\delta$ and $\frac{1}{2}$. However, the polytree constraint stops V_c^2, V_c^3 being parents of V_x^1 at the same time. So the best choice is S_1 with one of V_c^1, V_c^2, V_c^3.

After all redundant edges are removed, the remaining graph is a Markov blanket polytree with respect to the target vertex V_t. If the the graph has an edge from the clause vertex

V_c^j to a variable vertex V_x^i, then the clause C_i in the formula ϕ is satisfied by the variable X_i. If the formula ϕ is satisfiable, then the entropy reduction on all the variable vertices is $m(\frac{1}{2}-\delta)+n\delta$. If only $1-\varepsilon$ fraction of the total m clauses can be satisfied, then the total entropy reduction is $m(1-\varepsilon)(\frac{1}{2}-\delta)+n\delta$. Suppose the optimal total entropy is H^*, then if at most $(1-\varepsilon)m$ clauses can be satisfied, then the lowest total entropy is $H^*+\varepsilon m(\frac{1}{2}-\delta)\geq cH^*$, where c depends on ε but is a constant relative to size of the input formula ϕ. Therefore, the ML MBPT problem cannot be approximated to within a constant factor of the optimal value.

□

Appendix B

Optimal Moral Graph Triangulation

The preprint of this work is available at [50]. This work was rejected for publication because the claims can be proved in a much easier argument. For the completeness of this thesis, the original proofs are included in the appendix.

Moral graphs were introduced in the 1980s as an intermediate step when transforming a Bayesian network to a junction tree, on which exact belief propagation can be efficiently done. The moral graph of a Bayesian network can be trivially obtained by connecting non-adjacent parents for each node in the Bayesian network and dropping the direction of each edge. Perhaps because the moralization process looks simple, there has been little attention on the properties of moral graphs and their impact in belief propagation on Bayesian networks. This paper addresses the mistaken claim that it has been previously proved that optimal moral graph triangulation with the constraints of minimum fill-in, treewidth or total states is NP-complete. The problems are in fact NP-complete, but they have not previously been proved. We now prove these.

B.1 Introduction

One way to conduct an exact inference on a given Bayesian network is to transform it to a tree-like structure, called *junction tree*, and conduct inference on a junction tree instead. The process of obtaining a junction tree from a Bayesian network consists of moralization, triangulation and tree decomposition. *Moralization* was introduced by Lauritzen and Spiegelhalter [47] as connecting non-adjacent parents for each node in the Bayesian network and dropping all the directions. *Tree decomposition* maps a graph $G = (V,E)$ to a tree T, in which each tree node t is a subset V_t of vertices in V and satisfying the following three conditions: (1) $\cup_{t \in T} V_t = V$; (2) for each edge $e \in E$ there exists a tree node $t \in T$ s.t. $V(e) \subseteq t$; (3) if $V_{t_i} \cap V_{t_k} = I$ then $I \subseteq V_{t_j}$ for each t_j that appears on the path between t_i and t_k.

Any graph has a tree decomposition, but not all decomposed trees are junction trees. A junction tree is a tree decomposition s.t. each tree node is a complete subgraph. To ensure every DAG can be transformed into a junction tree, which represents a family of distributions that contains the distribution of the given Bayesian network, it is necessary to triangulate the DAG's moral graph. Here, *triangulation* finds a set of fill-in edges, whose addition makes a graph triangulated. When the marginal distribution of individual variables is of interest, the junction tree algorithm sums over the other variables in a tree node. Hence, the complexity of the junction tree algorithm is exponential in the size of a tree node.

Generally, a DAG may have more than one way of being triangulated. Then an optimal triangulation can be defined in terms of the following three constraints:

- minimum fill-in: deciding whether a graph can be triangulated by at most λ fill in edges [1];
- treewidth: deciding whether a graph has treewidth at most ω;

[1] Originally, the minimum fill-in problem for graphs was presented and proved by Yannakakis [98] as an optimization problem. But it can be revised to a decision problem, for which the original proof also works.

- total states: deciding whether a graph can be triangulated s.t. the total number of states is at most δ when summing over all maximal cliques in the triangulated graph.

The minimum fill-in problem is appealing because the treewidth usually increases exponentially in the number of fill-in edges. The total states problem incorporates both clique size and the number of states per variable, so is also essential to the complexity of the junction tree algorithm. The minimum fill-in and the treewidth problems for graphs were proved to be NP-complete by Yannakakis [98] and Arnborg et al. [8], respectively. Each proof stated a polynomial time reduction from a known NP-complete problem to optimally triangulating specially constructed graphs that are not moral. These reductions are sufficient to show the NP-completeness of the minimum fill-in and treewidth problems for graphs, but the difficulty of these problems do not automatically carry over to moral graphs. These works, however, were cited in [47] (Section 6 and discussion with Augustin Kong) during the discussion of triangulating moral graphs. So it gives the impression that the minimum fill-in and treewidth problems for moral graphs were proved to be NP-complete. Based on a similar reduction, Wen [95] presented proof of the NP-completeness of the total states problem for moral graphs. The proof is insufficient to support his claim, for the same reason above. Since then, all three works have been inaccurately cited as proving the NP-completeness of optimally triangulating moral graphs, e.g., [43, 46, 6, 28, 64, 48], etc.

This paper proves that the minimum fill-in, treewidth or total states problems for moral graphs are indeed NP-complete. It applies an additional step to each polynomial transformation to ensure the built graphs are moral after revision. Section B.2 introduces equivalent properties to graph morality and the necessary concepts for the proofs. Section B.3 demonstrates why the original constructions cannot produce moral graphs and presents a fix to each problem.

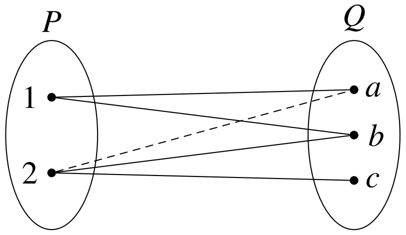


Fig. B.1 A bipartite non-chain graph with only the solid edges and a bipartite chain graph with all edges. The node 2 is a saturated node in the chain graph.

B.2 Preliminary

Denote a bipartite graph by $G = (P \sqcup Q, E)$, where $P \sqcup Q$ represents the disjoint union of two sets in vertices of G.

Definition B.2.1. *A bipartite graph $G = (P \sqcup Q, E)$ is **chain** if there is an ordering α : $\{1, \ldots, |P|\} \leftrightarrow P$ s.t. the neighbours of the vertices in P form a chain $N_G(\alpha(|P|)) \subseteq \cdots \subseteq N_G(\alpha(1))$.*

The definition is also well defined for the vertices in Q.

Definition B.2.2. *Let $G = (P \sqcup Q, E)$ be a bipartite graph. The **partition completion** of G is a function $C(\cdot)$ that makes each P and Q a clique.*

In particular, the partition completion $C_P(\cdot)$ restricted to P only makes P a clique.

Definition B.2.3. *A vertex in a bipartite graph is **saturated** if it is connected to every vertex in the other partition.*

Example B.2.1. *The bipartite graph in Fig. B.1 with only the solid edges is not a chain graph, because the neighbour sets of P's nodes do not form a chain. But the bipartite graph with all the edges is a chain graph, because $N(1) \subseteq N(2)$ w.r.t. the ordering $\alpha = \{2, 1\}$. In the chain graph, the node 2 is saturated, because it is adjacent to all nodes in Q.*

The following results have been proved in Chapter 3.

