UNIVERSITY OF CALIFORNIA
Los Angeles

Directed Acyclic Graphs:
Partitioned Hybrid Structure Learning and Bayesian Causal Bandits

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Statistics

by

Jireh Kyle Huang

2022

ABSTRACT OF THE DISSERTATION

Directed Acyclic Graphs:
Partitioned Hybrid Structure Learning and Bayesian Causal Bandits

by

Jireh Kyle Huang
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2022
Professor Qing Zhou, Chair

Bayesian networks are powerful models for probabilistic inference that compactly encode in their structures conditional independence and causal relationships amongst variables. In this dissertation, we propose novel methods for identifying and utilizing the structures of Bayesian networks.

We first develop a novel hybrid method for Bayesian network structure learning in the observational setting called partitioned hybrid greedy search (pHGS), composed of three distinct yet compatible new algorithms: Partitioned PC (pPC) accelerates skeleton learning via a divide-and-conquer strategy, p-value adjacency thresholding (PATH) effectively accomplishes parameter tuning with a single execution, and hybrid greedy initialization (HGI) maximally utilizes constraint-based information to obtain a high-scoring and well-performing initial graph for greedy search. We establish structure learning consistency of our algorithms in the large-sample limit, and empirically validate our methods individually and collectively through extensive numerical comparisons. The combined merits of pPC and PATH achieve significant computational reductions compared to the PC algorithm without

sacrificing the accuracy of estimated structures, and our generally applicable HGI strategy reliably improves the estimation structural accuracy of popular hybrid algorithms with negligible additional computational expense. Our empirical results demonstrate the competitive empirical performance of pHGS against many state-of-the-art structure learning algorithms.

Secondly, we turn our attention to exploiting Bayesian network structures in the causal bandit problem setting, a sequential decision-making framework where actions of interest correspond to interventions on variables in a system assumed to be governed by a causal model. The underlying causality may be exploited when investigating actions in the interest of optimizing the yield of the reward variable. Most existing approaches assume prior knowledge of the underlying causal graph, which is in practice restrictive and often unrealistic. In this paper, we develop a novel Bayesian framework for tackling causal bandit problems that does not rely on possession of the causal graph, but rather simultaneously learns the causal graph while exploiting causal inferences to optimize the reward. Our methods efficiently utilize joint inferences from interventional and observational data in a unified Bayesian model constructed with intervention calculus and causal graph learning. For the implementation of our proposed methodology in the discrete distributional setting, we derive an approximation of the sampling variance of the backdoor adjustment estimator. In the Gaussian setting, we characterize the interventional variance with intervention calculus and propose a simple graphical criterion to share information between arms. We validate our proposed methodology in an extensive empirical study, demonstrating compelling cumulative regret performance against state-of-the-art standard algorithms as well as optimistic implementations of their causal variants that assume strong prior knowledge of the causal structure.

The dissertation of Jireh Kyle Huang is approved.

Jingyi Jessica Li

Guido Montúfar

Ying Nian Wu

Qing Zhou, Committee Chair

University of California, Los Angeles

2022

Psalm 72:18-19

Romans 11:36

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Chapter 2 is based on Huang and Zhou (2022a). Chapter 3 is based on Huang and Zhou (2022b), which has been submitted for review. Work in both manuscripts was supported by NSF grant DMS-1952929. This work used computational and storage services associated with the Hoffman2 Shared Cluster provided by UCLA Institute for Digital Research and Education's Research Technology Group.

I am profoundly grateful for the supportive and encouraging mentorship of my advisor, Professor Qing Zhou. I have benefited incalculably from his adept and perceptive guidance, patient instruction and correction, passion for and excellence in research, generous availability and approachability, and earnest interest in the success of his students. It truly has been not only a distinct honor but also a sincere pleasure to study under his tutelage, and I could not have asked for a better advisor. I would also like to thank Professor Guido Montúfar, Professor Jingyi Jessica Li, and Professor Ying Nian Wu for generously sharing with me their time and expertise by serving on my committee. They have long commanded my respect and admiration, and I am honored to have them on my committee.

Additionally, my gratitude extends to many others in the UCLA Department of Statistics. The rigorous theoretical statistics courses taught by Dr. Nicolas Christou were instrumental in cementing my genuine interest in statistics. Teaching for Dr. Linda Zanontian gave me valuable experience in pedagogy and communicating statistical concepts with clarity. The department staff, especially Laurie Leyden, Chie Ryu, Enrique Reyes, Verghese Nallengara, and Glenda Jones, ensured that I had the opportunities and resources to succeed in this program. They supported my studies by helping me secure assistantships and housing, providing me with technology for research and teaching, and answering all my administrative questions. Finally, I could not imagine progressing this far in my academic career without the past eight years of encouraging and enriching camaraderie with my colleague and friend Stephen Smith.

My church family at Grace Community Church has been a fortress and lighthouse through every season for the deepest comfort to my soul and my greatest cause for rejoicing: “that Christ Jesus came into the world to save sinners, of whom I am the foremost” (1 Tim. 1:15b). I would not be the man I am today without the influence and discipleship of dear friends such as Joshua Lin, Christopher Law, and Stephen Smith. I continue to look to their examples of faithfulness, excellence, integrity, and devotion unto Christ.

I owe an immeasurable debt of love to my parents, who have charged me in every season to love the Lord my God with all my heart and with all my soul and with all my mind and and with all my strength (Mark 12:30). Thank you, Mom and Dad, for raising me in the discipline and admonition of the Lord, for the countless sacrifices you make to afford me every advantage and opportunity, and for your faithful prayers on my behalf.

Dearest Grace, thank you for standing steadfastly by my side in love, unwavering in affection and conviction, through each and every happiness, hardship, and happening of life. By your sacrificial love and selfless care you have purchased thousands of additional hours for my research, always with cheerfulness and without qualification. My whole being declares: “Many women have done excellently, but you surpass them all” (Prov. 31:29)! The unconditional nature of your love reminds me daily of the highest love which we enjoy in Christ: “In this is love, not that we have loved God but that he loved us and sent his Son to be the propitiation for our sins” (1 John 4:10).

Finally, to the Triune God: “Worthy are you, our Lord and God, to receive glory and honor and power, for you created all things, and by your will they existed and were created” (Rev. 4:11). You gave me breath and being, life and life eternal, my hands and my feet and my every faculty. You own my heart, soul, mind, and strength. “For from him and through him and to him are all things. To him be glory forever. Amen” (Rom. 11:36)!

VITA

2014-2017 B.S. Applied Mathematics, University of California, Los Angeles

2017-2022 Teaching Assistant, Department of Statistics, University of California, Los Angeles

PUBLICATIONS

Jireh Huang and Qing Zhou. Partitioned hybrid learning of Bayesian network structures. *Machine Learning*, 2022. https://doi.org/10.1007/s10994-022-06145-4.

CHAPTER 1

Introduction

1.1 Bayesian Networks

A *graph* $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a structure composed of a set of nodes $\mathbf{V} = \{1, \ldots, p\}$, and a set of edges $\mathbf{E}$. For a pair of distinct nodes $i, j \in \mathbf{V}$, we encode an undirected edge between i and j in $\mathcal{G}$ by an unordered connected pair $i - j \in \mathbf{E}$, and a directed edge from i to j in $\mathcal{G}$ by an ordered pair $i \to j \in \mathbf{E}$. A *directed acyclic graph* (DAG) has only directed edges and is oriented such that there are no directed cycles in $\mathcal{G}$. A DAG $\mathcal{G}$ defines the structure of a *Bayesian network* of a joint probability distribution P of variables $\mathbf{X}$ corresponding to $\mathbf{V}$ if P factorizes according to the structure of $\mathcal{G}$:

$$P(\mathbf{X}) = \prod_{i=1}^{p} P(X_i \mid \mathbf{Pa}_i^{\mathcal{G}}), \tag{1.1}$$

where $\mathbf{Pa}_i^{\mathcal{G}} = \{X_j : j \to i \in \mathbf{E}\}$ denotes the parents of X_i according to $\mathcal{G}$. In this work, we may refer to a node $i \in \mathbf{V}$ and its corresponding variable $X_i \in \mathbf{X}$ interchangeably. In a causal DAG, $i \to j$ asserts that i is a direct cause of j, whereas more generally, a DAG encodes in its structure a set of conditional independence statements between distinct variables according to the above factorization. For ease of notation, we let $\mathbf{X_k} = \{X_k \in \mathbf{X} : k \in \mathbf{k}\}$ for $\mathbf{k} \subseteq \mathbf{V}$.

Chapter 2 focuses on the setting in which P is a discrete probability distribution, although the presented strategies are not limited to such a domain. Each variable X_i probabilistically attains one of $r_i \geq 2$ states depending on the attained states of its parents $\mathbf{Pa}_i^{\mathcal{G}}$. The conditional probability distributions of the variables given each of their parent configurations

are multinomial distributions. In Chapter 3, we additionally consider the Gaussian setting where the conditional probability distributions are linear Gaussians.

Let $(X_i \perp\!\!\!\perp X_j \mid \mathbf{X_k})_P$ denote that X_i and X_j are independent given conditioning set $\mathbf{X_k} \subseteq \mathbf{X} \setminus \{X_i, X_j\}$ in P, and $(X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_{\mathcal{G}}$ that X_i and X_j are d-separated by $\mathbf{X_k}$ in $\mathcal{G}$. The factorization in (1.1) implies that $\mathcal{G}$ and P satisfy the *(global) Markov condition*: for disjoint sets of variables $\mathbf{A}, \mathbf{B}, \mathbf{C} \subseteq \mathbf{X}$,

$$(\mathbf{A} \perp\!\!\!\perp \mathbf{B} \,|\, \mathbf{C})_{\mathcal{G}} \Rightarrow (\mathbf{A} \perp\!\!\!\perp \mathbf{B} \,|\, \mathbf{C})_P. \tag{1.2}$$

1.1.1 Markov Equivalence

Multiple DAGs may encode the same set of d-separation statements and thus redundantly entail the same conditional independence statements. Such DAGs are said to be *Markov equivalent*. Formally, two DAGs $\mathcal{G}$ and $\mathcal{G}'$ are Markov equivalent if $(\mathbf{A} \perp\!\!\!\perp \mathbf{B} \,|\, \mathbf{C})_{\mathcal{G}} \Leftrightarrow (\mathbf{A} \perp\!\!\!\perp \mathbf{B} \,|\, \mathbf{C})_{\mathcal{G}'}$ for all mutually disjoint subsets $\mathbf{A}, \mathbf{B}, \mathbf{C} \subseteq \mathbf{X}$. We refer to Markov equivalent DAGs as simply *equivalent* and belonging to the same *equivalence class*. Given our distributional assumptions on P, equivalent DAGs are indistinguishable without background information or experimental data. As our interest lies in structure learning from observational data, the objective amounts to recovering the equivalence class of the underlying DAG.

The *skeleton* of a graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is the undirected graph obtained from replacing every adjacent (that is, connected) node pair in $\mathcal{G}$ with an undirected edge. A *v-structure* is a triplet $i, j, k \in \mathbf{V}$ oriented $i \rightarrow k \leftarrow j$ in $\mathcal{G}$ with i and j not adjacent. Let the *pattern* of $\mathcal{G}$ be the *partially directed acyclic graph* (PDAG) obtained by orienting all and only the v-structures of $\mathcal{G}$ in its skeleton, leaving all remaining edges undirected. The following theorem was adapted from Verma and Pearl (1991) to characterize equivalent DAGs.

Theorem 1 (Meek (1995))**.** *Two DAGs are equivalent if and only if they have the same patterns.*

Implied by Theorem 1 is the existence of *compelled* and *reversible* edges. An edge $i \to j$ in a DAG $\mathcal{G}$ is compelled if it exists oriented as stated in every DAG in the equivalence class of $\mathcal{G}$, whereas it is reversible if it is directed $j \to i$ in at least one DAG in the equivalence class of $\mathcal{G}$. Meek (1995) detailed a set of sound and complete rules known as *Meek's rules* (R1, R2, R3, and R4) that deterministically extend the pattern of a graph $\mathcal{G}$ to its *completed partially directed acyclic graph* (CPDAG), a PDAG featuring a directed edge for every compelled edge and an undirected edge for every reversible edge (Chickering, 2002a). As the unique representation of its equivalence class, the CPDAG is the structure of interest for structure learning methods in the observational setting.

1.1.2 Faithfulness

The global Markov property, as stated in (1.2), defines an avenue for inference regarding the conditional independence relationships in P according to information encoded in its Bayesian network structure $\mathcal{G}$. In order to recover $\mathcal{G}$ from data generated from and thus sample estimates of probability distribution P, we require the assumption of faithfulness to infer the structure of $\mathcal{G}$ from P.

Definition 1 (Faithfulness)**.** A distribution P and a DAG $\mathcal{G}$ are said to be *faithful* to each other if all and only the conditional independence relations true in P are entailed by the d-separation statements in $\mathcal{G}$, i.e.

$$(\mathbf{A} \perp\!\!\!\perp \mathbf{B} \,|\, \mathbf{C})_{\mathcal{G}} \Leftrightarrow (\mathbf{A} \perp\!\!\!\perp \mathbf{B} \,|\, \mathbf{C})_{P}.$$

Under faithfulness, we may say in such a case that $\mathbf{A}$ and $\mathbf{B}$ are separated by $\mathbf{C}$, regardless of whether we are referring to d-separation or conditional independence. If P is faithful to $\mathcal{G}$, then the existence of an edge between any distinct pair of nodes i and j can be necessarily and sufficiently determined by the nonexistence of a separation set of variables that render

i and j conditionally independent in P. In particular,

$$i, j \in \mathbf{V} \text{ are not connected in } \mathcal{G} \;\Leftrightarrow\; \exists \mathbf{k} \subseteq \mathbf{V} \setminus \{i, j\} \text{ such that } (X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P. \tag{1.3}$$

1.2 Structure Learning

The problem of recovering from data the structure of the true underlying Bayesian network that governs a domain of variables is notoriously challenging (Chickering et al., 2004). The space of Bayesian network structures grows super-exponentially with the number of variables, severely limiting exhaustive evaluation of all structures and motivating decades of work in developing efficient algorithms for structure learning (Robinson, 1977; Spirtes et al., 2000).

Generally, Bayesian network structure learning algorithms can be classified as one of the following three classes of algorithms. Constraint-based methods strategically test conditional independence relationships between pairs of variables, first determining the existence of edges before inferring orientations (Spirtes and Glymour, 1991; Meek, 1995). In the score-based approach, heuristics are designed to optimize some scoring criterion that evaluates the goodness-of-fit of a proposed structure to the available data (Heckerman et al., 1995; Chickering, 2002b; Russell and Norvig, 2009). Finally, hybrid methods combine the two strategies, optimizing a score over a reduced space of structures restricted through a constraint-based approach (Tsamardinos et al., 2006a; Gasse et al., 2014). We discuss score-based and hybrid structure learning in Section 2.3.1 during the exposition of our methodology, and we describe a number of popular structure learning algorithms in Section 2.4. In the what follows, we discuss relevant background on constraint-based structure learning.

1.2.1 The PC Algorithm

The well-known PC algorithm (Spirtes and Glymour, 1991), named after its authors, is widely considered the gold standard constraint-based structure learning method. The PC algorithm

first efficiently estimates a skeleton, reducing the criterion stated in (1.3) by leveraging sparsity. Let $\mathbf{N}_i^{\mathcal{G}} = \{X_j \in \mathbf{X} : i \text{ and } j \text{ are connected in } \mathcal{G}\}$ be the *neighbors*, or adjacencies, of node i in a graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$. If $\mathcal{G}$ is a DAG, the following is evident from the Markov condition:

$$\begin{aligned} i, j \in \mathbf{V} \text{ are not connected in } \mathcal{G} \;\Leftrightarrow\; & \exists \mathbf{X_k} \subseteq \mathbf{N}_i^{\mathcal{G}} \setminus \{X_j\} \text{ or } \exists \mathbf{X_k} \subseteq \mathbf{N}_j^{\mathcal{G}} \setminus \{X_i\} \\ & \text{such that } (X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P. \end{aligned} \tag{1.4}$$

For easy reference in our algorithm description, we detail an implementation of the skeleton estimation step of the PC algorithm known as PC-stable in Algorithm 1.1 (Colombo and Maathuis, 2014). Note that throughout the development of our methodology, in what we call the population versions of procedures, we assume possession of all conditional independence information in P denoted $\{\perp\!\!\!\perp_P\}$, thus having conditional independence oracles perfectly corresponding to d-separation. For inferring conditional independence from finite samples of discrete data $\mathcal{D}$ in the sample counterparts, we use the popular G^2 log-likelihood ratio

Algorithm 1.1 `PC-skeleton`$(\{\perp\!\!\!\perp_P\})$ (PC-stable; population version)

input: conditional independence information $\{\perp\!\!\!\perp_P\}$
output: undirected graph $\mathcal{G}$

form the complete undirected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ over nodes $\mathbf{V} = \{k : X_k \in \mathbf{X}\}$
initialize $\mathbf{S} = \emptyset$
initialize $l = 0$
repeat
    set $\mathcal{G}' = \mathcal{G}$ to fix adjacencies
    for all unordered node pairs $i - j$ adjacent in $\mathcal{G}'$ with $|\mathbf{N}_i^{\mathcal{G}'}| - 1$ or $|\mathbf{N}_j^{\mathcal{G}'}| - 1 \geq l$ **do**
        for all unique subsets $\mathbf{X_k} \subseteq \mathbf{N}_i^{\mathcal{G}'} \setminus \{X_j\}$ and $\mathbf{X_k} \subseteq \mathbf{N}_j^{\mathcal{G}'} \setminus \{X_i\}$ of size $|\mathbf{k}| = l$ **do**
            if $(X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P$ **then**
                store separation set $\mathbf{S}(i, j) = \mathbf{S}(j, i) = \mathbf{k}$
                disconnect i and j in $\mathcal{G}$ and continue to the next pair of nodes
            end if
        end for
    end for
    $l = l + 1$
until there is no node i with $|\mathbf{N}_i^{\mathcal{G}}| - 1 \geq l$, or $l > m$ for some user-specified m

test of independence for empirical estimation of conditional independence in P with some significance level threshold α, denoting the G^2 test statistic for testing $(X_i \perp\!\!\!\perp X_j \mid \mathbf{X_k})_P$ as $G^2_{ij|\mathbf{k}}$ (Spirtes et al., 2000). We briefly discuss the basic notation for and evaluation of the G^2 test in Section 1.2.3, referring details and examples to Neapolitan (2004, Section 10.3.1).

The key difference that distinguishes the PC-stable implementation from the original PC algorithm is that in line 5, the adjacencies are fixed in $\mathcal{G}'$ such that the considerations of adjacent node pairs within the outermost loop (lines 4-15) become order-independent and thus executable in parallel. We further discuss parallel execution of the PC algorithm in Section 2.2.1.2. Note that for every node i, $\mathbf{N}_i^{\mathcal{G}} \subseteq \mathbf{N}_i^{\mathcal{G}'}$ for $\mathbf{N}_i^{\mathcal{G}'}$ in any stage in Algorithm 1.1, preserving the general design of the original PC skeleton learning method by ensuring the exhaustive investigation of (1.4) and thus retaining its theoretical properties. Hereafter, when we discuss the PC algorithm, we refer to the PC-stable implementation.

After determining the skeleton of a DAG $\mathcal{G}$, knowledge about the conditional independence relationships between variables (namely, the accrued separation sets $\mathbf{S}$) can be used to detect the existence of v-structures and orient the skeleton to the pattern of $\mathcal{G}$. Recovery of the CPDAG of $\mathcal{G}$ can then be achieved by repeated application of Meek's rules (Meek, 1995). This process, which we refer to as `skel-to-cpdag` (Algorithm 1.2), is guaranteed to orient the skeleton of a DAG $\mathcal{G}$ to its CPDAG given accurate conditional independence information entailed by $\mathcal{G}$. We discuss the details regarding constraint-based edge orientation in the following section.

The complete PC(-stable) algorithm consists of skeleton estimation according to Algorithm 1.1 followed by edge orientation according to Algorithm 1.2, and is well-known to be sound and complete for CPDAG estimation (Kalisch and Bühlmann, 2007; Colombo and Maathuis, 2014).

1.2.2 Constraint-based Edge Orientation

We now review established constraint-based strategies for determining edge orientations, summarized in Algorithm 1.2. We begin by discussing the work of Verma and Pearl (1991) in determining edge orientations. Under faithfulness, knowledge about the conditional independence relationships between variables can be used to detect the existence of v-structures (defined in Section 1.1.1) as follows. A triplet of nodes (i, k, j) configured $i - k - j$ with i and j not adjacent, called an *unshielded triple*, is a v-structure if and only if

$$\exists \mathbf{k} \subseteq \mathbf{V} \setminus \{i, j, k\} \text{ such that } (X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P. \tag{1.5}$$

It is easy to see that any other directed configuration of $i - k - j$ requires k to separate i and j (Spirtes et al., 2000, Lemma 5.1.3). To investigate this criterion, the separation sets $\mathbf{S}$ are recorded throughout the estimation process (e.g., line 9 of Algorithm 1.1), defined as $\mathbf{S}(i, j) = \mathbf{S}(j, i) \subseteq \mathbf{V} \setminus \{i, j\}$ such that $(X_i \perp\!\!\!\perp X_j \mid \mathbf{X}_{\mathbf{S}(i,j)})_P$ for distinct nodes i and j. Assuming accurate separation sets, every v-structure in $\mathcal{G}$ can be recovered according to (1.5), guaranteeing extension of the skeleton of $\mathcal{G}$ to its pattern, to which Meek's rules can be straightforwardly applied to obtain its CPDAG (see Section 1.1.1).

Alternatively, considering the Markov condition, an unshielded triple (i, k, j) in $\mathcal{G}$ with adjacencies $\mathbf{N}_i^{\mathcal{G}}$ and $\mathbf{N}_j^{\mathcal{G}}$ can be correctly identified as a v-structure by investigating the criteria (1.5) limited to sets $\mathbf{k}$ such that $\mathbf{X_k} \subseteq \mathbf{N}_i^{\mathcal{G}}$ or sets $\mathbf{k}$ such that $\mathbf{X_k} \subseteq \mathbf{N}_j^{\mathcal{G}}$. Margaritis (2003) proposed the following criteria for identifying (i, k, j) as a v-structure:

$$(X_i \not\perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P \text{ for all } \mathbf{X_k} \subseteq \mathbf{N}_i^{\mathcal{G}} \text{ or for all } \mathbf{X_k} \subseteq \mathbf{N}_j^{\mathcal{G}} \text{ with } k \in \mathbf{k}, \tag{1.6}$$

investigating the smaller of $|\mathbf{N}_i^{\mathcal{G}}|$ and $|\mathbf{N}_j^{\mathcal{G}}|$ for efficiency. For constraint-based algorithms that do not record $\mathbf{S}$, we empirically prefer investigating (1.6) over (1.5) for both general well-performance and efficiency. As with (1.5), the correctness of (1.6) guarantees detection

Algorithm 1.2 `skel-to-cpdag(`$\mathcal{G}$`, S, {`$\perp\!\!\!\perp_P$`})`

input: undirected graph $\mathcal{G}$, and separation sets $\mathbf{S}$ or conditional independence information $\{\perp\!\!\!\perp_P\}$

output: CPDAG $\mathcal{G}$

initialize $\mathbf{U} = \emptyset$ to store v-structures
for all triplets of nodes configured $i - k - j$ in $\mathcal{G}$ where i and j are not adjacent **do**
    if $\mathbf{S}$ is supplied **then**
        if $k \notin \mathbf{S}(i, j)$ **then**
            store (i, k, j) in $\mathbf{U}$
        end if
    else
        let $q = \text{argmin}_{l \in \{i,j\}} |\mathbf{N}_l^{\mathcal{G}}|$
        if $(X_i \not\perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P$ for all $\mathbf{X_k} \subseteq \mathbf{N}_q^{\mathcal{G}}$ with $k \in \mathbf{k}$ **then**
            store (i, k, j) in $\mathbf{U}$
        end if
    end if
end for
for all triplets $(i, k, j) \in \mathbf{U}$ **do**
    orient $i \to k$ and $j \to k$ in $\mathcal{G}$
end for
repeatedly apply Meek's rules R1-4 to $\mathcal{G}$ until no rule is applicable

of all and only the v-structures in a DAG $\mathcal{G}$ faithful to P given its skeleton and conditional independence oracles.

1.2.3 Discrete Conditional Independence Testing

We develop here the notation and evaluation of the popular G^2 log-likelihood ratio test of independence for empirical estimation of conditional independence in discrete probability distribution P (Spirtes et al., 2000). For further details and examples, we refer to Neapolitan (2004) 10.3.1. Let $n[x_i]$ denote the number of counts for which random variable $X_i = x_i$ in n data samples $\mathcal{D}$, with the definition extending similarly to $n[x_i, x_j]$ and so on. For $\mathbf{k} \subseteq \mathbf{V}$, let $n[x_i, x_j, \mathbf{x_k}]$ denote the number of counts for which $X_i = x_i$, $X_j = x_j$, and $\mathbf{X_k}$ attains one of its $\prod_{k \in \mathbf{k}} r_k$ state configurations $\mathbf{x_k}$ in $\mathcal{D}$. Then the G^2 test statistic for testing

$H_0 : (X_i \perp\!\!\!\perp X_j \,|\, X_k)_P$ is calculated as

$$G^2_{ij|\mathbf{k}} = 2 \sum_{x_i, x_j, \mathbf{x_k}} n[x_i, x_j, \mathbf{x_k}] \log \left(\frac{n[x_i, x_j, \mathbf{x_k}] \cdot n[\mathbf{x_k}]}{n[x_i, x_j] \cdot n[x_j, \mathbf{x_k}]} \right). \tag{1.7}$$

The equation can be applied to test marginal independence with $\mathbf{k} = \emptyset$. Under H_0, the G^2 statistic is asymptotically χ^2_f distributed with

$$f = (r_i - 1)(r_j - 1) \prod_{k \in \mathbf{k}} r_k$$

degrees of freedom. Then for a chosen significance level α,

$$\Pr(\chi^2_f > G^2_{ij|\mathbf{k}}) \leq \alpha \;\Rightarrow\; \text{reject } H_0\text{: } (X_i \not\perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P.$$

Note that the PC algorithm (Algorithm 1.1) operates in a somewhat backwards fashion where the initialized complete graph on $\mathbf{V}$ assumes all distinct pairs of variables dependent rather than independent. A node pair i and j is then disconnected if

$$\Pr(\chi^2_f > G^2_{ij|\mathbf{k}}) > \alpha \;\Rightarrow\; \text{retain } H_0\text{: } (X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P$$

for some considered conditioning set $\mathbf{X_k}$.

Closely related to the G^2 test of independence and relevant to our work in Chapter 2 is the mutual information $I(X_i, X_j)$, which serves as a similarity measure between discrete random variables X_i and X_j. It may be interpreted as the Kullback-Leibler divergence between the joint probability distribution and the product of the marginals, and it is empirically calculated for n data observations as

$$\hat{I}(X_i, X_j) = \sum_{x_i, x_j} \frac{n[x_i, x_j]}{n} \log \left(\frac{n[x_i, x_j]/n}{n[x_i]/n \cdot n[x_j]/n} \right)$$

where $n[x_i, x_j]$ is the counts of the instances of the n observations that satisfy $X_i = x_i$ and $X_j = x_j$, with corresponding definitions for $n[x_i]$ and $n[x_j]$. It is easy to see that $G^2_{ij} = 2n \cdot \hat{I}(X_i, X_j)$.

1.3 Overview

The remainder of this dissertation is arranged as follows. In Chapter 2, we approach the problem of efficiently and accurately identifying large Bayesian network structures from observational data. We develop a sound and complete partitioned skeleton estimation strategy, a thresholding algorithm which efficiently accomplishes the task of parameter tuning both theoretically and empirically, and a hybrid initialization strategy for hybrid algorithms that effectively combines the merits of constraint-based and score-based structure learning. While individually and independently compelling, our contributions are additionally highly compatible, culminating in the form of a well-performing hybrid algorithm with desirable theoretical properties. In Chapter 3, we direct our attention to modeling and exploiting Bayesian network structures in the causal bandit problem setting. We develop a unified Bayesian hierarchical model for modeling the expected reward of deterministic atomic interventions that efficiently utilizes an ensemble of jointly interventional and observational data. Our proposed framework is able to effectively inform experimental investigations with observational data, accelerating identification of the best action. Finally, we summarize our work and discuss potential directions for future research in Chapter 4.

CHAPTER 2

Partitioned Hybrid Learning of Bayesian Network Structures

2.1 Motivation and Overview

The PC algorithm, named after its authors, is often considered state-of-the-art amongst constraint-based methods for Bayesian network structure learning because of its polynomial complexity for sparse graphs and attractive theoretical properties (Spirtes and Glymour, 1991; Kalisch and Bühlmann, 2007). Even with its favorable scaling, PC can quickly become unwieldy for large networks, motivating various developments to structure learning speed. Several works have contributed to accelerating its execution with various parallelization strategies, resulting in speed-ups ranging from up to ten times to over three orders of magnitude (Kalisch et al., 2012; Le et al., 2016; Madsen et al., 2017; Scutari, 2017; Zarebavani et al., 2020). However, these improvements are primarily feats of distributed processing implementation and are limited by the availability of required hardware. Gu and Zhou (2020) proposed a hybrid framework for partitioned estimation of Bayesian networks called partition, estimation, and fusion (PEF) in the interest of distributing learning by adopting a divide-and-conquer strategy. Unfortunately, its application to the PC algorithm does not in general retain the completeness of the PC algorithm and is limited in its capacity for parallel processing. Finally, none of these contributions tackle the practical problem that the performance of constraint-based algorithms can vary substantially with certain tuning parameters, potentially requiring multiple algorithm executions.

Prominent hybrid methods leverage the efficiency of constraint-based strategies to considerably reduce the space of Bayesian network models but sacrifice the asymptotic guarantees of constraint-based edge orientation for the generally superior empirical structural accuracy of restricted greedy search (Tsamardinos et al., 2006a). This is characteristic of members of what we call the generalized sparse candidate (GSC) framework, named after the sparse candidate algorithm (Friedman et al., 1999), in which a greedy search in the DAG space is executed from an empty graph restricted to a sparse set of candidate edges obtained through a constraint-based strategy. Hybrid algorithms belonging to GSC include max-min hill-climbing (MMHC) and hybrid hybrid parents and children (H2PC), which, despite being popular and widely regarded as well-performing, are well-known to be lacking in asymptotic guarantees (Tsamardinos et al., 2006a; Gasse et al., 2014). While the adaptively restricted greedy equivalence search (ARGES) stands out as a hybrid framework with established consistency (Nandy et al., 2018a), our simulations suggest that ARGES can likewise empirically benefit from the developments in our work. In particular, both GSC and ARGES initialize their respective greedy searches with an empty graph and, to our knowledge, no principled and well-performing initialization strategy without assuming expert knowledge has been proposed.

We propose an answer to these challenges by the development of the partitioned hybrid greedy search (pHGS) algorithm, a hybrid structure learning algorithm that can be considered the composition of three independent contributions to the computational efficiency, theoretical guarantees, and empirical performance of Bayesian network structure learning. In particular, pHGS accomplishes the following:

1. Restricts the search space with our proposed partitioned PC (pPC) algorithm that improves on the efficiency of the PC algorithm while retaining its soundness and completeness and capacity for parallel processing;

2. Mitigates the need for parameter tuning by automatically selecting the sparsity-controlling

threshold of conditional independence tests with our p-value adjacency thresholding (PATH) algorithm that extends the accessibility of constraint-based consistency;

3. Initializes the restricted greedy search with our hybrid greedy initialization (HGI) algorithm that elevates the asymptotic guarantees of existing hybrid algorithms such as members of the GSC framework to that of sound and complete constraint-based methods while improving empirical performance.

The novel components of pHGS are organized in the remainder of this chapter as follows. In Section 2.2, we develop the pPC algorithm which employs a partitioned estimation strategy to reduce the number of statistical tests required for the exhaustive conditional independence investigation in PC-like CPDAG learning. We additionally detail the PATH thresholding algorithm, which efficiently generates and selects from a set of CPDAG estimates with varying sparsity from a single execution of pPC (or PC) and extends the accessibility of classical asymptotic consistency results to more flexible parameter specification. We begin Section 2.3 with a brief review of score-based structure learning before developing HGI, a greedy initialization strategy which endears constraint-based edge orientation to the empirical setting with desirable theoretical guarantees.

We empirically validate pPC, PATH, and HGI in Section 2.4, first independently and then collectively in the form of pHGS through an extensive simulation study. We show that pPC generally requires significantly fewer statistical calls as compared to PC, and that PATH effectively accomplishes the task of parameter tuning from a single algorithm execution with practically negligible additional computational expense. Compared to repeated executions of PC, the combined effect of pPC and PATH consistently achieves significant computational reductions without sacrificing (and indeed often improving on) estimation accuracy. We demonstrate the effectiveness of HGI on several instantiations of the GSC hybrid framework, and validate the holistic merits of pHGS against several popular structure learning algorithms. Though the focus of this work is on the discrete case, we include

succinct comments and results for our methods on high-dimensional Gaussian data in the discussion in Section 2.5 and in Section 2.4.7.

2.2 The pPC and PATH Algorithms

In constraint-based methods, the computational expense of edge orientation has been noted to be generally insignificant compared to that of skeleton estimation (Chickering, 2002a; Madsen et al., 2017). As such, we develop the partitioned PC algorithm (pPC) to reduce the computational expense of skeleton estimation by imposing a partitioned ordering to the conditional independence tests. Similarly, we propose the p-value adjacency thresholding (PATH) algorithm that effectively accomplishes the task of parameter tuning by efficiently generating a solution path of estimates from a single execution of pPC or PC.

2.2.1 The Partitioned PC Algorithm

The pPC algorithm improves on the already desirable efficiency of the PC algorithm while retaining its attractive theoretical properties and empirical structure learning accuracy. The structure follows similarly to the partition, estimation, and fusion (PEF) strategy applied to the PC algorithm in Gu and Zhou (2020). We develop improvements and computational exploits to further increase performance, formulate pPC to retain soundness and completeness, and propose adaptations to address the challenges of learning the structure of discrete Bayesian networks.

The intuition motivating a partitioned strategy is that any structure learning algorithm that scales worse than linearly with p will be able to estimate $\kappa > 1$ subgraphs for node clusters that partition the p nodes faster than a single graph on all nodes. If the p nodes can be reliably partitioned such that the connectivity between clusters is weak relative to within clusters, then we can expect that there will not be many false positive edges (as a result of causal insufficiency) within subgraphs. Consequently, the adjacencies are expected

to be relatively well-estimated, providing a selective candidate set of neighbors to screen the edges between subgraphs. Coupled with the assumed weak connectivity between clusters, the process of determining the existence of edges amongst clusters is expected to be efficient.

The pPC algorithm estimation process proceeds as follows. We partition the p nodes into κ clusters using a version of the modified hierarchical clustering algorithm proposed in Gu and Zhou (2020) applied with a normalized discrete distance metric, additionally blacklisting marginally independent node pairs. We then apply the PC algorithm to estimate edges within clusters, and filter and refine edges between nodes in different clusters. Finally, we achieve completeness by applying a reduced PC algorithm before orienting the edges.

2.2.1.1 Clustering

As previously motivated, the task of obtaining an effective partition of the nodes is crucial for the success of the skeleton learning. A partition with many clusters κ is desirable for greatest computational benefit in subgraph estimation, but each cluster must be substantive so as to minimally violate causal sufficiency. To accomplish this, the distances between nodes are measured by a normalized mutual information distance metric, and the target number of clusters and initial clusters are chosen adaptively.

Mutual information, denoted $I(X_i, X_j)$, serves as a similarity measure between discrete random variables X_i and X_j and may be interpreted as the Kullback-Leibler divergence between the joint probability distribution and the product of their marginals. We obtain a distance measure by inverting the pairwise mutual information after normalizing using the joint entropy $H(X_i, X_j)$. In particular, we formulate the distance between each pair of variables X_i and X_j as

$$d_{ij} = 1 - \frac{I(X_i, X_j)}{H(X_i, X_j)} \in [0, 1]. \tag{2.1}$$

The proposed distance d_{ij} is a metric in the strict sense as shown by Kraskov et al. (2005),

meaning it is symmetric, non-negative, bounded, and satisfies the triangle inequality. In practice, we compute the empirical quantities of the mutual information and joint entropy ($\hat{I}$ and $\hat{H}$, respectively) between discrete variables (see Section 1.2.3).

Given our distance matrix $D = (d_{ij})_{p\times p}$, we apply Algorithm 1 in Gu and Zhou (2020) with average linkage to determine a cut l for the agglomerative hierarchical clustering of the p nodes. Succinctly described, we choose the highest cut such that the resulting cluster consists of the greatest number of large clusters, defined as node clusters of at least size $0.05p$ according to a loose suggestion by Hartigan (1981). We then merge clusters of size less than $0.05p$ with other small clusters or into large clusters sequentially, ordered by average linkage, until every cluster is a large cluster. For further details regarding the algorithm, we refer to the original paper. The clustering step partitions the p nodes into κ clusters, returning the cluster labels $\mathbf{c} = \{c_1, \ldots, c_p\}$, with $c_i \in \{1, \ldots, \kappa\}$ denoting the cluster label of node i.

While the pairwise computation of both the mutual information and the joint entropy may seem expensive for the purpose of obtaining a partition, we take advantage of two exploits to accomplish this economically. Observing that $I(X_i, X_j) = H(X_i) + H(X_j) - H(X_i, X_j)$, we need only compute the marginal entropies $H(X_i) = I(X_i, X_i)$ to derive the joint entropies from the pairwise mutual information, a reduction from $p(p-1)$ computations to $p(p+1)/2$. Further noting that the discrete unconditional G^2 test statistic for investigating the marginal independence between X_i and X_j is computed as $G^2_{ij} = 2n \cdot \hat{I}(X_i, X_j)$ (see (1.7) in Section 1.2.3), an initial edge screening can easily be obtained through the evaluation

$$\begin{aligned} \Pr(\chi^2_f > 2n \cdot \hat{I}(X_i, X_j)) > \alpha \;&\Rightarrow\; (X_i \perp\!\!\!\perp X_j)_P \\ &\Rightarrow\; \text{blacklist the edge } i - j, \end{aligned} \tag{2.2}$$

where f is the degrees of freedom corresponding to the test of independence between i and j conditioned on $\mathbf{k} = \emptyset$ (see Section 1.2.3). This effectively accomplishes the empty conditioning set ($l = 0$) testing step of the PC algorithm by separating all marginally independent pairs of variables (Algorithm 2.1 line 2).