Corollary B.2.1. *If a graph has no simplicial node, then it is not moral.*

Corollary B.2.2. *If a graph is chordal, then it is moral.*

Theorem B.2.1. *Let G be a graph. The following are equivalent:*

1. *G is moral.*
2. *G is weakly recursively simplicial.*
3. *G has a perfect elimination kit.*

B.3 Optimal Moral Graph Triangulation

Yannakakis [98], Arnborg et al. [8] and Wen [95] proved a polynomial reduction from a NP-complete problem to the minimum fill-in, treewidth or total states problems for graphs, respectively. The NP-complete problems used in these proofs are the optimal linear arrangement (OLA), minimum cut linear arrangement (MCLA) and elimination degree sequence (EDS), respectively. These problems are listed below and can be found in [36, page. 200-201].

OPTIMAL LINEAR ARRANGEMENT

INSTANCE: Graph $G = (V,E)$, positive integer $k \leq |V|$.

QUESTION: Is there an ordering $\alpha : \{1,\ldots,|V|\} \leftrightarrow V$ s.t. $c(\alpha) = \sum_{uv\in E} |\alpha^{-1}(u) - \alpha^{-1}(v)| \leq k$?

MINIMUM CUT LINEAR ARRANGEMENT

INSTANCE: Graph $G = (V,E)$, positive integer $k \leq |V|$.

QUESTION: Is there an ordering $\alpha : \{1,\ldots,|V|\} \leftrightarrow V$ s.t. $\forall i \in [1,|V|], |\{uv \in E \mid \alpha^{-1}(u) \leq i < \alpha^{-1}(v)\}| \leq k$?

ELIMINATION DEGREE SEQUENCE

INSTANCE: Graph $G = (V,E)$, sequence $< d_1,\ldots,d_{|V|} >$ of non-negative integers not exceeding $|V| - 1$.

QUESTION: Is there an ordering $\alpha : \{1,\ldots,|V|\} \leftrightarrow V$ s.t. $\forall i \in [1,|V|]$, if $\alpha^{-1}(v) = i$ then there are exactly d_i vertices u s.t. $\alpha^{-1}(u) > i$ and $uv \in E$?

Each of the above problems asks if there exists an ordering α of a given graph s.t. a certain constraint is satisfied. By fixing a node $u \in V$ at an arbitrary place in an ordering α, each of the OLA, MCLA and EDS problems seeks for a restricted ordering from the subset $A = \{\alpha \mid \alpha(i) = u\}$ of orderings of G to satisfy its constraint. It is easy to verify that these restricted problems remain NP-complete. Because if there is an $O(|V|^k)$ time algorithm to answer the question within the restricted domain A, it takes $|V| \times O(|V|^k)$ time to answer the original question in the entire set of orderings.

The motivation for creating a bipartite graph is the relation between chain graphs and chordal graphs stated next.

Lemma B.3.1. *[98]. $C(G')$ is chordal if and only if G' is a bipartite chain graph.*

It follows from the lemma that triangulation of the graph $C(G')$ is equivalent to making G' a chain graph. To address the matter that the original transformed graphs by Yannakakis [98], Arnborg et al. [8] or Wen [95] are not moral, Lemma B.3.3 proves an equivalent relation between bipartite graphs and moral graphs. In addition, it gives insights on how to revise the transformations to get moral graphs, on which the optimal triangulation can be solved.

Lemma B.3.2. *If $G' = (P \sqcup Q, E')$ is a bipartite graph, then $C_p(G')$ is triangulated.*

Proof. Trivial. □

Lemma B.3.3. *Let $G' = (P \sqcup Q, E')$ be a bipartite graph. The graph $C(G')$ is moral if and only if $\exists u \in P \sqcup Q$ s.t. $\forall v \in N_{G'}(u)$, the node v is saturated.*

Proof. Without loss of generality, consider a node $u \in Q$. The neighbour set $N_{C(G')}(u) = N_{G'}(u) \cup \{Q \setminus u\}$. The partition completion implies that both $N_{G'}(u)$ and $\{Q \setminus u\}$ are cliques in $C(G')$.

For any node in $P \sqcup Q$, if it has at least one unsaturated neighbour in the bipartite graph G', then $\exists v \in N_{G'}(u)$ s.t. $vw \notin E'$ for some node $w \in \{Q \setminus u\}$. It then implies that $N_{C(G')}(u)$ is not a clique in $C(G')$, so u is not a simplicial node in $C(G')$. This is true for all nodes in $C(G')$, so the graph has no simplicial node and by Corollary B.2.1 it is not moral.

If all neighbours of u are saturated in the bipartite graph G', then the neighbour set $N_{C(G')}(u)$ is a clique. It follows that u is a simplicial node in $C(G')$. The subgraph $H = G - u - \{vw \mid v, w \in Q \wedge v, w \neq u\}$ is the same as the subgraph $C_p(G-u)$, in which $G-u$ is bipartite. By Lemma B.3.2, the subgraph H is triangulated. Therefore the graph $C(G')$ is moral by Corollary B.2.2 and Theorem B.2.1. □

B.3.1 Minimum Fill-in

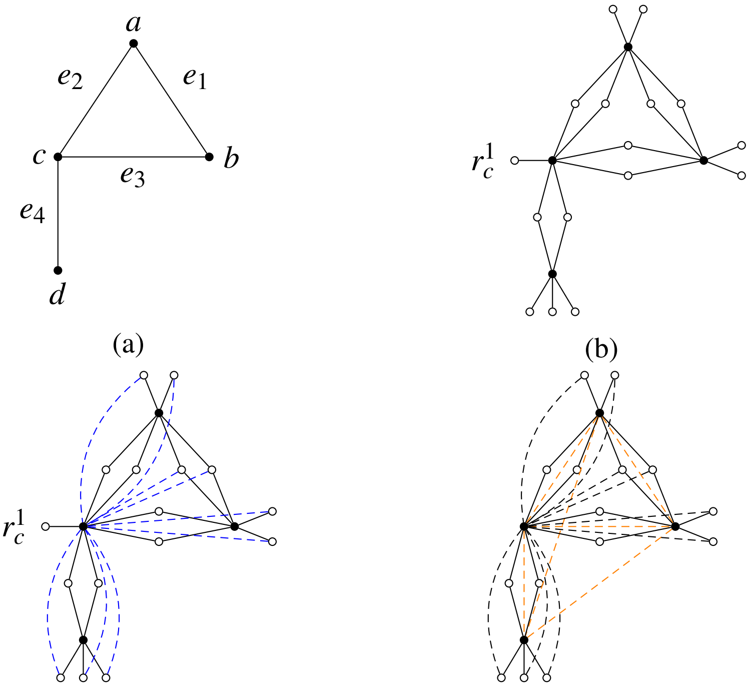


Fig. B.2 B.2a a graph G; B.2b the bipartite graph G' transformed from G by Y1-Y3; B.2c the bipartite graph $\hat{G}$ transformed from G' by saturating the node c using L4; B.2d the subgraph obtained from $C(\hat{G})$ by removing the simplicial node r_c^1 and its excess $\{vw \mid v, w \in Q \wedge v, w \neq r_c^1\}$.

Yannakakis [98] presented a polynomial transformation from an instance of the OLA problem into an instance of the minimum fill-in problem for graphs. The process first takes a graph $G = (V, E)$ (Fig. B.2a) and transforms it into a bipartite graph $G' = (P \sqcup Q, E')$ (Fig. B.2b) by the following steps:

Y1. $P = \{u \mid u \in V\}$,

Y2. $Q = \{e^j \mid e \in E \wedge j \in \{1,2\}\} \cup \{R(u) \mid u \in V\}$, where $R(u) = \{r_u^j \mid j \in \{1, \dots, |V| - d_G(u)\}\}$,

Y3. $E' = \{ue^j \mid u \in V(e) \wedge e \in E \wedge j \in \{1,2\}\} \cup \{uv \mid u \in P \wedge v \in R(u)\}$.

It then applies the partition completion on the bipartite graph G' to obtain the graph $C(G')$, on which the minimum fill-in problem is solved. As can be seen, each edge node $e_i^j \in Q$ is incident to exactly two nodes in P, so G' has no saturated node, unless $G = uv$. It follows from Lemma B.3.3 that $C(G')$ is not moral. The key to make $C(G')$ moral is to pick a node $u \in V$ and apply the additional step

L4. for a given node $u \in v$, let $\hat{E} = E' \cup S(u)$, where $S(u) = \{uv \notin E' \mid v \in Q\}$

that makes the corresponding node $u \in P$ saturated in $\hat{G}$ (Fig. B.2c).

Lemma B.3.4. *Let $\hat{G} = (P \sqcup Q, \hat{E})$ be the bipartite graph constructed from a graph $G = (V, E)$ by Y1-Y3 & L4 for a given node $u \in V$. Then $C(\hat{G})$ is moral.*

Proof. The proof follows from Lemma B.3.3. □

It is easy to see that the modified transformation can still be done in polynomial time, because the number of edges added by L4 is linear to the number of nodes in Q. It remains to show that a *Yes* instance of the restricted OLA problem is also a *Yes* instance of the minimum fill-in problem for moral graphs and vice versa. To do so, the next lemma calculates the difference between the cost of a graph G w.r.t. an ordering α and the number of fill-in edges that triangulates the corresponding moral graph $C(\hat{G})$ w.r.t. α. And it proves that the difference is constant for any restricted ordering α. Define the cost of an edge $e = uv \in E$ w.r.t. an ordering α to be $\delta(e,\alpha) = |\alpha^{-1}(u) - \alpha^{-1}(v)|$.

Lemma B.3.5. *Given a graph $G = (V,E)$ and a positive integer $k \leq |V|$, for any node $w \in V$ the minimum cost is k w.r.t. an ordering α of G s.t. $\alpha^{-1}(w) = 1$ if and only if the corresponding moral graph $C(\hat{G})$ can be triangulated by $\lambda = k + \frac{|V|(|V|-1)(|V|-2)}{2} - 2|E| + d_G(w)$ fill in edges w.r.t. α.*

Proof. Let $\hat{G}$ be the polynomial transformed graph from $G = (V,E)$ using Y1-Y3 & L4. For a given node $w \in V$, define $A = \{\alpha \mid \alpha(1) = w\}$ to be a subset of orderings of G. Then any ordering $\alpha \in A$ uniquely specifies a set $F_{\hat{G}}(\alpha)$ of fill-in edges to make $\hat{G}$ a chain by the following two steps:

(a) for each node $u \in Q$ calculate $\sigma(u) = \max\{i \mid u\alpha(i) \in E\}$,

(b) for any ordering α, define $F_{\hat{G}}(\alpha) = \{u\alpha(j) \notin \hat{E} \mid u \in Q \wedge j < \sigma(u)\}$.

It is easy to see that $F_{\hat{G}}(\alpha)$ is minimal because any edge deletion from it stops the neighbour sets of P's nodes in $\hat{G} + F_{\hat{G}}(\alpha)$ from forming a chain. Lemma B.3.1 implies $C(\hat{G}) + F_{\hat{G}}(\alpha)$ is triangulated, so $F_{\hat{G}}(\alpha)$ is a minimal triangulation of $C(\hat{G})$. It remains to show that every ordering $\alpha \in A$ yields a set of fill in edges with cardinality

$$f_{\hat{G}}(\alpha) = c(\alpha) + \frac{|V|(|V|-1)(|V|-2)}{2} - 2|E| + d_G(w), \tag{B.1}$$

where $c(\alpha)$ is the total cost of G w.r.t. α. Yannakakis [98] proved that for every ordering π (not necessarily in A), the number of fill-in edges

$$f_{G'}(\pi) = c(\pi) + \frac{|V|^2(|V|-1)}{2} - 2|E|. \tag{B.2}$$

The following is a brief explanation of Yannakakis [98]'s proof of equation (B.2). For every $v \in V$, each $x \in R(v)$ connects to $\pi^{-1}(v) - 1$ nodes in P, whose orderings are smaller than $\sigma(x)$. For any edge $e = uv \in E$, assume without loss of generality that $\pi^{-1}(u) < \pi^{-1}(v)$. Since each e^j in $\hat{G}$ is adjacent to the two end nodes of the edge e, the fill in edges in $F_{G'}(\alpha)$ connect e^j to $\pi^{-1}(v) - 2 = \pi^{-1}(u) + [\pi^{-1}(v) - \pi^{-1}(u)] - 2 = \pi^{-1}(u) + \delta(e, \pi) - 2$ nodes in P. Hence, $F_{G'}(\pi)$ contains $\pi^{-1}(v) + \pi^{-1}(u) + \delta(e, \pi) - 4$ edges incident to both e^1 and e^2. Therefore, the number of fill in edges w.r.t to π is calculated by

$$\begin{aligned}
f_{G'}(\pi) = &\sum_{v \in V} \sum_{x \in R(v)} [\pi^{-1}(v) - 1] + \sum_{e = uv \in E} [\pi^{-1}(v) + \pi^{-1}(u) + \delta(e, \pi) - 4] \\
= &\sum_{v \in V} [|V| - d_G(v)][\pi^{-1}(v) - 1] + \sum_{v \in V} d_G(v)\pi^{-1}(v) + \sum_{e \in E} \delta(e, \pi) - 4|E| \\
= &\sum_{v \in V} |V|[\pi^{-1}(v) - 1] + \sum_{v \in V} d_G(v) + c(\pi) - 4|E| \\
= &c(\pi) + \frac{|V|^2(|V| - 1)}{2} - 2|E|,
\end{aligned}$$

where by definition $\sum_{e \in E} \delta(e, \pi) = c(\pi)$ and the last equality obtains because $\sum_{v \in V} d_G(v) = 2|E|$ and $\sum_{v \in V} (\pi^{-1}(v) - 1) = |V|(|V| - 1)/2$. Note that the reason to make two edge nodes and the residual nodes is to cancel the term $d_G(v)$ during the derivation of $f_{G'}(\pi)$, so that the difference between $f_{G'}(\pi)$ and $c(\pi)$ is constant, regardless of the corresponding ordering.

The size difference between the two sets of edges $\hat{E}$ and E' is

$$\begin{aligned}
|S(w)| = &2(|E| - d_G(w)) + \sum_{u \in V}^{u \neq w} |R(u)| \\
= &2(|E| - d_G(w)) + \sum_{u \in V}^{u \neq w} (|V| - d_G(u)) \\
= &2(|E| - d_G(w)) + |V|(|V| - 1) - \sum_{u \in V} d_G(u) + d_G(w) \\
= &|V|(|V| - 1) - d_G(w).
\end{aligned} \tag{B.3}$$

Since equation (B.2) is true for every ordering, it certainly holds for orderings in A. It follows from $\alpha^{-1}(w) = 1$ that $S(w) \subseteq F_{G'}(\alpha)$. Hence, equation (B.1) is obtained by subtracting equation (B.3) from equation (B.2). If there exists an ordering α of G, w.r.t. which the minimum cost of G is k, then α produces a set of fill-in edges that triangulates the moral graph $C(\hat{G})$ with λ edges. Conversely, if the moral graph $C(\hat{G})$ can be triangulated w.r.t. an ordering α with λ fill-in edges, α indicates the minimum cost of the graph G is k. □

Theorem B.3.1. *The minimum fill-in problem for moral graphs is NP-complete.*

Proof. Since any set of fill-in edges that triangulates a moral graph can be verified for whether or not it contains at most λ edges in polynomial time, the minimum fill-in problem for moral graphs is in NP. Given a graph G can be polynomially transformed to the corresponding moral graph $C(\hat{G})$, Lemma B.3.5 proves the NP-hardness of the problem. □