Note that for continuous data, we recommend using correlation as a similarity measure, which Gu and Zhou (2020) found to lead to reasonable partitions of nodes for divide-and-conquer strategies for learning Gaussian Bayesian networks. This design may also take advantage of (2.2) by testing for zero correlation. After the clustering step, the pPC algorithm and other methods developed in this chapter can be generalized to continuous cases with straightforward substitutions of conditional independence tests and score functions.

2.2.1.2 Partitioned Skeleton Estimation

We now apply the PC algorithm skeleton learning phase to estimate κ disconnected undirected subgraphs according to the partition obtained in the clustering step. Practically, independently applying the PC algorithm to each node cluster benefits from at most κ processors if distributed as such for parallel processing. Furthermore, the speed-up is limited by the longest estimation runtime, usually corresponding to the largest node cluster. In contrast, the design of the PC-stable implementation (Algorithm 1.1) by Colombo and Maathuis (2014) allows for parallel investigation of adjacent node pairs in lines 6-13, provided that updating the graph estimate is deferred to a synchronization step between iterations of l. Several contributions and implementations exist for this approach, referred to as vertical parallelization, which addresses the case where the number of variables p is large (Kalisch et al., 2012; Le et al., 2016; Scutari, 2017; Zarebavani et al., 2020). Alternatively, a horizontal parallelization approach parallelizes across data observations and is preferred when the sample size n is large (Madsen et al., 2017). In these parallelization paradigms, due to the large number of distributed tasks, the number of utilizable computing processors is not practically limited, and the computational load is reasonably expected to be evenly distributed. To take advantage of these developments in parallelizing the PC algorithm, we estimate subgraphs within the node clusters by executing Algorithm 1.1 with the following modifications: (i) form the initial complete undirected graph by only connecting nodes within clusters, (ii) delete edges according to (2.2), and (iii) begin investigating candidate conditioning sets of

size $l = 1$. The result is an undirected graph $\mathcal{G}$ on $\mathbf{V}$ consisting of κ disconnected subgraphs.

At this stage, given a good partition, we expect the node adjacencies to be relatively well-estimated, with the exception of extraneous edge connections within clusters due to the violation of causal sufficiency and missing edge connections between clusters that are disconnected by the partition. Recall from Section 2.2.1.1 that many pairs, including those between clusters, were removed from consideration via the initial marginal independence filtering according to (2.2). We further refine the pairs between clusters through a two-step screening process, our strategy being similar to Algorithm 2 designed by Gu and Zhou (2020), with modifications made to accommodate discrete structure learning. Note that this is where we anticipate to derive the most computational advantage over the PC algorithm. Assuming a block structure was successfully detected, we aim to circumvent the lower order conditional independence tests in our investigation of (1.4) by separating as many between cluster pairs as possible with the currently estimated adjacencies.

The proposed between cluster screening process is summarized in the following operations to the currently estimated skeleton $\mathcal{G} = (\mathbf{V}, \mathbf{E})$:

$$\mathbf{E} \leftarrow \mathbf{E} \cup \left\{ i - j : c_i \neq c_j, \, (X_i \not\perp\!\!\!\perp X_j)_P, \text{ and } (X_i \not\perp\!\!\!\perp X_j \,|\, \mathbf{N}_i^{\mathcal{G}} \cup \mathbf{N}_j^{\mathcal{G}})_P \right\}, \tag{2.3}$$

$$\begin{aligned} \mathbf{E} \leftarrow \mathbf{E} \setminus \Big\{ i - j \in \mathbf{E} : c_i \neq c_j, \text{ and } & (X_i \perp\!\!\!\perp X_j \,|\, \mathbf{N}_i^{\mathcal{G}} \setminus \{X_j\})_P \\ & \text{or } (X_i \perp\!\!\!\perp X_j \,|\, \mathbf{N}_j^{\mathcal{G}} \setminus \{X_i\})_P \Big\}, \end{aligned} \tag{2.4}$$

where c_i is the cluster label of X_i. The first screen in (2.3) constructively connects the between cluster edges that are dependent marginally, as assessed according to (2.2), as well as conditioned on the union of the neighbor sets. With the addition of edges between clusters, the second screen in (2.4) disconnects pairs that are separated by the newly updated adjacencies. As in Algorithm 1.1, we fix adjacencies to retain the capacity for parallel investigation of the considered node pairs. Note that every between-cluster edge that is present in the

underlying DAG will be connected by (2.3), and (2.4) can only prune false positives. Thus, after this step every node pair will have been considered and, in the population setting, every truly connected edge in the underlying DAG will be connected in $\mathcal{G}$.

Remark 1. The formulation of the discrete G^2 tests of independence require enumerating and counting across all conditioning variable configurations. An unavoidable consequence is that the computational complexity and memory requirement increase dramatically with increasing conditioning variables, especially when they have a large number of discrete levels. As such, it is of practical interest to restrict the size of conditioning sets to some user-specified $m > 0$. Thus, for the evaluations in (2.3) and (2.4) (Algorithm 2.1 lines 5 and 13), if for example $|\mathbf{N}_i^{\mathcal{G}} \cup \mathbf{N}_j^{\mathcal{G}}| > m$, we instead investigate the unique sets $\mathbf{X_k} \subseteq \mathbf{N}_i^{\mathcal{G}} \cup \mathbf{N}_j^{\mathcal{G}}$ such that $|\mathbf{k}| = m$. Furthermore, in the case that the considered neighborhood is large due to an unfavorable partition, the investigated conditioning sets may be limited to those contained within only a subset of the neighborhood, provided the subsequent step is adjusted accordingly ((b) in (2.5)).

However, it is important to note that the constraint-based criterion for edge existence expressed in (1.4) has not yet been fully investigated for all node pairs. The tests for edges within clusters only considered conditioning sets consisting of nodes within the same clusters, and tests for edges between clusters were limited to the empty set, $\mathbf{N}_i^{\mathcal{G}} \cup \mathbf{N}_j^{\mathcal{G}}$, $\mathbf{N}_i^{\mathcal{G}} \setminus \{X_j\}$, and $\mathbf{N}_j^{\mathcal{G}} \setminus \{X_i\}$. In particular, for each remaining adjacent node pair $i - j \in \mathbf{E}$, the possible separation sets that have not been evaluated are limited to either of the following cases, if any:

$$\begin{aligned} &\text{(a) } c_i = c_j \text{ (within clusters), sets } \mathbf{k} \text{ where } \exists k \in \mathbf{k} \text{ such that } c_k \neq c_i; \\ &\text{(b) } c_i \neq c_j \text{ (between clusters), sets } \mathbf{k} \neq \emptyset \text{ not defined in (2.3) or (2.4).} \end{aligned} \tag{2.5}$$

The most straightforward continuation to achieve completeness in our partitioned skeleton learning process would be to exhaustively evaluate the dependence of the remaining con-

Algorithm 2.1 `pPC(`$\{\perp\!\!\!\perp_P\}$`)` (population version)

input: conditional independence information $\{\perp\!\!\!\perp_P\}$

output: CPDAG estimate $\mathcal{G}$

partitioning

assign cluster labels $\mathbf{c}$ by partitioning all nodes into κ clusters as in Section 2.2.1.1

within cluster edge estimation

execute $\mathcal{G} \leftarrow$ `PC-skeleton(`$\{\perp\!\!\!\perp_P\}$`)` (Algorithm 1.1), with the following modifications:

line 1: form undirected graph by only connecting node pairs $i, j \in \mathbf{V}$ with $c_i = c_j$

line 3: delete marginally independent pairs from $\mathcal{G}$ according to (2.2) and initialize $l = 1$

between cluster edge screening

initialize $\mathcal{G}' = \mathcal{G}$ to fix adjacencies
for all unordered node pairs i, j such that $c_i \neq c_j$ and $(X_i \not\perp\!\!\!\perp X_j)_P$ **do**
    if $(X_i \not\perp\!\!\!\perp X_j \,|\, \mathbf{N}_i^{\mathcal{G}'} \cup \mathbf{N}_j^{\mathcal{G}'})_P$ **then**
        connect i and j in $\mathcal{G}$
    else
        store separation set in $\mathbf{S}(i, j) = \mathbf{S}(j, i)$
    end if
end for
set $\mathcal{G}' = \mathcal{G}$ to update fixed adjacencies
for all unordered node pairs i, j adjacent in $\mathcal{G}$ with $c_i \neq c_j$ **do**
    if $(X_i \perp\!\!\!\perp X_j \,|\, \mathbf{N}_i^{\mathcal{G}'} \setminus \{X_j\})_P$ or $(X_i \perp\!\!\!\perp X_j \,|\, \mathbf{N}_j^{\mathcal{G}'} \setminus \{X_i\})_P$ **then**
        store separation set in $\mathbf{S}(i, j) = \mathbf{S}(j, i)$
        disconnect i and j in $\mathcal{G}$
    end if
end for

complete edge estimation

execute $\mathcal{G} \leftarrow$ `PC-skeleton(`$\{\perp\!\!\!\perp_P\}$`)` (Algorithm 1.1), continuing from $\mathcal{G}$ and restricting to (2.5) with the following replacements:

line 1: initialize undirected graph with current estimate $\mathcal{G}$

line 2: omit this line

line 3: initialize $l = 1$

line 7: **for all** unique subsets $\mathbf{X_k} \subseteq \mathbf{N}_i^{\mathcal{G}'} \setminus \{X_j\}$ and $\mathbf{X_k} \subseteq \mathbf{N}_j^{\mathcal{G}'} \setminus \{X_i\}$ satisfying $|\mathbf{k}| = l$ and criteria (2.5) **do**

edge orientation

execute $\mathcal{G} \leftarrow$ `skel-to-cpdag(`$\mathcal{G}, \mathbf{S}$`)` (Algorithm 1.2) to orient $\mathcal{G}$

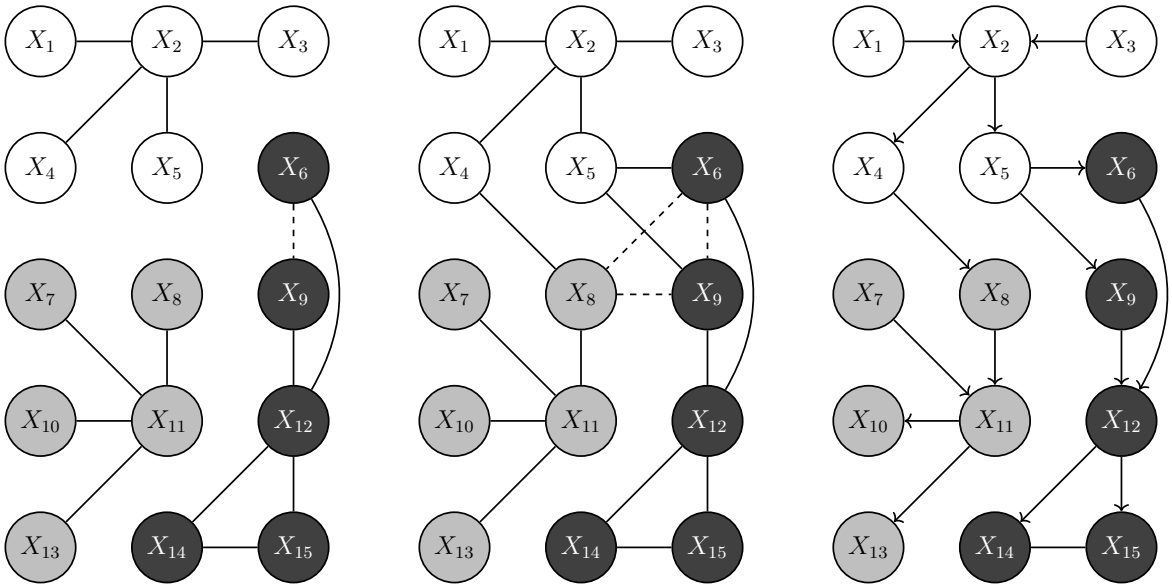
nected node pairs in $\mathcal{G}$ conditioned on the remaining conditioning sets. We accomplish this by restarting the PC algorithm on the current skeleton, evaluating independence conditioned on sets restricted to criteria (2.5), before finally orienting the resulting skeleton to a CPDAG with `skel-to-cpdag` (Algorithm 1.2) to complete structure learning.

The resulting pPC algorithm is detailed in Algorithm 2.1, and an example of its execution is illustrated in Figure 2.1, which provides intuition for its theoretical properties expressed in Theorem 2. The pPC algorithm first estimates the edges within the clusters to obtain $\kappa = 3$ disconnected subgraphs (Figure 2.1a). False positives may be present in these subgraphs in the case that conditioning sets that separate truly disconnected variables contain variables belonging to different clusters, such as in the case of $X_6 - X_9$ for which the common parent X_5 is in a different cluster. Similarly, two false positive edges are constructively connected along with all true positives according to (2.3) (Figure 2.1b), but these are quickly pruned by the second screening of edges between clusters (2.4) (not shown). $X_6 - X_9$ remains to be separated, requiring $(X_6 \perp\!\!\!\perp X_9 \,|\, X_5)_P$ to be investigated in line 18 to recover the underlying skeleton before orientation to the true CPDAG $\mathcal{G}^*$ in line 19 (Figure 2.1c).

The pPC algorithm (Algorithm 2.1) is sound and complete, formalized in Theorem 2, which we prove in Section 2.6.

Theorem 2. *Suppose that probability distribution P and DAG $\mathcal{G}^*$ are faithful to each other. Then given conditional independence oracles, the output of the partitioned PC algorithm (Algorithm 2.1) is the CPDAG that represents the equivalence class of $\mathcal{G}^*$.*

Its implication is the asymptotic consistency of pPC for fixed p as $n \to \infty$, given a consistent conditional independence test such as the G^2 test (Cressie and Read, 1989). Note that while the computational savings may depend heavily on the quality of the partitioning, Theorem 2 holds regardless of the obtained clusters.



(a) $\mathcal{G}$ after line 2 **(b)** $\mathcal{G}$ after line 10 **(c)** Output $\mathcal{G} \equiv \mathcal{G}^*$

Figure 2.1: Example of various stages of pPC (Algorithm 2.1). Node shading denotes clusters, lines represent estimated edges, with dashed lines representing false positive edges. The estimated graph after line 2 and after line 10 are shown in **(a)** and **(b)**, respectively. The final output is equivalent to the CPDAG of the underlying DAG in **(c)**

2.2.2 *p*-value Adjacency Thresholding

Despite the attractive theoretical properties of algorithms such as pPC and PC, it is well-known that in practice, constraint-based algorithms suffer from the multiple testing problem, a challenge exacerbated when p is large (Spirtes, 2010). The effect of individual errors can compound throughout the conditional independence testing process, leading to erroneous inferences regarding both the existence and orientation of edges (Koller and Friedman, 2009; Spirtes, 2010). In addition to deteriorated quality of structures estimated, mistakes in conditional independence inferences can result in invalid PDAG estimates that do not admit a consistent extension (see Remark 2). In practical applications, the choice of conditional independence test threshold α can significantly control the sparsity and quality of the resulting estimate. To the best of our knowledge, proposed theoretical thresholds depend on

unknown quantities and are not practically informative, such as in Kalisch and Bühlmann (2007). Empirically, the optimal choice of α varies depending on factors such as the sample size and the structure and parameters of the underlying Bayesian network, and no universally well-performing value is known.

We propose the p-value adjacency thresholding (PATH) algorithm to generate and select from a CPDAG solution path across various values of α from a single execution of the pPC algorithm, or indeed from any constraint-based structure learning algorithm that is able to obtain the following. Define the maximum p-values $\Phi = (\Phi_{ij})$ such that Φ_{ij} is the maximum p-value obtained by the conditional independence test between i and j across all conditioning sets $\mathbf{k} \in \mathcal{K}_{ij}$ visited in the algorithm, as well as the corresponding separation sets, thus extending the definition of $\mathbf{S}$. That is, for all distinct node pairs i, j,

$$\begin{aligned} \Phi_{ij} = \Phi_{ji} &\coloneqq \max_{\mathbf{k} \in \mathcal{K}_{ij}} \Pr\left(\chi^2_f > G^2_{ij|\mathbf{k}}\right), \\ \mathbf{S}(i,j) = \mathbf{S}(j,i) &\coloneqq \operatorname*{argmax}_{\mathbf{k} \in \mathcal{K}_{ij}} \Pr\left(\chi^2_f > G^2_{ij|\mathbf{k}}\right). \end{aligned} \tag{2.6}$$

For a connected node pair $i - j$, $\mathbf{S}(i,j)$ may be considered the conditioning set closest to separating i and j, and Φ_{ij} measures how close.

The process itself is straightforward: for a sequence of significance levels $\{\alpha^{(t)}\}$, we obtain updated PDAG estimates $\mathcal{G}^{(t)}$ by thresholding the maximum achieved p-values Φ to obtain skeleton estimates with edge sets $\mathbf{E}^{(t)} = \{i - j : \Phi_{ij} \leq \alpha^{(t)}\}$ and then orienting them to CPDAGs according `skel-to-cpdag` (Algorithm 1.2) with the corresponding separation information $\mathbf{S}^{(t)} = \{\mathbf{S}(i,j) : \Phi_{ij} > \alpha^{(t)}\}$. The quality of the estimates are then evaluated by a scoring criterion and the highest-scoring network is returned. In what follows, we develop the choice of the threshold values and present the strategy for estimate generation and selection.

We begin with a graph estimate obtained by executing the pPC algorithm with some maximal threshold value α. The goal is to start with the densest graph so that the elements of Φ and $\mathbf{S}$ in (2.6) are maximized over a larger number of visited conditioning sets $|\mathcal{K}|$.

We generate a sequence of τ values decreasing from $\alpha^{(1)} \coloneqq \alpha$ to some minimum threshold value $\alpha^{(\tau)}$. This sequence may be incremented according to some linear or log-linear scale, but we choose to achieve maximal difference in sparsity amongst estimates by utilizing the information in Φ. Given each $\alpha^{(t)}$ corresponding to estimate $\mathcal{G}^{(t)} = (\mathbf{V}, \mathbf{E}^{(t)})$, we choose $\alpha^{(t+1)}$ such that

$$|\mathbf{E}^{(t)}| - |\mathbf{E}^{(t+1)}| \approx \frac{|\mathbf{E}^{(1)}| - |\mathbf{E}^{(\tau)}|}{\tau - 1}. \tag{2.7}$$

Noting that $|\mathbf{E}^{(t)}| = \sum_{i<j} \mathbb{1}\left\{\Phi_{ij} \leq \alpha^{(t)}\right\}$ for indicator function $\mathbb{1}\{\cdot\}$, it is easy to see that the sequence $\alpha^{(1)}, \ldots, \alpha^{(\tau)}$ can be straightforwardly obtained using the order statistics of the elements of Φ.

Once a solution path of CPDAG estimates $\{\mathcal{G}^{(t)} : t \in \{1, \ldots, \tau\}\}$ is obtained, we select the highest quality estimate by means of score-based selection. The *Bayesian information criterion* (BIC) is a penalized log-likelihood score derived from the asymptotic behavior of Bayesian network models, with established consistency (Schwarz, 1978). The formulation of the BIC score makes clear its *score decomposability* (see (2.9) in Section 2.3.1); that is, the score $\phi(\mathcal{G}, \mathcal{D})$ can be computed as the sum of the scores of the individual variables with respect to their parents in $\mathcal{G}$: $\phi(\mathcal{G}, \mathcal{D}) = \sum_{i=1}^{p} \phi(X_i, \mathbf{Pa}_i^{\mathcal{G}})$. The BIC score is additionally *score equivalent*, evaluating all Markov equivalent DAGs as of identical quality, with a higher value indicating a better fit to the data.

Due to score equivalence, it is sufficient to evaluate each CPDAG $\mathcal{G}^{(t)}$ with any arbitrary DAG $\tilde{\mathcal{G}}^{(t)}$ in its equivalence class, called a *consistent extension* of $\mathcal{G}^{(t)}$. Dor and Tarsi (1992) proposed a simple algorithm, which we refer to as `pdag-to-dag`, that obtains such an extension by orienting the undirected (reversible) edges in $\mathcal{G}^{(t)}$ without inducing directed cycles or introducing new v-structures, and is guaranteed to succeed if a consistent extension exists (further discussed in Section 2.3.2). After obtaining these DAG extensions, in practice, score decomposability can be leveraged to avoid scoring p nodes for τ estimates by setting $\Delta^{(1)} = 0$

Algorithm 2.2 `PATH(`Φ`, S, `$\mathcal{D}$`, `τ`, `$\alpha^{(\tau)}$`)` (sample version)

input: maximum p-values Φ, separation sets $\mathbf{S}$, data $\mathcal{D}$, number of estimates τ, minimum threshold $\alpha^{(\tau)}$

output: CPDAG $\mathcal{G}^{(t^*)}$, CPDAG solution path $\{\mathcal{G}^{(t)} : t \in \{1, \dots, \tau\}\}$

generate a decreasing sequence of τ values $\{\alpha^{(t)}\}$ from $\alpha^{(1)} \coloneqq \max_{i,j} \Phi_{ij}$ to $\alpha^{(\tau)}$ by (2.7)
for $t = 1, 2, \dots, \tau$ **do**
    obtain skeleton $\mathcal{G}^{(t)} = (\mathbf{V}, \mathbf{E}^{(t)})$ by thresholding: $\mathbf{E}^{(t)} = \{i - j : \Phi_{ij} \leq \alpha^{(t)}\}$
    obtain separation sets $\mathbf{S}^{(t)} = \{\mathbf{S}(i, j) : \Phi_{ij} > \alpha^{(t)}\}$
    execute $\mathcal{G}^{(t)} \leftarrow$ `skel-to-cpdag(`$\mathcal{G}^{(t)}, \mathbf{S}^{(t)}$`)` (Algorithm 1.2 in Section 1.2.2)
    obtain DAG extension $\tilde{\mathcal{G}}^{(t)} \leftarrow$ `pdag-to-dag(`$\mathcal{G}^{(t)}$`)` (Dor and Tarsi, 1992)
    if $t = 1$ **then**
        $\Delta^{(t)} = 0$
    else
        compute score difference $\Delta^{(t)} = \phi(\tilde{\mathcal{G}}^{(t)}, \mathcal{D}) - \phi(\tilde{\mathcal{G}}^{(t-1)}, \mathcal{D})$
    end if
end for
select the best estimate $t^* = \operatorname{argmax}_{t \in \{1, \dots, \tau\}} \sum_{r=1}^{t} \Delta^{(r)}$
return $\mathcal{G}^{(t^*)}$ and $\{\mathcal{G}^{(t)} : t \in \{1, \dots, \tau\}\}$

and computing the score differences between estimates $\Delta^{(t)} = \phi(\tilde{\mathcal{G}}^{(t)}, \mathcal{D}) - \phi(\tilde{\mathcal{G}}^{(t-1)}, \mathcal{D})$ for $t = 2, \dots, \tau$, caching computed node score to avoid redundant computations. The best solution can then be straightforwardly obtained according to

$$t^* = \operatorname*{argmax}_{t \in \{1, \dots, \tau\}} \phi(\mathcal{G}^{(t)}, \mathcal{D}) = \operatorname*{argmax}_{t \in \{1, \dots, \tau\}} \sum_{r=1}^{t} \Delta^{(r)}.$$

We detail the PATH solution path strategy in Algorithm 2.2.

Remark 2. While `pdag-to-dag` is guaranteed to extend a PDAG $\mathcal{G}$ to a DAG if any consistent extension exists, in the presence of finite-sample error, Algorithm 1.2 may obtain a PDAG estimate $\mathcal{G}$ for which no such extension exists. In such a case, we say that $\mathcal{G}$ does not admit a consistent extension, and refer to it as an invalid CPDAG. Such PDAGs contain undirected edges that cannot be oriented without inducing cycles or constructing additional v-structures, and do not encode any probabilistic model of P. To account for these, we

restrict the candidate graphs considered in line 13 to valid CPDAGs. In the case that no valid CPDAG is obtained, we obtain semi-arbitrary DAG extensions $\tilde{\mathcal{G}}^{(t)}$ by first applying the algorithm by Dor and Tarsi (1992) and randomly directing as many remaining undirected edges as possible without introducing any cycles, finally removing edges that cannot be oriented. The resulting DAGs are used to score the PDAGs, and the original PDAGs are returned in the output as these structures are nonetheless interpretable even as incomplete dependency structures.

The computational expense of executing Algorithm 2.2 can reasonably be expected to be insignificant compared to any constraint-based algorithm to which it is attached, supported by our empirical results in Section 2.4.2. Our exploitation of score decomposability reduces the order of score computations far below the worst case of $O(\tau p)$, which is already much more efficient than any favorable order of conditional independence tests such as polynomial with p. As for the τ executions of `skel-to-cpdag` and `pdag-to-dag`, Chickering (2002a) found the computational cost of applying the edge orientation heuristics to be insignificant, as did Madsen et al. (2017) for `skel-to-cpdag` in comparison to skeleton learning. As such, the computational cost for score-based selection from the solution path can be expected to be essentially inconsequential, an assertion further validated in our experiments.

In the large-sample limit, the correctness of (1.4) and `skel-to-cpdag` implies the asymptotic consistency of PC and pPC under certain conditions without any solution path (see Lemma 2 in Section 2.6). This property can be achieved with a consistent conditional independence test by controlling type I error with $\alpha_n \to 0$ as $n \to \infty$, thus rendering the test Chernoff-consistent (Cressie and Read, 1989; Shao, 2003, Definition 2.13). As α_n is not practically informative due to its implicit dependence on n, Algorithm 2.2 contributes an element of accessibility to this asymptotic result in the form of the following theorem, a proof for which can be found in Section 2.6.

Theorem 3. *Suppose the distribution P is fixed and faithful to a DAG with CPDAG $\mathcal{G}^*$, $\mathcal{D}$ is data containing n i.i.d. samples from P, and ϕ is a consistent score. Let $\Phi_n = (\Phi_{n,ij})$ and*

$\mathbf{S}_n$ *be the maximum p-values and corresponding separation sets recorded for any exhaustive investigation of* (1.4)*, executed with a consistent test applied with threshold* α_n*. Let* $\hat{\mathcal{G}}_n^{(t^*)}(\alpha_n)$ *be the selected estimate of Algorithm 2.2 with parameters* $\tau = 1 + \sum_{i<j} \mathbb{1}\{\Phi_{n,ij} \leq \alpha_n\}$ *and* $\alpha^{(\tau)} = 0$ *applied to* Φ_n *and* $\mathbf{S}_n$*. Then there exists* $a_n \to 0$ *as* $n \to \infty$ *such that if* $\alpha_n \geq a_n$ *when* n *is large,*

$$\lim_{n\to\infty} \Pr\left[\hat{\mathcal{G}}_n^{(t^*)}(\alpha_n) = \mathcal{G}^*\right] = 1. \tag{2.8}$$

In particular, (2.8) *holds if* α_n *is fixed to some constant* $\alpha \in (0, 1)$ *for all* n*.*

In the finite-sample setting, the thresholded estimate $\mathcal{G}^{(t)}$ corresponding to $\alpha^{(t)} < \alpha^{(1)}$ typically does not correspond exactly to the estimate obtained by directly executing PC or pPC with significance level $\alpha^{(t)}$. Adjacencies that would be disconnected in the earlier stages of the learning process with threshold $\alpha^{(t)}$ may survive to the later stages with a more lenient threshold $\alpha^{(1)}$. The enlarged neighborhoods may persist as additional (potentially false positive) connections in the output structure, but may also cause more conditioning sets to be considered and thus lead to the deletion of edges that would not have been deleted in the execution with $\alpha^{(t)}$. Thus, while the estimated structure from executing with $\alpha^{(1)}$ may reasonably be expected to be denser than that of $\alpha^{(t)}$, it may not be a supergraph of the latter, and the p-values for each node pair (2.6) are optimized over different conditioning sets, so the thresholded estimate with PATH is likely to differ. Nonetheless, our empirical results demonstrate the potential of the solution path generated by thresholding, showing that PATH applied to pPC and PC is able to produce estimates of competitive quality to the best of those obtained by multiple executions with various α.

To the best of our knowledge, there is currently no easy way to choose the optimal threshold α_n for PC (or pPC), and thus repeated executions are often needed for parameter tuning. The application of PATH pragmatically allows for a single execution of PC or pPC with fixed threshold α while returning estimates with both theoretical guarantees

(Theorem 3) and empirical well-performance (Section 2.4.2).

2.3 Consistent Hybrid Structure Learning

Despite the accessibility provided by PATH (Algorithm 2.2), the asymptotic guarantees of constraint-based learning strategies do not necessarily translate to well-performance in practice. For this reason, based on their experiments with which our simulation results are in agreement, Tsamardinos et al. (2006a) preferred greedy search in the DAG space over constraint-based orientation in their development of their algorithm, even though the former is lacking in comparable theoretical guarantees.

In the interest of elevating the class of algorithms that share such an compromise, we develop the hybrid greedy initialization (HGI) strategy to preserve the asymptotic guarantees of sound and complete constraint-based structure learning while improving on the empirical well-performance of the current standard hybrid framework. We motivate and develop HGI in this section, first reviewing relevant standard score-based and hybrid structure learning before describing the HGI strategy in detail.

2.3.1 Score-based and Hybrid Structure Learning

Chickering (2002a) distills score-based Bayesian network learning into two general problems: the evaluation problem and the identification problem. In this section, we begin by introducing the relevant tenets of score-based structure learning under these categories.

We have briefly interacted with the evaluation problem in our discussion of the BIC score in Section 2.2.2, which more broadly involves the development of scoring criteria to evaluate the goodness-of-fit of a Bayesian network to data. The existence of equivalence classes (Section 1.1.1) motivates the design of metrics that evaluate all structures within an equivalence class as of equal quality, satisfying the score equivalence property. The BIC score that we utilize satisfies this property, and is equivalent to the (negative) minimum

description length (MDL) in Rissanen (1978). Other scores that are score equivalent include log-likelihood, Akaike's information criterion (AIC), and Bayesian Dirichlet equivalence score (BDeu) (Akaike, 1974; Buntine, 1991; Heckerman et al., 1995). Notwithstanding, prominent scores that are not score equivalent exist as well, such as the K2 score (Cooper and Herskovits, 1991) and, more recently, ℓ_1-regularized likelihood scores (Fu and Zhou, 2013; Gu et al., 2019). In their investigation, Liu et al. (2012) found BIC to have favorable model selection performance relative to a number of other scores.

The BIC score is a special case of the penalized multinomial log-likelihood score. Following the notation from (1.7), let $\mathbf{pa}_i^{\mathcal{G}}$ represent one of the q_i unique state configurations of $\mathbf{Pa}_i^{\mathcal{G}}$. Given the Bayesian network DAG structure $\mathcal{G}$ and empirical data $\mathcal{D}$ from P, the penalized score with penalty parameter λ is computed as

$$\begin{aligned}
\phi(\mathcal{G}, \mathcal{D}) &= \sum_{i, x_i, \mathbf{pa}_i^{\mathcal{G}}} n[x_i, \mathbf{pa}_i^{\mathcal{G}}] \log\left(\frac{n[x_i, \mathbf{pa}_i^{\mathcal{G}}]}{n[\mathbf{pa}_i^{\mathcal{G}}]}\right) - \lambda \sum_i (r_i - 1) q_i \\
&= \sum_{i=1}^{p} \left(\sum_{x_i, \mathbf{pa}_i^{\mathcal{G}}} n[x_i, \mathbf{pa}_i^{\mathcal{G}}] \log\left(\frac{n[x_i, \mathbf{pa}_i^{\mathcal{G}}]}{n[\mathbf{pa}_i^{\mathcal{G}}]}\right) - \lambda (r_i - 1) q_i \right) \\
&= \sum_{i=1}^{p} \phi(X_i, \mathbf{Pa}_i^{\mathcal{G}}).
\end{aligned} \tag{2.9}$$

In the penalty term, $\sum_{i=1}^{p} (r_i - 1) q_i$ encodes model complexity, encouraging sparsity in the structure of $\mathcal{G}$. In the BIC score, $\lambda = \frac{1}{2}\log(n)$ (Schwarz, 1978).

The consistency property of the BIC score guarantees that in the large-sample limit, $\mathcal{G}^* = \text{argmax}_{\mathcal{G}}\, \phi(\mathcal{G}, \mathcal{D})$ is in the equivalence class of the underlying DAG. BIC additionally retains the property of *local consistency* (Chickering, 2002b), meaning for any DAG $\mathcal{G}$ and another DAG $\mathcal{G}'$ resulting from adding the edge $X_i \to X_j$ to $\mathcal{G}$, the following two properties

hold asymptotically:

$$(X_j \not\perp\!\!\!\perp X_i | \mathbf{Pa}_j^{\mathcal{G}})_P \Rightarrow \phi(\mathbf{X} \mid \mathcal{G}', \mathcal{D}) > \phi(\mathbf{X} \mid \mathcal{G}, \mathcal{D}), \text{ and} \tag{2.10}$$

$$(X_j \perp\!\!\!\perp X_i | \mathbf{Pa}_j^{\mathcal{G}})_P \Rightarrow \phi(\mathbf{X} \mid \mathcal{G}', \mathcal{D}) < \phi(\mathbf{X} \mid \mathcal{G}, \mathcal{D}). \tag{2.11}$$

We have discussed the BIC score as having desirable qualities for evaluating Bayesian network structures, being decomposable, score equivalent, consistent, locally consistent, and empirically well-performing. However, finding the global optimum $\mathcal{G}^*$ is highly non-trivial, reminding us of the problem of identification.

Relevant to our work is the general *greedy search* algorithm which repeatedly moves from the current state to the neighboring state that maximally improves the optimization criterion (in our application, BIC) until no improvement can be thusly achieved (Russell and Norvig, 2009). That is, the algorithm is guaranteed to terminate in a locally optimal state, where locality is determined by the chosen definitions of a state and its neighborhood. The popular hill-climbing (HC) algorithm is a greedy search in the state space of DAGs, with neighboring states defined as DAGs obtainable by a single directed edge addition, deletion, or reversal applied to the current DAG (Heckerman et al., 1995; Russell and Norvig, 2009). The greedy equivalence search (GES) algorithm is a sound and complete variation of greedy search in which the state space is CPDAGs representing equivalence classes, with a forward-stepping edge addition phase followed by an edge deletion phase (Meek, 1997; Chickering, 2002a,b).

While widely applied and regarded as efficient and well-performing, the locality of the hill-climbing search in the DAG space unavoidably risks the common problem of accepting locally optimal yet globally suboptimal solutions. Gámez et al. (2011) showed that under certain conditions, hill-climbing returns a minimal independence map of the probability distribution P, but it does not guarantee a globally optimal result. Hill-climbing can be augmented to more thoroughly search the DAG space with one or both of *tabu list* and *random restarts*, respectively governed by parameters (t_0, t_1) and (r_0, r_1). In what is known as the *tabu*

search, a solution is obtained through hill-climbing while a tabu list stores the last t_1 DAG structures visited during the search. Then, the hill-climbing procedure is continued for up to t_0 iterations while allowing for minimal score decreases, with a local neighborhood restricted by the tabu list to avoid previously visited structures. In hill-climbing with random restarts, the hill-climbing procedure is repeated r_0 times after the initial execution by perturbing the current solution with r_1 random local changes. In our work, we prefer augmenting hill-climbing with a tabu list rather than random restarts due to its generally superior efficiency and its reliable and deterministic well-performance.

As mentioned in Section 2.1, prominent hybrid structure learning algorithms are instantiations of what we call the GSC framework, in which hill-climbing is executed from an empty graph restricted to a sparse set of candidate edge connections. That is, for a graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ estimated using a constraint-based approach, define $\mathbf{A} = \{(i, j) : i \text{ and } j \text{ are connected in } \mathcal{G}\}$ as the set of candidates: the unordered node pairs that have not been determined to be conditionally independent. Hill-climbing is then executed from an empty graph on $\mathbf{V}$, considering adding an edge $i \to j$ only if $(i, j) \in \mathbf{A}$. MMPC and H2PC are two well-known examples, obtaining $\mathbf{A}$ according to sound skeleton estimation algorithms max-min parents and children (MMPC) and hybrid parents and children (HPC), respectively (Tsamardinos et al., 2006a; Gasse et al., 2014). The GSC strategy guarantees estimation of a valid DAG restricted to $\mathbf{A}$, but often accepts locally optimal solutions that are structurally inaccurate due to the connectivity of the DAG space induced by the hill-climbing neighborhood. As will be seen in Section 2.4.3, this problem persists even when the search space is well-restricted and a tabu list is utilized, leaving much to be desired. In this work, we primarily focus on improving upon the theoretical properties and empirical performance of the GSC framework, though our contributions may be applied to another hybrid approach that uses greedy search in the space of equivalence classes instead of DAGs, which we discuss briefly in Section 2.5 and Section 2.4.7.

2.3.2 Hybrid Greedy Initialization

We now develop our proposed HGI strategy to overcome the aforementioned difficulties for hybrid algorithms belonging to the GSC framework. Our method is designed to retain the soundness and completeness of constraint-based structure learning while empirically improving structural estimation accuracy and achieving higher-scoring structures as compared to those obtained by the GSC framework. The primary novel contribution is the introduction of a score-based ordering to the application of orientation heuristics to obtain a favorable initialization for hill-climbing. Given the skeleton output of a constraint-based algorithm, we sequentially add v-structures that most improve the score, scored with respect to directed edges. We then make greedy determinations for the remaining undirected edges according to efficient criteria from `pdag-to-dag`, assisted by Meek's rules R1-4 (Dor and Tarsi, 1992; Meek, 1995). Finally, we execute hill-climbing initialized by the resulting DAG. From a score-based learning perspective, the formulation of HGI may be understood as a principled strategy for obtaining a good starting point for greedy search. In what follows, we further detail and discuss the HGI algorithm.

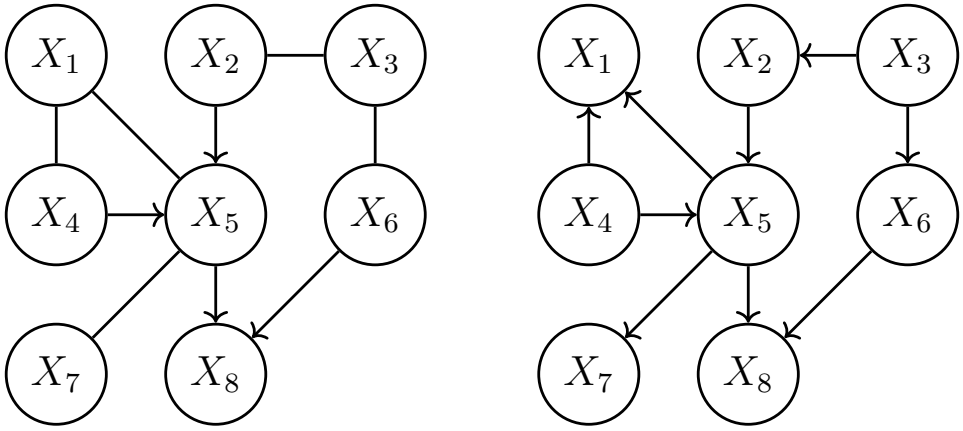
Recall that `pdag-to-dag` (introduced in Section 2.2.2) is guaranteed to obtain a consistent extension of a PDAG if one exists, and thus implicitly includes Meek's rules R1-4 when the given PDAG is a valid pattern that admits a consistent extension. Let $\mathcal{G}_0$ and $\mathcal{G}$ be identical copies of a PDAG to be oriented. The algorithm repeatedly searches for a node j satisfying the following conditions in a PDAG $\mathcal{G}_0$:

(a) j is a *sink*: that is, j has no edges directed outwards in $\mathcal{G}_0$;

(b) For every vertex k connected to j by an undirected edge in $\mathcal{G}_0$, k is adjacent to all the other vertices which are adjacent to j in $\mathcal{G}_0$.

If such a node j can be found, all undirected edges adjacent to j are oriented into j in $\mathcal{G}$ and $\mathcal{G}_0$. Node j is then a *complete sink*, a node satisfying (a) with no undirected edges

incident to it, and is removed from $\mathcal{G}_0$ along with all edges incident to it in order to uncover subsequent candidate nodes. This process is repeated until $\mathcal{G}$ is fully oriented to a DAG, or until no such node j can be found, in which case the initial PDAG does not admit a consistent extension. Briefly exposited, (a) ensures acyclicity by requiring that all directed paths induced by considered orientations terminate in sinks, and (b) ensures that considered orientations do not create new v-structures if applied.



(a) Pattern PDAG structure **(b)** DAG extension

Figure 2.2: Example of `pdag-to-dag` (Dor and Tarsi, 1992) applied to a PDAG pattern structure

Consider an example of `pdag-to-dag` applied to a PDAG in Figure 2.2. The algorithm proceeds as follows. Starting from the PDAG structure in Figure 2.2a, nodes X_1, X_7, and X_8 satisfy conditions (a) and (b), with X_8 additionally a complete sink. Nodes X_2, X_4, and X_6 violate condition (a), and X_3 and X_5 violate condition (b). Since X_1, X_7, and X_8 are not adjacent to each other, they may be selected by the algorithm in an arbitrary order without affecting the particular outcome of the DAG extension, resulting in orientations $X_4 \to X_1$, $X_5 \to X_1$, and $X_5 \to X_7$. Once these nodes are removed from consideration, X_5 and X_6 are likewise removed as complete sinks, and the remaining undirected edge $X_2 - X_3$ may be oriented in either direction as both X_2 and X_3 satisfy (a) and (b).

Since the implementation of `pdag-to-dag`, as proposed, does not straightforwardly lend

itself to greedy application, we accomplish this by developing a decomposed version of `pdag-to-dag`. Let $\mathcal{G}_0$ be a PDAG with only v-structures oriented, and let $\mathcal{G}$ be a DAG consisting of only the directed edges in $\mathcal{G}_0$. We prioritize checking for and removing all complete sinks from consideration by deleting all edges incident to such nodes in $\mathcal{G}_0$, and we greedily consider orienting $i \to j$ in $\mathcal{G}_0$ and $\mathcal{G}$ if $i - j$ is an undirected edge in $\mathcal{G}_0$ and conditions (a) and (b) are satisfied for node j. For example, in a single greedy step applied to Figure 2.2a, we would first remove X_8 from $\mathcal{G}_0$ as a complete sink, resulting in nodes X_1, X_6, and X_7 satisfying (a) and (b). We then greedily consider the individual edge orientations $X_4 \to X_1$, $X_5 \to X_1$, $X_5 \to X_7$, and $X_3 \to X_6$, applying the orientation that most improves the score computed with respect to the structure of $\mathcal{G}$ (i.e., all edges that have determined orientations). This design essentially decomposes the node-centric operations in `pdag-to-dag` into single edge operations (e.g., $X_4 \to X_1$ and $X_5 \to X_1$ are considered as individual orientations instead of both being considered with node X_1), and its result is a DAG in the same equivalence class as the output of `pdag-to-dag` given a valid PDAG. In practice, as with the sequential v-structure application, the greedy ordering filters edges and selects between ambiguous orientations. In the case that undirected edges still exist in $\mathcal{G}_0$ and no node satisfying (a) and (b) can be found, we likewise greedily consider transformations compelled by Meek's rules R1-4 applied to $\mathcal{G}_0$. We detail the HGI strategy in Algorithm 2.3.

Important to note is that in the population setting, v-structure detection and orientation is order-independent, and while the particular DAG obtained by (our decomposed) `pdag-to-dag` is order-dependent, it will always recover a DAG in the same equivalence class if successful (i.e., a consistent extension of the input PDAG exists). In such a case, whatever ordering imposed on both or either of the heuristics has no meaningful effect on the result. Furthermore, given a greedy criterion, a locally consistent score will asymptotically accept proposed additions of truly connected edges due to (2.10), thus preserving guaranteed identification of the equivalence class of the underlying DAG. Indeed, in such a setting, a lenient score that prefers denser graphs is sufficient as only property (2.10) is required.

Algorithm 2.3 `HGI(`$\mathcal{G}_0$`, `$\mathcal{D}$`, `$\mathbf{U}$`)` (sample version)

input: undirected graph $\mathcal{G}_0$, data $\mathcal{D}$, and v-structures $\mathbf{U}$
output: DAG $\mathcal{G}$

1: initialize $\mathcal{G}$ as the empty graph on $\mathbf{V}$
2: **repeat**
3: orient $i \to k$ and $j \to k$ in $\mathcal{G}_0$ and $\mathcal{G}$ for $(i, k, j) \in \mathbf{U}$ that most improves $\phi(\mathcal{G}, \mathcal{D})$ and does not induce any cycles or conflict with any v-structures in $\mathcal{G}$
4: **until** no such improvement possible
5: **repeat**
6: delete all edges incident to complete sinks in $\mathcal{G}_0$
7: apply the first applicable of the following operations for $i - j$ in $\mathcal{G}_0$:
(i) apply $i \to j$ to $\mathcal{G}_0$ and $\mathcal{G}$ satisfying (a) and (b) that most improves $\phi(\mathcal{G}, \mathcal{D})$
(ii) delete $i - j$ from $\mathcal{G}_0$ satisfying (a) and (b) that if applied, would most deteriorate $\phi(\mathcal{G}, \mathcal{D})$
(iii) apply $i \to j$ to $\mathcal{G}_0$ and $\mathcal{G}$ compelled by R1-4 that most improves $\phi(\mathcal{G}, \mathcal{D})$
(iv) delete $i - j$ from $\mathcal{G}_0$ with $i \to j$ compelled by R1-4 such that if applied, would most deteriorate $\phi(\mathcal{G}, \mathcal{D})$
8: **until** no such operation possible

In the finite-sample setting, incorrectly inferred conditional independence information can result in the determination of incomplete or extraneous and even conflicting v-structures, and could result in PDAGs that do not admit a consistent extension (Remark 2). The outcome of a naive non-greedy application of v-structures and (our decomposed) `pdag-to-dag` empirically varies in quality depending on the order by which the operations are applied due to conflicting operations and obstacles induced by the acyclicity constraint, providing the primary incentive for greedy decisions regarding proposed constraint-based orientations. From a constraint-based learning perspective, greedy forward stepping imposes a greedy ordering on the application of v-structures and other potentially conflicting or ambiguous edge orientations, while additionally providing an element of selectivity by disregarding operations that deteriorate the score.

As already discussed, Algorithm 2.3 asymptotically preserves sound and complete orien-

tation of the skeleton of the underlying DAG $\mathcal{G}$ to a DAG in its equivalence class, straightforwardly evident from our discussion thus far.

Lemma 1. *Suppose that probability distribution P is fixed and faithful to a DAG $\mathcal{G}^*$, $\mathcal{D}_n$ is data containing n i.i.d. samples from P, and ϕ is a score satisfying local consistency. Let $\hat{\mathcal{G}}_n$ be the output of Algorithm 2.3. If $\mathcal{G}_0$ is the skeleton of $\mathcal{G}^*$ and $\mathbf{U}$ contains the v-structures of $\mathcal{G}^*$, then $\hat{\mathcal{G}}_n$ is in the same equivalence class as $\mathcal{G}^*$ with probability approaching one as $n \to \infty$.*

Note that while we state Lemma 1 assuming possession of all v-structures $\mathbf{U}$ that are present in the underlying DAG, these may be correctly obtained asymptotically depending on what information is available from the skeleton estimation method, which we discuss in Section 1.2.2.

Indeed, neither operations (ii)-(iv) in line 7 nor any subsequent score-based search is necessary for Lemma 1 to hold, but rather serve in a corrective capacity in the finite-sample setting. The process of completing and deleting sinks to uncover subsequent sinks in $\mathcal{G}_0$ requires decisions for each undirected edge participating in a node satisfying (a) and (b) in order to continually progress in the algorithm. Operation (ii) discards each proposed edge addition $i \to j$ that deteriorates the score when no improvement according to (i) is possible so that j can be completed and removed. In the case that no node satisfying both (a) and (b) can be found, we apply the same greedy criterion to all edges compelled by Meek's rules R1-4 in operations (iii) and (iv). These rules are not subject to a leaf-to-root construction and often help resume applications of (i) and (ii), for example by deleting an undirected edge $i - j$ participating in an unshielded triple from $\mathcal{G}_0$ so that j can satisfy (b).

Note that in finite-sample applications, repeated application of (i)-(iv) does not guarantee orientation or deletion of all undirected edges in $\mathcal{G}_0$ (e.g. consider an undirected square where no vertex satisfies (a) and (b) and no edge is compelled by R1-4), though we empirically find it to typically address most if not all edges. Furthermore, while the adjacency criterion

(b) exists to prevent the creation of additional v-structures, it is still possible for new v-structures to be created by deletion. Consider an undirected triangle in $\mathcal{G}_0$ where all three vertices i, j, and k satisfy (a) and (b). The greedy ordering may orient $i \to k$ and $j \to k$, remove node k once it is a complete sink, and eventually delete $i - j$, leaving $i \to k \leftarrow j$ as a new v-structure in $\mathcal{G}$.

While essentially equivalent in the large sample setting, directly executing a greedy decomposed `pdag-to-dag` poses a number of pragmatic advantages over first greedily applying Meek's rules in the presence of finite-sample error. The sink criterion (a) effectively accomplishes acyclicity checks for each proposed edge orientation, which grow increasingly computationally burdensome for larger networks. It additionally induces a leaf-to-root construction with operations that minimally conflict with subsequent operations, with $i \to j$ only denying i from satisfying (a) until j is removed. This further strengthens the effect of the greedy ordering in minimizing ambiguity in the initial DAG construction process. Indeed, considering Theorem 1, the order of greedy v-structure application is unambiguous given a score equivalent metric. In contrast, hill-climbing from an empty graph restricted to sparse candidates $\mathbf{A}$ begins with $O(|\mathbf{A}|^2)$ ambiguous edge additions where, for any distinct node pair $(i, j) \in \mathbf{A}$, adding the edge $i \to j$ or $j \to i$ results in the same score improvement, again evident from Theorem 1. Hill-climbing may encounter many such non-unique edge additions which are typically decided according to a node ordering that is often arbitrary, and their compounding effect can result in conflicts that, together with the acyclicity constraint, entrap hill-climbing in local solutions.

Remark 3. Relevant to our work is the partition, estimation, and fusion (PEF) framework by Gu and Zhou (2020), a hybrid strategy consisting of a final fusion step that is conceptually analogous to a non-greedy form of sparse candidate hill-climbing initialized with the directed edges of an estimated PDAG rather than an empty graph. The algorithm removes all undirected edges from a PDAG input and performs local edge additions, reversals, and deletions to the resulting DAG that improve the overall score by repeatedly iterating through

the surviving node pairs in a semi-arbitrary order, simultaneously testing for conditional independence. While they empirically demonstrated this process to correct many of the errors in the estimated structure, we find that the order with which the edges are visited can result in varying degrees of improvement, and the testing strategy performs redundant conditional independence tests. Furthermore, naively initializing a score-based search with the PDAG output of a constraint-based algorithm may prove volatile given the sensitivity of `skel-to-cpdag` to erroneous conditional independence inferences as well as its order-dependence. Lastly, even if initialized by the DAG consisting of the compelled edges of the underlying DAG and perfectly restricted to true connections, neither PEF nor hill-climbing with a consistent score guarantees asymptotic orientation to a DAG in the equivalence class of $\mathcal{G}$.

Finally, we detail in Algorithm 2.4 the partitioned hybrid greedy search (pHGS) algorithm, a composition of pPC, PATH, and HGI. The pHGS algorithm efficiently restricts the search space with the pPC algorithm (Algorithm 2.1), obtaining Φ and $\mathbf{S}$ as in (2.6) for use in PATH. Instead of generating τ CPDAG estimates with `skel-to-cpdag`, PATH instead obtains τ DAG estimates by detecting v-structures $\mathbf{U}^{(t)}$ in each thresholded skeleton $\mathcal{G}^{(t)}$ and

---

Algorithm 2.4 `pHGS(`$\mathcal{D}$`, `α`, `τ`, `$\alpha^{(\tau)}$`)` (sample version)

---

input: data $\mathcal{D}$, threshold α, number of estimates τ, minimum threshold $\alpha^{(\tau)}$
output: DAG $\mathcal{G}$

1: execute `pPC(`$\mathcal{D}, \alpha$`)` (the sample version of Algorithm 2.1), omitting edge orientation in line 19, to obtain Φ and $\mathbf{S}$ as in (2.6)

2: execute $\mathcal{G}^{(t^*)} \leftarrow$ `PATH(`$\Phi, \mathbf{S}, \mathcal{D}, \tau, \alpha^{(\tau)}$`)` (Algorithm 2.2) with the following modifications:

line 5: detect v-structures $\mathbf{U}^{(t)}$ with $\mathbf{S}^{(t)}$ and execute $\mathcal{G}^{(t)} \leftarrow$ `HGI(`$\mathcal{G}^{(t)}, \mathcal{D}, \mathbf{U}^{(t)}$`)` (Algorithm 2.3)

line 6: copy $\tilde{\mathcal{G}}^{(t)} \leftarrow \mathcal{G}^{(t)}$

3: execute hill-climbing (or otherwise greedy search) to obtain $\mathcal{G}$, initialized with the selected estimate $\mathcal{G}^{(t^*)}$ and restricted to $\mathbf{A} = \{(i, j) : \Phi_{ij} \leq \alpha^{(1)}\}$ where $\alpha^{(1)} = \alpha$ is the maximum p-value (see Section 2.2.2)

---

executing HGI (Algorithm 2.3). See Section 1.2.2 for details on v-structure detection. The highest-scoring of the τ estimates $\mathcal{G}^{(t^*)}$ is selected to initialize hill-climbing (or an alternate score-based search algorithm) restricted to the active set $\mathbf{A} = \{(i,j) : \Phi_{ij} \leq \alpha^{(1)}\}$. We choose the maximum threshold $\alpha^{(1)}$ (see Section 2.2.2) for the restriction instead of $\alpha^{(t^*)}$ corresponding to the highest-scoring estimate to reduce false negatives that excessively restrict the score-based exploration in the finite-sample setting.

In the large-sample limit, under the same conditions and parameter specifications as Theorem 3 and Lemma 1, the output of pHGS (Algorithm 2.4) is a DAG that is Markov equivalent to the underlying DAG. Indeed, this result is already achieved by the modified PATH in line 2, and in such a case the subsequent greedy search exists only to verify its optimality.

2.4 Numerical Results

We conducted extensive simulations to demonstrate the merits of pPC, PATH, HGI, and pHGS alongside a number of other popular structure learning algorithms. In addition to considering data simulated from various reference Bayesian network configurations, which we introduce in Section 2.4.1, we analyze real data in the form of a well-known flow cytometry dataset in Section 2.4.4. Additionally, we provide extended simulation results investigating the effect of the clustering quality of pPC on its subsequent structure learning in Section 2.4.5, experiments evaluating the performance of PATH for varying sample sizes in Section 2.4.6, and preliminary simulations evaluating HGI applied to ARGES in the Gaussian setting in Section 2.4.7. Detailed tables with additional metrics and visual comparisons are available in Appendix A.

2.4.1 Simulation Setup

The performance of our methods were primarily evaluated in comparison to several structure learning algorithms on numerous reference Bayesian networks obtained from the Bayesian network repository compiled alongside the `R` package `bnlearn` (Scutari, 2010, 2017). Most available discrete networks were considered, with the MUNIN networks represented by MUNIN1 and the LINK network omitted because certain minuscule marginal probabilities required much larger sample sizes to generate complete sets of valid variables. The following preprocessing procedures were applied to each network. For each random variable X_i, non-informative states x_i with $\Pr(X_i = x_i) = 0$ were removed, and non-informative variables X_i with $|r_i| = 1$ were likewise removed. Furthermore, each variable X_i was restricted to $|r_i| \leq 8$, with the extraneous discrete states of excessively granular variables removed by randomly merging states. The conditional probability distributions imposed by merged states were averaged, weighted according to their marginal probabilities.

In order to demonstrate the effectiveness of our methods for learning large discrete networks, we generated larger versions of each network with a house implementation of the tiling method proposed by Tsamardinos et al. (2006b), modified to approximately preserve the average in-degree amongst non-minimal nodes. In particular, let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be the structure consisting of κ disconnected subgraphs to be connected by tiling. For a minimal node k (that is, k has no parents), instead of probabilistically choosing the number of added interconnecting edges, denoted by e_k, according to $e_k \sim \text{Unif}\{0, d \coloneqq \max_{i \in \mathbf{V}} |\mathbf{Pa}_i^{\mathcal{G}}|\}$, we let $\Pr(e_k = a) = \sum_{i \in \mathbf{V}} \mathbb{1}\left\{|\mathbf{Pa}_i^{\mathcal{G}}| = a\right\} / |\mathbf{V}|$ for $a = 0, \ldots, \min\{d, 4\}$. Note that in this process we did not enforce any block structure on the tiled structures.

The considered networks, along with select descriptive characteristics, are presented in Table 2.1, ordered by increasing complexity. The MIX network consists of the 14 networks from Table 2.1 with the least complexity, tiled in random order. For each network configuration, we generated $N = 100$ datasets with $n = 25000$ data samples each, for a total of

Table 2.1: Networks for data generation consisting of κ connected sub-networks with p nodes, $|\mathbf{E}|$ edges, average number of neighbors $\overline{|\mathbf{N}^{\mathcal{G}}|}$, maximum in-degree $\max_i |\mathbf{Pa}_i^{\mathcal{G}}|$, and $|\boldsymbol{\Theta}|$ number of parameters

	Network	κ	p	$\lvert\mathbf{E}\rvert$	$\overline{\lvert\mathbf{N}^{\mathcal{G}}\rvert}$	$\max_i \lvert\mathbf{Pa}_i^{\mathcal{G}}\rvert$	$\lvert\boldsymbol{\Theta}\rvert$
1	EARTHQUAKE	200	1000	1103	2.206	2	2380
2	CANCER	200	1000	1123	2.246	2	2399
3	ASIA	125	1000	1243	2.486	2	2544
4	SURVEY	167	1002	1347	2.689	2	4444
5	ANDES	5	1115	2245	4.027	6	7082
6	WIN95PTS	14	1064	2184	4.105	7	9525
7	CHILD	50	1000	1309	2.618	2	11649
8	ALARM	28	1036	1689	3.261	4	16574
9	MIX	14	1011	1877	3.713	7	17319
10	SACHS	91	1001	1822	3.640	3	18746
11	PIGS	3	1323	2182	3.299	2	20102
12	HEPAR2	15	1050	2077	3.956	6	23007
13	INSURANCE	38	1026	2120	4.133	3	40918
14	HAILFINDER	18	1008	1541	3.058	4	54322
15	WATER	39	1014	2303	4.542	5	72316
16	MUNIN1	7	1064	1776	3.338	3	83208
17	PATHFINDER	10	1090	1968	3.611	5	96037
18	DIABETES	3	1239	2035	3.285	2	302008
19	MILDEW	29	1015	1913	3.769	3	345575
20	BARLEY	21	1008	2101	4.169	4	1771800

2000 datasets. The p columns of each dataset were randomly permuted so as to obfuscate any information regarding the causal ordering. Note that while only one sample size was considered for all the networks of similar order in p, the networks vary significantly with respect to sparsity, structure, and complexity, thus representing a wide variety of conditions.

Algorithm implementations of competing algorithms MMPC, HPC, HITON, IAMB, MMHC, and H2PC, which we briefly introduce in their respective featuring sections, were obtained from the `R` package `bnlearn`, which is written in `R` with computationally intensive operations delegated to `C` (R Core Team, 2021; Scutari, 2010, 2017). We also compared against GES and FGES from `rcausal`, the `R` package wrapper for the Tetrad Library (Wongchokprasitti, 2019). Our pPC, PATH, HGI, and pHGS implementations were built in `R` and

`Rcpp` using tools from the `bnlearn` package, and the results for PC were obtained by executing pPC restricted to $\kappa = 1$ for fair comparison. We have made our methods publicly available in the form of an `R` package at the following link:

`https://github.com/jirehhuang/phsl`

We evaluate the quality of a graph estimate $\hat{\mathcal{G}} = (\mathbf{V}, \hat{\mathbf{E}})$ with respect to the underlying DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ by considering the Jaccard index (JI) of the CPDAG of $\hat{\mathcal{G}}$ in comparison to the CPDAG of $\mathcal{G}$. JI is a normalized measure of accuracy (higher is better) that is monotonically associated with the F1 score, computed as JI = TP / $(|\mathbf{E}| + |\hat{\mathbf{E}}| - \text{TP})$ where TP is the number of true positive edges: the number of edges in the CPDAG of $\hat{\mathcal{G}}$ that coincide exactly with the CPDAG of $\mathcal{G}$ (both existence and orientation).

We use the JI as our primary accuracy metric over the popular structural Hamming distance (SHD) as we find it to be consistent with SHD (higher JI almost always indicates lower SHD) and for its convenience as a normalized metric. The choice of evaluating the CPDAG estimates rather than DAG estimates is motivated foremost by the fact that given that our estimates are inferred from observational data, the orientation of reversible edges in DAGs provide no meaningful interpretation (see Section 1.1.1). Additionally, the aforementioned metrics allow for evaluation of the quality of estimated PDAGs that do not admit a consistent extension (see Remark 2).

Regarding efficiency, execution time is confounded by factors such as hardware, software platform, and implementation quality. Even if the aforementioned variables are accounted for, performing all simulations on the same device cannot guarantee consistent performance over all simulations and instead severely constrains the feasible scope of study. Thus, we evaluate the estimation speed of structure learning algorithms by the number of statistical calls to conditional independence tests or local score differences, with fewer calls indicating greater efficiency. For pPC, we additionally include mutual information and entropy evaluations to account for the expense of clustering (see Section 2.2.1.1).

Remark 4. In general, we find the relative execution time comparisons to be approximately equivalent to the relative number of statistical calls, within comparable implementations. For example, the relative estimation speed of pPC as compared to PC is approximately equivalently characterized by their relative execution time and relative number of statistical calls. The implementation of hill-climbing, however, does not scale as well as that of pPC, resulting in disproportionately greater relative execution times compared to relative number of statistical calls.

2.4.2 pPC and PATH

As the pPC algorithm can be considered an augmentation of the PC algorithm by imposing an ordering to the conditional independence tests by partitioning, we highlight its performance against the PC algorithm. We additionally apply the PATH augmentation to pPC and PC.

Note that our proposed HGI strategy motivates the design of high-performing constraint-based algorithms that not only efficiently restrict the search space, but also demonstrate potential for good score-based search initialization with HGI by producing structurally accurate estimates. As such, we further validate the performance of pPC and PATH against four other established constraint-based structure learning algorithms, all local discovery methods, modified with a symmetry correction for skeleton learning (Aliferis et al., 2010). Max-min parents and children (MMPC) uses a heuristic that selects variables that maximize a minimum association measure before removing false positives by testing for conditional independence (Tsamardinos et al., 2003a, 2006a). The fast version of the incremental association Markov blanket algorithm (Fast-IAMB; IAMB in this work) is a two-phase algorithm that consists of a speculative stepwise forward variable selection phase designed to reduce the number of conditional independence tests as compared to single forward variable selection, followed by a backward variable pruning phase by testing for conditional independence (Tsamardinos et al., 2003b; Yaramakala and Margaritis, 2005). The semi-interleaved implementation

of HITON[1] parents and children (SI-HITON-PC; HITON in this work) iteratively selects variables based on maximum pairwise marginal association while attempting to eliminate selected variables by testing for conditional independence (Aliferis et al., 2003, 2010). Finally, hybrid parents and children (HPC) is comprised of several subroutines designed to efficiently control the false discovery rate while reducing false negatives by increasing the reliability of the tests (Gasse et al., 2014). For each of these methods, following skeleton estimation, we orient edges by detecting and orienting v-structures according to (1.6) (in Section 1.2.2) and applying Meek's rules R1-4.

The maximum size of considered conditioning sets m was chosen empirically for a balance between efficiency and well-performance: $m = 3$ for pPC and PC, $m = 4$ for HPC, and $m = 5$ for the remaining methods. We note that HPC insignificantly varies in efficiency with m and produces the most accurate estimates in our simulations with $m = 4$, and the remaining competing methods are more efficient but significantly less accurate with $m < 5$. Additionally for pPC, we restricted the maximum set size to 3 for the evaluations in (2.3) and (2.4), and the maximum considered neighborhood size to 5 (see Remark 1). We executed each algorithm with each of the following ten choices of significance level thresholds:

$$\alpha \in \mathcal{A} \coloneqq \{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001, 0.00005, 0.00001, 0.000005\}. \quad (2.12)$$

For each execution of pPC and PC, estimates for $\tau = 10$ thresholding values were automatically generated with PATH (Algorithm 2.2) according to (2.7), restricted to a minimum value of $\alpha^{(\tau)} = 10^{-5}$.

The comparison results for pPC and PATH are reported in Table 2.2. We first compare pPC against PC in terms of computational efficiency and estimation accuracy. Unsurprisingly, pPC demonstrates the greatest computational benefit over PC for large α, typically halving the number of conditional independence tests for $\alpha = 0.1$, as seen from the normal-

[1]From the Greek word "*Χιτον*", pronounced "*hee-tón*", meaning "cover", "cloak", or "blanket".

Table 2.2: Accuracy (JI) and efficiency (Normalized Calls) comparison between pPC and PC, without and with PATH (indicated by None and $\tau = 10$, respectively)

	JI						**Normalized Calls**[a]		
	pPC		PC		pPC*	PC*	PC	pPC*	PC*
	$\alpha = 0.1$		$\alpha = 0.1$		$\alpha \in \mathcal{A}$	$\alpha \in \mathcal{A}$	$\alpha = 0.1$	$\alpha \in \mathcal{A}$	$\alpha \in \mathcal{A}$
	None	$\tau = 10$	None	$\tau = 10$	None	None	$\tau = 10$	None	None
1	0.447	**0.725**	0.390	0.697	0.712	0.707	3.261	8.516	11.384
2	0.243	**0.333**	0.259	0.332	0.292	0.292	7.905	7.264	15.643
3	0.218	**0.284**	0.211	0.275	0.261	0.261	1.435	4.580	5.122
4	0.322	**0.362**	0.309	**0.362**	0.360	0.360	2.821	7.914	10.321
5	0.497	0.560	0.487	**0.561**	0.515	0.505	4.384	7.668	12.707
6	0.403	**0.411**	0.382	0.404	0.408	0.398	2.500	8.358	10.889
7	0.564	**0.579**	0.536	0.562	0.569	0.548	1.586	8.446	9.355
8	0.474	0.478	0.469	**0.480**	0.477	0.473	1.812	8.150	9.845
9	0.517	**0.546**	0.504	0.536	0.519	0.506	1.297	7.456	8.068
10	0.542	**0.551**	0.506	0.522	0.543	0.508	1.896	8.204	9.908
11	0.851	0.852	0.867	**0.871**	0.851	0.867	1.562	8.910	10.641
12	0.204	**0.233**	0.197	0.224	0.204	0.199	2.236	7.675	9.815
13	0.404	**0.424**	0.401	0.419	0.408	0.404	1.900	8.123	9.886
14	0.337	0.360	0.341	**0.361**	0.340	0.346	1.466	8.582	9.591
15	**0.305**	0.298	0.291	0.272	**0.305**	0.291	1.875	8.501	10.223
16	0.086	0.086	0.087	0.087	0.088	**0.090**	2.009	8.573	13.332
17	0.053	0.053	0.053	0.052	**0.054**	0.053	1.266	8.805	9.860
18	0.219	0.219	0.262	0.262	0.239	**0.263**	1.765	8.705	11.573
19	0.324	0.297	0.312	0.274	**0.326**	0.313	1.665	8.676	10.256
20	0.158	0.157	0.149	0.149	**0.168**	0.160	1.684	8.239	9.662

Columns pPC* and PC* provide the highest JI and total statistical calls of executions for all ten $\alpha \in \mathcal{A}$. Rows correspond to the networks in Table 2.1, and best values are provided in boldface

[a]Normalized by pPC-PATH($\alpha = 0.1, \tau = 10$)

ized calls of PC with $\alpha = 0.1$ and PATH ($\tau = 10$) in the table. Note that our partitioning strategy is solely responsible for this computational improvement as parallelization has no bearing on the efficiency metric, and by design pPC can, like PC, further benefit from parallel execution. The reduction suffers from diminishing returns with decreasing α, with an average speed-up of about 22% across the ten α. Notwithstanding, we found pPC and PC,

even without PATH, to generally prefer larger α. In particular, estimates with thresholds $\alpha = 0.1$ and 0.05 respectively produced the best estimates (highest JI) for over 50% and 17% of datasets, resulted in the highest average JI scores for 13 and 3 network configurations, and achieved the highest two JI scores averaged across all datasets. As such, algorithm executions with large significance level thresholds are not unreasonable in practice, which coincides with the general strategy of PATH. We note that while pPC appears to be typically slightly more accurate than PC, we do not find the impprovement to be substantial and primarily assert that pPC performs comparably to PC.

Additionally, we see from Table 2.2 that PATH applied to pPC and PC is able to obtain, from a single execution with $\alpha = 0.1$ and $\tau = 10$, estimates of similar and often superior quality compared to the best estimates without PATH (pPC* and PC*) obtained from ten executions with the various $\alpha \in \mathcal{A}$. Important to note is that the solution path automatically selects an estimate based on a BIC selection strategy restricted to valid CPDAG estimates, if any (see Remark 2), whereas for the multiple executions the maximum JI (as computed with respect to the CPDAG of the underlying DAG) for each dataset was chosen. The BIC selection strategy appears less effective for a couple of networks (e.g., 15 and 19), where on average the original estimates without PATH were more structurally accurate than those chosen from a solution path. One explanation for the worse performance could be the presence of invalid CPDAG estimates. PATH prefers valid estimates, and may prefer lower-scoring valid estimates over more structurally accurate invalid estimates. In the case that all estimates are invalid, the semi-arbitrary DAG extension process can be volatile, resulting in structurally inaccurate estimates being selected. As anticipated in Section 2.2.2, the computational expense required to execute Algorithm 2.2 is practically negligible in comparison to skeleton estimation. The statistical calls for pPC and PC with PATH ($\alpha = 0.1$ and $\tau = 10$) include the scores evaluated for BIC selection from the generated solutions by PATH, and are practically indistinguishable from those without PATH, with the score evaluations typically consisting of less than 0.1% of the statistical calls.

Table 2.3: Accuracy (JI) and efficiency (Normalized Calls) comparison amongst constraint-based methods

	JI					Normalized Calls[a]			
	pPC[b]	MMPC	HPC	IAMB	HITON	MMPC	HPC	IAMB	HITON
1	0.725	0.759	**0.760**	0.749	0.759	33.933	86.837	53.532	17.189
2	0.333	0.757	0.805	**0.870**	0.701	30.377	157.542	39.762	16.831
3	0.284	**0.306**	0.245	0.260	0.299	15.271	47.117	21.783	7.758
4	0.362	0.773	0.818	**0.835**	0.734	33.130	97.750	40.456	20.167
5	0.560	0.563	**0.692**	0.532	0.606	34.973	168.566	33.262	61.176
6	**0.411**	0.295	0.379	0.251	0.243	30.886	101.774	44.803	17.194
7	0.579	0.087	**0.717**	0.317	0.077	29.551	67.678	49.607	15.100
8	**0.478**	0.200	0.434	0.314	0.158	29.016	78.442	59.514	14.939
9	0.546	0.423	**0.633**	0.360	0.316	12.150	53.621	14.422	7.762
10	0.551	0.259	**0.724**	0.352	0.358	27.139	100.016	30.732	14.306
11	0.852	0.536	**0.974**	0.385	0.276	12.454	69.107	12.761	6.200
12	0.233	0.259	**0.399**	0.289	0.262	21.322	82.737	31.368	12.408
13	**0.424**	0.189	0.310	0.252	0.114	24.916	95.332	38.883	12.967
14	**0.360**	0.246	0.325	0.287	0.220	32.241	59.202	40.034	16.646
15	**0.298**	0.283	0.287	0.291	0.232	32.291	88.191	48.761	16.602
16	**0.086**	0.004	0.010	0.032	0.004	11.962	72.271	12.746	6.757
17	0.053	0.067	**0.077**	0.050	0.062	7.282	19.078	10.687	3.737
18	0.219	0.067	**0.221**	0.073	0.061	28.936	74.800	26.148	14.965
19	**0.297**	0.066	0.252	0.232	0.064	32.707	82.580	36.943	16.629
20	0.157	0.055	**0.244**	0.093	0.055	24.945	96.625	22.784	12.777

Rows correspond to the networks in Table 2.1, and the best values are provided in boldface

[a]Normalized with respect to pPC-PATH($\alpha = 0.1, \tau = 10$)

[b]pPC-PATH($\alpha = 0.1, \tau = 10$), while all other methods report the highest JI from and the (normalized) total statistical calls for the executions across the ten $\alpha \in \mathcal{A}$

In Table 2.3, we compare pPC with PATH against other constraint-based structure learning algorithms. We exclude PC as its comparison with pPC is thoroughly demonstrated in Table 2.2. Again, competing methods report optimal results and total statistical calls for the executions across the ten significance levels $\alpha \in \mathcal{A}$. In terms of structural accuracy, the only algorithm that can compete against pPC is HPC, which outperforms pPC in twelve of the network configurations, sometimes by a substantial margin. However, when it comes

to efficiency, there is no contest against the pPC algorithm, in most cases even if the number of calls were averaged across the ten executions instead of summed. Additionally, pPC most often produced valid CPDAG estimates, succeeding with 44% of the datasets with only $\alpha = 0.1$ and $\tau = 10$ in contrast to from 10% by HPC to up to 41% by IAMB with $\alpha \in \mathcal{A}$.

In Appendix A, we provide additional results for the comparisons of pPC against three constraint-based methods, PC, MMPC, and HPC, thus far discussed in Table 2.2 and Table 2.3. We include detailed results with additional metrics as well as standard deviations in Table A.1, and a figure that visualizes the comparisons and variability of the accuracy and efficiency results across executions by including the results for individual datasets for each network (Figure A.1).

2.4.3 HGI and pHGS

Having discussed the theoretical merits of HGI in Section 2.3.2, we demonstrate the empirical performance of HGI and pHGS in this section, beginning with the comparison and application to the GSC framework. We refer to unrestricted hill-climbing as simply HC and perfectly restricted hill-climbing as GSC*. In general, for a restriction of the search space with constraint-based algorithm Alg, we refer to the GSC version as Alg-HC, and the version with HGI as Alg-HGI-HC. However, we refer to the versions of established algorithms MMHC and H2PC that are augmented with HGI as MMHC-HGI and H2PC-HGI.

The hill-climbing phase of each algorithm was augmented with a tabu list to avoid $t_1 = 100$ previously visited DAG structures for $t_0 = 100$ suboptimal iterations (see Section 2.3.1). All score-based methods evaluated structures with the BIC score. We executed pHGS with significance level $\alpha = 0.01$, and generated and selected from $\tau = 10$ HGI estimates in PATH by thresholding to a minimum of $\alpha^{(\tau)} = 10^{-5}$.

The accuracy results for HGI and pHGS are summarized in Table 2.4. All methods other than HC, GSC*, and pHGS report the highest JI for the executions across the ten $\alpha \in \mathcal{A}$.