B.3.2 Treewidth

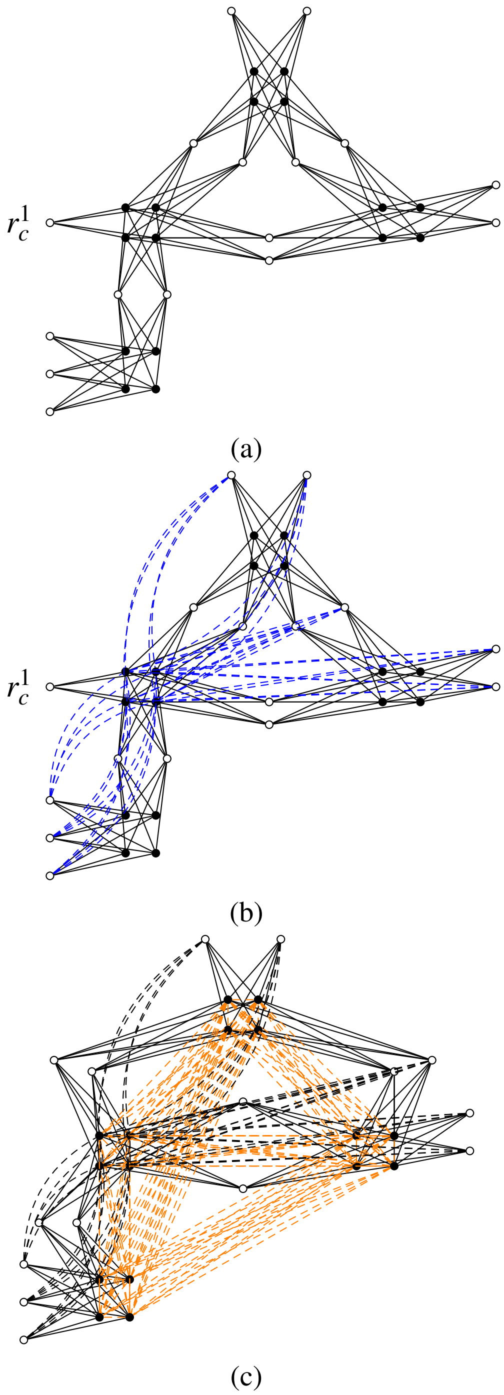
Arnborg et al. [8] reduced the MCLA problem to the decision problem of whether or not a graph has a bounded treewidth. Below are the steps (A1 through A3) of Arnborg et al. [8]'s polynomial transformation from a graph $G = (V, E)$ (Fig. B.2a) that is an instance of the MCLA problem to a bipartite graph $G' = (P \sqcup Q, E')$ (Fig. B.3a), w.r.t. which $C(G')$ is an instance of the treewidth problem for graphs.

A1. $P = \{u^i \mid u \in V \wedge i \in \{1, \ldots, \Delta(G) + 1\}\}$,

A2. $Q = \{e^j \mid e \in E \wedge j \in \{1, 2\}\} \cup \{R(u) \mid u \in V\}$, where $R(u) = \{r_u^j \mid j \in \{1, \ldots, \Delta(G) + 1 - d_G(u)\}\}$,

A3. $E' = \{u^i e^j \mid u^i \in P \wedge u \in V(e) \wedge e \in E \wedge j \in \{1, 2\}\} \cup \{u^i v \mid u \in P \wedge v \in R(u)\}$.

The transformation is similar to that of [98] but produces multiple copies for the nodes in G. The bipartite graph G' built via the above three steps has no saturated node (unless $G = uv$)



(a)

(b)

(c)

Fig. B.3 B.3a the bipartite graph G' transformed from G (Fig. B.2a) by A1-A3; B.3b the bipartite graph $\hat{G}$ transformed from G' by saturating the nodes $\{c^1, c^2, c^3, c^4\}$ using L*4; B.3c the subgraph obtained from $C(\hat{G})$ by removing the simplicial node r_c^1 and its excess $\{vw \mid v, w \in Q \wedge v, w \neq r_c^1\}$.

for the same reason discussed in the preceding subsection, so $C(G')$ is not moral by Lemma B.3.3. To make it moral, the following step

L*4. for a given node $u \in V$, let $\hat{E} = E' \cup \{S(u^j) \mid j \in [1, \Delta(G)+1]\}$, where $u^j \in P$ is the corresponding node to u and $S(u^j) = \{u^j v \notin E' \mid v \in Q\}$

is applied to each copy u^j of a given node $u \in V$ (Fig. B.3b) to make valid of any residual node of u being simplicial in $C(\hat{G})$.

Lemma B.3.6. *Let $\hat{G} = (P \sqcup Q, \hat{E})$ be the bipartite graph constructed from a graph $G = (V, E)$ by A1-A3 & L*4 for a given node $u \in V$. Then $C(\hat{G})$ is moral.*

Proof. The proof follows from Lemma B.3.3. □

Although this additional step is applied to all copies of u, the polynomial time is guaranteed by the bounded number $\Delta(G)+1$ of copies of u. Before proceeding, it is necessary to draw the connection between an ordering w.r.t. which a chain graph G' is defined and a perfect elimination ordering (PEO) of the corresponding triangulated graph $C(G')$.

Lemma B.3.7. *Let $G' = (P \sqcup Q, E')$ be a chain graph w.r.t. an ordering α of P and π_P be the reverse of α. Then for any ordering π_Q of Q, the elimination ordering $\{\pi_P, \pi_Q\}$ is perfect for the graph $C(G')$.*

Proof. The neighbour set $N_{C(G')}(\alpha(|P|)) = \{P \setminus \alpha(|P|)\} \cup N_{G'}(\alpha(|P|))$, where each of the two subsets is a clique because of the partition completion. G' is a chain graph implies that for each $i \in [1, |P|-1]$, $N_{G'}(\alpha(|P|)) \subseteq N_{G'}(\alpha(i))$. It follows that each $\alpha(i)$ is adjacent to all nodes in $N_{G'}(\alpha(|P|))$, so $\alpha(|P|)$ is simplicial in $C(G')$. By the same argument, the node $\alpha(|P|-1)$ is simplicial in the subgraph $C(G') - \alpha(|P|)$. Hence, the subgraph of $C(G')$ induced by P can be eliminated recursively according to π_P. The remaining part is a complete subgraph over Q. Hence, any ordering of Q appends to π_P forms a PEO of $C(G')$. □

It has been shown that A1-A3, L*4 and partition completion polynomially transform an instance of the restricted MCLA problem to an instance of the treewidth problem for moral graphs. Based on this transformation, the next lemma proves that a *Yes* answer to the restricted MCLA problem is also a *Yes* answer to the treewidth problem for moral graphs and vice versa. Define the *linear cut value* of G w.r.t. an ordering α as $\max_{1\leq i<|V|}|\{uv \in E \mid \alpha(u) \leq i < \alpha(v)\}|$.

Lemma B.3.8. *Given a graph $G = (V,E)$ and a positive integer $k \leq |V|$, for any node $v \in V$ the minimum linear cut value is k w.r.t. an ordering α of G s.t. $\alpha^{-1}(v) = |V|$ if and only if the treewidth of the corresponding moral graph $C(\hat{G})$ is $\omega = (\Delta(G)+1)\times(|V|+1)+k$.*

Proof. $\hat{G} = (P \sqcup Q, \hat{E})$ is the bipartite graph constructed from G using A1-A3 & L*4 for a given node $v \in V$. Let π_P be an ordering of P s.t. for any node $u_i = \alpha(i) \in V$, the corresponding node $u_i^j \in P$ has order

$$\pi_P^{-1}(u_i^j) = (\Delta(G)+1)\times i - j + 1, \tag{B.4}$$

where $j \in [1, \Delta(G)+1]$. Furthermore, let β be the reverse order of π_P. Then steps (a) and (b) in Lemma B.3.5 specify a set $F_{\hat{G}}(\beta)$ of fill-in edges w.r.t. β to triangulate $C(\hat{G})$, because the neighbour sets of P's nodes in $\hat{G}+F_{\hat{G}}(\beta)$ form the chain $N(\beta(|P|)) \subseteq \cdots \subseteq N(\beta(1))$. By Lemma B.3.7, for any ordering π_Q of Q, the ordering $\{\pi_P, \pi_Q\}$ is a PEO of the triangulated graph $T = C(\hat{G})+F_{\hat{G}}(\beta)$. For each $i \in [1, (\Delta(G)+1)\times|V|]$, the node $\pi_P(i)$ and its neighbours in the elimination graph T^{i-1} form a clique K^i. By going through $\{\pi_P, \pi_Q\}$, we get a list of cliques that include all maximal cliques in T and consequently the maximum clique. Since each v^j is a saturated node in P, all maximal cliques correspond to nodes in P only.