Table 2.4: Accuracy (JI) comparisons for CPDAGs estimated without and with HGI (indicated by EG for empty graph and HGI, respectively) given restrictions obtained by various skeleton methods

Restrict	None	True Skeleton		pPC		MMPC		HPC	
Initial	EG	EG	HGI	EG	HGI	EG	HGI	EG	HGI
Alias	HC	GSC*			pHGS[a]	MMHC		H2PC	
1	0.341	0.478	0.806	0.501	**0.746**	0.500	0.761	0.548	0.762
2	0.419	0.596	0.954	0.577	**0.729**	0.570	0.874	0.590	0.903
3	**0.323**	0.504	0.802	0.255	0.282	0.254	0.302	0.224	0.255
4	0.649	0.770	0.931	0.751	**0.786**	0.726	0.819	0.760	0.887
5	0.610	0.910	0.970	0.715	**0.746**	0.544	0.598	0.686	0.738
6	0.295	0.564	0.736	0.414	**0.467**	0.254	0.306	0.330	0.403
7	0.450	0.535	0.995	0.528	**0.851**	0.191	0.199	0.532	0.854
8	0.361	0.567	0.889	0.411	**0.515**	0.288	0.329	0.493	0.636
9	0.630	0.823	0.902	0.686	**0.726**	0.388	0.441	0.695	0.738
10	0.320	0.417	1.000	0.413	**0.807**	0.276	0.308	0.418	0.921
11	0.827	1.000	1.000	0.992	**0.991**	0.450	0.535	0.990	0.991
12	0.529	0.634	0.763	0.513	0.552	0.247	0.294	**0.585**	0.622
13	0.368	0.513	0.811	0.417	**0.610**	0.212	0.227	0.404	0.423
14	0.456	0.730	0.920	0.387	**0.449**	0.279	0.315	0.455	0.488
15	0.234	0.361	0.581	0.293	**0.399**	0.268	0.366	0.328	0.419
16	**0.261**	0.576	0.655	0.045	0.056	0.007	0.007	0.096	0.102
17	**0.358**	0.407	0.478	0.070	0.072	0.081	0.081	0.345	0.336
18	0.222	0.618	0.946	0.164	0.197	0.063	0.069	**0.241**	0.304
19	0.375	0.549	0.690	0.276	0.320	0.106	0.114	**0.440**	0.481
20	0.260	0.524	0.607	0.144	0.165	0.068	0.080	**0.289**	0.327

Rows correspond to the networks in Table 2.1, and best values amongst pHGS, HC, MMHC, and H2PC are provided in boldface

[a]pHGS($\alpha = 0.01$, $\tau = 10$); all other methods, excluding HC and GSC*, report the highest JI for the executions across the ten $\alpha \in \mathcal{A}$

In the first column, the performance of (unrestricted) HC leaves much to be desired with its generally lackluster structural accuracy in comparison with the hybrid methods. Exceptions exist, as anticipated by Table 2.3 in which constraint-based methods struggle to produce good estimates for some higher complexity networks, often inferring excessive false negatives that erroneously reduce the search space. In such cases, hybrid approaches are limited by

their constraint-based component and thus perform worse than HC.

We first highlight the demonstrated improvement of initialization with HGI compared to the empty graph (EG) (as in the GSC framework) for different skeleton restriction methods. A perfect restriction to the true skeleton represents the most optimistic scenario for GSC and HGI, wherein all true positives are accessible and no false positives are considered. HGI additionally enjoys consideration of all true v-structures when detecting v-structures amongst unshielded triples according to (1.6). Unsurprisingly, perfectly restricted GSC uniformly improves on the performance of unrestricted HC. The addition of HGI achieves further improvements to structural accuracy of typically 28% and up to 139% (Table 2.4, True Skeleton). This same trend persists when comparing GSC without and with HGI for empirical skeleton estimation methods pPC, MMPC, and HPC, demonstrating the effectiveness of HGI. For the various skeleton estimation methods, the addition of HGI achieves estimates that are typically 14% and up to 120% more structurally accurate.

We now compare pHGS with established algorithms HC, MMHC, and H2PC, the best values amongst which are indicated in boldface in Table 2.4. In most of the network configurations, a single execution of pHGS learns estimates of higher quality than the best of the ten executions with $\alpha \in \mathcal{A}$ of MMHC and H2PC. Note that for the GSC framework, the goal of parameter tuning for α is to obtain a balance between true positives and true negatives. MMHC does not outperform pHGS in any meaningful capacity, and H2PC only substantially outperforms pHGS for higher complexity networks due to a mechanism in HPC to reduce false negative edges (Gasse et al., 2014). While the addition of HGI dramatically improves the general accuracy of MMHC, only H2PC-HGI performs competitively with pHGS, reflective of the results in Table 2.3 where HPC rivaled pPC in the realm of structural accuracy. However, as we will see from our discussion of Table 2.5, the speed comparisons in Table 2.3 generally hold for these hybrid variants as well, with pHGS on average nearly an order of magnitude more efficient than H2PC and around 2.5 times more efficient than MMHC per execution, with or without HGI. HC only outperforms the hybrid methods for a few struc-

tures in which the latter overly restrict the search space. Overall, we find pHGS to be most well-performing method, followed by H2PC, MMHC, and HC. We provide detailed results for these methods in Appendix A, with Figure A.2 visualizing the accuracy comparisons across datasets for each network as well as Table A.2 with additional metrics.

In Table 2.5, we compare the computational expense of GSC and HGI-HC with respect to skeleton learning. Under HGI-HC, the Detect column refers to the computational expense of detecting v-structures necessary to execute HGI (Algorithm 2.3) (see Section 1.2.2). While HGI-HC typically comes at greater computational cost compared to the HC step in GSC, the expense of either is largely negligible in comparison to that of skeleton learning, generally (and often significantly) fewer than an additional 2% statistical calls. Rare exceptions exist, in particular extreme cases where MMPC or HPC required a significant number of additional tests to detect v-structures. Here, pPC has a clear computational advantage, having the ability to detect v-structures using separation sets accrued throughout skeleton learning (see (1.5) in Section 1.2.2), resulting in fewer than 4.8% additional calls for every dataset to execute HGI $\tau = 10$ times in PATH and perform hill-climbing from the chosen initial DAG. Other algorithms must conduct additional conditional independence tests to detect v-structures via (1.6), which can quickly add up if the learned skeleton structure has a significant number of unshielded triples $i - k - j$, or if either or both of $|\mathbf{N}_i^{\mathcal{G}}|$ and $|\mathbf{N}_j^{\mathcal{G}}|$ are

Table 2.5: Median and 95% precision intervals of percent additional statistical calls by GSC and HGI-HC with respect to skeleton learning

	GSC	HGI-HC		
	HC	Detect	HGI	HC
pPC[a]	1.00 (0.04, 1.88)	0.00 (0, 0)	1.12 (0.05, 3.87)	0.52 (0.03, 1.21)
MMPC[b]	0.13 (0.01, 0.28)	0.90 (0.02, 51.64)	0.02 (0, 0.06)	0.06 (0.01, 0.16)
HPC[b]	0.22 (0.02, 0.41)	0.92 (0, 32.56)	0.04 (0.01, 0.08)	0.12 (0.02, 0.21)

Each data point represents one algorithm execution for each dataset

[a]pPC under HGI-HC represents pHGS executed with $\tau = 10$ and $\alpha = 0.01$

[b]MMPC and HPC include the results from every individual execution across the ten $\alpha \in \mathcal{A}$

large. On the topic of efficiency, unrestricted HC typically requires three to five times the number of statistical calls to execute as compared to pHGS, providing further validation for the hybrid approach.

In general, we find the initial DAG obtained by HGI (Algorithm 2.3) through the greedy application of v-structures and greedy decomposed `pdag-to-dag` to be typically superior in structural accuracy compared to the direct application of `skel-to-cpdag` (Algorithm 1.2), the standard edge orientation strategy of constraint-based algorithms. HGI exhibits the greatest median improvement of 16.7% over `skel-to-cpdag` when applied to pPC, followed by 9.5% with HPC, 7.3% restricted to the skeleton of the underlying DAG (True Skeleton), and 1.9% with MMPC. In general, pPC detects the most v-structures as its detection criterion (1.5) may be considered less strict than (1.6) used by True Skeleton, MMPC, and HPC. Consequently, pPC generally detects a significant number of false positive v-structures, thus benefiting most significantly from the greedy v-structure determinations. The poor skeleton estimation performance of MMPC is likely responsible for its lackluster improvement, with its estimated skeletons generally containing the fewest unshielded triples corresponding to true v-structures in the underlying DAGs in comparison to pPC and HPC.

Next, we compare pHGS against GES, introduced in Section 2.3.1, along with one of its variants, which serve well as points of reference for evaluating pHGS as they share comparable theoretical guarantees in our investigative setting, being sound and complete for Bayesian network structure learning (Meek, 1997; Chickering, 2002a,b). Ramsey (2015) proposed the fast greedy (equivalence) search algorithm (in this work, FGES), which featured a number of optimizations and adjustments to scale up GES to handle large networks. In addition to techniques such as parallelization and addressing redundant scoring, perhaps the most prominent improvement was that of blacklisting zero correlation edges when computing initial scores for the addition of edges to the empty graph. This drastically reduces the number of computations of local score differences. A highly optimized implementation of FGES is available in the `R` package `rcausal`, and we roughly recover an efficient implementation of

the original GES algorithm by omitting the blacklisting step (Wongchokprasitti, 2019).

In the detailed results in Table 2.6, in addition to the Jaccard index (JI) and $\log_{10}$ number of statistical calls (Calls), we report the number of predicted edges (P), true positives (TP), edges correctly present but with incorrect orientation (R), false positives (FP), and the structural Hamming distance (SHD), which may be interpreted as the number of edge additions, deletions, and reversals required to traverse from one graph to another.

For most networks, pHGS executes significantly faster than GES, with a few exceptions, most notably in networks with higher complexity. With respect to FGES, we find pHGS to be approximately equivalently efficient in most cases, though considerably slower in some. The initial blacklisting feature in FGES functions similarly to that of (2.2) in the pPC algorithm, drastically reducing the number of considered edge connections. Furthermore, (F)GES conducts its greedy search in the reduced space of equivalence classes represented by CPDAGs rather than DAGs.

In terms of structural accuracy, pHGS performs competitively against GES and FGES. Consistent with the observations of Ramsey (2015), we find that the significant computational reductions achieved by FGES generally do not compromise the quality of learned structures compared to GES. The results for FGES are quite similar to that of GES for all metrics apart from number of statistical calls, though FGES does appear to obtain marginally sparser networks. As a hybrid algorithm that estimates the skeleton with a constraint-based approach, pHGS is much more restrictive in determining edge connections, resulting in significantly fewer true positives than (F)GES in some networks. As discussed previously in this section, in these scenarios, the pPC component excessively inferred false negatives that erroneously reduced the set of candidate edges. On the other hand, however, pHGS produced substantially fewer false positives than (F)GES for not only these networks but indeed in every network configuration. In such a way, pHGS appears more appropriate in applications where conservative estimation of edge connections is desired.

Table 2.6: Results comparing pHGS against GES and FGES

	pHGS	GES	FGES	pHGS	GES	FGES	pHGS	GES	FGES
	EARTHQUAKE			CANCER			ASIA		
P	911.3	1099.5	1082.0	1124.1	1203.3	1179.2	876.3	1199.7	1176.4
TP	**860.7**	680.3	686.8	947.3	**949.5**	947.0	466.3	**947.9**	941.1
R	**21.9**	227.5	220.1	**126.5**	135.0	131.3	383.7	177.4	**175.6**
FP	**28.7**	191.7	175.1	**50.2**	118.8	101.0	**26.3**	74.4	59.7
SHD	**271.1**	614.4	591.2	**225.9**	292.4	276.9	803.1	369.6	**361.7**
JI	**0.746**	0.447	0.459	**0.729**	0.690	0.699	0.282	0.634	**0.637**
Calls	**5.724**	6.395	5.735	5.739	6.326	**5.724**	**5.745**	6.374	5.751
	SURVEY			ANDES			WIN95PTS		
P	1329.2	1431.0	1418.7	1853.6	2398.4	2341.8	1308.6	2023.0	1973.0
TP	**1177.5**	992.4	987.3	1751.8	**2009.2**	1934.6	1111.9	**1403.3**	1366.5
R	**145.2**	340.5	340.6	**57.8**	83.8	94.3	**158.2**	225.3	224.1
FP	**6.5**	98.2	90.7	**44.0**	305.4	312.9	**38.5**	394.4	382.4
SHD	**175.9**	452.8	450.4	**537.2**	541.1	623.4	**1110.6**	1175.1	1199.9
JI	**0.786**	0.556	0.556	0.746	**0.763**	0.730	0.467	**0.501**	0.490
Calls	5.742	6.237	**5.722**	5.924	6.438	**5.918**	**5.864**	6.327	5.893
	CHILD			ALARM			MIX		
P	1298.8	1312.0	1304.6	1282.9	1649.4	1631.2	1547.8	1850.7	1829.6
TP	1198.1	**1239.0**	1200.5	1010.1	**1322.2**	1313.2	1440.1	**1592.2**	1567.6
R	100.7	**66.7**	99.4	269.9	203.8	**203.5**	**94.2**	126.0	130.2
FP	**0.1**	6.3	4.7	**3.0**	123.4	114.5	**13.5**	132.5	131.8
SHD	111.0	**76.3**	113.2	681.9	**490.2**	490.3	450.4	**417.4**	441.3
JI	0.851	**0.897**	0.853	0.515	**0.656**	0.654	0.726	**0.746**	0.733
Calls	5.798	6.277	**5.785**	**5.837**	6.306	5.862	6.223	6.317	**5.907**
	SACHS			PIGS			HEPAR2		
P	1797.5	1864.4	1862.3	2170.6	2280.2	2280.6	1424.5	1745.2	1685.5
TP	**1615.7**	1544.3	1537.8	**2166.0**	2124.7	2122.2	1245.9	**1375.9**	1325.1
R	**181.9**	264.3	270.0	**4.6**	49.7	51.2	**158.5**	239.2	243.1
FP	**0.0**	55.8	54.4	**0.0**	105.7	107.2	**20.1**	130.1	117.3
SHD	**206.3**	333.6	338.6	**16.0**	163.0	167.0	851.2	**831.3**	869.2
JI	**0.807**	0.722	0.717	**0.991**	0.909	0.907	0.552	**0.563**	0.544
Calls	**5.847**	6.451	5.855	6.523	6.441	**6.139**	5.978	6.317	**5.822**

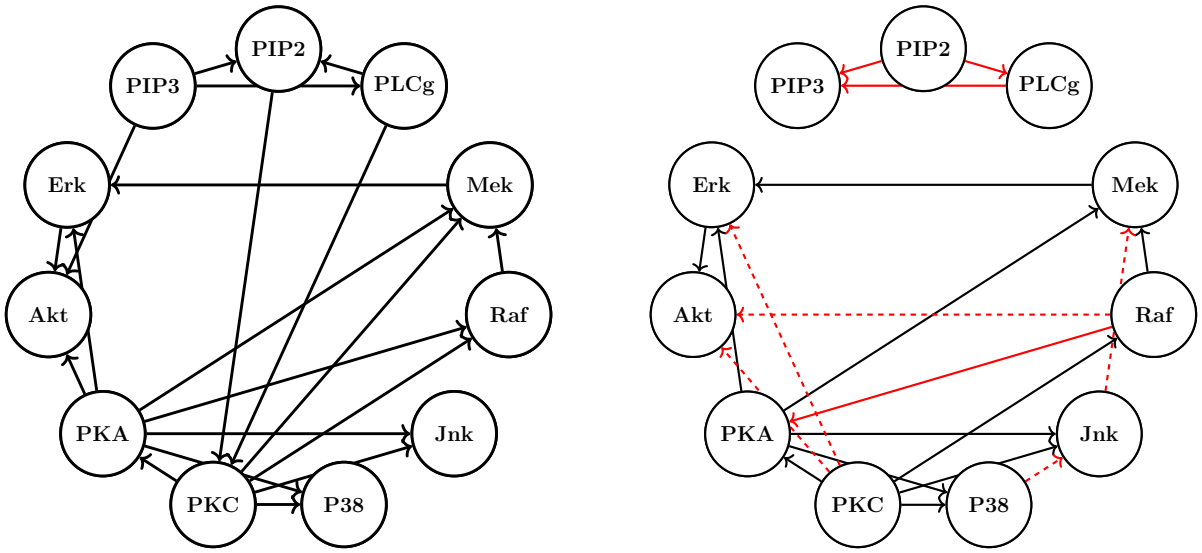
Table 2.6: (continued)

	INSURANCE			HAILFINDER			WATER		
P	1512.0	1899.5	1848.4	986.8	1493.8	1480.8	1321.1	1672.4	1642.3
TP	**1375.2**	1198.9	1178.4	783.3	1191.0	**1192.9**	1032.5	**1048.8**	1033.8
R	**133.8**	479.0	479.5	**117.1**	143.6	135.1	**288.3**	427.3	427.5
FP	**3.0**	221.6	190.4	**86.4**	159.3	152.8	**0.3**	196.3	181.0
SHD	**747.8**	1142.8	1131.9	844.1	509.3	**500.9**	**1270.7**	1450.5	1450.2
JI	**0.610**	0.425	0.422	0.449	0.646	**0.653**	**0.399**	0.358	0.355
Calls	5.910	6.368	**5.868**	**5.790**	6.267	5.819	**5.779**	6.250	5.783
	MUNIN1			PATHFINDER			DIABETES		
P	518.2	1629.6	1614.4	290.6	1458.5	1458.5	1053.3	1988.2	1972.2
TP	120.8	890.5	**891.3**	151.6	**1017.9**	1006.9	508.6	1303.6	**1305.0**
R	364.2	**215.7**	216.0	**135.6**	262.0	274.5	523.0	399.1	**395.2**
FP	**33.3**	523.4	507.1	**3.4**	178.7	177.1	**21.7**	285.5	272.0
SHD	1688.5	1408.8	**1391.8**	1819.8	**1128.8**	1138.2	1548.1	1016.8	**1002.0**
JI	0.056	0.354	**0.357**	0.072	**0.423**	0.417	0.197	0.480	**0.483**
Calls	6.437	6.423	**6.194**	6.531	6.560	**6.328**	**6.040**	6.496	6.043
	MILDEW			BARLEY					
P	1094.8	1526.1	1502.7	758.6	1481.1	1462.7			
TP	729.1	**1241.5**	1232.8	404.6	**1120.8**	1117.6			
R	365.4	178.2	**178.0**	332.6	**212.9**	213.6			
FP	**0.2**	106.4	91.9	**21.3**	147.4	131.5			
SHD	1184.2	777.9	**772.1**	1717.7	1127.7	**1114.9**			
JI	0.320	**0.565**	**0.565**	0.165	0.455	**0.457**			
Calls	5.772	6.138	**5.752**	5.869	6.171	**5.782**			

Calls reports $\log_{10}$(calls)

2.4.4 Real Data Application

To evaluate our methods applied to real data, we analyzed a flow cytometry dataset generated by Sachs et al. (2005) through a series of experiments measuring $p = 11$ phosphorylated proteins and phospholipids. In particular, we considered their preprocessed dataset containing $n = 5400$ observations discretized to low, medium or high levels of the phosphorylated components, available in the `R` package `sparsebn` (Aragam et al., 2019). The structure of the currently accepted signalling network, which we refer to as the consensus network, is



(a) Consensus network **(b)** Output of pHGS

Figure 2.3: Comparison of **(a)** the consensus network and **(b)** the DAG estimated by pHGS. In **(b)**, edges correctly present in the consensus graph are drawn in solid black, edges with incorrect orientation in solid red, and false positive edges in dashed red

shown in Figure 2.3a, and contains 20 directed edges.

We executed pHGS, MMHC, H2PC, HC, and FGES on this dataset, reporting the structure learning performance with respect to the consensus DAG. Since the dataset contains a mixture of observational and interventional data, we directly evaluate the DAG estimates against the consensus network instead of comparing their respective CPDAGs. For FGES, which outputs a CPDAG representing an equivalence class, we obtained a DAG extension by applying lines 5-8 of Algorithm 2.3 to evaluate the structure learning accuracy of FGES, which we note is significantly more favorable for FGES than comparing its CPDAG to the CPDAG of the consensus network. The hybrid algorithms pHGS, MMHC, and H2PC were executed with $\alpha \in \mathcal{A}$ in (2.12), with other parameters unchanged from Section 2.4.2 with the exception of $\alpha^{(\tau)} = 10^{-6}$ to ensure $\alpha^{(\tau)} < \alpha$ for all executions of pHGS. For these methods, we present the results for the estimates closest in sparsity to the consensus graph (20 edges), which coincide with those with the highest JI. The results are reported in Table 2.7, and the

Table 2.7: Structure learning performance on discretized cytometry data

	pHGS	MMHC	H2PC	HC	FGES
P	20	16	20	22	22
TP	**11**	8	8	7	6
R	**4**	**4**	7	7	7
FP	5	**4**	5	8	9
SHD (DAG)	**14**	16	17	21	23
SHD (skeleton)	**10**	12	**10**	14	16
JI	**0.379**	0.286	0.250	0.200	0.167

pHGS estimate is shown in Figure 2.3b.

By a substantial margin, pHGS produced the most structurally accurate estimate with respect to the consensus network, recovering 11 out of the 20 well-established causal relationships. The score-based algorithms FGES and HC obtained the densest networks as they do not conduct any restriction of the search space. In contrast, MMHC estimated the sparsest graph and indeed estimated the exact same graph for all ten $\alpha \in \mathcal{A}$, obtaining different restrictions of the search space via MMPC yet arriving at the same network structure in the restricted hill-climbing step.

2.4.5 Effect of pPC Clustering Quality

The pPC algorithm endeavors to discover some block dependence structure amongst variables by way of clustering in order to impose a partitioned ordering on the conditional independence investigation. As such, it is of interest to investigate the sensitivity of the pPC algorithm to the clustering quality in terms of both accuracy and efficiency. We conducted a simulation study, varying the partition by randomly permuting a proportion of group labels obtained by the clustering algorithm. By such a design, the group sizes are still determined by the clustering algorithm, but some groups may be perturbed. A permuted proportion of 0 corresponds to the default pPC algorithm, and 1 corresponds to randomly rearranging all group labels in the partition. The simulation was executed with

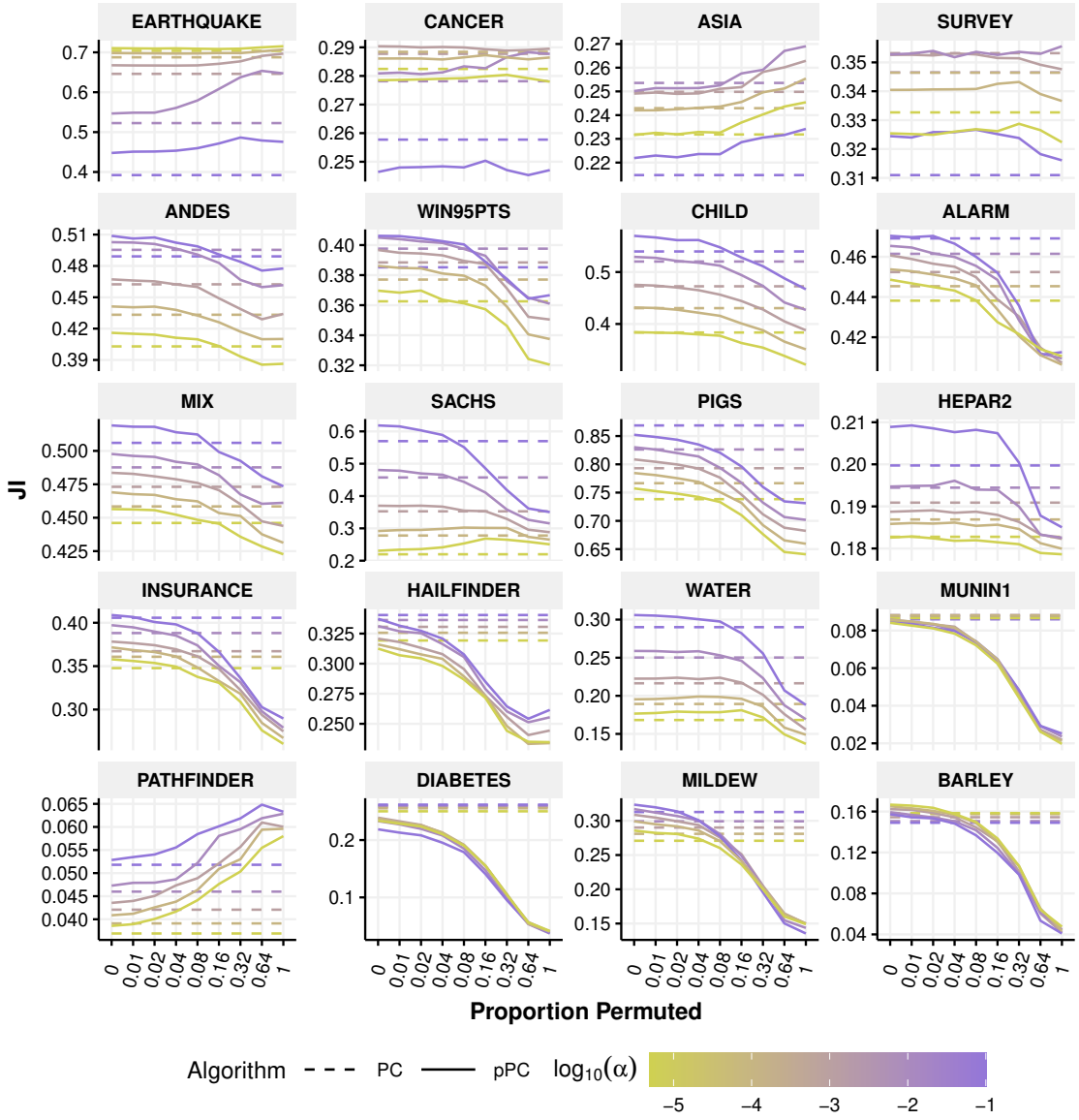


Figure 2.4: Accuracy (JI) results for the pPC algorithm across a varying proportion of the partition randomly permuted. Five α were investigated, and for reference, the dashed lines represent the results for the PC algorithm

$\alpha \in \{0.1, 0.01, 0.001, 0.0001, 0.000005\}$ on a limited set of 20 datasets for each of the 20 networks. The accuracy results are visualized in Figure 2.4.

For the majority of networks, the accuracy of estimated structures drops sharply following permuted proportions of greater than 0.08, unsurprisingly. Several exceptions exist,

with CANCER and SURVEY appearing to be largely agnostic to the clustering quality, and EARTHQUAKE, ASIA, and PATHFINDER even preferring randomly rearranged partitions for some settings. Generally, the performance of pPC only drops below that of PC (represented by the dashed lines) when the permuted proportion is 0.16 or greater, if at all. This

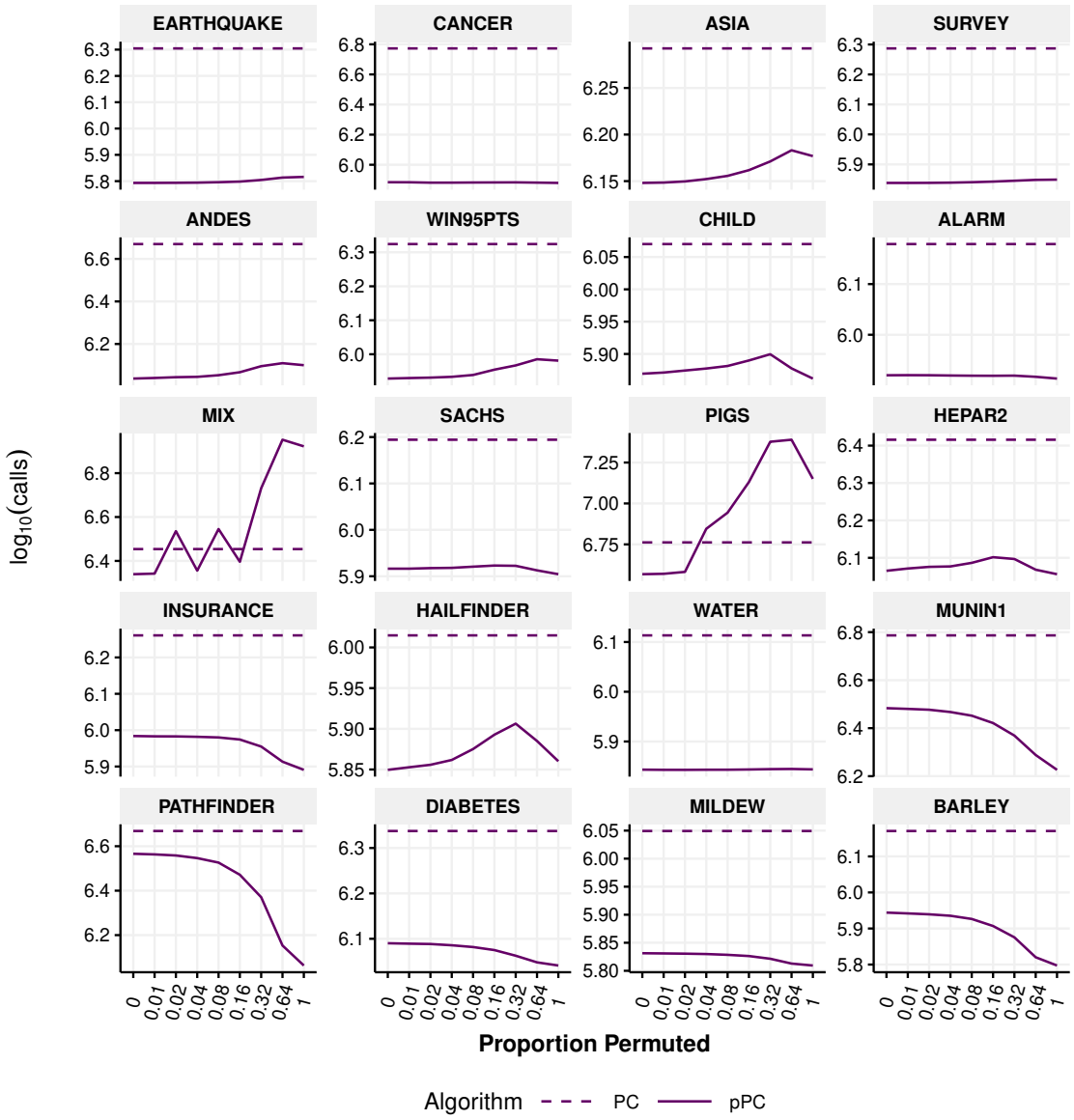


Figure 2.5: Efficiency ($\log_{10}$(calls)) results for the pPC algorithm with $\alpha = 0.1$ across a varying proportion of the partition randomly permuted. For reference, the dashed lines represent the PC algorithm

result shows that pPC is quite robust against varying clustering quality.

The corresponding computational efficiency results for $\alpha = 0.1$ are visualized in Figure 2.5. Again, for the majority of networks, the clustering quality does not appear to significantly influence the computational expense of pPC relative to that of PC. Notable exceptions are the PIGS network which becomes significantly slower than PC and the MIX network which contains the PIGS network. Many high complexity networks even exhibit varying degrees of increased computational efficiency with higher permute proportions, likely due to increased false negatives. Regardless, the effect of clustering quality on the computational expense of pPC appears to vary according to the particular Bayesian network model.

Note that while there may be a *correct* partition according to the tiling of smaller Bayesian networks to create a large network (as discussed in Section 2.4.1), that is not necessarily the optimal clustering, and multiple partitions that result in well-performance may exist.

2.4.6 Extended Empirical Validation of PATH

In this section, we provide extended results supporting the performance of PATH compared to parameter tuning. In Section 2.4.2, we demonstrated that for $n = 25000$ data samples, pPC and PC with PATH using parameters $\alpha = 0.1$, $\tau = 10$, and $\alpha^{(\tau)} = 10^{-5}$ are able to obtain estimates of competitive quality compared to optimal parameter tuning, in particular the best estimates from multiple executions with various α. Here, we discuss an extended simulation study that we conducted, wherein we generated 20 datasets at sample sizes

$$n \in \{1000, 2500, 5000, 10000, 25000, 50000\}$$

for 16 out of the 20 networks in Table 2.1. The networks WIN95PTS, INSURANCE, PATHFINDER, and BARLEY were excluded due to minute marginal probabilities that made it difficult to generate datasets with at least two unique values for each variable at smaller sample sizes.

The results for pPC and PC are reported in figures Figure 2.6 and Figure 2.7, respectively. For parameter tuning, we executed algorithms pPC and PC on these datasets with the same ten α specified in (2.12), and we report the highest JI obtained for each dataset across the parameters as (p)PC* (JI). For PATH, we executed pPC and PC with $\tau = 10$ and $\alpha^{(\tau)} = 10^{-5}$, and we report the JI of the estimate selected by PATH as (p)PC-PATH (BIC)

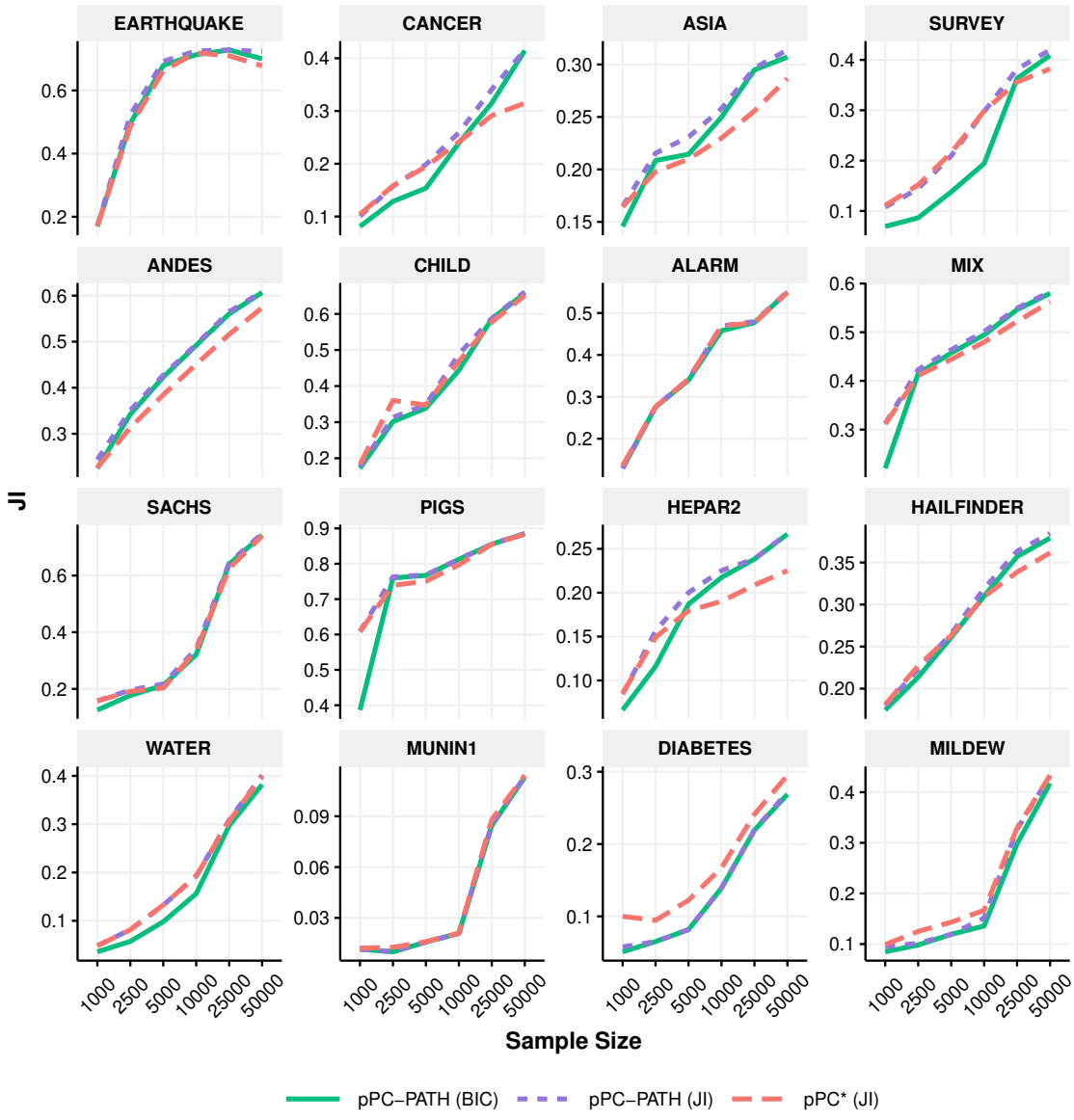


Figure 2.6: Accuracy (JI) of PATH compared to parameter tuning applied to the pPC algorithm with varying sample sizes

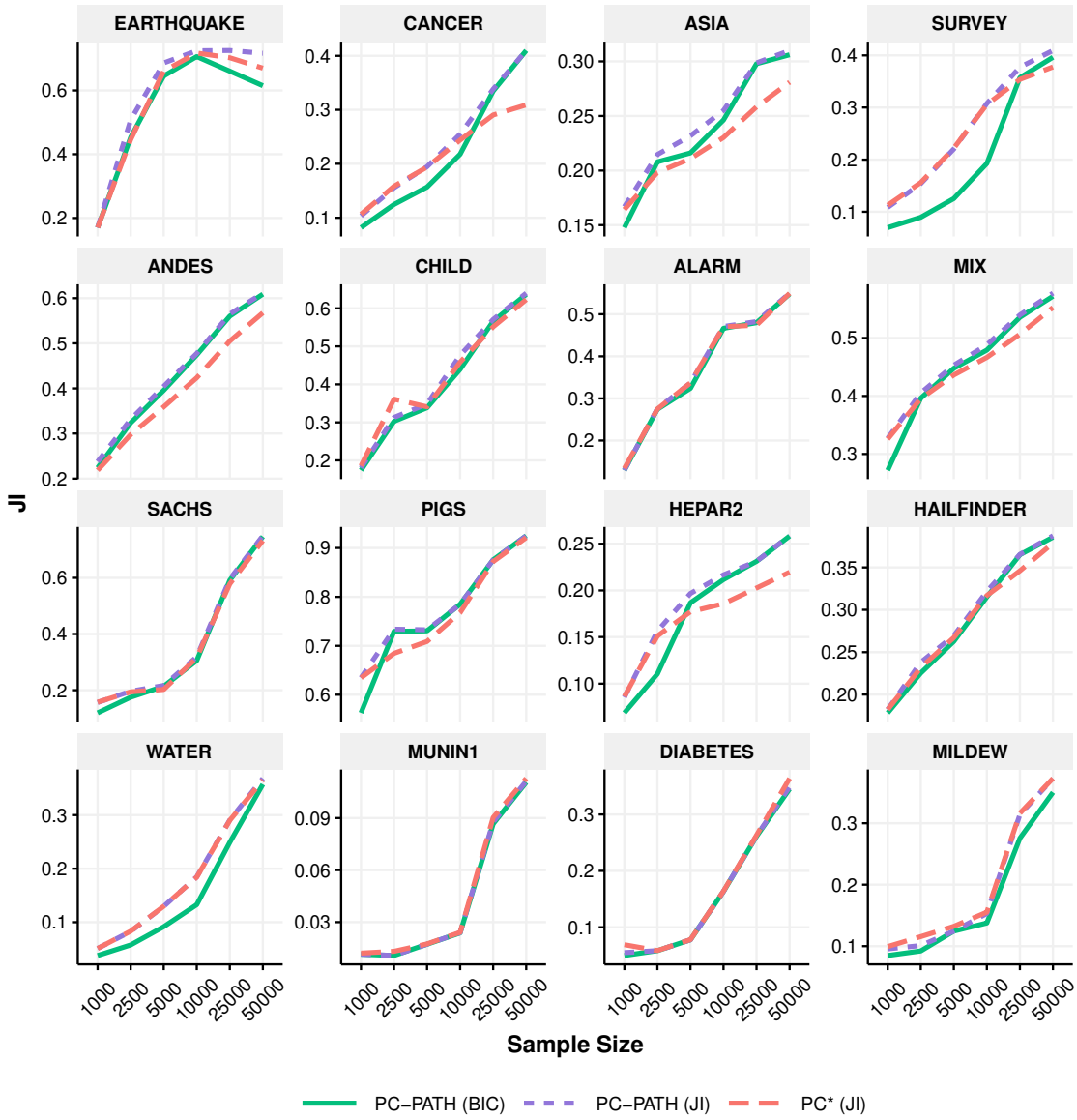


Figure 2.7: Accuracy (JI) of PATH compared to parameter tuning applied to the PC algorithm with varying sample sizes

as well as the highest JI out of all estimates in the solution path generated by PATH as (p)PC-PATH (JI).

For almost every case, (p)PC-PATH (JI) performs at least as well as (p)PC* (JI), demonstrating that the solution path generated by PATH contains highly accurate structures. (p)PC-PATH (BIC) performs approximately equivalently but sometimes slightly worse than

(p)PC* (JI), which is expected behavior due to the dependence on BIC selection. Note also that BIC selection in PATH selects from valid CPDAG estimates only if any are available, while (p)PC* (JI) chooses the highest JI regardless of whether the estimates are valid CPDAGs or not (see Remark 2 and Section 2.4.2).

2.4.7 Gaussian HGI Results

In this section, we report and briefly discuss some preliminary results on the application of HGI to ARGES in the Gaussian setting. Adaptively restricted greedy equivalence search (ARGES) is a hybrid adaptation of greedy equivalence search (GES) that achieves asymptotic recovery of the CPDAG of the underlying DAG given either a consistent estimation algorithm for the skeleton or the conditional independence graph (CIG), the two variants respectively referred to as ARGES-skeleton and ARGES-CIG (Nandy et al., 2018b). The CIG of the joint distribution of variables $\mathbf{X} = \{X_1, \ldots, X_p\}$ represented by nodes $\mathbf{V} = \{1, \ldots, p\}$ is the undirected graph where i and j are adjacent if and only if $(X_i \not\perp\!\!\!\perp X_j \,|\, \mathbf{X} \setminus \{X_i, X_j\})_P$, and is a supergraph of the skeleton.

For our simulations, we again considered the network structures in Table 2.1. For each DAG structure, the edge weights β_{ij} were sampled uniformly from $[-1, -0.1] \cup [0.1, 1]$ for $i \to j \in \mathbf{E}$, with $\beta_{ij} = 0$ otherwise. Similarly, the zero mean error variances σ_i^2, $i = 1, \ldots, p$ were drawn uniformly from $[1, 2]$. For each network configuration, we generated $N = 10$ datasets with sample sizes $n \in \{100, 250, 500, 1000, 2500, 5000\}$. For each dataset and algorithm, we applied the algorithm with ten score penalties $\lambda \in \mathcal{L} \coloneqq \{0.5, 0.7, \ldots, 2.3\}$ (BIC penalty being $\lambda = 0.5$). Note that due to the restriction to the true skeleton, λ functions largely to adjust the search traversal through the space of equivalence classes rather than control the sparsity. The detection of v-structures for HGI was accomplished with a single significance level threshold $\alpha = 0.01$.

The results are summarized in Figure 2.8. For ARGES without HGI, we report the highest JI of ARGES-skeleton and ARGES-CIG. For ARGES with HGI, we report the JI obtained

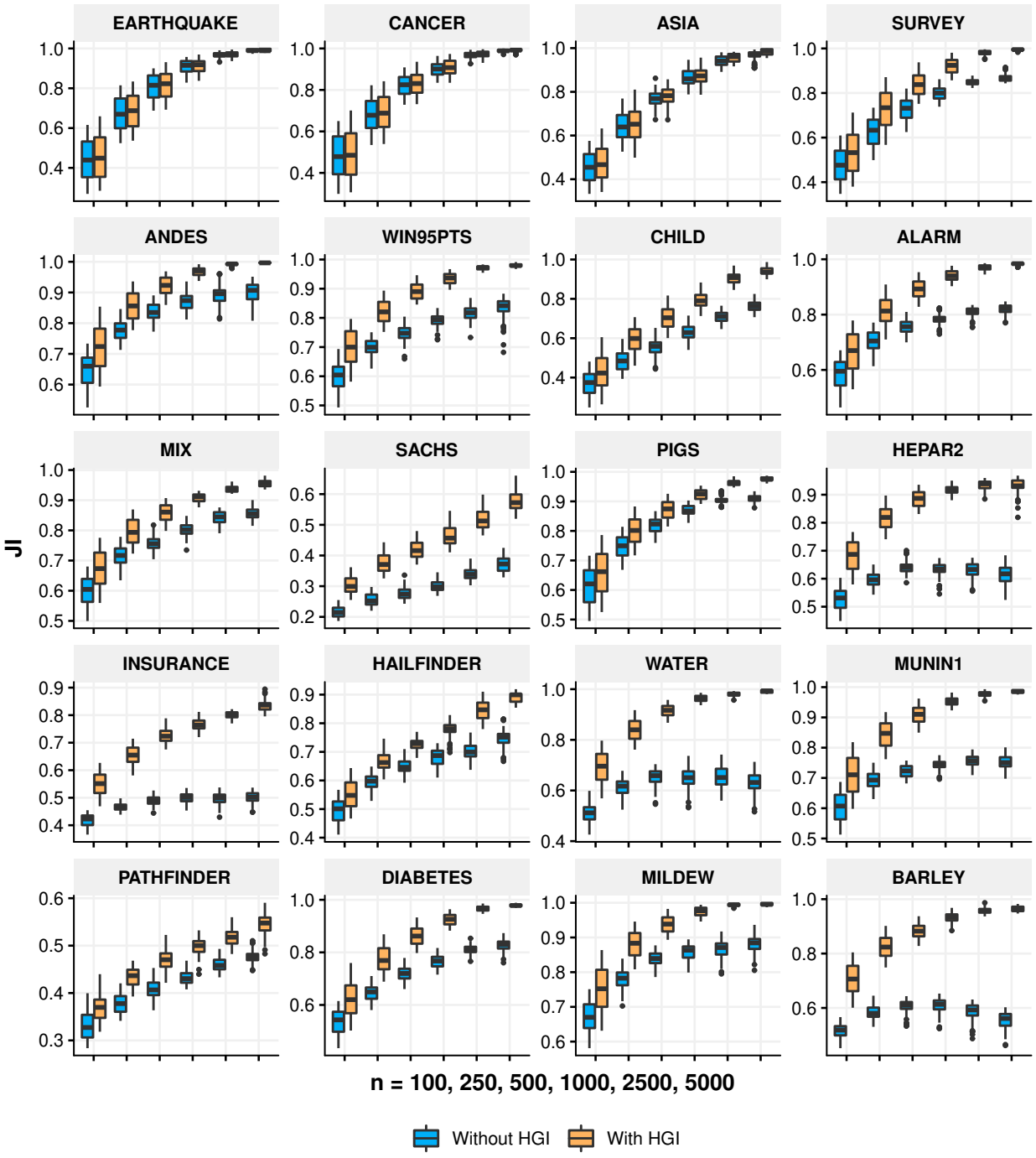


Figure 2.8: Boxplots comparing the JI of ARGES without and with HGI for various networks and sample sizes, restricted to the skeleton or CIG of the underlying DAG

by executing ARGES-skeleton initialized with the CPDAG of the initial DAG obtained by HGI. Both ARGES and ARGES with HGI achieve greater structural accuracy as n increases, though ARGES with HGI is generally more accurate and approaches JI = 1 more quickly. The addition of HGI consistently results in typically 14-18% improvement in JI, typically slightly increasing with sample size. ARGES with HGI only obtained worse JI results than ARGES for 2.7% of the executions, with fewer than 0.4% executions more than 4% worse.

2.5 Conclusion

In this chapter, we developed three independent yet compatible contributions to the general well-performance of discrete Bayesian network structure learning, culminating in the form of the pHGS algorithm.

First, the pPC algorithm improves on the empirical efficiency of the PC algorithm while retaining its soundness and completeness as well as its empirical performance. Though it is difficult to quantify the expected computational reduction, our simulation results in Table 2.2 indicate that for the empirically preferred significance level threshold, pPC typically requires less than half the number of conditional independence tests per execution compared to PC. This speed-up is enjoyed at no compromise to structural accuracy, with pPC performing comparably with PC. Second, the PATH algorithm effectively accomplishes the task of parameter tuning for certain constraint-based structure learning algorithms such as pPC and PC, theoretically preserving consistency in the classical setting and empirically achieving highly competitive structural accuracy at negligible computational expense. In the current landscape, the asymptotic result for sound and complete constraint-based structure learning asserts the existence of some uninformative sequence of significance levels $\alpha_n \to 0$ as $n \to \infty$ that recovers the underlying equivalence class in the large-sample limit. We prove that appropriately applied to pPC or PC executed with any fixed threshold $\alpha \in (0, 1)$, PATH asymptotically includes and correctly selects the underlying CPDAG in its generated

solution path. We demonstrate an analogous result in the empirical setting, wherein pPC with PATH returns estimates of competitive quality to that of optimistic parameter tuning, achieving significant computational reductions compared to the current standard.

Third, the HGI algorithm provides a principled strategy for initializing score-based search in hybrid methods that asymptotically preserves soundness and completeness of constraint-based structure learning, elevating the GSC framework to consistency in the classical setting while significantly improving its empirical performance. While popular hybrid algorithms MMHC and H2PC forego asymptotic consistency for empirical performance, our HGI strategy makes no such compromise. When applied to GSC with various skeleton estimation strategies (including MMHC and H2PC), HGI significantly improves the estimation accuracy with typically negligible additional computational expense. Notably, a more recent development in hybrid structure learning is adaptively restricted greedy equivalence search (ARGES), which adaptively relaxes the restricted search space in order to ensure a search path in the space of equivalence classes (CPDAGs) to the optimal solution (Nandy et al., 2018a). Though in this chapter we take primary interest in improving upon the GSC framework which operates in the space of DAGs, we present preliminary simulation results in Section 2.4.7 that indicate significant potential for empirical improvement to ARGES through the initialization provided by HGI.

Altogether, we combined pPC, PATH, and HGI to create the pHGS algorithm, which enjoys the skeleton estimation efficiency of pPC, the parameter tuning by PATH, and the empirical well-performance afforded by HGI. In comparison to MMHC and H2PC, pHGS learns more accurate DAGs in nearly every considered underlying network configuration with significantly reduced computational cost.

2.6 Proofs

In this section, we prove Theorem 2 and Theorem 3.

The proof of Theorem 2 regarding the soundness and completeness of the pPC algorithm (Algorithm 2.1) follows from the thorough investigation of the criterion expressed in (1.4) by the pPC algorithm.

Proof of Theorem 2. Let $\mathbf{c}$ be any clustering labels obtained in the clustering step. Let $\hat{\mathcal{G}}_1$ denote the estimated structure after estimating edges within clusters at line 3, and $\tilde{\mathcal{G}}$ denote the final estimated skeleton at line 18.

We first show that every truly adjacent node pair (that is, adjacent in $\mathcal{G}^*$) is also adjacent in the estimated structure at every stage of Algorithm 2.1 following line 10. An implication of the Markov condition (Section 1.1) is that no truly adjacent pair of nodes $i, j \in \mathbf{V}$ are independent conditioned on any set of variables not containing X_i or X_j. Thus, every truly adjacent pair of nodes belonging to the same cluster are never disconnected after being connected in the first execution of Algorithm 2.1 in line 2:

$$\mathbf{N}_i^{\hat{\mathcal{G}}_1} \supseteq \{X_k \in \mathbf{N}_i^{\mathcal{G}^*} : c_k = c_i\} \text{ for all } i \in \mathbf{V}. \tag{2.13}$$

Similarly, every truly adjacent pair of nodes belonging to different clusters will necessarily be connected in the first screening of edges between clusters, and are never disconnected. Thus, $\tilde{\mathcal{G}}$ is equivalent to or a supergraph of the skeleton of $\mathcal{G}^*$, with

$$\mathbf{N}_i^{\tilde{\mathcal{G}}} \supseteq \mathbf{N}_i^{\mathcal{G}^*} \text{ for all } i \in \mathbf{V}. \tag{2.14}$$

Next, we show that no truly nonadjacent node pair (that is, not adjacent in $\mathcal{G}^*$) will be connected in $\tilde{\mathcal{G}}$. Restating (1.4), for every truly nonadjacent distinct node pair $i, j \in \mathbf{V}$ there exists at least one conditioning set $\mathbf{X_k} \subseteq \mathbf{N}_i^{\mathcal{G}^*} \setminus X_j$ or $\mathbf{X_k} \subseteq \mathbf{N}_j^{\mathcal{G}^*} \setminus X_i$ such that $(X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P$. Thus, it is sufficient to show that for every truly nonadjacent node pair, $(X_i \perp\!\!\!\perp X_j \mid \mathbf{X_k})_P$ for one such separation set $\mathbf{S}(i, j) \coloneqq \mathbf{k}$ is evaluated by Algorithm 2.1. Consider the following for any truly nonadjacent node pair $i, j \in \mathbf{V}$.

Suppose $c_i = c_j$. If there exists a conditioning set $\mathbf{X_k}$ such that $(X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P$ and $c_k = c_j$ for all $k \in \mathbf{k}$, then due to (2.13), $(X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P$ is evaluated by the first modified execution of PC in line 2. Otherwise, if $\exists k \in \mathbf{k}$ such that $c_k \neq c_j$, then for any and every conditioning set $\mathbf{X_k}$ such that $(X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P$, $\mathbf{X_k}$ satisfies criteria (a) of (2.5). In such a case, because of (2.14), $(X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P$ for one such conditioning set $\mathbf{X_k}$ is evaluated by the second modified execution of PC in line 18.

Suppose $c_i \neq c_j$. Ideally, i and j will be separated during the screening of edges between clusters in lines 3-17. If they are not, then for any and every conditioning set $\mathbf{X_k}$ such that $(X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P$, $\mathbf{X_k}$ satisfies criteria (b) of (2.5). Since (2.14) holds, $(X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P$ for one such $\mathbf{X_k}$ is evaluated by the second modified execution of PC in line 18.

We have shown that Algorithm 2.1 correctly estimates the skeleton of $\mathcal{G}^*$ and, in particular, for every truly nonadjacent $i, j \in \mathbf{V}$, $(X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_P$ for some conditioning set $\mathbf{X_k}$ has been evaluated, with separation set $\mathbf{S}(i, j) \coloneqq \mathbf{k}$ stored as specified in Algorithm 2.1. Since these separation sets are guaranteed to be accurate with conditional independence oracles, orientation of $\tilde{\mathcal{G}}$ to the CPDAG of $\mathcal{G}^*$ in line 19 follows directly from the correctness of `skel-to-cpdag` (Algorithm 1.2) (Spirtes and Glymour, 1991; Meek, 1995; Kalisch and Bühlmann, 2007).

□

To prove Theorem 3, we begin by stating and proving a well-known consistency result for constraint-based structure learning in the classical setting.

Lemma 2. *Suppose the distribution P is fixed and faithful to a DAG with CPDAG $\mathcal{G}^*$ and $\mathcal{D}$ is data containing n i.i.d. samples from P. Let $\hat{\mathcal{G}}_n(\alpha_n) = (\mathbf{V}, \hat{\mathbf{E}}_n)$ be the graph output of any exhaustive investigation of* (1.4) *followed by Algorithm 1.2 executed with a consistent test applied with threshold α_n. Then there exists $\alpha_n \to 0$ $(n \to \infty)$ such that*

$$\lim_{n\to\infty} \Pr\left[\hat{\mathcal{G}}_n(\alpha_n) = \mathcal{G}^*\right] = 1.$$

Proof of Lemma 2. Define $p_{n;i,j|\mathbf{k}}$ and $p^*_{n;i,j|\mathbf{k}}$ be the p-values for testing the independence between i and j conditioned on $\mathbf{k}$ with n data samples when i and j are and are not d-separated by $\mathbf{k}$ in $\mathcal{G}^*$, respectively. For example,

$$p_{n;i,j|\mathbf{k}} = \Pr(\chi^2_f > G^2_{n;ij|\mathbf{k}} \mid (X_i \perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_{\mathcal{G}}), \text{ and}$$
$$p^*_{n;i,j|\mathbf{k}} = \Pr(\chi^2_f > G^2_{n;ij|\mathbf{k}} \mid (X_i \not\perp\!\!\!\perp X_j \,|\, \mathbf{X_k})_{\mathcal{G}})$$

for the G^2 test of independence. Given faithfulness, d-separation in $\mathcal{G}^*$ corresponds one-to-one with conditional independence in P. In the case that conditional independence holds (i.e. the null hypothesis is true), $p_{n;i,j|\mathbf{k}} \sim \text{Unif}(0,1)$, so $\Pr(p_{n;i,j|\mathbf{k}} \leq \alpha_n) = \alpha_n \to 0$ $(n \to \infty)$. Whereas whenever conditional independence does not hold, the consistency of the conditional independence test implies that $p^*_{n;i,j|\mathbf{k}} = o(1)$. As these statements hold for every $i, j \in \mathbf{V}$ and $\mathbf{k} \subseteq \mathbf{V} \setminus \{i, j\}$ with fixed $p = |\mathbf{V}|$, what follows is the existence of $\alpha_n \to 0$ such that the probability of making any erroneous conditional independence inference in the investigation of (1.4) decays to zero when $n \to \infty$. That is,

$$\lim_{n\to\infty} \Pr\left[\bigcup_{i,j,\mathbf{k}} \left(p_{n;i,j|\mathbf{k}} \leq \alpha_n\right) \cup \left(p^*_{n;i,j|\mathbf{k}} > \alpha_n\right)\right] = 0. \tag{2.15}$$

Given perfect conditional independence inferences asymptotically, the consistency follows straightforwardly from the correctness of (1.4) for skeleton identification and the soundness and completeness of Algorithm 1.2 for CPDAG orientation.

□

The proof of Theorem 3 then proceeds from Lemma 2 and its proof as follows.

Proof of Theorem 3. (2.15) indicates the existence of some $a_n \to 0$ $(n \to \infty)$ such that

$$\lim_{n\to\infty} \Pr\left[\sup_{i,j,\mathbf{k}} p^*_{n;i,j|\mathbf{k}} \leq a_n < \inf_{i,j,\mathbf{k}} p_{n;i,j|\mathbf{k}}\right] = 1. \tag{2.16}$$

That is, in the large-sample limit, a_n perfectly discriminates true positives from true negatives. For conditional independence inferred with threshold $\alpha_n \geq a_n \geq \sup_{i,j,\mathbf{k}} p^*_{n;i,j|\mathbf{k}}$, only false positive edges are possible. This ensures that for every truly nonadjacent distinct node pair $i, j \in \mathbf{V}$ (that is, $i - j \notin \mathbf{E}^*$ where $\mathbf{E}^*$ is the edge set of the skeleton of $\mathcal{G}^*$), at least one conditioning set that separates i and j is considered in the investigation of (1.4) and stored in (2.6) since (2.16) holds asymptotically.

$$\max_{i-j\in\mathbf{E}^*} \Phi_{n;ij} \leq \sup_{i,j,\mathbf{k}} p^*_{n;i,j|\mathbf{k}} \leq a_n < \inf_{i,j,\mathbf{k}} p_{n;i,j|\mathbf{k}} \leq \min_{i-j\notin\mathbf{E}^*} \Phi_{n;ij}. \tag{2.17}$$

Inspecting (2.17) confirms the existence of a threshold value $\hat{a}_n$, in particular $\hat{a}_n = \max_{i-j\in\mathbf{E}^*} \Phi_{n;i,j}$, that perfectly discriminates true positives from true negatives. Consequently, $\mathbf{E}^* = \{i - j : \Phi_{n;ij} \leq \hat{a}_n\}$, and $\{\mathbf{S}_n(i,j) : \Phi_{n;ij} > \hat{a}_n\}$ obtains correct separation sets for complete and sound orientation to $\mathcal{G}^*$ according to Algorithm 1.2.

Having verified the existence of a threshold $\hat{a}_n$ that recovers $\mathcal{G}^*$ by thresholding Φ_n and $\mathbf{S}_n$, it remains to show that $\hat{a}_n$ is recovered and selected by Algorithm 2.2. Parameter settings $\tau = 1 + \sum_{i<j} \mathbb{1}\{\Phi_{n;ij} \leq \alpha_n\} = 1 + |\hat{\mathbf{E}}^{(1)}_n|$ and $\alpha^{(\tau)} = 0$ regulate the generation of $\alpha^{(t)}$ such that the skeletons of sequential estimates differ in sparsity by exactly one edge. The resulting values $\mathcal{A} \coloneqq \{\alpha^{(1)}, \ldots, \alpha^{\tau}\}$ correspond to the order statistics of the upper triangular elements of Φ_n decreasing from $\alpha^{(1)}_n = \alpha_n \geq a_n \geq \hat{a}_n$ to $\alpha^{(\tau)}_n = 0$, ensuring that $\hat{a}_n = \max_{i-j\in\mathbf{E}^*} \Phi_{n;i,j} \in \mathcal{A}$ and so $\mathcal{G}^* \in \{\mathcal{G}^{(t)} : t \in \{1, 2, \ldots, \tau\}\}$. The consistency of ϕ ensures that in the large-sample limit, $t^* = \operatorname{argmax}_{t\in\{1,\ldots,\tau\}} \phi(\mathbf{X} \mid \hat{\mathcal{G}}^{(t)}, \mathcal{D})$ selects $\alpha^{(t^*)} = \hat{a}_n$ that results in the true CPDAG $\hat{\mathcal{G}}^{(t^*)}_n(\alpha_n) = \mathcal{G}^*$, with high probability.

Given that the arguments hold for any α_n as long as $\alpha_n \geq a_n$ when n is large, it is not necessary for α_n to converge to zero. In particular, (2.8) holds for any fixed $\alpha_n = \alpha \in (0, 1)$.

□

Note that while Theorem 3 generously allows for any $\alpha_n \in (0, 1)$, in practice a small threshold is desirable in the interest of efficiency. Since $\Pr(p_{n;i,j|\mathbf{k}} \leq \alpha_n) = \alpha_n$, the expected

length of the solution path τ may be approximately bounded between $|\mathbf{E}^*|$ and $|\mathbf{E}^*|+\frac{p(p-1)}{2}\alpha_n$. The bounds may be further regulated if approximate knowledge of the sparsity of $\mathcal{G}^*$ is known.

Furthermore, an inspection of the proof of Theorem 3 more generally indicates that the solution path returned by Algorithm 2.2 with the same τ and $\alpha^{(\tau)}$ contains the CPDAG of the underlying DAG $\mathcal{G}^*$ if there exists some $a \leq \alpha$ such that

$$\max_{i,j \in \mathbf{E}^*} \Phi_{ij} \leq a < \min_{i,j \notin \mathbf{E}^*} \Phi_{ij}$$

holds when (1.4) is investigated with threshold α, which can hold even for a finite sample size n.

CHAPTER 3

Bayesian Causal Bandits with Backdoor Adjustment Prior

3.1 Motivation and Overview

In Chapter 2, we focused on learning the equivalence class of large Bayesian network structures in the observational setting, proposing algorithms for identifying the CPDAG of the underlying DAG based on asymptotic properties. As discussed in Friedman and Koller (2003), in many practical applications where identification of the underlying causal graph is desirable for inference or decision making, there may not be sufficient data to distinguish a single structure with high assurance. We now direct our attention to consider exploitation of causal assumptions in smaller domains with modest amounts of observational data for the purposes of efficiently allocating resources when conducting experimental investigations.

The multi-armed bandit (MAB) problem is a well-known sequential allocation framework for experimental investigations (Berry and Fristedt, 1985). Classically, the MAB problem formulation features an action set $\mathcal{A}$ consisting of $|\mathcal{A}| = K$ actions, also called arms, typically corresponding to interventions. Each arm $a \in \mathcal{A}$ defines a real-valued distribution for the reward signal, with expected reward μ_a. The objective of an allocation policy is to sequentially pick arms in a manner that maintains a balance between exploration and exploitation in the interest of identifying and obtaining the greatest reward. Maximally and effectively utilizing all available information is imperative, especially when investigating interventions that are either or both resource-demanding and time consuming.

Lattimore et al. (2016) proposed the causal bandit (CB) problem setting wherein a non-trivial probabilistic causal model is assumed to govern the distribution of the reward variable and its covariates (Pearl, 2000). The addition of causal assumptions introduces avenues by which interventional distributions may be inferred from observational distributions and information may be shared between arms. Most works addressing the CB problem exploit strong assumptions as to prior knowledge of the underlying causal model to achieve improvements over standard MAB algorithms. In this work, we develop a Bayesian CB framework that does not require prior knowledge of the underlying causal structure, but instead efficiently utilizes previously available observational data and acquired interventional data to inform exploitation and guide exploration.

3.1.1 Related Work

In its original formulation by Lattimore et al. (2016), the CB problem presupposes knowledge of the underlying causal graph. Accordingly, most proposed CB algorithms require knowledge of the causal graph structure (Lattimore et al., 2016; Lee and Bareinboim, 2018; Maiti et al., 2021; Yabe et al., 2018), and some additionally assume certain model parameters are given (Lu et al., 2020; Nair et al., 2021). Furthermore, many approaches are dependent on some restrictive form or class of graphs. These assumptions are restrictive and often unrealistic in practice.

More recently, Lu et al. (2021) proposed a central node approach based on the work of Greenewald et al. (2019) that does not assume prior knowledge of the causal graph, but rather asymptotic knowledge of the observational distribution. Their approach is restrictive in terms of structural and distributional assumptions, and while it is generally reasonable to assume that observational data is much more accessible than interventional data (Greenewald et al., 2019), the large-sample observational setting is not often realistic. de Kroon et al. (2022) proposed an estimator using separating sets to share information between arms without assuming prior knowledge of nor requiring discovery of the causal graph. Their methodology

makes no attempt to learn the causal graph, and makes use of observational data only to strengthen conditional independence testing to identify separating sets.

Relevant to our work is the estimation of interventional quantities from observational data using intervention calculus (Pearl, 2000). In the CB setting, Lattimore et al. (2016) and Nair et al. (2021) consider graph structures with no confounding such that the interventional distributions are equivalent to conditional distributions, and Maiti et al. (2021) proposed a consistent estimator for the expected reward for discrete variables using both interventional and observational data in the presence of confounding. Our work extends the Bayesian model averaging approach proposed by Pensar et al. (2020) wherein the possible causal effect estimates are averaged across an observational posterior distribution of graphs.

3.1.2 Our Contributions

We approach the CB problem from a Bayesian perspective, assuming simply that finite samples of observational data are available. Importantly, we do not assume the causal graph is known, nor are we restrictive as to the class of graphs. We design a novel Bayesian CB framework called **B**ayesian **B**ackdoor **B**andit (BBB) that efficiently utilizes the entirety of evidence from an ensemble of observational and interventional data in a unified Bayesian model. Our proposed BBB methodology quantifies the uncertainty in the expected reward estimates as contributed to by the reward signal and the causal model to identify potentially profitable exploration, simultaneously learning the causal graph in addition to and for the purposes of improving estimates to exploit. Through extensive numerical experiments, we validate our methodology by demonstrating compelling empirical performance against both non-causal and causal algorithms. In particular, even with modest amounts of observational data, our BBB approach achieves substantially superior cumulative regret performance compared to standard algorithms, as well as against a generously optimistic version of the causal central node approach proposed by Lu et al. (2021) that assumes large-sample observational data.

Additionally, in detailing the application of our methods to the discrete and Gaussian distributional settings, we propose various developments that are of independent interest. In the discrete setting, we derive an approximation for the sampling variance of the backdoor adjustment probability estimate. In the Gaussian setting, we characterize the interventional variance of a target variable using intervention calculus and correspondingly propose an estimator, and we propose a simple graphical criterion for sharing causal information between arms to perform intervention calculus with jointly observational and interventional data.

The remainder of the chapter is arranged as follows. We first review relevant background and notation in Section 3.2. Then, we develop the formulation of our proposed Bayesian backdoor prior and posterior update in Section 3.3, discussing the design of informative conditional priors given a graph and Bayesian model averaging across graph structures. In Section 3.4, we develop our proposed algorithms by applying established MAB algorithms under the BBB framework, and we discuss details regarding the implementation of BBB in the discrete and Gaussian settings in Section 3.5. We provide extensive empirical results in Section 3.6 and summarize our work in Section 3.7. Lastly, we include detailed proofs for our theoretical results in Section 3.8 and technical derivations in Section 3.9.

3.2 Preliminaries

We consider the setting where the generative model governing a joint probability distribution P of a set of p variables $\mathbf{X} = \{X_1, \ldots, X_p\}$ is a causal Bayesian network (CBN). A CBN consists of its structure $\mathcal{G}$, which takes the form of a directed acyclic graph (DAG), and its parameters $\Theta_{\mathcal{G}}$. Its DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, often referred to as the underlying causal graph, is composed of a set of nodes $\mathbf{V} = \{1, \ldots, p\}$ in one-to-one correspondence with the variables, and a set of directed edges $\mathbf{E}$ oriented such that there are no directed cycles. As is standard in causal literature, we may refer to a node $i \in \mathbf{V}$ and its corresponding variable $X_i \in \mathbf{X}$ interchangeably.

The probability distribution P imposed by the CBN factorizes according to structure $\mathcal{G}$, with local probability distributions defined by $\Theta_{\mathcal{G}}$. In particular, $P(\mathbf{X}) = \prod_{i=1}^{p} P(X_i \mid \mathbf{Pa}_i^{\mathcal{G}}, \theta_i^{\mathcal{G}})$, where $\mathbf{Pa}_i^{\mathcal{G}} = \{X_j : j \to i \in \mathbf{E}\}$ is the parents of X_i in $\mathcal{G}$, and the local parameters $\theta_i^{\mathcal{G}} \in \Theta_{\mathcal{G}}$ specify the conditional probability distribution (CPD) of X_i given its parents. In our work, we assume that $\mathbf{X}$ is a causally sufficient system with no unobserved confounders.

The action set $\mathcal{A}$ consists of $|\mathcal{A}| = K$ arms that correspond to interventions on variables in $\mathbf{X} \setminus Y$, where $Y = X_p$ is the reward variable (Lattimore et al., 2016). In particular, let arm $a \in \mathcal{A}$ correspond to a deterministic atomic intervention denoted $do(X_{\langle a \rangle} = x_a)$ (Pearl, 1995), fixing $X_{\langle a \rangle}$ to some value $x_a \in \mathrm{Dom}(X_{\langle a \rangle})$, where $\langle a \rangle \in \mathbf{V}$ is the node corresponding to the intervened variable. The expected reward of each arm $a \in \mathcal{A}$ is given by $\mu_a \coloneqq \mathrm{E}[Y \mid do(X_{\langle a \rangle} = x_a)]$, and there is some optimal arm $a^* \coloneqq \mathrm{argmax}_{a \in \mathcal{A}} \mu_a$ corresponding to the optimal reward $\mu^* \coloneqq \mu_{a^*}$. Given a horizon of time steps T, let $a_t \in \mathcal{A}$ be the arm pulled by an algorithm at time step $t \in \{1, \ldots, T\}$. Denote by $n_a(t) = \sum_{l=1}^{t} \mathbb{1}\{a_l = a\}$ the number of times arm a has been pulled in t time steps. We take interest in optimizing the cumulative regret, where the objective of the algorithm is to pull arms over T time steps with a balance between exploring different arms and exploiting the reward signal to minimize the expected cumulative regret $\mathrm{E}[R_T] = T\mu^* - \sum_{a \in \mathcal{A}} \mu_a \mathrm{E}[n_a(T)]$.

In our problem formulation, we assume possession of n_0 samples of observational data $\mathcal{D}_0$ prior to investigating arms. We denote by $\mathcal{D}^{(t)}$ the interventional data acquired by pulling arm a_t at time t, and by $\mathcal{D}_a[t] = \bigcup_{l \le t, a_l = a} \mathcal{D}^{(l)}$ the accumulated interventional data from arm a through time t. The combined observational and interventional data accrued through time t is $\mathcal{D}[t] = \mathcal{D}_0 \cup \bigcup_{a \in \mathcal{A}} \mathcal{D}_a[t]$, which we refer to as ensemble data.

We now describe a Bayesian approach to the general MAB problem, with some notation adapted from Kaufmann et al. (2012). The parameters $\Theta_{\mathcal{A}} = (\theta_a)_{a \in \mathcal{A}}$, assumed to mutually independently define the corresponding marginal reward distributions $p_{\theta_a}(y) \coloneqq P[Y = y \mid do(X_{\langle a \rangle} = x_a)]$, jointly follow a modular prior distribution $\Pi^0(\Theta_{\mathcal{A}}) = \prod_{a \in \mathcal{A}} \pi_a^0(\theta_a)$. Typically, $(\pi_a^0)_{a \in \mathcal{A}}$ are chosen to be all equal and uninformative. When arm $a_t \in \mathcal{A}$ is pulled at time

step t and a realization $y_t \leftarrow Y \mid do(X_{\langle a_t \rangle} = x_{a_t})$ is observed, the posterior Π^t is computed by updating according to $\pi^t_{a_t}(\theta_{a_t}) \propto p_{\theta_{a_t}}(y_t)\ \pi^{t-1}_{a_t}(\theta_{a_t})$, while $\pi^t_a = \pi^{t-1}_a$ for $a \neq a_t$. For each arm $a \in \mathcal{A}$, the posterior π^t_a induces a posterior distribution for the expected reward μ_a, which is simply a marginal or transformation of π^t_a since, in general, μ_a is a function of θ_a. These posteriors are utilized by Bayesian MAB algorithms, which we discuss and apply under our proposed framework in Section 3.4.

3.3 Designing Informative Priors with Intervention Calculus

In this section, we detail the development of our proposed BBB model consisting of an informative prior Π^0 constructed using observational data that seamlessly integrates interventional data to obtain the posterior Π^t. For each arm $a \in \mathcal{A}$, we construct conditional priors $\pi^0_{a|\mathbf{Z}}(\theta_a)$ using the backdoor adjustment given sets $\mathbf{Z} \subseteq \mathbf{X} \setminus X_{\langle a \rangle}$ as follows. If $\mathbf{Z}$ satisfies the backdoor criterion relative to $X_{\langle a \rangle}$ and Y (Pearl, 2000, Definition 3.3.1), then the interventional distribution $Y \mid do(X_{\langle a \rangle} = x_a)$ may be expressed in terms of the joint observational distribution of $\{X_{\langle a \rangle}, Y\} \cup \mathbf{Z}$ via the backdoor adjustment (Pearl, 2000, Theorem 3.3.2):

$$P[Y = y \mid do(X_{\langle a \rangle} = x_a)] = \sum_{\mathbf{z} \in \mathrm{Dom}(\mathbf{Z})} P(Y = y \mid X_{\langle a \rangle} = x_a, \mathbf{Z} = \mathbf{z}) P(\mathbf{Z} = \mathbf{z}). \tag{3.1}$$

Eq. (3.1) provides an avenue through which an estimator for μ_a using observational data may be derived, which we denote $\hat{\mu}_{a,\mathrm{bda}}(\mathbf{Z})$ and with which we design an informative prior $\pi^0_{a|\mathbf{Z}}$ such that the induced prior distribution of the expected reward μ_a satisfies

$$\mathrm{E}_{\pi^0_{a|\mathbf{Z}}}[\mu_a] = \hat{\mu}_{a,\mathrm{bda}}(\mathbf{Z}), \quad \mathrm{Var}_{\pi^0_{a|\mathbf{Z}}}[\mu_a] = \hat{\mathrm{SE}}^2\left[\hat{\mu}_{a,\mathrm{bda}}(\mathbf{Z})\right]. \tag{3.2}$$

The matching of the prior variance with the sampling variance of the backdoor adjustment estimator endeavors to assign the appropriate prior effective sample size. When arm $a_t = a$ is pulled at time step $t \in \{1, \ldots, T\}$ and a realization of the reward $y_t \leftarrow Y \mid do(X_{\langle a \rangle} = x_a)$

is observed in $\mathcal{D}^{(t)}$, the posterior $\pi^t_{a|\mathbf{Z}}$ is computed by updating according to $\pi^t_{a|\mathbf{Z}}(\theta_a) \propto p_{\theta_a}(y_t)\ \pi^{t-1}_{a|\mathbf{Z}}(\theta_a)$.

Thus far we have taken for granted the possession of adjustment set $\mathbf{Z}$, the validity of which is dependent on the underlying causal structure $\mathcal{G}$ which we assume to be unknown. If $Y \notin \mathbf{Pa}^{\mathcal{G}}_{\langle a \rangle}$, then $\mathbf{Z} = \mathbf{Pa}^{\mathcal{G}}_{\langle a \rangle}$ satisfies the backdoor criterion relative to $X_{\langle a \rangle}$ and Y, and its uncertainty is quantified by the posterior probability $P(\mathbf{Pa}_{\langle a \rangle} = \mathbf{Z} \mid \mathcal{D}[t])$ given the ensemble data at time t. Accordingly, the posterior of θ_a is determined by averaging over all possible parent sets for $X_{\langle a \rangle}$:

$$\pi^t_a(\theta_a) = \sum_{\mathbf{Z} \subseteq \mathbf{X} \setminus X_{\langle a \rangle}} \pi^t_{a|\mathbf{Z}}(\theta_a) P(\mathbf{Pa}_{\langle a \rangle} = \mathbf{Z} \mid \mathcal{D}[t]), \tag{3.3}$$

which is the key posterior distribution to be updated at each time step t in the Bayesian CB problem. Note that if $Y \in \mathbf{Pa}^{\mathcal{G}}_{\langle a \rangle}$, then $P[Y = y \mid do(X_{\langle a \rangle} = x_a)] = P(Y = y)$ holds straightforwardly for $y \in \mathrm{Dom}(Y)$. Accordingly, if $Y \in \mathbf{Z}$, we compute $\hat{\mu}_{a,\mathrm{bda}}(\mathbf{Z})$ with the marginal distribution of Y for the design of $\pi^0_{a|\mathbf{Z}}$.