To calculate the size of each corresponding maximal clique, consider the node u_i^1 in the elimination graph w.r.t. π_P by removing from T the initial $(\Delta(G)+1)\times(i-1)$ nodes in P. The partition completion $C(\hat{G})$ connects u_i^1 to $\Delta(G)$ nodes corresponding to u_i and $\Delta(G)+1$

nodes corresponding to each of the remaining $|V| - i$ nodes in V. In addition, the edge set $\hat{E}$ and the fill-in edges $F_{\hat{G}}(\beta)$ connects u_i^1 to $\Delta(G) + 1 - d_G(u_l)$ residual nodes in each $R(u_l)$ for $\sigma(u_l) \geq \alpha^{-1}(u_i)$ and the two edge nodes for each edge $e \in E$ for $\sigma(e^j) \geq \alpha^{-1}(u_i)$. Let $e = xy \in E$ and assume without loss of generality that $\alpha^{-1}(x) < \alpha^{-1}(y)$. Then define $E_1^i = \{xy \in E \mid \alpha^{-1}(x) \leq i < \alpha^{-1}(y)\}$ and $E_2^i = \{xy \in E \mid \alpha^{-1}(y) \leq i\}$. The degree of u_i^j in the corresponding elimination graph can be calculated by

$$\begin{aligned} d(u_i^j) =& \Delta(G) + [(\Delta(G)+1) \times (|V| - i)] + \left[(\Delta(G)+1) \times i - \sum_{k=1}^{i} d_G(u_k)\right] + \quad 2|E_1^i| + 2|E_2^i| \\ =& (\Delta(G)+1) \times (|V|+1) - 1 + |E_1^i|, \end{aligned} \tag{B.5}$$

because $\sum_{k=1}^{i} d_G(u_k) = |E_1^i| + 2|E_2^i|$. Note that the reason for having $\Delta(G) + 1$ copies of each node in P and two edge nodes for each edge in G is to cancel the terms containing i and E_2^i in the final answer.

It is obvious that $\max\{E_1^i \mid i \in [1, |V|]\}$ is the linear cut value of G. If an ordering α gives the *Yes* answer to the restricted MCLA problem of a graph G, the treewidth of the corresponding moral graph $C(\hat{G})$ is equal to ω when triangulating it w.r.t. the ordering $\{\pi_P, \pi_Q\}$, where π_P is generated according to α by equation (B.4). Conversely, if the treewidth of the moral graph $C(\hat{G})$ is ω w.r.t. the ordering $\{\pi_P, \pi_Q\}$, the minimum linear cut value of G is k w.r.t. the ordering of V induced from π_P. □

Theorem B.3.2. *The treewidth problem for moral graphs is NP-complete.*

Proof. Let F be a set of fill-in edges to triangulate a moral graph G. It takes polynomial time to find the maximum clique in $G + F$ and test if it is at most k, so the treewidth problem is in NP. Hence, the theorem follows from Lemma B.3.8 and the polynomial transformation from G to $C(\hat{G})$. □

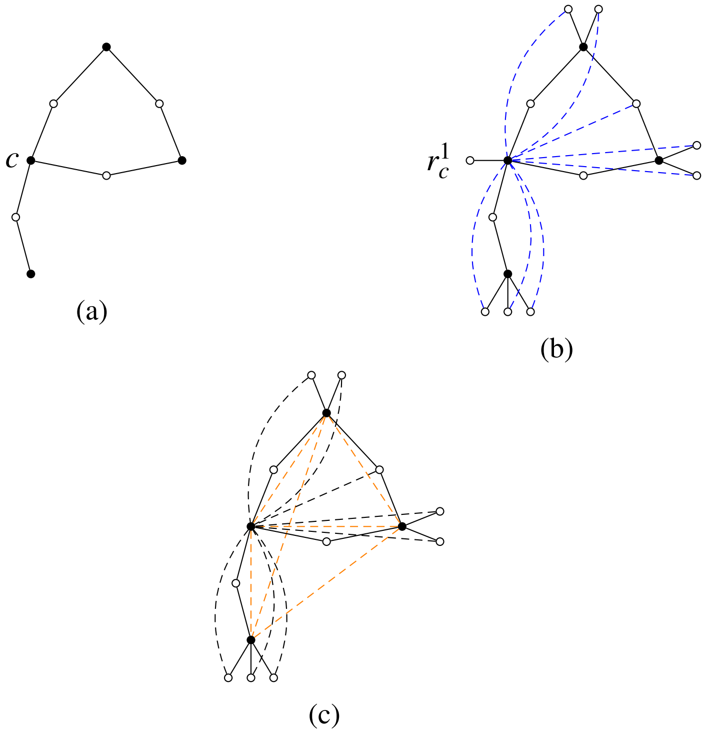


Fig. B.4 B.4a the bipartite graph G' transformed from G (Fig. B.2a) by W1-W3; B.4b the bipartite graph $\hat{G}$ transformed from G by W1 & L2-L4 for a given node c; B.4c the subgraph obtained from $C(\hat{G})$ by removing the simplicial node r_c^1 and its excess $\{vw \mid v,w \in Q \land v,w \neq r_c^1\}$.

B.3.3 Total States

By considering a constraint on the number of states per variable, Wen [95] introduced the total states problem for moral graphs and presented a proof for its difficulty by polynomially reducing the EDS problem to it. His transformation is simpler than the previous two cases, using one copy of the nodes in G and one edge node for each edge in G, without creating residual nodes (Fig. B.4a). The detailed transformation from a graph G (Fig. B.2a) that is an instance of the EDS problem to a bipartite graph G' is given by the following steps:

W1. $P = \{u \mid u \in V\}$,

W2. $Q = \{e^1 \mid e \in E\}$,

W3. $E' = \{ue^1 \mid u \in V(e) \land e \in E\}$.

Applying the partition completion on G' to transform it to $C(G')$, yields an instance of the total states problem for graphs but not for moral graphs by Lemma B.3.3. Therefore, Wen [95]'s reduction does not prove that the total states problem for moral graphs is NP-complete.

To prove the EDS problem is reducible to the total states problem for moral graphs in polynomial time, W2 and W3 need to be replaced by the following two steps

L2. $Q = \{e^1 \mid e \in E\} \cup \{R(u) \mid u \in V\}$, where $R(u) = \{r_u^j \mid j \in \{1, \ldots, \Delta(G) + 1 - d_G(u)\}\}$,

L3. $E' = \{ue^1 \mid u \in V(e) \land e \in E\} \cup \{uv \mid u \in P \land v \in R(u)\}$,

to create residual nodes before applying the same step L4 (as in Section B.3.1) for a given node to get the bipartite graph $\hat{G}$ (Fig. B.4b).