The parent set distribution in (3.3) is obtained according to a posterior distribution of DAG structures:

$$P(\mathbf{Pa}_i = \mathbf{Z} \mid \mathcal{D}[t]) = \sum_{\mathcal{G}' : \mathbf{Pa}^{\mathcal{G}'}_i = \mathbf{Z}} P(\mathcal{G}' \mid \mathcal{D}[t]). \tag{3.4}$$

The structure posterior is given by $P(\mathcal{G} \mid \mathcal{D}[t]) \propto P(\mathcal{D}[t] \mid \mathcal{G}) P(\mathcal{G})$, where $P(\mathcal{G})$ is the structure prior, and the marginal likelihood $P(\mathcal{D}[t] \mid \mathcal{G}) = \int P(\mathcal{D}[t] \mid \mathcal{G}, \Theta_{\mathcal{G}}) P(\Theta_{\mathcal{G}} \mid \mathcal{G}) d\Theta_{\mathcal{G}}$ is obtained by integrating the likelihood function over the support of a conjugate prior of the parameters as follows. Let $m \in \mathcal{I} \coloneqq \{1, \ldots, M\}$ index the $M = n_0 + t$ samples of data in $\mathcal{D}[t]$, and let $\mathcal{O}_i \subseteq \mathcal{I}$ represent the data points for which X_i is not fixed by intervention. We make standard assumptions of global and local parameter independence and parameter modularity (see Heckerman et al. (1995) and Friedman and Koller (2003) for details). These

allow us to express the marginal likelihood as $P(\mathcal{D}[t] \mid \mathcal{G}) = \prod_{i=1}^{p} P(x_i[\mathcal{O}_i] \mid \mathbf{pa}_i^{\mathcal{G}}[\mathcal{O}_i])$, where $x_i[\cdot]$ and $\mathbf{pa}_i^{\mathcal{G}}[\cdot]$ represent indexed samples of X_i and $\mathbf{Pa}_i^{\mathcal{G}}$ in $\mathcal{D}[t]$, respectively. Assuming a conjugate prior and complete data, each conditional likelihood $P(x_i[\mathcal{O}_i] \mid \mathbf{pa}_i^{\mathcal{G}}[\mathcal{O}_i])$ can be calculated in closed form by integrating over the parameters:

$$P(x_i[\mathcal{O}_i] \mid \mathbf{pa}_i^{\mathcal{G}}[\mathcal{O}_i]) = \int \left[\prod_{m \in \mathcal{O}_i} P\left(x_i[m] \mid \mathbf{pa}_i^{\mathcal{G}}[m], \theta_i^{\mathcal{G}}\right) \right] P(\theta_i^{\mathcal{G}}) d\theta_i^{\mathcal{G}} \tag{3.5}$$

where $\theta_i^{\mathcal{G}} = \theta_{X_i|\mathbf{Pa}_i^{\mathcal{G}}}$ is the parameters specifying the conditional distribution of X_i given its parents (Eaton and Murphy, 2007).

Assuming the distribution P is faithful to $\mathcal{G}$ (that is, all and only the conditional independence relationships in P are entailed by $\mathcal{G}$), the posterior probability $P(\mathcal{G} \mid \mathcal{D}[t])$ will concentrate around the Markov equivalence class with increasing samples of observational data n_0. The equivalence class consists of the identification of all direct edge connections (that is, the skeleton of $\mathcal{G}$) and some edge orientations called compelled edges, but even with infinite observational data, in general, not all edge orientations are identifiable without interventional data. The effect on $P(\mathcal{G} \mid \mathcal{D}[t])$ of pulling arm $a \in \mathcal{A}$ and observing interventional data according to the intervention $do(X_{\langle a \rangle} = x_a)$ is primarily though not limited to that of clarifying the orientation of the edges incident to $X_{\langle a \rangle}$ in $\mathcal{G}$.

Considering that the DAG space grows super-exponentially with the number of variables (Robinson, 1977), computation of the parent set probabilities $P(\mathbf{Pa}_i \mid \mathcal{D}[t])$ is admittedly challenging, even when the maximum number of parents is restricted. It is generally computationally advantageous to additionally assume a structure prior satisfying modularity, that is $P(\mathcal{G}) = \Pi_{i=1}^{p} P(\mathbf{Pa}_i^{\mathcal{G}})$, so that the posterior distribution is proportional to decomposable weights consisting of the product of local scores depending only on a node and its parents (Friedman and Koller, 2003). This property of score decomposability is crucial for the efficient implementation of Markov Chain Monte Carlo (MCMC) methods in which the probability distribution of features in $\mathcal{G}$ may be estimated by sampling DAGs from a Markov

chain with stationary distribution $P(\mathcal{G} \mid \mathcal{D}[t])$ (Madigan et al., 1995; Friedman and Koller, 2003; Kuipers and Moffa, 2017). Particularly useful for our purposes is an algorithm developed by Pensar et al. (2020) to compute the exact parent set probabilities for a graph in time $O(3^p p)$ that also takes advantage of score decomposability.

3.4 Bayesian Backdoor Bandit Algorithms

In this section, we apply our proposed BBB framework to several state-of-the-art MAB algorithms, namely upper confidence bound (UCB), Thompson sampling (TS), and Bayesian UCB (Bayes-UCB). Each method is concerned with designing and computing some criterion $U_a(t)$ to maintain a balance between exploration and exploitation when selecting arms according to $a_t \in \text{argmax}_{a \in \mathcal{A}} U_a(t)$. In what follows, we briefly introduce these methods and discuss their application under the BBB framework.

The general UCB family of algorithms operates under the principle of optimism in the face of uncertainty (Lai and Robbins, 1985). Arms that have not been investigated as many times as others have more uncertain reward estimates and thus optimistically have potential for greater reward, motivating the design of a padding function $F_a(t)$ for computing the selection criterion $U_a(t) = \hat{\mu}_a(t) + F_a(t)$. Intuitively, the combination of the expected reward estimate $\hat{\mu}_a(t)$ and the uncertainty $F_a(t)$ maintains a balance between high confidence exploitation and potentially profitable exploration. Perhaps the most well-known and typically the default instantiation of UCB algorithms is UCB1 (Agrawal, 1995; Auer et al., 2002) which computes the following criterion:

$$U_a(t) = \hat{\mu}_a(t-1) + c\sqrt{\log(t-1)/n_a(t-1)}. \tag{3.6}$$

The confidence tuning parameter $c > 0$, discussed in Sutton and Barto (2018), controls the desired degree of exploration, where $c = \sqrt{2}$ in Auer et al. (2002). Hereafter, when discussing UCB, we refer to the policy expressed by the criterion in (3.6).

In what we refer to as BBB-UCB, we estimate the expected reward with the posterior mean $\mathrm{E}_{\pi_a^{t-1}}[\mu_a]$ with respect to (3.3), and we replace $1/n_a(t-1)$ with the posterior variance $\mathrm{Var}_{\pi_a^{t-1}}[\mu_a] \sim 1/(n_0 + n_a(t-1))$. In particular, for each arm $a \in \mathcal{A}$, we compute

$$U_a(t) = \mathrm{E}_{\pi_a^{t-1}}[\mu_a] + c\sqrt{\mathrm{Var}_{\pi_a^{t-1}}[\mu_a]\log(t)}, \tag{3.7}$$

where, for outer expectation with respect to the parent set distribution $\mathbb{P}_{t-1}(\mathbf{Z}) \coloneqq P(\mathbf{Pa}_i = \mathbf{Z} \mid \mathcal{D}[t-1])$ in (3.4),

$$\mathrm{E}_{\pi_a^{t-1}}[\mu_a] = \mathrm{E}_{\mathbb{P}_{t-1}}\left[\mathrm{E}_{\pi_{a|\mathbf{Z}}^{t-1}}[\mu_a]\right], \qquad \mathrm{Var}_{\pi_a^{t-1}}[\mu_a] = \mathrm{E}_{\mathbb{P}_{t-1}}\left[\mathrm{Var}_{\pi_{a|\mathbf{Z}}^{t-1}}[\mu_a]\right] + \mathrm{Var}_{\mathbb{P}_{t-1}}\left[\mathrm{E}_{\pi_{a|\mathbf{Z}}^{t-1}}[\mu_a]\right].$$

The Bayesian procedures of Bayes-UCB and TS are especially amenable to straightforward application under the BBB framework. These methods follow the Bayesian MAB formulation introduced in Section 3.2, typically taking as input uninformative priors in Π^0 that are equivalent for each arm. At each time step t, TS samples the expectations from the posterior $U_a(t) \leftarrow \pi_a^{t-1}(\mu_a)$, effectively selecting arm $a \in \mathcal{A}$ with probability equal to the posterior probability that μ_a is the highest expectation (Thompson, 1933). Bayes-UCB instead computes for each arm at time t an upper quantile of μ_a based on its posterior distribution induced by π_a^{t-1}:

$$U_a(t) = Q\left(1 - \frac{1}{t(\log T)^c}, \pi_a^{t-1}\right), \tag{3.8}$$

where $Q(r, \rho)$ is the quantile function defining $P_\rho(X \leq Q(r, \rho)) = r$ for probability distribution ρ and random variable $X \sim \rho$, and c is a constant for computing the quantile used in the theoretical analysis of Bayes-UCB, with $c = 0$ empirically preferred (Kaufmann et al., 2012). For the BBB variants of Bayes-UCB and TS, we need simply to supply our designed backdoor adjustment prior Π^0 and make appropriate Bayesian updates to obtain the posterior Π^t. A detailed algorithm outline of our BBB methodology is provided in Algorithm 3.1.

Algorithm 3.1 `BBB-Alg`$(T, \mathcal{A}, \mathcal{D}_0, c)$

input: Horizon T, action set $\mathcal{A}$, observational data $\mathcal{D}_0$, confidence level c

Compute the observational parent set posteriors (3.4)
for all $a \in \mathcal{A}$ and $\mathbf{Z} \subseteq \mathbf{X} \setminus X_{\langle a \rangle}$ **do**
    Compute $\pi^0_{a|\mathbf{Z}}$ according to (3.2)
end for
for all $t = 1, \ldots, T$ **do**
    for all $a \in \mathcal{A}$ **do**
        Compute criterion $U_a(t)$ according to `Alg`:
        - `Bayes-UCB`: $U_a(t) = Q(1 - 1/(t(\log T)^c), \pi_a^{t-1})$ as in (3.8)
        - `TS`: Sample $U_a(t) \leftarrow \pi_a^{t-1}(\mu_a)$
        - `UCB`: $U_a(t) = \mathrm{E}_{\pi_a^{t-1}}[\mu_a] + c\sqrt{\mathrm{Var}_{\pi_a^{t-1}}[\mu_a] \log(t)}$ as in (3.7)
    end for
    Pull arm $a_t \in \operatorname{argmax}_{a \in \mathcal{A}} U_a(t)$ and observe $\mathcal{D}^{(t)}$
    for all $\mathbf{Z} \subseteq \mathbf{X} \setminus X_{\langle a \rangle}$ where $a = a_t$ **do**
        Update $\pi^t_{a|\mathbf{Z}}$ according to $\pi^t_{a|\mathbf{Z}}(\theta_a) \propto p_{\theta_a}(y_t)\ \pi^{t-1}_{a|\mathbf{Z}}(\theta_a)$
    end for
    Compute or update the parent set posteriors (3.4)
end for

3.5 Implementation Details

3.5.1 Nonparametric Discrete Setting

We now detail the application of our proposed construction of $\pi^0_{a|\mathbf{Z}}$ to the setting where the CPDs are multinomials, with each variable $X_i \in \mathbf{X}$ probabilistically attaining its states depending on the attained state configuration of its parents $\mathbf{Pa}_i^{\mathcal{G}}$. The reward variable $Y = X_p$ is a binary variable with $\mathrm{Dom}(Y) = \{0, 1\}$. If $Y \notin \mathbf{Z}$, μ_a may be estimated with observational data through straightforward empirical estimation of (3.1):

$$\hat{\mu}_{a,\mathrm{bda}}(\mathbf{Z}) = \hat{P}[Y = 1 \mid do(X_{\langle a \rangle} = x_a)] = \frac{1}{n_0} \sum_{\mathbf{z}} \frac{n_0[1, x_a, \mathbf{z}] n_0[\mathbf{z}]}{n_0[x_a, \mathbf{z}]}, \tag{3.9}$$

where $n_0[1, x_a, \mathbf{z}]$ represents the number of the n_0 samples of $\mathcal{D}_0$ in which $Y = 1$, $X = x_a$, and $\mathbf{Z} = \mathbf{z}$, with corresponding definitions for $n[x_a, \mathbf{z}]$ and $n[\mathbf{z}]$.

Analysis of the sampling distribution of (3.9) is admittedly challenging. To design an appropriately weighted informative prior as proposed in (3.2), we require some characterization of the sampling variability of $\hat{\mu}_{a,\text{bda}}(\mathbf{Z})$. Hence, we derive an approximation of the variance of (3.9), $\hat{\text{SE}}^2[\hat{\mu}_{a,\text{bda}}(\mathbf{Z})]$. We accomplish this by first re-expressing the joint counts $n_0[\cdot]$ as sums of elements of a multinomial random vector. The term within the sum may then be expressed as a product and ratio of intersecting random quantities, which we approximate through a first-order Taylor series expansion. The details of the derivation are delegated to Section 3.9. It is appropriate to acknowledge that Maiti et al. (2021) proposed a provably unbiased strategy for empirical estimation of (3.1) through splitting the sample into independent partitions. However, this approach suffers from severe loss of precision through what some may consider underutilization of the observed data. In our experiments detailed in Section 3.6.2, we find the empirical performance of (3.9) in our applications to be acceptable. We additionally provide extensive empirical validation of our derived approximation, demonstrating coverage probabilities comparable to empirical estimates of the sampling variability for modest sample sizes.

Since the reward variable under arm a is a Bernoulli random variable with probability parameter $\mu_a = P[Y = 1 \mid do(X_{\langle a \rangle} = x_a)]$, we assume a conjugate prior $\pi^0_{a|\mathbf{Z}} = \text{Beta}(\alpha_0, \beta_0)$ for $\theta_a = \mu_a$ designed according to (3.2), resulting in prior hyperparameters

$$\alpha_0 = \hat{\mu}_{a,\text{bda}}(\mathbf{Z}) \left(\frac{\hat{\mu}_{a,\text{bda}}(\mathbf{Z})[1 - \hat{\mu}_{a,\text{bda}}(\mathbf{Z})]}{\hat{\text{SE}}^2 [\hat{\mu}_{a,\text{bda}}(\mathbf{Z})]} - 1 \right), \qquad \beta_0 = \alpha_0 \left(\frac{1 - \hat{\mu}_{a,\text{bda}}(\mathbf{Z})}{\hat{\mu}_{a,\text{bda}}(\mathbf{Z})} \right).$$

3.5.2 Gaussian Unit Deviation Setting

In this section, we consider the setting where the causal model may be expressed as a set of Gaussian structural equations:

$$X_j = \sum_{i=1}^{p} \beta_{ij} X_i + \varepsilon_j, \quad \varepsilon_j \sim \text{N}(0, \sigma_j^2), \quad j = 1, \ldots, p. \tag{3.10}$$

There is no intercept term, which is analogous to having prior knowledge of the observational means, and we consider interventions $x_a \in \{-1, 1\}$, which may be interpreted as investigating unit deviations from the observational means. In this setting, the causal effect of $X_{\langle a \rangle}$ on Y is given by

$$\psi_{\langle a \rangle} \coloneqq \mathrm{E}[Y \mid do(X_{\langle a \rangle} = x' + 1)] - \mathrm{E}[Y \mid do(X_{\langle a \rangle} = x')]$$

for any $x' \in \mathbb{R}$, derived via a special case of (3.1). Note that in our problem formulation, $Y \mid do(X_{\langle a \rangle} = 1)$ and $-Y \mid do(X_{\langle a \rangle} = -1)$ are identically distributed, so all data generated from interventions on $X_{\langle a \rangle}$ may be combined to estimate $\psi_{\langle a \rangle}$. Since $\mu_a = x_a \psi_{\langle a \rangle}$, we focus our efforts on estimating and modeling $\psi_{\langle a \rangle}$. Accordingly, in constructing our priors using intervention calculus, we design priors $\pi^0_{\langle a \rangle | \mathbf{Z}}$ for $\theta_{\langle a \rangle}$ corresponding to estimating $\psi_{\langle a \rangle}$, and allow $\pi^0_{a|\mathbf{Z}}$ to be the induced priors for θ_a corresponding to $\mu_a = \psi_{\langle a \rangle} x_a$, detailed as follows.

If $Y \notin \mathbf{Z}$, then a consistent estimator of $\psi_{\langle a \rangle}$, denoted $\hat{\psi}_{\langle a \rangle, \mathrm{bda}}$, may be obtained with observational data by the least squares regression

$$Y = \psi_{\langle a \rangle} X_{\langle a \rangle} + \boldsymbol{\gamma}^\top \mathbf{Z} + e, \quad e \sim \mathrm{N}(0, \eta^2), \tag{3.11}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^{|\mathbf{Z}|}$ is the coefficients of the parents $\mathbf{Z}$ (Maathuis et al., 2009; Pensar et al., 2020), and some dependence on a is omitted for simplicity. Correspondingly, we express the desired interventional distribution as $Y \mid do(X_{\langle a \rangle} = x_a) \sim \mathrm{N}(\psi_{\langle a \rangle} x_a, \omega^2)$. Claiming no prior knowledge of the interventional variance, we assume a Normal-inverse-gamma (N-Γ^{-1}) conjugate prior $\pi^0_{\langle a \rangle | \mathbf{Z}}$ for $\theta_{\langle a \rangle} = (\psi_{\langle a \rangle}, \omega^2)$:

$$\psi_{\langle a \rangle} \mid \omega^2 \sim \mathrm{N}\left(m_0, \omega^2 \nu_0^{-1}\right), \quad \omega^2 \sim \Gamma^{-1}(u_0, v_0). \tag{3.12}$$

Since in general, the residual variance η^2 in (3.11) is not equivalent to ω^2, we propose the following to estimate ω^2 from observational data.

Proposition 1. Suppose that $\mathbf{X}$ follows the causal structural equation model in (3.10). Let $Y, X \in \mathbf{X}$, and denote by ψ the causal effect of X on Y. Then for any $x \in \mathrm{Dom}(X)$,

$$\mathrm{Var}[Y \mid do(X = x)] = \mathrm{Var}[Y - \psi X].$$

Note that the variance on the right side is with respect to the observational distribution of $\mathbf{X}$. Intuitively, subtracting by ψX negates the noise variances σ^2 in (3.10) propogated through and from X to Y. We include a detailed proof for Proposition 1 in Section 3.8.

Thus, to estimate ω^2 from $\mathcal{D}_0$, we propose the estimator $\hat{\omega}^2 = \sum_i (\tilde{y}_i - \bar{\tilde{y}})^2/(n_0 - |\mathbf{Z}| - 2)$ where $\tilde{y}_i$ are realizations of $\tilde{Y} \coloneqq Y - \hat{\psi}_{\langle a \rangle, \mathrm{bda}}(\mathbf{Z}) X_{\langle a \rangle}$ in $\mathcal{D}_0$, and $n_0 - |\mathbf{Z}| - 2$ is the degrees of freedom resulting from estimating $\bar{\tilde{y}}$ in addition to $|\mathbf{Z}| + 1$ coefficients in (3.11). Accordingly, we design the prior $\omega^2 \sim \Gamma^{-1}(u_0, v_0)$ to have prior mean $\mathrm{E}[\omega^2] = v_0/(u_0 - 1) = \hat{\omega}^2$, resulting in hyperparameters $u_0 = (n_0 - |\mathbf{Z}|)/2$ and $v_0 = \sum_i (\tilde{y}_i - \bar{\tilde{y}})^2/2$. After marginalizing out ω^2, $\psi_{\langle a \rangle} \sim t_{2a_0}(m_0, v_0(u_0\nu_0)^{-1})$, so we set $\mathrm{E}_{\pi^0_{\langle a \rangle | \mathbf{Z}}}[\psi_{\langle a \rangle}] = m_0 = \hat{\psi}_{\langle a \rangle, \mathrm{bda}}(\mathbf{Z})$ and solve to obtain $\nu_0 = v_0/(u_0 \hat{\mathrm{SE}}^2[\hat{\psi}_{\langle a \rangle, \mathrm{bda}}(\mathbf{Z})])$.

To maximally utilize the ensemble data, we further generalize the estimation of $\psi_{\langle a \rangle}$ via regression in (3.11) to include eligible samples of intervention data. This is achieved through the following proposition, which we prove in Section 3.8.

Proposition 2. Suppose that $\mathbf{X}$ follows the causal structural equation model in (3.10), and $X, Y \in \mathbf{X}$. Suppose that $W \in \mathbf{X} \setminus \{X, Y\}$ does not block any directed path from X to Y in $\mathcal{G}$. Then for any $w \in \mathbb{R}$,

$$\frac{\partial}{\partial x} \mathrm{E}[Y \mid do(X = x)] = \frac{\partial}{\partial x} \mathrm{E}[Y \mid do(X = x), do(W = w)].$$

Proposition 2 asserts a simple graphical criterion which, if satisfied, defines an avenue by which information can be shared between arms. In our work, we check the graphical criterion for estimating the causal effect of $X_{\langle a \rangle}$ on Y with interventional data generated

from intervening on X_j as follows. Using another algorithm proposed by Pensar et al. (2020) for computing exact ancestor posterior probabilities, we consider the criterion satisfied at time step t if the event that X_j blocks a directed path from $X_{\langle a \rangle}$ to Y has low posterior probability:

$$P(X_{\langle a \rangle} \rightsquigarrow X_j \rightsquigarrow Y \mid \mathcal{D}[t]) \leq \min\{P(X_{\langle a \rangle} \rightsquigarrow X_j \mid \mathcal{D}[t]), P(X_j \rightsquigarrow Y \mid \mathcal{D}[t])\} \leq \tau \tag{3.13}$$

where $X_j \rightsquigarrow Y$ denotes that X_j is an ancestor of Y and the threshold is set to $\tau = 0.1$ in our application. If (3.13) holds at time step t, we combine the observational data and the data from interventions on X_j when conducting the regression (3.11). While independent samples of observational and interventional data are not guaranteed to have identically distributed errors in the regression (3.11), we provide extensive empirical validation of our proposed regression with ensemble data in Section 3.6.2, confirming indistinguishable performance for the purposes of estimating $\psi_{\langle a \rangle}$ and its sampling variability compared to that of purely observational data.

3.6 Numerical Results

In this section, we provide empirical results validating our methodology on causal bandit problems. Additionally, we present the results from additional experiments designed to evaluate firstly our proposed approximation of the sampling variance of the discrete backdoor adjustment estimator (3.9), and secondly the application of Proposition 2 by way of Gaussian backdoor adjustment with jointly interventional and observational data. The complete code for reproducing the experiments in this section has been made available at the following link:

`https://github.com/jirehhuang/bcb`

3.6.1 Causal Bandit Experiments

For our experiments evaluating the empirical performance of our methods on causal bandit problems, we generated CBN models for $p = 10$ variables with reward variable $Y = X_p$. In order to investigate interesting structures with diverse non-trivial confounding relationships, we randomly generated graph structures using the following process adapted from de Kroon et al. (2022). Given a fixed topological sort of the variables $X_1 \prec \cdots \prec X_p$ where the reward variable is $Y = X_p$, we sequentially considered nodes in reverse topological order: $i = p-1, \ldots, 1$. We uniformly sampled the maximum out-degree of X_i, denoted d_i, from 1 to $p - i$. Then, for d_i times, we randomly selected X_j from $\{X_j \in \mathbf{X} : X_i \prec X_j\}$, adding $X_i \to X_j$ to the graph only if the edge was not already present and $|\mathbf{Pa}_j^{\mathcal{G}}| < 3$. We imposed an additional requirement that $|\mathbf{Pa}_p^{\mathcal{G}}| = 3$, randomly adding parents if necessary. If the generated structure consisted of multiple disconnected components, we rejected the structure and reattempted the process.

The conditional probability distributions of each CBN were likewise generated randomly. For discrete networks, the variables were all assumed to be binary, and the conditional probability tables were randomly generated uniformly and normalized, and were accepted only if for every edge $X_j \to X_i$, there is a sufficiently large causal effect, with $|P[X_i = x_i \mid do(X_j = x_j)] - P(X_i = x_i)| \geq 0.05$ for some $x_i \in \mathrm{Dom}(X_i)$ and $x_j \in \mathrm{Dom}(X_j)$. Additionally, we required the marginal probability of any single discrete level to be at least 0.01, and that the reward signal of the optimal intervention a^* be sufficiently large with respect to the observational mean: $\mu^* - \mathrm{E}[Y] \geq 0.05$. For Gaussian networks, according to the model expressed in (3.10), we sampled coefficients uniformly from $[-1, -0.5] \cup [0.5, 1]$ for $X_i \in \mathbf{Pa}_j^{\mathcal{G}}$ and standard deviations from $[\sqrt{0.5}, 1]$, and we normalized the system to have unit variance.

We evaluated our BBB methodology against algorithms designed to optimize cumulative regret, including popular standard MAB algorithms Bayes-UCB, TS, and UCB that do not utilize causal assumptions (see Section 3.4). Additionally, we compared against what can

be interpreted as a highly optimistic version of the central node approach by Lu et al. (2021), introduced in Section 3.1, by presupposing knowledge of the direct causes of the reward variable. In particular, for Bayes-UCB*, TS*, and UCB*, we executed the respective algorithms over the reduced action set $\mathcal{A}' = \{a \in \mathcal{A} : \langle a \rangle \in \mathbf{Pa}_p^{\mathcal{G}}\}$. We found that $\langle a^* \rangle \in \mathbf{Pa}_p^{\mathcal{G}}$ held for 98% of the discrete models that we randomly generated, though only for 65% of the random Gaussian models. Accordingly, for the cases where $\langle a^* \rangle \notin \mathbf{Pa}_p^{\mathcal{G}}$, we redefined the optimal intervention to $a^* = \text{argmax}_{a \in \mathcal{A}'} \mu_a$ when evaluating the regret of TS*, Bayes-UCB*, and UCB*.

Using the process described above, we generated 100 CBN models for each distributional setting. For each CBN, we executed the competing methods 10 times but our BBB methods only 5 times due to their greater computational expense, with $T = 5000$ time steps. Atomic interventions as described in Section 3.5 were allowed on all variables excluding the reward variable, resulting in $|\mathcal{A}| = 2(p-1) = 18$ actions. Note that in the Gaussian setting, there are effectively $|\mathcal{A}| = 9$ actions given that interventional data on the same variable may be combined as discussed in Section 3.5.2, which we implement for the competing methods as well. The results presented are averaged across all simulations for each time step, with the cumulative regret normalized by the optimal reward μ^* to ensure that each CBN model contributes comparably.

In preference to the competing methods, we tuned for their best-performing parameters where relevant and applied them to our BBB implementations. For Bayes-UCB(*), the best quantile constant in (3.8) was $c = 0$, in agreement with the empirical recommendation by Kaufmann et al. (2012). The best exploration parameter for UCB in (3.6) was $c = 1/(2\sqrt{2})$ for UCB(*) in the discrete setting. In the Gaussian setting, UCB and UCB* preferred $c = 1/2$ and $c = 1/\sqrt{2}$, respectively, the latter of which we applied for BBB. We used standard uninformative priors for TS(*), with $\alpha_0 = \beta_0 = 1$ for the Beta prior and $m_0 = 0$, $\nu_0 = 1$, and $u_0 = v_0 = 1$ for the N-Γ^{-1} prior. For BBB, we computed exact parent set

probabilities (3.4) using the program[1] implementing the efficient algorithm developed by and applied in Pensar et al. (2020), restricting the maximum size of parent sets to three and using the Bayesian Dirichlet equivalent uniform and Bayesian Gaussian equivalent scores. For the Gaussian setting, we checked the graphical criterion in Proposition 2 according to (3.13) with $\tau = 0.1$.

While we focused in Section 3.3 on designing the marginal posteriors according to (3.3), a notable difference between our proposed Bayesian CB framework and the Bayesian MAB approach described in Section 3.2 is that in our design, the posterior distribution is not modular, with the marginals $(\pi_a^t)_{a\in\mathcal{A}}$ mutually dependent on the distribution of graph structures. However, because of software limitations and for simplicity, we sampled the criterion $U_a(t)$ for each arm independently in the implementation of BBB-TS in our experiments (line 7 in Algorithm 3.1). Although preliminary results have shown the difference in empirical performance to be negligible, a more precise implementation would first sample a DAG $\mathcal{G}$ from the posterior distribution $P(\mathcal{G} \mid \mathcal{D}[t])$ and subsequently for each arm $a \in \mathcal{A}$, sample $U_a(t)$ from $\pi^t_{a|\mathbf{Pa}^{\mathcal{G}}_{\langle a\rangle}}$.

The empirical cumulative regret results in Figure 3.1 demonstrate that in both the discrete and Gaussian settings and for all algorithms, our BBB methodology is able to reliably outperform the non-causal variants with finite samples of observational data. The improvement increases monotonically with increasing sample sizes of observational data (n_0). While corresponding variants of Bayes-UCB and TS perform comparably, UCB achieves substantially lower regret because the parameter c in (3.6) was tuned to maintain a balance between exploration and exploitation that is most empirically preferred. In particular, UCB is able to avoid excessive exploration by scaling its padding term with a relatively small constant, whereas Bayes-UCB maintains a relatively high minimum exploration rate according to its formulation in (3.8), as does TS.

[1]Pensar et al. (2020) distributed their code under the MIT License at `https://github.com/jopensar/BIDA`.

In comparison to the optimistic central node versions of the algorithms, BBB generally achieves lower cumulative regret with $n_0 \geq 800$ in the discrete setting and $n_0 \geq 40$ in the Gaussian setting. Recall that, in practical applications, the central node approach relies on the availability of large-sample observational data as well as a sequence of interventions to recover the reward generating variables $\mathbf{Pa}_p^{\mathcal{G}}$. Based on our simulation settings, this reduces the action set from $|\mathcal{A}| = 18$ arms to only $|\mathcal{A}'| = 2|\mathbf{Pa}_p^{\mathcal{G}}| = 6$ arms, and we additionally restrict $a^* \in \mathcal{A}'$ to evaluate the regret. Furthermore, the regret results reported for these methods do not include the interventions required to identify $\mathbf{Pa}_p^{\mathcal{G}}$, thus representing a kind of best case scenario for the central node approach. In contrast, our methodology derives substantial benefit from modest amounts of observational data samples n_0.

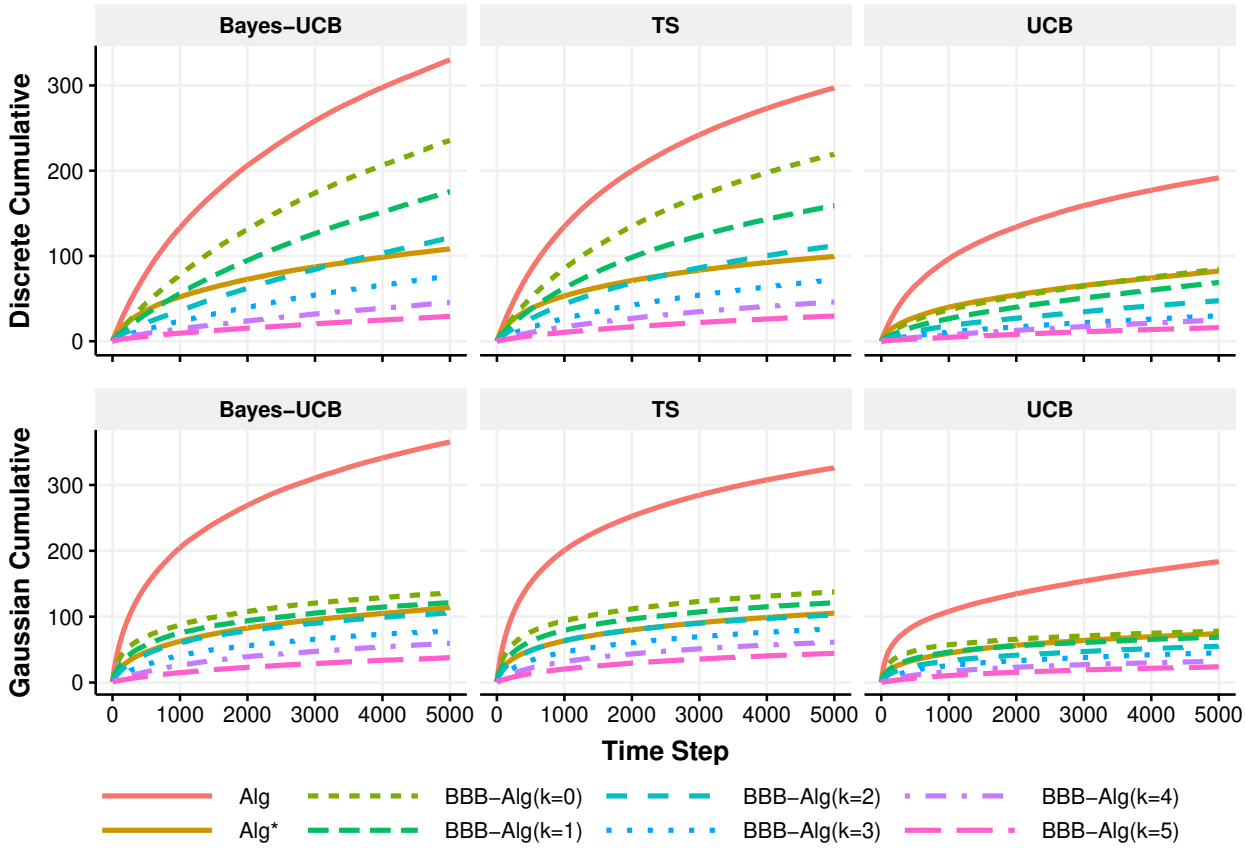


Figure 3.1: Cumulative regret results against $T = 5000$ time steps comparing Alg, Alg*, and BBB-Alg for Alg $\in \{\text{Bayes-UCB}, \text{TS}, \text{UCB}\}$. BBB methods were executed with $n_0 = 100 \cdot 2^k$ in the discrete setting and $n_0 = 10 \cdot 2^k$ in the Gaussian setting

We find that our BBB methods are able to outperform standard algorithms and perform competitively against the optimistic central node approach even when the competing methods are given n_0 time steps to explore arms before incurring regret. To compensate for the fact that BBB utilizes n_0 samples of observational data prior to investigating arms, we present the results where the competing algorithms TS(*) and (Bayes-)UCB(*) are given a head start of $n_0 \in \{100 \cdot 2^k : k = 0, 1, \ldots, 5\}$ time steps to explore arms before incurring regret. The results for the discrete setting are shown in Figure 3.2. The Gaussian results are omitted because $n_0 \leq 320$ is relatively small, so the head start does not offer substantial benefit to the competing methods. In all cases, BBB still significantly outperforms the standard algorithms TS and (Bayes-)UCB given the head start. Given sufficient samples of observational data, BBB still performs comparably to if not better than the optimistic central node variants in terms of cumulative regret, for which the head start is only an additional undeserved advantage given that they already require significantly more observational

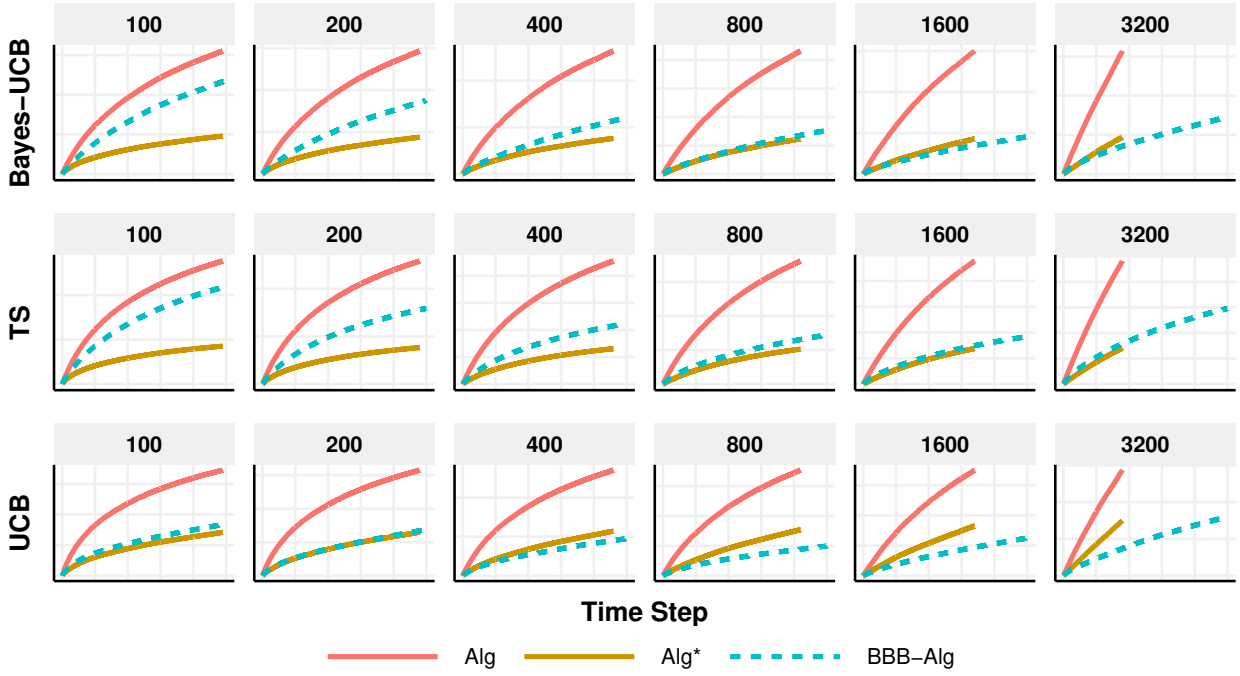


Figure 3.2: Discrete cumulative regret for Alg $\in \{$Bayes-UCB, TS, UCB$\}$ with a head start of $n_0 \in \{100 \cdot 2^k : k = 0, 1, \ldots, 5\}$ time steps for competing methods

data as well as additional interventions.

In addition to the cumulative regret performance, it is of interest to consider the structure identification behavior of the BBB approach in our experiments. We measure the concentration of the posterior probability across DAGs $\mathcal{G}$ with respect to the underlying causal graph $\mathcal{G}^*$ using the edge support sum of absolute errors (ESSAE), which is given at time t by

$$\sum_{i=1}^{p} \sum_{j \neq i} \left| P(j \in \mathbf{Pa}_i^{\mathcal{G}} \mid \mathcal{D}[t]) - \mathbb{1}\left\{ j \in \mathbf{Pa}_i^{\mathcal{G}^*} \right\} \right|.$$

This quantity may be understood as a probabilistic version of the structural hamming distance, a common metric in Bayesian network structure learning literature. Lower ESSAE corresponds to greater concentration of the posterior probability around the causal graph $\mathcal{G}^*$. The results are provided in Figure 3.3.

In the discrete results for BBB-Bayes-UCB and BBB-TS, the initial ESSAE is unsurprisingly lower for the larger sample sizes, but the trend quickly reverses as the time steps progress. This effect is also observed occurring in the Gaussian results, but at an accelerated pace. This behavior is perhaps best understood in complement to the cumulative regret results in Figure 3.1. If P is faithful to $\mathcal{G}$, then if n_0 is large, the structure prior $P(\mathcal{G} \mid \mathcal{D}_0)$ is expected to concentrate around the Markov equivalence class, which entails identification of the skeleton and in general, partial identification of the orientations. Additionally, the conditional priors $\pi^0_{a|\mathbf{Z}}$ are precise models, allowing BBB to quickly identify and select arm(s) $a \in \mathcal{A}$ with small regret $\mu^* - \mu_a$, which has the effect of clarifying the orientation of edges incident to such $X_{\langle a \rangle}$. The policies take no interest in determining the orientation of the remaining edges if the uncertainty does not indicate potential to identify more profitable actions. In contrast, when n_0 is small, the greater uncertainty in both the structure prior and the conditional priors encourage the exploration of many different arms, thus incurring greater cumulative regret. In addition to clarifying the orientation of the incident edges, selecting arm(s) $a \in \mathcal{A}$ contributes to identifying the direct edge connections excluding those

from $\mathbf{Pa}^{\mathcal{G}}_{\langle a \rangle}$ to $X_{\langle a \rangle}$, as can be seen in (3.5). Thus, the skeleton is recovered and more edge orientations are identified than in the case where n_0 is large, achieving lower ESSAE at the cost of greater cumulative regret. Notably, while this reversal appears to be absent in the discrete results for BBB-UCB, in actuality it has simply not yet been realized even after $T = 5000$ time steps due to the small exploration constant c in (3.7).

3.6.2 Backdoor Adjustment Experiments

We first describe and present experiments evaluating the behavior of $\hat{\mu}_{a,\text{bda}}(\mathbf{Z})$ where $\mathbf{Z} = \mathbf{Pa}^{\mathcal{G}}_{\langle a \rangle}$ as in (3.9), as well as our proposed approximation of its variance, derived in detail in Section 3.9. Four variance estimation methods were investigated. In the naive approach,

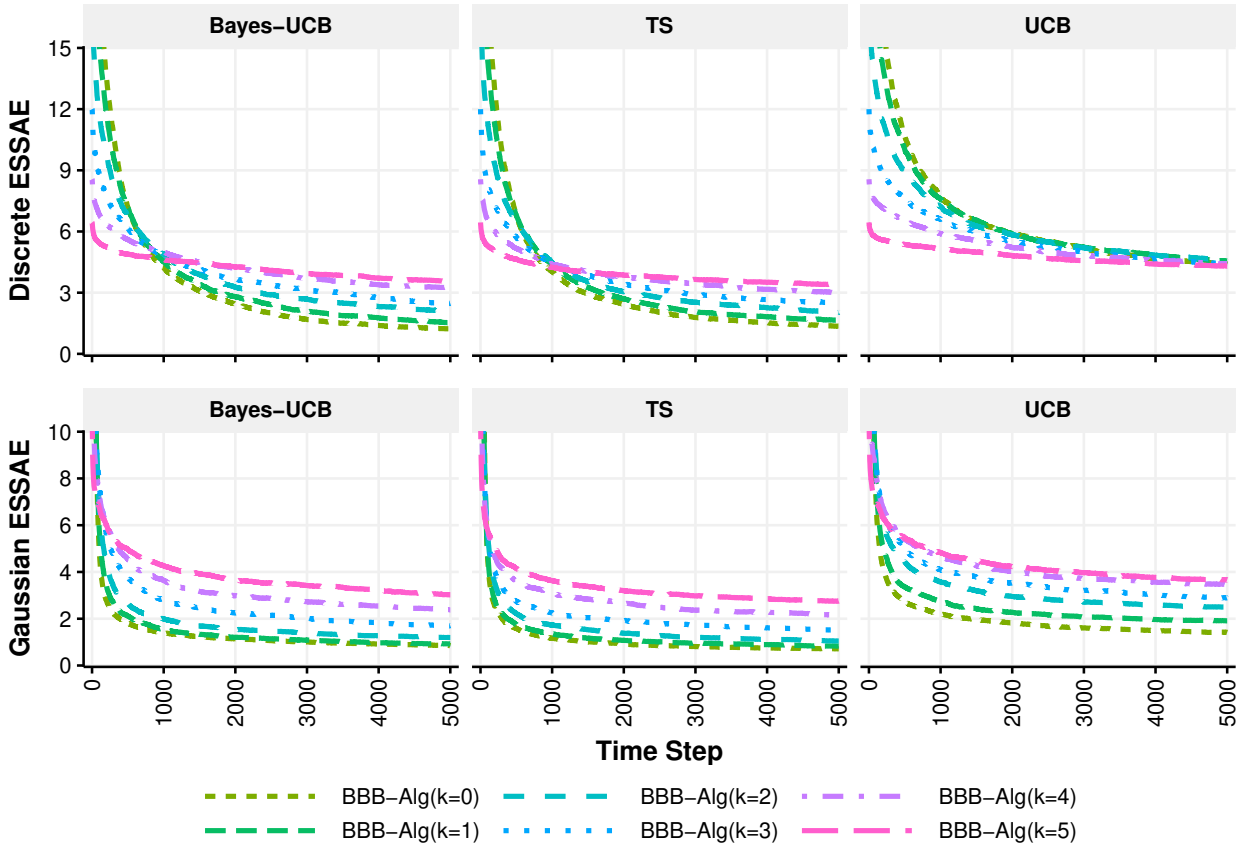


Figure 3.3: Discrete ($n_0 = 100 \cdot 2^k$) and Gaussian ($n_0 = 10 \cdot 2^k$) results of the ESSAE of the full graph structure for BBB-Alg, Alg $\in$ {Bayes-UCB, TS, UCB}

$\hat{\mu}_{a,\text{bda}}(\mathbf{Z})$ is treated as a conditional proportion as is the case when $|\mathbf{Z}| = 0$, and the variance is estimated as $\hat{\mu}_{a,\text{bda}}(\mathbf{Z})[1 - \hat{\mu}_{a,\text{bda}}(\mathbf{Z})]/n[x_a]$ where $n[x_a]$ is the number of samples of data where $X_{\langle a \rangle} = x_a$. The sampling approach estimates the variance from samples from the population distribution, and the bootstrap approach conducts resampling from each sample distribution, each with 10^3 repetitions.

The generation of discrete CBNs for the simulation scenarios was designed as follows. The graph structure was generated simply by initializing a structure where there is a direct edge from the intervened node $X_{\langle a \rangle}$ to the reward variable Y and $X_{\langle a \rangle}$ has $|\mathbf{Z}| = m$ parents. For each parent $X_j \in \mathbf{Z}$, an edge $X_j \to Y$ was randomly added with 0.5 probability to create backdoor paths. Finally, conditional probability tables were generated uniformly as described in Section 3.6.

For observational sample sizes $n_0 \in \{100 \cdot 2^k : k = 0, 1, \ldots, 5\}$ and parent set sizes $|\mathbf{Z}| \in \{0, 1, 2, 3\}$, 10^3 scenarios were created by randomly generating CBNs as described above and the methods were assessed under each scenario through the following process. First, 10^6 datasets were generated, each with n_0 samples of observational data, and for each dataset, $\hat{\mu}_{a,\text{bda}}(\mathbf{Z})$ was computed for some arbitrary $x_a \in \text{Dom}(X_{\langle a \rangle})$. Then, for each of the four methods, the variance was estimated corresponding to the first 10^3 estimates of $\hat{\mu}_{a,\text{bda}}(\mathbf{Z})$, and from those the 2 standard deviation interval coverage probability of the true μ_a was computed.

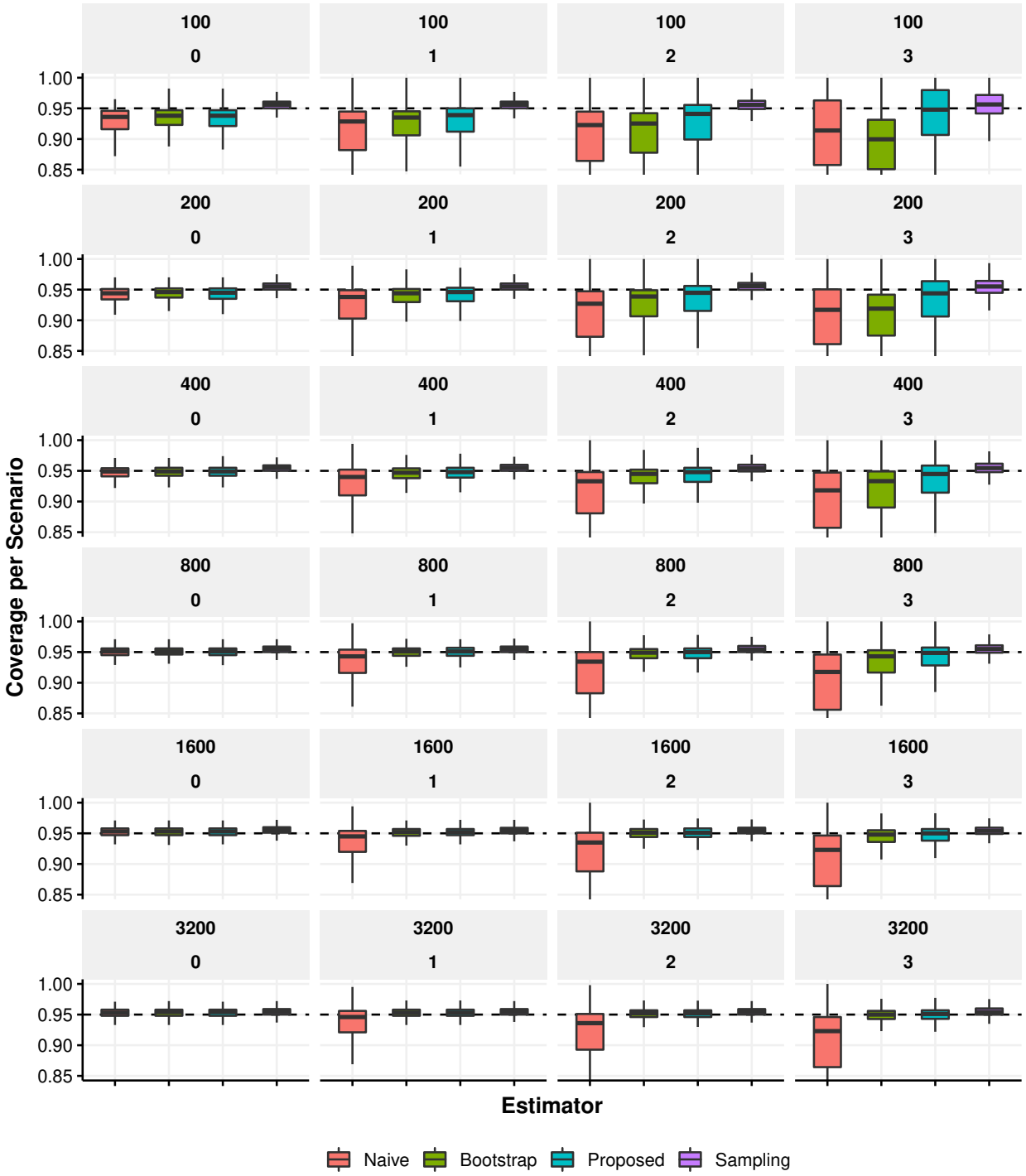


Figure 3.4: Coverage probability per scenario using various estimators of $\text{Var}[\hat{\mu}_{a,\text{bda}}(\mathbf{Z})]$ across $n_0 \in \{100 \cdot 2^k : k = 0, 1, \ldots, 5\}$ samples of observational data and $|\mathbf{Z}| \in \{0, 1, 2, 3\}$ adjustment set sizes

The estimator $\hat{\mu}_{a,\text{bda}}(\mathbf{Z})$ itself was found to be generally unbiased, with the average of the 10^6 estimates deviating from the true μ_a by less than 2% in over 99% of the 24,000 scenarios. The coverage probability results are shown in Figure 3.4, where each boxplot visualizes the coverage probability of a method across 10^3 scenarios randomly generated under the given simulation setting. The outliers and invalid values, which typically corresponded to extreme scenarios, were removed. The naive approach is only correct when $|\mathbf{Z}| = 0$ and performs poorly when otherwise. The general results may be summarized as Naive $<$ Bootstrap $\approx$ Proposed $<$ Sampling, though our proposed estimator appears to outperform the bootstrap approach for larger $|\mathbf{Z}|$ and perform comparably with the population sampling approach for larger n_0.

Next, we empirically validate our methodology of conducting the regression (3.11) with jointly interventional and observational data to estimate $\psi_{\langle a \rangle}$, as discussed in Section 3.5.2. In particular, we compare the coverage probability of $\hat{\psi}_{\langle a \rangle,\text{bda}}(\mathbf{Z})$ where $\mathbf{Z} = \mathbf{Pa}^{\mathcal{G}}_{\langle a \rangle}$ estimated using purely observational data and ensemble data. The ensemble data was generated by allowing each data sample to be generated by one of the possible interventions $\{do(X_j = x_j) : X_j \in \mathbf{Z}, x_j \in \{-1, 1\}\}$ or by passive observation, with equal probability given to each of the $2|\mathbf{Z}| + 1$ options.

For sample sizes $n \in \{10 \cdot 2^k : k = 0, 1, \ldots, 5\}$ and parent set sizes $|\mathbf{Z}| \in \{1, 2, 3, 4\}$, 10^3 scenarios were created by randomly generating CBNs. The network structures were generated as described for the discrete case, and the parameters as in Section 3.6. Each data generation method was evaluated for each scenario by generating 10^5 datasets with n samples and estimating $\hat{\psi}_{\langle a \rangle,\text{bda}}(\mathbf{Z})$ and $\hat{\text{SE}}^2[\hat{\psi}_{\langle a \rangle,\text{bda}}(\mathbf{Z})]$ for each dataset by conducting the regression (3.11). From those estimates, 95% confidence interval coverage probabilities were computed for each scenario.

The average of the 10^5 estimates of $\hat{\psi}_{\langle a \rangle,\text{bda}}(\mathbf{Z})$ deviated from the true $\psi_{\langle a \rangle}$ by at most 0.9% across all 24,000 simulation scenarios for both data generation methods. The coverage probability results are shown in Figure 3.5. Since the results did not vary across parent

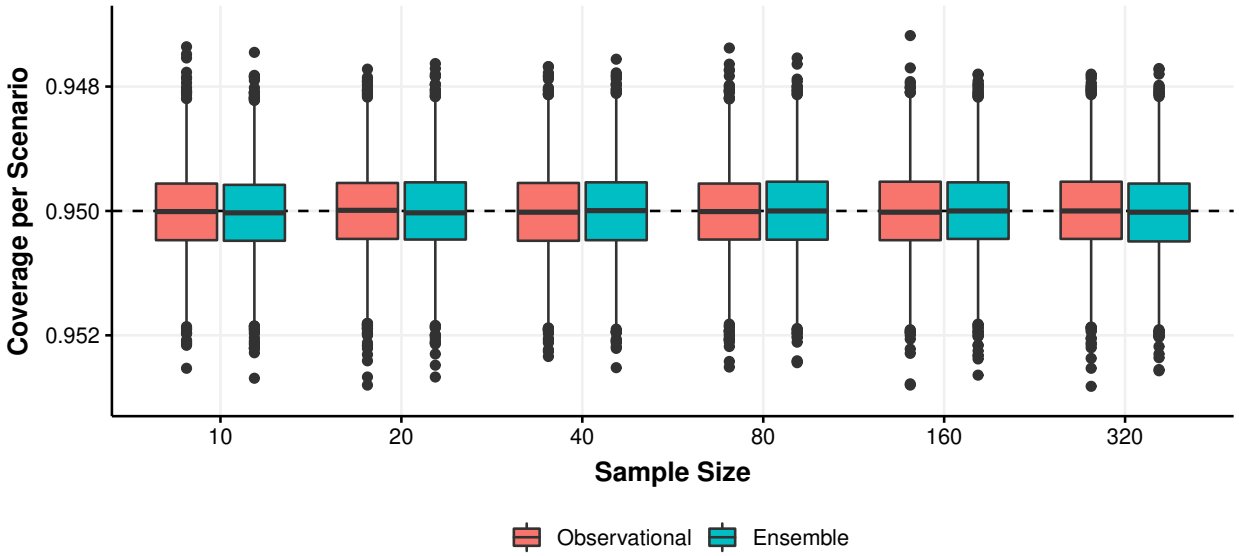


Figure 3.5: Coverage probability per simulation scenario across sample sizes for observational and ensemble data generating methods

set sizes, each boxplot visualizes the coverage probability of a method across the $4,000$ simulation scenarios at each sample size. It is easy to see equivalent performance of the estimator computed with ensemble data compared to observational data, with consistent coverage across all sample sizes.

3.7 Conclusion

In this chapter, we proposed the BBB framework for enhancing experimental investigations with observational data. BBB consists of an aggregation of various strategies for estimating and modeling the parameters of interest with jointly interventional and observational data in order to efficiently utilize all available data to inform exploitation and exploration. Applied in our methodology but also of independent interest, we derived a well-performing approximation for the variance of the discrete backdoor adjustment estimator, and in the Gaussian setting, we characterized the interventional variance using the observational distribution and proposed a simple graphical criterion for sharing information between arms.

We empirically validated our proposed algorithms through extensive numerical experiments against standard MAB algorithms as well as a generously optimistic version of a recently proposed CB approach.

3.8 Proofs

In this section, we prove Proposition 1 and Proposition 2.

Proof of Proposition 1. Let $\mathbf{Z}$ be the parent set of X. Then by a special case of (3.1),

$$\begin{aligned} p(y \mid do(x)) &= \int p(y \mid x, \mathbf{z}) p(\mathbf{z}) d\mathbf{z} \\ &= \int \phi(y \mid \psi x + \boldsymbol{\gamma}^\top \mathbf{z}, \sigma^2) \phi(\mathbf{z} \mid 0, \Sigma_{\mathbf{Z}}) d\mathbf{z} \\ &= \phi(y \mid \psi x, \boldsymbol{\gamma}^\top \Sigma_{\mathbf{Z}} \boldsymbol{\gamma} + \sigma^2), \end{aligned}$$

where $\phi(\cdot \mid \mu, \Sigma)$ is the probability density function of $\mathrm{N}(\mu, \Sigma)$ and $\Sigma_{\mathbf{Z}}$ is the covariance matrix of $\mathbf{Z}$. Thus,

$$Y \mid do(X = x) \sim \mathrm{N}(\psi x, \boldsymbol{\gamma}^\top \Sigma_{\mathbf{Z}} \boldsymbol{\gamma} + \sigma^2).$$

Now representing $Y \sim X + \mathbf{Z}$ by a linear regression:

$$Y = \psi X + \boldsymbol{\gamma}^\top \mathbf{Z} + \varepsilon,$$

where $\varepsilon \sim N(0, \sigma^2) \perp \mathbf{Z} \sim N(0, \Sigma_{\mathbf{Z}})$. Then we have

$$\begin{aligned} \mathrm{Var}(Y - \psi X) &= \mathrm{Var}(\boldsymbol{\gamma}^\top \mathbf{Z} + \varepsilon) \\ &= \boldsymbol{\gamma}^\top \Sigma_{\mathbf{Z}} \boldsymbol{\gamma} + \sigma^2 = \mathrm{Var}(Y \mid do(X = x)). \end{aligned}$$

□

Proof of Proposition 2. The result follows straightforwardly from a simple graphical argument. Let $\Xi_{XY}^{\mathcal{G}}$ denote the distinct directed paths from X to Y in the causal graph $\mathcal{G}$ given the model (3.10), where $\xi \in \Xi_{XY}^{\mathcal{G}}$ consists of all the directed edges $i \to j \in \mathbf{E}$ on the given path from X to Y. Then the causal effect of X on Y can be expressed as the sum of propagated direct effects along all directed paths from X to Y:

$$\psi_{XY} \coloneqq \frac{\partial}{\partial x}\mathrm{E}[Y \mid do(X = x)] = \sum_{\xi \in \Xi_{XY}^{\mathcal{G}}} \prod_{i \to j \in \xi} \beta_{ij}.$$

We denote the variables under the intervention $do(W = w)$, $w \in \mathbb{R}$ as $\tilde{\mathbf{X}}$, with resulting causal model

$$\tilde{X}_j = \sum_{i=1}^{p} \tilde{\beta}_{ij}\tilde{X}_i + \tilde{\varepsilon}_j, \quad j = 1 \ldots, p,$$

where

$$\tilde{\beta}_{ij} = \begin{cases} 0 & \text{if } X_j = W \\ \beta_{ij} & \text{otherwise,} \end{cases} \qquad \tilde{\varepsilon}_j = \begin{cases} w & \text{if } X_j = W \\ \varepsilon_j & \text{otherwise.} \end{cases}$$

The corresponding causal graph for $\tilde{\mathbf{X}}$ is the mutilated graph $\tilde{\mathcal{G}}$ resulting from deleting all edges into W. The causal effect of $\tilde{X}$ on $\tilde{Y}$ is then

$$\psi_{\tilde{X}\tilde{Y}} \coloneqq \frac{\partial}{\partial x}\mathrm{E}[\tilde{Y} \mid do(\tilde{X} = x)] = \frac{\partial}{\partial x}\mathrm{E}[Y \mid do(X = x), do(W = w)] = \sum_{\xi \in \Xi_{XY}^{\tilde{\mathcal{G}}}} \prod_{i \to j \in \xi} \tilde{\beta}_{ij}.$$

Since W does not block any directed path from X to Y, the mutilated graph $\tilde{\mathcal{G}}$ retains all the directed paths from X to Y in $\mathcal{G}$, so $\Xi_{XY}^{\tilde{\mathcal{G}}} = \Xi_{XY}^{\mathcal{G}}$. By the same reasoning, $\tilde{\beta}_{ij} = \beta_{ij}$ for all $i \to j \in \xi$ where $\xi \in \Xi_{XY}^{\tilde{\mathcal{G}}}$. Therefore, for any $w \in \mathbb{R}$,

$$\frac{\partial}{\partial x}\mathrm{E}[Y \mid do(X = x)] = \frac{\partial}{\partial x}\mathrm{E}[Y \mid do(X = x), do(W = w)].$$

□

3.9 Derivation of the Discrete Backdoor Adjustment Variance Approximation

In this section, we derive the approximation of the sampling variance of (3.9):

$$\hat{\mu}_{a,\text{bda}}(\mathbf{Z}) = \frac{1}{n_0} \sum_{\mathbf{z}} \frac{n_0[1, x_a, \mathbf{z}] n_0[\mathbf{z}]}{n_0[x_a, \mathbf{z}]}.$$

3.9.1 Introduction

For simplicity, we redefine some notation. The backdoor adjustment to estimate the interventional distribution of $Y \mid do(X = x)$ with parent set $\mathbf{Z} = \mathbf{Pa}_X^{\mathcal{G}}$ with r parent configurations is given by:

$$P[Y = y \mid do(X = x)] = \sum_{\mathbf{z}} P(Y = y \mid X = x, \mathbf{Z} = \mathbf{z}) P(\mathbf{Z} = \mathbf{z}).$$

Empirically, given n samples of observational data, this quantity is estimated using counts:

$$\hat{P}[Y = y \mid do(X = x)] = \sum_{\mathbf{z}} \frac{n[y, x, \mathbf{z}]}{n[x, \mathbf{z}]} \frac{n[\mathbf{z}]}{n} = \frac{1}{n} \sum_{\mathbf{z}} \frac{n[y, x, \mathbf{z}] n[\mathbf{z}]}{n[x, \mathbf{z}]} \tag{3.14}$$

where $n[y, x, \mathbf{z}]$ represents the number of samples in which $Y = y$, $X = x$, and $\mathbf{Z} = \mathbf{z}$, with corresponding definitions for $n[x, \mathbf{z}]$ and $n[\mathbf{z}]$. The joint probability distribution of X, Y, and $\mathbf{Z}$ may be lumped into a multinomial random vector $\mathbf{N} = (N_1, N_1', N_1'', \ldots, N_r, N_r', N_r'') \in \mathbb{R}^{3r}$ where for $i = 1, \ldots, r$,

$$N_i = n[y, x, \mathbf{z}_i], \quad N_i' = n[\neg y, x, \mathbf{z}_i], \quad N_i'' = n[\neg x, \mathbf{z}_i].$$

Note that $N_i + N_i' + N_i'' = n[\mathbf{z}_i]$, so $\sum_{i=1}^{r}(N_i + N_i' + N_i'') = n$, so $\mathbf{N}$ may be thought of as a repartitioning of the joint probability distribution of X, Y, and $\mathbf{Z}$ into $3r$ disjoint levels:

$$\begin{aligned}
&\mathbf{N} = (N_1, N_1', N_1'', \ldots, N_r, N_r', N_r'') \sim \text{Multinom}(n, \mathbf{p}),\\
&\mathbf{p} = (p_1, p_1', p_1'', \ldots, p_r, p_r', p_r''), \quad \text{where}\\
&p_i = \mathrm{E}\left[\frac{n[y, x, \mathbf{z}_i]}{n}\right], \quad p_i' = \mathrm{E}\left[\frac{n[\neg y, x, \mathbf{z}_i]}{n}\right], \quad p_i'' = \mathrm{E}\left[\frac{n[\neg x, \mathbf{z}_i]}{n}\right] \quad \text{for } i = 1, \ldots, r.
\end{aligned} \tag{3.15}$$

The advantage of such a representation is so that for each $\mathbf{z}_i$, the term within the summation may be expressed as a function of three disjoint elements of a multinomial random vector:

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{r}\frac{n[y, x, \mathbf{z}_i]n[\mathbf{z}_i]}{n[x, \mathbf{z}_i]} &= \frac{1}{n}\sum_{i=1}^{r}\frac{n[y, x, \mathbf{z}_i]\left(n[y, x, \mathbf{z}_i] + n[\neg y, x, \mathbf{z}_i] + n[\neg x, \mathbf{z}_i]\right)}{n[y, x, \mathbf{z}_i] + n[\neg y, x, \mathbf{z}_i]}\\
&= \frac{1}{n}\sum_{i=1}^{r}\frac{N_i(N_i + N_i' + N_i'')}{N_i + N_i'}.
\end{aligned} \tag{3.16}$$

Note that each term is not straightforward to compute. An obvious challenge is that the denominator of each term in the summation in (3.16) can be zero, so there is no analytical solution for its mean, variance, and covariance.

3.9.2 Taylor Series Expansion for Ratio Distribution

To circumvent this challenge, we approximate the ratio in (3.16) with the Taylor series approximation. We begin by defining

$$\begin{aligned}
M_i &= \frac{N_i(N_i + N_i' + N_i'')}{n^2},\\
W_i &= \frac{N_i + N_i'}{n},\\
Q_i &= f(M_i, W_i) = \frac{M_i}{W_i}.
\end{aligned}$$

This allows us to express the variance of (3.16) in terms of Q_i:

$$\begin{aligned}\mathrm{Var}\left[\hat{P}[Y = y \mid do(X = x)]\right] &= \mathrm{Var}\left[\sum_{i=1}^{r} Q_i\right] \\ &= \sum_{i}^{r} \mathrm{Var}\left[Q_i\right] + 2\sum_{i=1}^{r}\sum_{j>i} \mathrm{Cov}\left[Q_i, Q_j\right].\end{aligned} \tag{3.17}$$

By Taylor series expansion around $\mu_i = (\mu_{M_i}, \mu_{W_i}) = (\mathrm{E}[M_i], \mathrm{E}[W_i])$:

$$\begin{aligned}Q_i &= f(M_i, W_i) \\ &\approx f(\mu_i) + (M_i - \mu_{M_i})\frac{\partial f}{\partial M_i}(\mu_i) + (W_i - \mu_{W_i})\frac{\partial f}{\partial W_i}(\mu_i) \\ &\quad + \frac{1}{2}(M_i - \mu_{M_i})^2\frac{\partial^2 f}{\partial M_i^2}(\mu_i) + \frac{1}{2}(W_i - \mu_{W_i})^2\frac{\partial^2 f}{\partial W_i^2}(\mu_i) \\ &\quad + (M_i - \mu_{M_i})(W_i - \mu_{W_i})\frac{\partial^2 f}{\partial M_i \partial W_i}(\mu_i),\end{aligned} \tag{3.18}$$

where

$$\begin{aligned}\frac{\partial f}{\partial M_i}(M_i, W_i) &= \frac{1}{W_i}, & \frac{\partial^2 f}{\partial M_i^2}(M_i, W_i) &= 0, \\ \frac{\partial f}{\partial W_i}(M_i, W_i) &= -\frac{M_i}{W_i^2}, & \frac{\partial^2 f}{\partial W_i^2}(M_i, W_i) &= \frac{2M_i}{W_i^3}, \\ \frac{\partial^2 f}{\partial M_i \partial W_i}(M_i, W_i) &= \frac{\partial^2 f}{\partial W_i \partial M_i}(M_i, W_i) = \frac{1}{W_i^2}\end{aligned} \tag{3.19}$$

Given (3.18), we obtain an approximate expected value:

$$\mathrm{E}[Q_i] \approx f(\mu_i) + \frac{1}{2}\frac{\partial^2 f}{\partial M_i^2}(\mu_i)\mathrm{Var}[M_i] + \frac{1}{2}\frac{\partial^2 f}{\partial W_i^2}(\mu_i)\mathrm{Var}[W_i] + \frac{\partial^2 f}{\partial M_i \partial W_i}(\mu_i)\mathrm{Cov}[M_i, W_i]. \tag{3.20}$$

For variance and covariance, we use a simpler approximation:

$$Q_i = f(M_i, W_i) \approx f(\mu_i) + (M_i - \mu_{M_i})\frac{\partial f}{\partial M_i}(\mu_i) + (W_i - \mu_{W_i})\frac{\partial f}{\partial W_i}(\mu_i), \tag{3.21}$$

resulting in

$$\begin{aligned}\mathrm{Var}[Q_i] \approx \frac{\partial f}{\partial M_i}(\mu_i)^2\mathrm{Var}[M_i] + \frac{\partial f}{\partial W_i}(\mu_i)^2\mathrm{Var}[W_i] \\ + 2\frac{\partial f}{\partial M_i}(\mu_i)\frac{\partial f}{\partial W_i}(\mu_i)\mathrm{Cov}[M_i, W_i],\end{aligned} \tag{3.22}$$

and

$$\begin{aligned}\mathrm{E}[Q_iQ_j] \approx f(\mu_i)f(\mu_j) \\ + \frac{\partial f}{\partial M_i}(\mu_i)\frac{\partial f}{\partial M_j}(\mu_j)\mathrm{Cov}[M_i, M_j] + \frac{\partial f}{\partial M_i}(\mu_i)\frac{\partial f}{\partial W_j}(\mu_j)\mathrm{Cov}[M_i, W_j] \\ + \frac{\partial f}{\partial W_i}(\mu_i)\frac{\partial f}{\partial M_j}(\mu_j)\mathrm{Cov}[W_i, M_j] + \frac{\partial f}{\partial W_i}(\mu_i)\frac{\partial f}{\partial W_j}(\mu_j)\mathrm{Cov}[W_i, W_j],\end{aligned}$$

so

$$\begin{aligned}\mathrm{Cov}[Q_i, Q_j] &= \mathrm{E}[Q_iQ_j] - \mathrm{E}[Q_i]\mathrm{E}[Q_j] \\ &= \frac{\partial f}{\partial M_i}(\mu_i)\frac{\partial f}{\partial M_j}(\mu_j)\mathrm{Cov}[M_i, M_j] + \frac{\partial f}{\partial M_i}(\mu_i)\frac{\partial f}{\partial W_j}(\mu_j)\mathrm{Cov}[M_i, W_j] \\ &\quad + \frac{\partial f}{\partial W_i}(\mu_i)\frac{\partial f}{\partial M_j}(\mu_j)\mathrm{Cov}[W_i, M_j] + \frac{\partial f}{\partial W_i}(\mu_i)\frac{\partial f}{\partial W_j}(\mu_j)\mathrm{Cov}[W_i, W_j].\end{aligned} \tag{3.23}$$

In what follows, we first derive important quantities from the multinomial distribution in Section 3.9.3 and apply them to compute the quantities in (3.17).

3.9.3 Multinomial Derivations

For this subsection, in an abuse of notation, let $\mathbf{N} = (N_1, \ldots, N_r) \sim \mathrm{Multinom}(n, \mathbf{p})$ and $u, v, w, x \in \{1, \ldots, r\}$ are distinct values. It is well-known that $\mathrm{E}[N_u] = np_u$, $\mathrm{Var}[N_u] = np_u(1 - p_u)$, and $\mathrm{Cov}(N_u, N_v) = -np_up_v$. Furthermore,

$$\begin{aligned}\mathrm{E}[N_uN_v] &= \mathrm{Cov}[N_u, N_v] + \mathrm{E}[N_u]\mathrm{E}[N_v] \\ &= n(n-1)p_up_v,\end{aligned} \tag{3.24}$$

and the first four moments from derivating the moment generating function are:

$$
\begin{aligned}
\mathrm{E}[N_u] &= np_u, \\
\mathrm{E}[N_u^2] &= n(n-1)p_u^2 + \mathrm{E}[N_u] \\
&= np_u[1 + (n-1)p_u], \\
\mathrm{E}[N_u^3] &= n(n-1)[(n-2)p_u^3 + 2p_u^2] + \mathrm{E}[N_u^2] \\
&= np_u\left[1 + (n-1)p_u(3 + (n-2)p_u)\right], \\
\mathrm{E}[N_u^4] &= n(n-1)(n-2)\left[(n-3)p_u^4 + 3p_u^3\right] + 2n(n-1)\left[(n-2)p_u^3 + 2p_u^2\right] + \mathrm{E}[N_u^3] \\
&= np_u\left[1 + (n-1)p_u(7 + (n-2)p_u[6 + (n-3)p_u])\right].
\end{aligned}
\tag{3.25}
$$

Define indicator random variable U_i such that $U_i = 1$ if the outcome for trial i is $u \in \{1, \ldots, r\}$ and $U_i = 0$ otherwise. Similarly define V_i for $v \neq u$, W_i for $w \neq v \neq u$, and X_i for $x \neq w \neq v \neq u$. Then N_u, N_v, N_w, and N_x may be expressed as

$$
N_u = \sum_{i=1}^{n} U_i, \quad N_v = \sum_{i=1}^{n} V_i, \quad N_w = \sum_{i=1}^{n} W_i, \quad N_x = \sum_{i=1}^{n} X_i.
$$

We are interested in $\mathrm{E}[N_u^2 N_v^2]$, $\mathrm{E}[N_u^3 N_v]$, $\mathrm{E}[N_u^2 N_v N_w]$, $\mathrm{E}[N_u N_v N_w N_x]$, $\mathrm{E}[N_u^2 N_v]$, and $\mathrm{E}[N_u N_v N_w]$.

$$
\begin{aligned}
\mathrm{E}[N_u^2 N_v^2] &= \mathrm{E}\left[\left(\sum_{i=1}^{n} U_i\right)^2 \left(\sum_{i=1}^{n} V_i\right)^2\right] \\
&= \mathrm{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} U_i U_j V_k V_l\right] && \text{by distributing} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} \mathrm{E}\left[U_i U_j V_k V_l\right] && \text{by linearity of expectation} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\substack{k\neq i\\ k\neq j}}\sum_{\substack{l\neq i\\ l\neq j}} \mathrm{E}\left[U_i U_j V_k V_l\right] && \text{since } U_i V_i = 0 \text{ for all } i = 1, \ldots, n \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\substack{k\neq i\\ k\neq j}}\sum_{\substack{l\neq i\\ l\neq j}} \mathrm{E}\left[U_i U_j\right]\mathrm{E}\left[V_k V_l\right] && \text{by independence between trials} \\
&= \sum_{i=j}\sum_{\substack{k=l\\ k\neq i}} \mathrm{E}\left[U_i U_j\right]\mathrm{E}\left[V_k V_l\right] + \sum_{i}\sum_{j\neq i}\sum_{\substack{k\neq i\\ k\neq j}}\sum_{\substack{l\neq k\\ l\neq i\\ l\neq j}} \mathrm{E}\left[U_i U_j\right]\mathrm{E}\left[V_k V_l\right] \\
&\quad + \sum_{i=j}\sum_{k\neq i}\sum_{\substack{l\neq k\\ l\neq i}} \mathrm{E}\left[U_i U_j\right]\mathrm{E}\left[V_k V_l\right] + \sum_{k=l}\sum_{i\neq k}\sum_{\substack{j\neq i\\ j\neq k}} \mathrm{E}\left[U_i U_j\right]\mathrm{E}\left[V_k V_l\right] && \text{reexpressed} \\
&= \sum_{i}\sum_{\substack{k=l\\ k\neq i}} \mathrm{E}[U_i^2]\mathrm{E}[V_k^2] + \sum_{i}\sum_{j\neq i}\sum_{\substack{k\neq i\\ k\neq j}}\sum_{\substack{l\neq k\\ l\neq i\\ l\neq j}} \mathrm{E}[U_i]\mathrm{E}[U_j]\mathrm{E}[V_k]\mathrm{E}[V_l] && \text{reexpressed; independence; and} \\
&\quad + \sum_{i}\sum_{k\neq i}\sum_{\substack{l\neq k\\ l\neq i}} \mathrm{E}[U_i^2]\mathrm{E}[V_k V_l] + \sum_{k}\sum_{i\neq k}\sum_{\substack{j\neq i\\ j\neq k}} \mathrm{E}[U_i]\mathrm{E}[U_j]\mathrm{E}[V_k^2] && \text{since } \mathrm{E}[U_i U_j] = \mathrm{E}[U_i]\mathrm{E}[U_j],\ i \neq j \\
&= n(n-1)p_u p_v + n(n-1)(n-2)(n-3)p_u^2 p_v^2 && \text{since } \mathrm{E}[U_i^2] = \mathrm{E}[U_i] = p_u \\
&\quad + n(n-1)(n-2)p_u p_v^2 + n(n-1)(n-2)p_u^2 p_v \\
&= n(n-1)p_u p_v\left[1 + (n-2)(p_u + p_v + (n-3)p_u p_v)\right] && \text{simplified.}
\end{aligned}
$$

Hence,

$$
\mathrm{E}[N_u^2 N_v^2] = n(n-1)p_u p_v\left[1 + (n-2)(p_u + p_v + (n-3)p_u p_v)\right]. \tag{3.26}
$$

Following the same derivation strategy,

$$\mathrm{E}[N_u^3 N_v] = n(n-1)p_u p_v \left[1 + (n-2)p_u(3 + (n-3)p_u)\right], \tag{3.27}$$

$$\mathrm{E}[N_u^2 N_v N_w] = n(n-1)(n-2)p_u p_v p_w \left[1 + (n-3)p_u\right], \tag{3.28}$$

$$\mathrm{E}[N_u N_v N_w N_x] = n(n-1)(n-2)(n-3)p_u p_v p_w p_x, \tag{3.29}$$

$$\mathrm{E}[N_u^2 N_v] = n(n-1)p_u p_v \left[1 + (n-2)p_u\right], \tag{3.30}$$

$$\mathrm{E}[N_u N_v N_w] = n(n-1)(n-2)p_u p_v p_w. \tag{3.31}$$

3.9.4 Numerator and Denominator of Ratio

We now turn to the task of deriving expressions for $\mathrm{Var}[M_i]$, $\mathrm{Var}[W_i]$, and $\mathrm{Cov}[M_i, W_i]$ in order to compute (3.22), and additionally for $\mathrm{Cov}[M_i, M_j]$, $\mathrm{Cov}[M_i, W_j]$, $\mathrm{Cov}[W_i, M_j]$, and $\mathrm{Cov}[W_i, W_j]$ for (3.23). For this subsection, return to the notation for $\mathbf{N}$ expressed in (3.15).