Lemma B.3.9. *Let $\hat{G} = (P \sqcup Q, \hat{E})$ be the bipartite graph constructed from a graph $G = (V, E)$ by W1 & L2-L4 for a given node $u \in V$. Then $C(\hat{G})$ is moral.*

Proof. The proof follows from Lemma B.3.3. □

For simplicity, define $N(i) = |\{u \mid u\alpha(i) \in E \land \alpha^{-1}(u) > i\}|$. Assume all variables considered are binary, the following lemma proves the hardness of the total states problem for moral graphs by polynomially reducing it from the restricted EDS problem.

Lemma B.3.10. *Given a graph $G = (V, E)$ and a sequence of non-negative integers $< d_1, \ldots, d_{|V|} >$ not exceeding $|V| - 1$, for any node $w \in V$ each value in the sequence satisfies $d_i = N(i)$ w.r.t. an ordering α of G s.t. $\alpha^{-1}(w) = |V|$ if and only if the corresponding moral graph $C(\hat{G})$ has the total number of states equal to $\delta = \sum_{i=1}^{|V|} 2^{\delta_i}$, where $\delta_i = |V| + \Delta(G) \times i + 1 + \sum_{j=1}^{i} [d_j - d_G(\alpha(j))]$.*

Proof. Let $\hat{G} = (P \sqcup Q, \hat{E})$ be the bipartite graph constructed from G using W1 & L2-L4. Let β be the reverse order of α. According to steps (a) and (b) in Lemma B.3.5, β specifies a set $F_{\hat{G}}(\beta)$ of fill-in edges to make $\hat{G}$ a chain graph. Hence, $T = C(\hat{G}) + F_{\hat{G}}(\beta)$ is a triangulated graph. By Lemma B.3.7, for any ordering α_Q of Q the ordering $\{\alpha, \alpha_Q\}$ is a PEO of T. As stated in the proof of Lemma B.3.8, the maximal cliques of T only correspond to nodes in P, so the rest of this proof does not consider the degrees of the nodes in Q.

For $i \in [1,|V|]$, the neighbour set $N_{T^{i-1}}(\alpha(i)) = N_P \cup N_Q$, where $N_P = \{u \in P \mid \alpha^{-1}(u) > i\}$ and $N_Q = \cup_{j=1}^{i}(N_{T^{j-1}}(\alpha(j)) \cap Q)$ because $\hat{G} + F_{\hat{G}}(\beta)$ is a chain graph. The cardinality of N_P can be easily calculated by $|V| - i + 1$. The set N_Q consists of the union of the neighbours of $\alpha(j)$ restricted to Q in the elimination graph T^{j-1} for all $j \in [1,i]$. The restricted neighbour set $N_{T^{j-1}}(\alpha(j)) \cap Q$ contains $N(j)$ edge nodes incident to $\alpha(j)$ because there is exactly one edge node for each edge in G, and $\Delta(G) + 1 - d_G(\alpha(j))$ residual nodes of $\alpha(j)$. So the total size $|N_Q| = \sum_{j=1}^{i} N(j) + \Delta(G) + 1 - d_G(\alpha(j))$. To show $N_P \cup N_Q$ forms a clique, it is easy to see that each of these subsets is a clique because of the partition completion. For any $u \in N_P$, the condition $\alpha^{-1}(u) > i$ implies $\beta^{-1}(u) < i$. The definition of $\sigma(\cdot)$ (Lemma B.3.5 step (a)) implies that $\sigma(v) \geq i$ for any $v \in N_Q$. It follows that $\sigma(v) > \beta^{-1}(u)$ for any $u \in N_P$ and $v \in N_Q$, so each node in N_P is connected to each node in N_Q by the edges in $F_{\hat{G}}(\beta)$. Therefore, the closed neighbourhood $N_{T^{i-1}}[\alpha(i)] = N_{T^{i-1}}(\alpha(i)) \cup \{\alpha(i)\}$ is a clique in T^{i-1}. It is in fact a maximal clique, because the set $R(\alpha(i))$ is not incident to $\alpha(i-1)$ for $i \in [2,|V|]$. The size k_i of the maximal clique corresponding to $\alpha(i)$ is thus

$$\begin{aligned} k_i &= |V| - i + 1 + \sum_{j=1}^{i} [N(j) + \Delta(G) + 1 - d_G(\alpha(j))] \\ &= |V| + \Delta(G) \times i + 1 + \sum_{j=1}^{i} [N(j) - d_G(\alpha(j))]. \end{aligned} \tag{B.6}$$

Since all variables are binary, the total number of states summing over all $|V|$ maximal cliques is $\sum_{i=1}^{|V|} 2^{k_i}$.

If there exists an ordering α of G, w.r.t. which the EDS answer is *Yes*, substituting $N(i)$ by d_i in equation (B.6) entails that the total states of $C(\hat{G})$ is δ w.r.t. α. That is, α is a *Yes* answer to the total states problem for the corresponding moral graph $C(\hat{G})$. Conversely, if the answer to the total states problem for a moral graph $C(\hat{G})$ is *Yes* w.r.t. an ordering $\{\alpha, \alpha_Q\}$, it follows from equation (B.6) that $d_i = N(i)$, so α gives a *Yes* answer to the EDS problem for the graph G. □

Theorem B.3.3. *The total states problem for moral graphs is NP-complete.*

Proof. Since the maximal cliques of a triangulated graph can be found in polynomial time, the total states of any triangulated graph can be verified in polynomial time to be greater than δ or not. Hence, the problem is in NP. Given the moral graph $C(\hat{G})$ can be transformed from a graph G by W1 & L2-L4 in polynomial time, Lemma B.3.10 proves the NP-hardness of the total states problem for moral graphs. □

B.4 Conclusion

Optimal moral graph triangulation plays an important role in determining the computational complexity of the junction tree algorithm for belief propagation on Bayesian networks. The minimum number of fill-in edges is closely related to the maximum clique size of the triangulated moral graph. The treewidth of a moral graph directly determines the efficiency of the junction tree algorithm when computing probabilities of unobserved variables by marginalizing out observed variables in the largest clique. The total number of states when summing over all maximal cliques in a triangulated moral graph takes into account the number of states per variable. The optimal moral graph triangulation with the objective of minimizing the number of fill-in edges, the maximum clique size or the total number of states has proved to be NP-complete in each case in this paper. Thus, we show that previous claims that optimal moral graph triangulation is NP-complete were, in fact, correct, by supplying the missing proofs.

Appendix C

More Plots on Markov Blanket Discovery

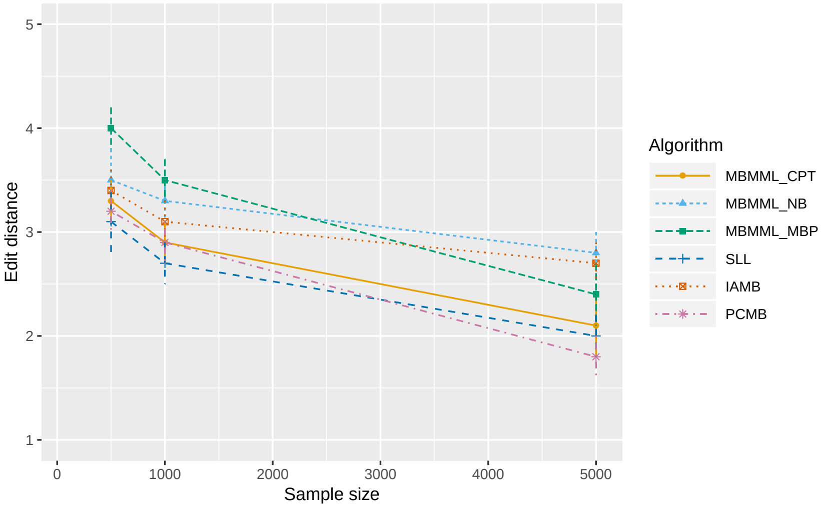


Fig. C.1 MB learners' edit distances (with 95% confidence intervals) vs. sample size on the INSURANCE network.

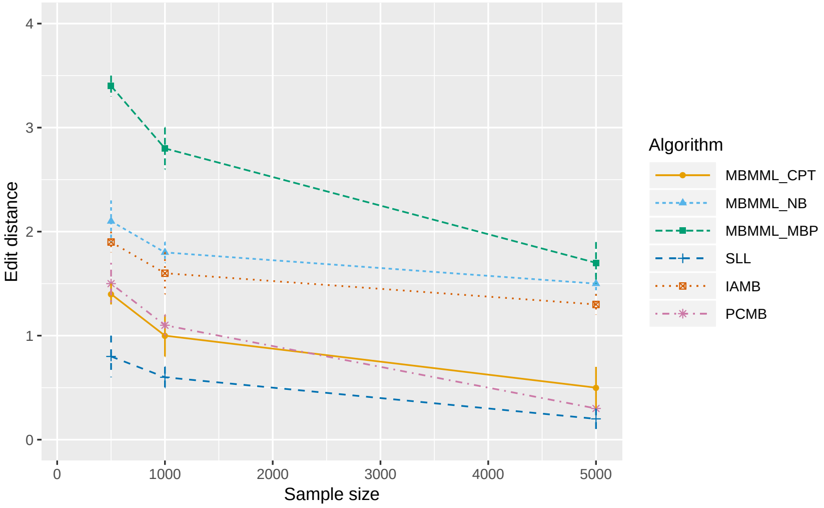


Fig. C.2 MB learners' edit distances (with 95% confidence intervals) vs. sample size on the ALARM network.

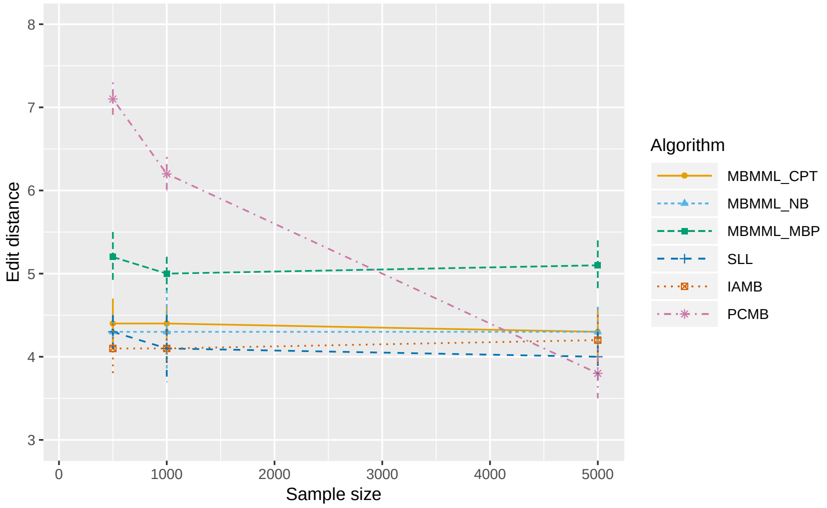


Fig. C.3 MB learners' edit distances (with 95% confidence intervals) vs. sample size on the HAILFINDER network.

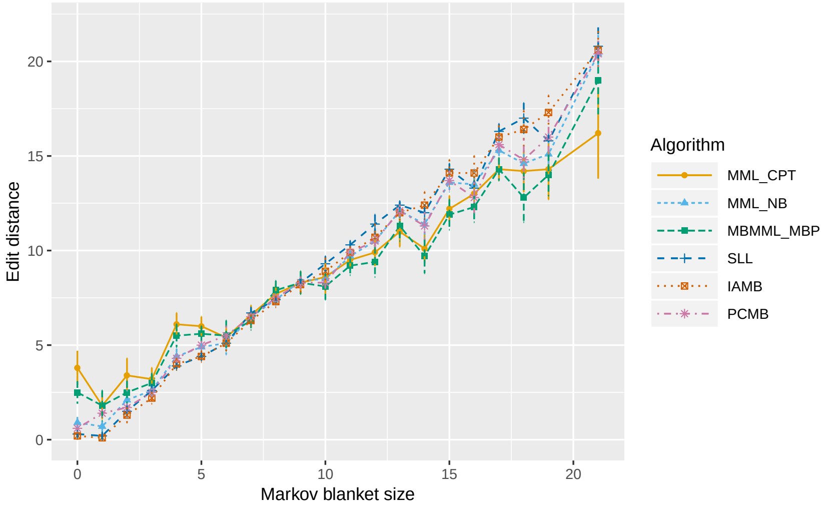


Fig. C.4 MB learners' edit distances (with 95% confidence intervals) vs. average MB size on artificial BNs (30-5-4-1) with 100 samples.

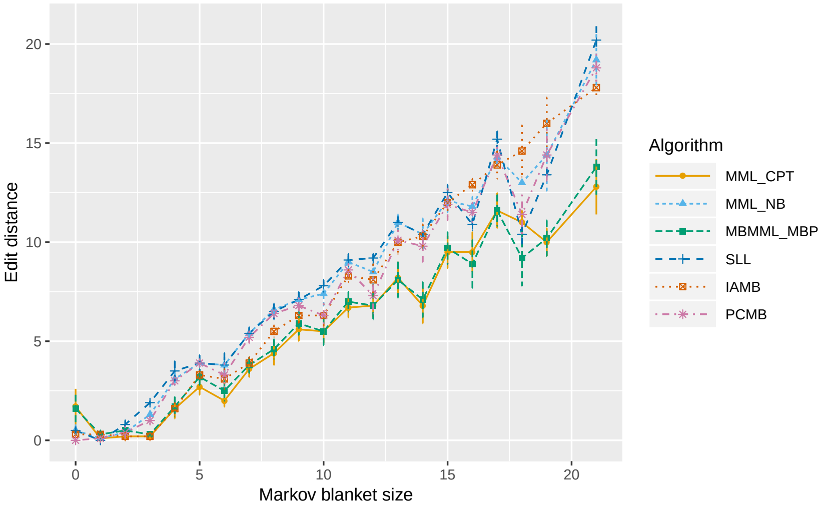


Fig. C.5 MB learners' edit distances (with 95% confidence intervals) vs. average MB size on artificial BNs (30-5-4-1) with 500 samples.

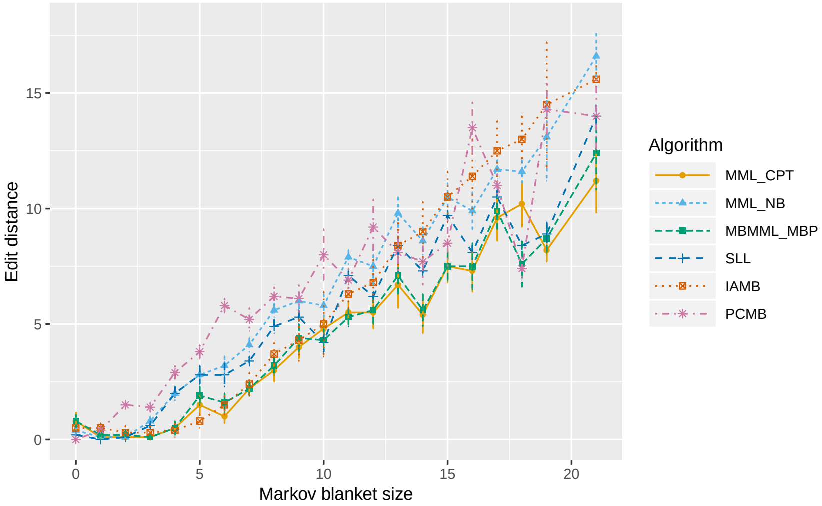


Fig. C.6 MB learners' edit distances (with 95% confidence intervals) vs. average MB size on artificial BNs (30-5-4-1) with 2000 samples.

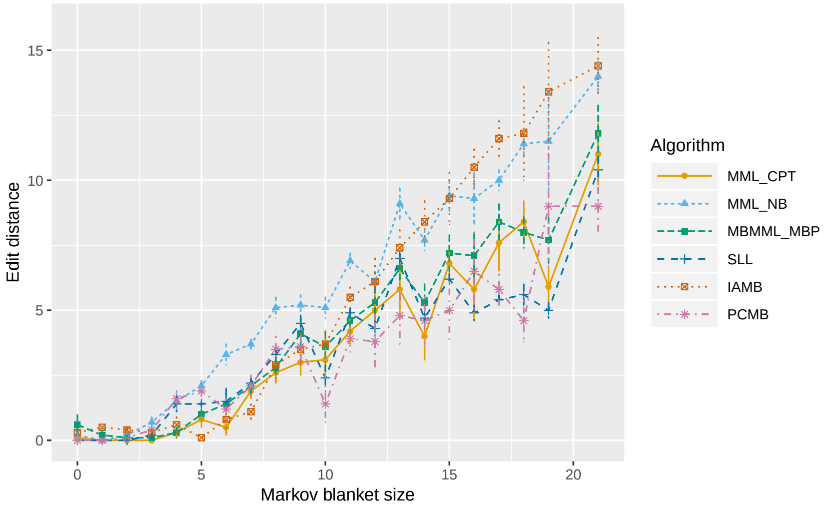


Fig. C.7 MB learners' edit distances (with 95% confidence intervals) vs. average MB size on artificial BNs (30-5-4-1) with 5000 samples.

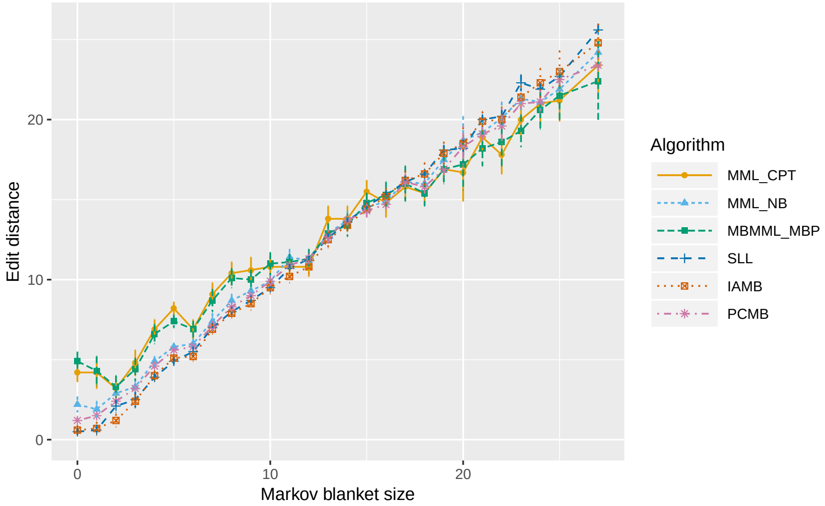


Fig. C.8 MB learners' edit distances (with 95% confidence intervals) vs. average MB size on artificial BNs (50-5-4-1) with 100 samples.

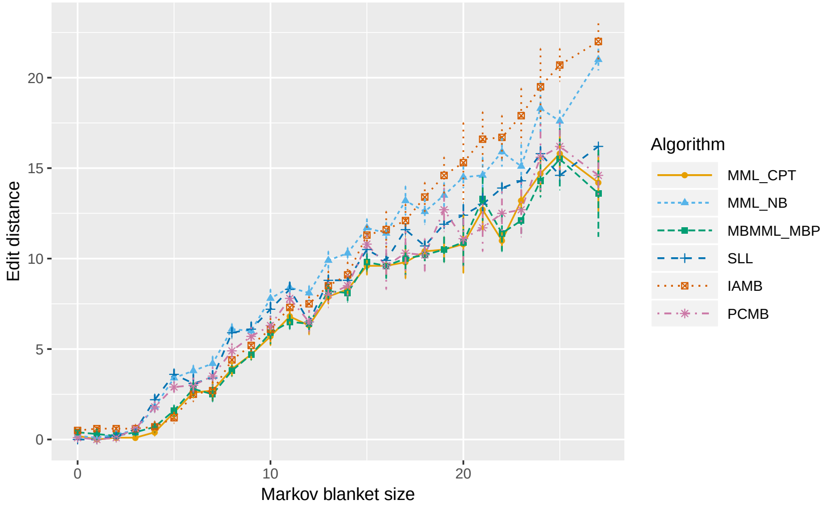


Fig. C.9 MB learners' edit distances (with 95% confidence intervals) vs. average MB size on artificial BNs (50-5-4-1) with 2000 samples.

Glossary

faithful A DAG and a joint probability distribution together satisfy the faithful condition if the DAG entails all and only the conditional independencies in the distribution. 7

junction tree A tree decomposition such that each tree node is a clique. 12

Markov blanket The Markov Blanket of a target variable is the smallest subset of variables conditioning on which renders the target independent of the remaining variables of a model. ix

Markov blanket consistency A set of Markov blankets over all variables must be consistent with at least one DAG. 4

Markov blanket polytree A Markov blanket polytree of a target variable is a polytree, in which every other variable is in the Markov blanket of the target. x

Markov equivalence class A set of all Markov equivalent DAGs. 2, 17

Minimum Message Length A way of balancing the complexity of a statistical model with the fit of the model to a given data set. ix

moral graph The moral graph of a given DAG is the undirected graph obtained by connecting all the non-adjacent parents of each node and dropping all the directions in the DAG. x

moralization The process of obtaining the moral graph of a given DAG. 5, 12

pattern A partially directed graph that uniquely represents a Markov equivalence class. It has the same edges as the DAGs in the equivalence class and directed all and only the edges common to all of the DAGs in the equivalence class. 2

regional structure A network structure over a subset of variables. 2

tree decomposition Tree decomposition maps a graph $G = (V,E)$ to a tree T , in which each tree node t is a subset V_t of vertices in V and satisfying the following three conditions: (1) $\cup_{t \in T} Vt = V$; (2) for each edge $e \in E$ there exists a tree node $t \in T$ such that $V(e) \in T$; (3) if $V_{t_i} \cap V_{t_k} = I$ then $I \subseteq V_{t_j}$ for each t_j that appears on the path between t_i and t_k. 13

treewidth The minimum size of the largest clique over all triangulated graphs of a given graph. 20

triangulation The process of making an undirected graph triangulated (or chordal). 5, 12

Acronyms

AIC Akaike Information Criteria. 19

BDe Bayesian Dirichlet equivalent. 14

BDeu Bayesian Dirichlet equivalent with uniform parameter prior. 18, 19

BIC Bayesian Information Criteria. 19

CaMML Causal MML. x, 4, 5

CPT Conditional Probability Table. 12, 70

DAG Directed Acyclic Graph. 2, 24

IAMB Incremental Association Markov Blanket. 9, 85

LASSO Least Absolute Shrinkage and Selection Operator. 12

LGL Local-to-Global. ix, x, 1–3, 20

MBPs Markov Blanket Polytrees. 70, 74

MDL Minimum Description Length. 19

MH Metropolis-Hasting. 18

MMHC Max-Min Hill-Climbing. 2

MML Minimum Message Length. ix, 4, 5, 12, 63

MMMB Max-Min Markov Blanket. 10

MMPC Max-Min Parents and Children. 10

NB Naive Bayes. 70, 72

PCMB Parents and Children based Markov Blanket. 10, 11, 86

SLL Score-based Local Learning. 2, 11, 86

TOMs Totally Ordered Models. 18, 112, 130

WRS Weakly Recursively Simplicial. 23, 29