The distribution of $W_i = n^{-1}(N_i + N_i')$ is most simple. By the lumping property of multinomial random vectors,

$$\begin{aligned} \mathrm{E}[W_i] &= p_i + p_i', \\ \mathrm{Var}[W_i] &= \frac{(p_i + p_i')(1 - p_i - p_i')}{n}, \\ \mathrm{Cov}[W_i, W_j] &= -\frac{(p_i + p_i')(p_j + p_j')}{n}. \end{aligned} \tag{3.32}$$

The distribution of $M_i = n^{-2}N_i(N_i + N_i' + N_i'')$ is more challenging. From (3.25) and (3.24), the expectation is given by:

$$\begin{aligned} \mathrm{E}[M_i] &= n^{-2}\mathrm{E}[N_i(N_i + N_i' + N_i'')] \\ &= n^{-2}\left(\mathrm{E}[N_i^2] + \mathrm{E}[N_i N_i'] + \mathrm{E}[N_i N_i'']\right) \\ &= n^{-2}\left(np_i[1 + (n-1)p_i] + n(n-1)p_i p_i' + n(n-1)p_i p_i''\right) \\ &= n^{-1}p_i[1 + (n-1)(p_i + p_i' + p_i'')]. \end{aligned} \tag{3.33}$$

Next, the variance is given by:

$$\begin{aligned}
\mathrm{Var}[M_i] &= n^{-4}\mathrm{Var}[N_i(N_i + N_i{}' + N_i{}'')] \\
&= n^{-4}\mathrm{Var}[N_i^2 + N_iN_i{}' + N_iN_i{}''] \\
&= n^{-4}\big(\mathrm{Var}[N_i^2] + \mathrm{Var}[N_iN_i{}'] + \mathrm{Var}[N_iN_i{}''] \\
&\qquad + 2\mathrm{Cov}[N_i^2, N_iN_i{}'] + 2\mathrm{Cov}[N_i^2, N_iN_i{}''] + 2\mathrm{Cov}[N_iN_i{}', N_iN_i{}'']\big).
\end{aligned}$$

The terms in the expression above are given below. From the moments of the multinomial distribution (3.25):

$$\begin{aligned}
\mathrm{Var}[N_i^2] &= \mathrm{E}[N_i^4] - \mathrm{E}[N_i^2]^2 \\
&= np_i\left[1 + (n-1)p_i(7 + (n-2)p_i[6 + (n-3)p_i])\right] - (np_i[1 + (n-1)p_i])^2 \\
&= np_i\left[1 + (n-1)p_i(7 + (n-2)p_i[6 + (n-3)p_i]) - np_i(1 + (n-1)p_i)^2\right].
\end{aligned}$$

From (3.26) and (3.24):

$$\begin{aligned}
\mathrm{Var}[N_iN_i{}'] &= \mathrm{E}[N_i^2N_i{}'^2] - \mathrm{E}[N_iN_i{}']^2 \\
&= n(n-1)p_ip_i{}'[1 + (n-2)(p_i + p_i{}' + (n-3)p_ip_i{}')] - [n(n-1)p_ip_i{}']^2 \\
&= n(n-1)p_ip_i{}'[1 + (n-2)(p_i + p_i{}' + (n-3)p_ip_i{}') - n(n-1)p_ip_i{}'], \\
\mathrm{Var}[N_iN_i{}''] &= n(n-1)p_ip_i{}''[1 + (n-2)(p_i + p_i{}'' + (n-3)p_ip_i{}'') - n(n-1)p_ip_i{}''].
\end{aligned}$$

From (3.27), (3.25), and (3.24):

$$\begin{aligned}
\mathrm{Cov}[N_i^2, N_i N_i{}'] &= \mathrm{E}[N_i^3 N_i{}'] - \mathrm{E}[N_i^2]\mathrm{E}[N_i N_i{}'] \\
&= n(n-1)p_i p_i{}'\left[1 + (n-2)p_i(3 + (n-3)p_i)\right] \\
&\quad - np_i[1 + (n-1)p_i]n(n-1)p_i p_i{}' \\
&= n(n-1)p_i p_i{}'\left[1 + (n-2)(3p_i + (n-3)p_i^2) - np_i(1 + (n-1)p_i)\right] \\
\mathrm{Cov}[N_i^2, N_i N_i{}''] &= n(n-1)p_i p_i{}''\left[1 + (n-2)(3p_i + (n-3)p_i^2) - np_i(1 + (n-1)p_i)\right].
\end{aligned}$$

From (3.28) and (3.24):

$$\begin{aligned}
\mathrm{Cov}[N_i N_i{}', N_i N_i{}''] &= \mathrm{E}[N_i^2 N_i{}' N_i{}''] - \mathrm{E}[N_i N_i{}']\mathrm{E}[N_i N_i{}''] \\
&= n(n-1)(n-2)p_i p_i{}' p_i{}''[1 + (n-3)p_i] - n(n-1)p_i p_i{}' n(n-1)p_i p_i{}'' \\
&= n(n-1)p_i p_i{}' p_i{}''\left[(n-2)[1 + (n-3)p_i] - n(n-1)p_i\right].
\end{aligned}$$

Hence, $\mathrm{Var}[M_i]$ is derived:

$$\begin{aligned}
\mathrm{Var}[M_i] = n^{-4}\big(&np_i\left[1 + (n-1)p_i(7 + (n-2)p_i[6 + (n-3)p_i]) - np_i(1 + (n-1)p_i)^2\right] \\
&+ n(n-1)p_i p_i{}'[1 + (n-2)(p_i + p_i{}' + (n-3)p_i p_i{}') - n(n-1)p_i p_i{}'] \\
&+ n(n-1)p_i p_i{}''[1 + (n-2)(p_i + p_i{}'' + (n-3)p_i p_i{}'') - n(n-1)p_i p_i{}''] \\
&+ 2n(n-1)p_i p_i{}'\left[1 + (n-2)(3p_i + (n-3)p_i^2) - np_i(1 + (n-1)p_i)\right] \\
&+ 2n(n-1)p_i p_i{}''\left[1 + (n-2)(3p_i + (n-3)p_i^2) - np_i(1 + (n-1)p_i)\right] \\
&+ n(n-1)p_i p_i{}' p_i{}''\left[(n-2)[1 + (n-3)p_i] - n(n-1)p_i\right]\big).
\end{aligned} \tag{3.34}$$

Next, consider $\mathrm{Cov}[M_i, M_j]$.

$$\begin{aligned}
\mathrm{Cov}[M_i, M_j] &= n^{-4}\mathrm{Cov}\left[N_i(N_i + N_i{}' + N_i{}''), N_j(N_j + N_j{}' + N_j{}'')\right] \\
&= n^{-4}\mathrm{Cov}\left[N_i^2 + N_iN_i{}' + N_iN_i{}'', N_j^2 + N_jN_j{}' + N_jN_j{}''\right] \\
&= n^{-4}\big(\mathrm{Cov}[N_i^2, N_j^2] \\
&\qquad + \mathrm{Cov}[N_i^2, N_jN_j{}'] + \mathrm{Cov}[N_i^2, N_jN_j{}''] + \mathrm{Cov}[N_iN_i{}', N_j^2] + \mathrm{Cov}[N_iN_i{}'', N_j^2] \\
&\qquad + \mathrm{Cov}[N_iN_i{}', N_jN_j{}'] + \mathrm{Cov}[N_iN_i{}', N_jN_j{}''] \\
&\qquad + \mathrm{Cov}[N_iN_i{}'', N_jN_j{}'] + \mathrm{Cov}[N_iN_i{}'', N_jN_j{}'']\big).
\end{aligned}$$

The terms in the expression above are given below. From (3.26) and (3.25):

$$\begin{aligned}
\mathrm{Cov}[N_i^2, N_j^2] &= \mathrm{E}[N_i^2N_j^2] - \mathrm{E}[N_i^2]\mathrm{E}[N_j^2] \\
&= n(n-1)p_ip_j\left[1 + (n-2)(p_i + p_j + (n-3)p_ip_j)\right] \\
&\quad - np_i[1 + (n-1)p_i]np_j[1 + (n-1)p_j] \\
&= np_ip_j[(n-1)(1 + (n-2)(p_i + p_j + (n-3)p_ip_j)) \\
&\qquad - n(1 + (n-1)p_i)(1 + (n-1)p_j)]
\end{aligned}$$

From (3.28), (3.25), and (3.24):

$$\begin{aligned}
\mathrm{Cov}[N_i^2, N_jN_j{}'] &= \mathrm{E}[N_i^2N_jN_j{}'] - \mathrm{E}[N_i^2]\mathrm{E}[N_jN_j{}'] \\
&= n(n-1)(n-2)p_ip_jp_j{}'[1 + (n-3)p_i] - np_i(1 + (n-1)p_i)n(n-1)p_jp_j{}' \\
&= n(n-1)p_ip_jp_j{}'\left[(n-2)[1 + (n-3)p_i] - n(1 + (n-1)p_i)\right], \\
\mathrm{Cov}[N_i^2, N_jN_j{}''] &= n(n-1)p_ip_jp_j{}''\left[(n-2)[1 + (n-3)p_i] - n(1 + (n-1)p_i)\right], \\
\mathrm{Cov}[N_iN_i{}', N_j^2] &= n(n-1)p_jp_ip_i{}'\left[(n-2)[1 + (n-3)p_j] - n(1 + (n-1)p_j)\right], \\
\mathrm{Cov}[N_iN_i{}'', N_j^2] &= n(n-1)p_jp_ip_i{}''\left[(n-2)[1 + (n-3)p_j] - n(1 + (n-1)p_j)\right].
\end{aligned}$$

From (3.29) and (3.24):

$$
\begin{aligned}
\mathrm{Cov}[N_i N_i', N_j N_j'] &= \mathrm{E}[N_i N_i' N_j N_j'] - \mathrm{E}[N_i N_i']\mathrm{E}[N_j N_j'] \\
&= n(n-1)(n-2)(n-3)p_i p_i' p_j p_j' - n(n-1)p_i p_i' n(n-1)p_j p_j' \\
&= n(n-1)p_i p_i' p_j p_j' \left[(n-2)(n-3) - n(n-1)\right], \\
\mathrm{Cov}[N_i N_i', N_j N_j''] &= n(n-1)p_i p_i' p_j p_j'' \left[(n-2)(n-3) - n(n-1)\right], \\
\mathrm{Cov}[N_i N_i'', N_j N_j'] &= n(n-1)p_i p_i'' p_j p_j' \left[(n-2)(n-3) - n(n-1)\right], \\
\mathrm{Cov}[N_i N_i'', N_j N_j''] &= n(n-1)p_i p_i'' p_j p_j'' \left[(n-2)(n-3) - n(n-1)\right].
\end{aligned}
$$

Hence, $\mathrm{Cov}[M_i, M_j]$ is derived:

$$
\begin{aligned}
&\mathrm{Cov}[M_i, M_j] \\
&\quad = n^{-4}\big(np_ip_j\big[(n-1)(1+(n-2)(p_i+p_j+(n-3)p_ip_j)) \\
&\qquad\qquad - n(1+(n-1)p_i)(1+(n-1)p_j)\big] \\
&\qquad + n(n-1)p_ip_jp_j{}'\,[(n-2)[1+(n-3)p_i] - n(1+(n-1)p_i)] \\
&\qquad + n(n-1)p_ip_jp_j{}''\,[(n-2)[1+(n-3)p_i] - n(1+(n-1)p_i)] \\
&\qquad + n(n-1)p_jp_ip_i{}'\,[(n-2)[1+(n-3)p_j] - n(1+(n-1)p_j)] \\
&\qquad + n(n-1)p_jp_ip_i{}''\,[(n-2)[1+(n-3)p_j] - n(1+(n-1)p_j)] \\
&\qquad + n(n-1)p_ip_i{}'p_jp_j{}'\,[(n-2)(n-3) - n(n-1)] \\
&\qquad + n(n-1)p_ip_i{}'p_jp_j{}''\,[(n-2)(n-3) - n(n-1)] \\
&\qquad + n(n-1)p_ip_i{}''p_jp_j{}'\,[(n-2)(n-3) - n(n-1)] \\
&\qquad + n(n-1)p_ip_i{}''p_jp_j{}''\,[(n-2)(n-3) - n(n-1)]\,\big) \\
&\quad = n^{-3}p_ip_j\big[(n-1)\big(1+(n-2)(p_i+p_j+(n-3)p_ip_j) \\
&\qquad\qquad + (p_j{}'+p_j{}'')[(n-2)[1+(n-3)p_i] - n(1+(n-1)p_i)] \\
&\qquad\qquad + p_jp_i(p_i{}'+p_i{}'')[(n-2)[1+(n-3)p_j] - n(1+(n-1)p_j)] \\
&\qquad\qquad + (p_i{}'+p_i{}'')(p_j{}'+p_j{}'')[(n-2)(n-3) - n(n-1)]\big) \\
&\qquad - n(1+(n-1)p_i)(1+(n-1)p_j)\big]
\end{aligned}
\tag{3.35}
$$

Finally, we turn our attention to $\mathrm{Cov}[M_i, W_i]$, $\mathrm{Cov}[M_i, W_j]$, and $\mathrm{Cov}[W_i, M_j]$. Beginning

with $\mathrm{Cov}[M_i, W_i]$:

$$\begin{aligned}
\mathrm{Cov}[M_i, W_i] &= n^{-3}\mathrm{Cov}\left[N_i(N_i + N_i{}' + N_i{}''), N_i + N_i{}'\right] \\
&= n^{-3}\mathrm{Cov}\left[N_i^2 + N_iN_i{}' + N_iN_i{}'', N_i + N_i{}'\right] \\
&= n^{-3}\big(\mathrm{Cov}[N_i^2, N_i] + \mathrm{Cov}[N_i^2, N_i{}'] \\
&\qquad + \mathrm{Cov}[N_iN_i{}', N_i] + \mathrm{Cov}[N_iN_i{}', N_i{}'] + \mathrm{Cov}[N_iN_i{}'', N_i] + \mathrm{Cov}[N_iN_i{}'', N_i{}']\big).
\end{aligned}$$

The terms in the expression above are given below. From (3.25):

$$\begin{aligned}
\mathrm{Cov}[N_i^2, N_i] &= \mathrm{E}[N_i^3] - \mathrm{E}[N_i^2]\mathrm{E}[N_i] \\
&= np_i\left[1 + (n-1)p_i(3 + (n-2)p_i)\right] - np_i[1 + (n-1)p_i]np_i \\
&= np_i\left[1 + (n-1)p_i(3 + (n-2)p_i) - np_i(1 + (n-1)p_i)\right] \\
&= np_i\left[1 + p_i((n-1)[3 - 2p_i] - n)\right].
\end{aligned}$$

From (3.30) and (3.25):

$$\begin{aligned}
\mathrm{Cov}[N_i^2, N_i{}'] &= \mathrm{E}[N_i^2N_i{}'] - \mathrm{E}[N_i^2]\mathrm{E}[N_i{}'] \\
&= n(n-1)p_ip_i{}'[1 + (n-2)p_i] - np_i(1 + (n-1)p_i)np_j \qquad (3.36) \\
&= np_ip_i{}'\left[(n-1)(1 + (n-2)p_i) - n(1 + (n-1)p_i)\right].
\end{aligned}$$

From (3.30) and (3.25):

$$\begin{aligned}
\mathrm{Cov}[N_iN_i{}', N_i] &= \mathrm{E}[N_i^2N_i{}'] - \mathrm{E}[N_iN_i{}']\mathrm{E}[N_i] \\
&= n(n-1)p_ip_i{}'[1 + (n-2)p_i] - n(n-1)p_ip_i{}'np_i \\
&= n(n-1)p_ip_i{}'[1 - 2p_i], \\
\mathrm{Cov}[N_iN_i{}', N_i{}'] &= n(n-1)p_i{}'p_i[1 - 2p_i{}'], \\
\mathrm{Cov}[N_iN_i{}'', N_i] &= n(n-1)p_ip_i{}''[1 - 2p_i].
\end{aligned}$$

From (3.31), (3.24), and (3.25):

$$\begin{aligned}\mathrm{Cov}[N_iN_i'', N_i'] &= \mathrm{E}[N_iN_i''N_i'] - \mathrm{E}[N_iN_i'']\mathrm{E}[N_i'] \\ &= n(n-1)(n-2)p_ip_i''p_i' - n(n-1)p_ip_i''np_i' \\ &= -2n(n-1)p_ip_i''p_i'.\end{aligned} \tag{3.37}$$

Hence, $\mathrm{Cov}[M_i, W_i]$ is derived:

$$\begin{aligned}\mathrm{Cov}[M_i, W_i] = n^{-3}\big(&np_i[1 + p_i((n-1)[3-2p_i] - n)] \\ &+ np_ip_i'[(n-1)(1+(n-2)p_i) - n(1+(n-1)p_i)] \\ &+ n(n-1)p_ip_i'[1-2p_i] \\ &+ n(n-1)p_i'p_i[1-2p_i'] \\ &+ n(n-1)p_ip_i''[1-2p_i] \\ &- 2n(n-1)p_ip_i''p_i'\big). \\ = n^{-2}p_i\big(&1 + p_i((n-1)[3-2p_i] - n) \\ &+ p_i'[(n-1)(1+(n-2)p_i) - n(1+(n-1)p_i)]\big) \\ &+ n^{-2}(n-1)(p_ip_i'[2-2p_i-2p_i'] + p_ip_i''[1-2p_i-2p_i']).\end{aligned} \tag{3.38}$$

Then, moving on to $\mathrm{Cov}[M_i, W_j]$ and $\mathrm{Cov}[W_i, M_j]$:

$$\begin{aligned}\mathrm{Cov}[M_i, W_j] &= n^{-3}\mathrm{Cov}\left[N_i(N_i + N_i' + N_i''), N_j + N_j'\right] \\ &= n^{-3}\mathrm{Cov}\left[N_i^2 + N_iN_i' + N_iN_i'', N_j + N_j'\right] \\ &= n^{-3}\big(\mathrm{Cov}[N_i^2, N_j] + \mathrm{Cov}[N_i^2, N_j'] \\ &\qquad + \mathrm{Cov}[N_iN_i', N_j] + \mathrm{Cov}[N_iN_i', N_j'] + \mathrm{Cov}[N_iN_i'', N_j] + \mathrm{Cov}[N_iN_i'', N_j']\big).\end{aligned}$$

The terms in the expression above are given below. From (3.36):

$$\begin{aligned}
\mathrm{Cov}[N_i^2, N_j] &= np_ip_j\left[(n-1)(1+(n-2)p_i) - n(1+(n-1)p_i)\right], \\
\mathrm{Cov}[N_i^2, N_j{}'] &= np_ip_j{}'\left[(n-1)(1+(n-2)p_i) - n(1+(n-1)p_i)\right].
\end{aligned}$$

From (3.37):

$$\begin{aligned}
\mathrm{Cov}[N_iN_i{}', N_j] &= -2n(n-1)p_ip_i{}'p_j, \\
\mathrm{Cov}[N_iN_i{}', N_j{}'] &= -2n(n-1)p_ip_i{}'p_j{}', \\
\mathrm{Cov}[N_iN_i{}'', N_j] &= -2n(n-1)p_ip_i{}''p_j, \\
\mathrm{Cov}[N_iN_i{}'', N_j{}'] &= -2n(n-1)p_ip_i{}''p_j{}'.
\end{aligned}$$

Hence, $\mathrm{Cov}[M_i, W_j]$ and $\mathrm{Cov}[W_i, M_j]$ are derived:

$$\begin{aligned}
\mathrm{Cov}[M_i, W_j] &= n^{-3}\big[np_i(p_j+p_j{}')[(n-1)(1+(n-2)p_i) - n(1+(n-1)p_i)] \\
&\qquad - 2n(n-1)p_i(p_i{}'p_j + p_i{}'p_j{}' + p_i{}''p_j + p_i{}''p_j{}')\big] \\
&= n^{-2}p_i(p_j+p_j{}')\big[(n-1)(1+(n-2)p_i - 2(p_i{}'+p_i{}'')(p_j+p_j{}')) \\
&\qquad - n(1+(n-1)p_i)\big], \\
\mathrm{Cov}[W_i, M_j] &= n^{-2}p_j(p_i+p_i{}')\big[(n-1)(1+(n-2)p_j - 2(p_j{}'+p_j{}'')(p_i+p_i{}')) \\
&\qquad - n(1+(n-1)p_j)\big].
\end{aligned} \tag{3.39}$$

Thus, all quantities necessary to compute (3.17) are derived.

CHAPTER 4

Summary and Discussion

In this dissertation, we contributed a number of developments to the identification and exploitation of Bayesian network structures. First, we considered learning equivalence classes of large Bayesian network structures from observational data, proposing a well-performing and asymptotically consistent hybrid algorithm called partitioned hybrid greedy search which is composed of individually compelling independent contributions. The partitioned PC algorithm improves on the efficiency of the PC algorithm while retaining soundness and completeness for CPDAG learning. The p-value adjacency thresholding algorithm, from a single execution of pPC or PC, generates and selects from a solution path of DAG estimates at comparatively negligible additional computational expense. PATH requires only a single execution of an appropriate constraint-based skeleton leearning algorithm, extending the accessibility of asymptotic consistency results and empirically accomplishing parameter tuning. Finally, the hybrid greedy initialization algorithm endears constraint-based edge orientation to the empirical setting while preserving large-sample guarantees, providing a favorable starting point to improve the score-based search phase in hybrid algorithms.

Second, we considered informing sequential experimental investigations with observational data by developing the Bayesian Backdoor Bandit framework for tackling the causal bandit problem. Our proposed BBB methodology efficiently utilizes jointly interventional and observational data to model the underlying causal graph as well as the parameters of interest. In our detailed application of BBB to the discrete and Gaussian distributional settings, we additionally propose useful estimators and an avenue for sharing information

between arms. We have left a number of directions for future work, which we discuss in what follows.

While we have empirically demonstrated a significant reduction to the number of conditional independence tests executed by pPC in comparison to PC, we have not established any concrete theoretical complexity results. Indeed, the extent of computational reduction inevitably depends on the quality of the partition with respect to the underlying structure. As such, it is of interest to determine under what conditions pPC is guaranteed to perform fewer tests than PC, and to quantify the reduction. Such an investigation is crucial to establish structure learning consistency of pPC in the sparse high-dimensional setting with multivariate Gaussian distributions. The high-dimensional Gaussian consistency of PC proved in Kalisch and Bühlmann (2007) relies on the number of conditional independence tests performed, determined by the maximum size of conditioning sets reached by PC with no errors. The same result holds for pPC under the same assumptions if the number of tests investigated by pPC is not greater than that of PC. Note that empirically, we find pPC to always perform fewer tests than PC.

In Section 2.4, we observed superior structure learning performance by HPC and H2PC-HGI against pPC-PATH and pHGS, respectively, though not without significantly greater computational expense. We note that since the skeleton learning phase of HPC involves independently estimating the parents and children of each variable and aggregating the results, there are many obvious opportunities to more efficiently execute a synchronized global version of HPC for skeleton learning that avoids redundant computations. For instance, the first subroutine estimates a supergraph by identifying, for each node $i \in \mathbf{V}$, every node $j \in \mathbf{V} \setminus \{i\}$ that cannot be separated from i by conditioning sets $\{\mathbf{X} \subseteq \mathbf{V} \setminus \{i, j\} : |\mathbf{k}| \leq 1\}$. When considered node-wise, every conditional independence test is executed twice, once in the consideration of i as a neighbor of j and again in the consideration of j as a neighbor of i. A global partition-based strategy, such as the pPC algorithm with the restriction to conditioning sets $|\mathbf{k}| \leq 1$, can be utilized to substantially reduce the computational expense

compared to the original formulation.

As formulated, pPC is limited to obtaining $\kappa \leq 20$ clusters as per a loose suggestion by Hartigan (1981). Note that by our design, pPC is not limited to parallel processing utility from at most κ processors, having comparable capacity for parallel execution as PC. Nonetheless, a future direction for further improvement would be the development of an unsupervised criterion to more flexibly determine the target number of clusters that optimizes the efficiency of pPC.

A substantial limitation of our BBB methodology is the computational expense of computing the parent set probabilities and the conditional parameter posteriors corresponding to each parent set. However, such an investment is justified when interventions are particularly expensive or time-consuming. In these settings, it is crucial to utilize all available evidence, motivating future work in scaling our methods to feasibly operate for larger causal systems. Note that parent set probabilities with trivial support may be thresholded to zero and the corresponding conditional posteriors need not be updated. This significantly reduces the computational load of causal parameter modeling, especially for systems with sparse structures, so the limiting factor tends to the structure posterior. A potential scalable alternative to exact computation of parent set probabilities would be to estimate the structure posterior distribution using the hybrid MCMC approach proposed by Kuipers et al. (2022) in which DAGs are sampled from a restricted space estimated by a structure learning algorithm. This approach would additionally provide a way to directly approximate the left hand side of (3.13).

In addition to scaling, there are a number of interesting directions for future investigations. Though we validated our proposed BBB methodology numerically by demonstrating compelling empirical performance against well-studied algorithms, we leave formal theoretical analysis of the BBB framework to future work. Finally, it would be interesting to consider how to share information between arms in the discrete setting as in Proposition 2 with an equally simple graphical criterion.

APPENDIX A

Additional Results

In this section, we provide additional figures and tables further summarizing the results from the simulation study described in Section 2.4. In particular, Figure A.1 visualizes the accuracy and efficiency of pPC compared against popular constraint-based algorithms, and Figure A.2 visualizes the accuracy of pHGS against HC and GSC algorithms MMHC and H2PC. The tables report extensive results containing numerous metrics comparing pPC with PATH against PC, MMPC, and HPC in Table A.1, pHGS against HC, MMHC, and H2PC in Table A.2, and pHGS against GES and FGES in Table A.3.

We report the Jaccard index (JI), $\log_{10}$ number of statistical calls (Calls), number of predicted edges (P), true positives (TP), edges correctly present but with incorrect orientation (R), false positives (FP), and the structural Hamming distance (SHD), which may be interpreted as the number of edge additions, deletions, and reversals required to traverse from one graph to another. In Tables A.1-A.3, the means are provided alongside standard deviations (provided in parenthesis) across the $N = 100$ datasets for each of the 20 networks in Table 2.1.

The results for pPC are that of pPC with PATH with parameters $\alpha = 0.1$, $\tau = 10$, nd $\alpha^{(\tau)} = 10^{-5}$, and the results for pHGS are with $\alpha = 0.01$, $\tau = 10$, and $\alpha^{(\tau)} = 10^{-5}$. The results for the estimates with the highest JI out of the ten executions with $\alpha \in \mathcal{A}$ for each dataset are reported for PC, MMPC, HPC, MMHC, and H2PC, alongside the $\log_{10}$ total number of statistical calls for all ten executions.

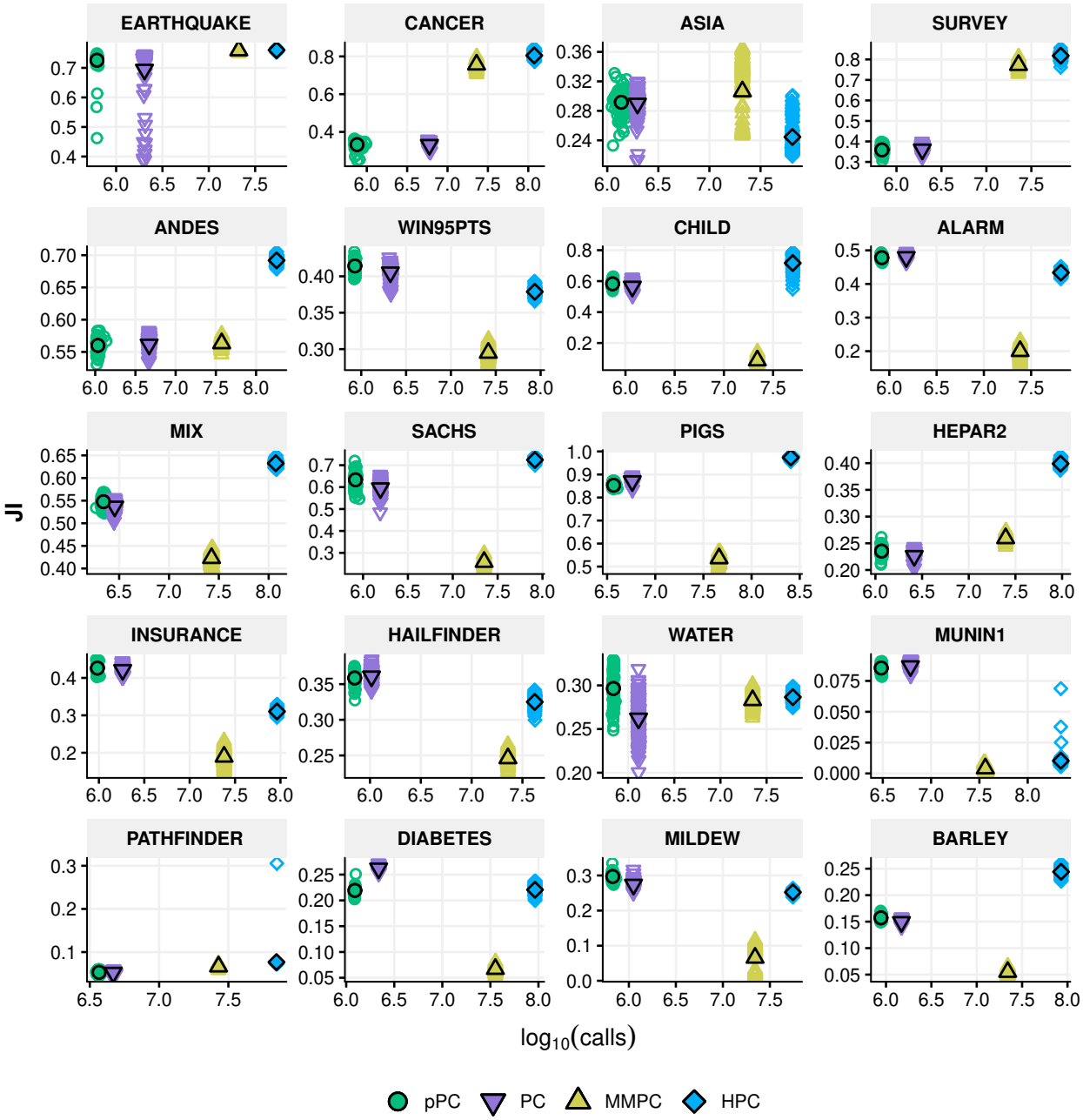


Figure A.1: Visual accuracy (JI) and efficiency ($\log_{10}$(calls)) comparisons amongst pPC, PC, MMPC, and HPC. Filled shapes with black outlines indicate averages, whereas colored outlines represent results for individual datasets

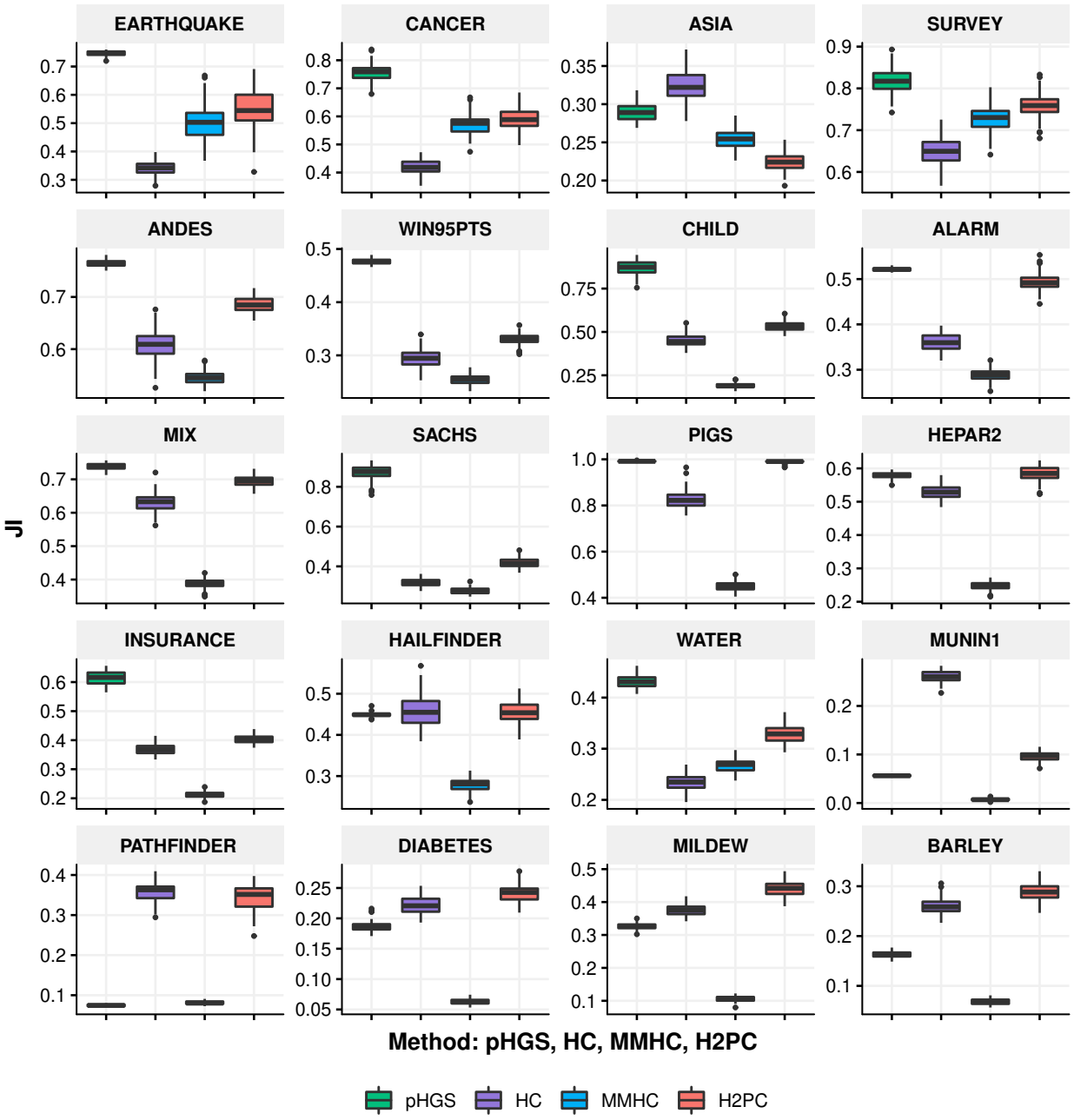


Figure A.2: Visual accuracy (JI) comparisons amongst pHGS, HC, MMHC, and H2PC

Table A.1: Detailed results comparing pPC with PATH against PC, MMPC, and HPC

	pPC		PC		MMPC		HPC	
	EARTHQUAKE							
P	850.8	(23.8)	852.8	(2.4)	874.0	(9.0)	858.2	(4.7)
TP	820.3	(19.1)	809.8	(5.0)	853.1	(4.8)	847.0	(3.3)
R	27.6	(26.3)	43.0	(5.9)	16.6	(2.3)	8.8	(1.2)
FP	2.8	(16.0)	0.0	(0.2)	4.3	(3.4)	2.3	(2.2)
SHD	285.5	(34.6)	293.2	(5.0)	254.3	(3.7)	258.3	(3.2)
JI	0.725	(0.036)	0.707	(0.008)	0.759	(0.003)	0.760	(0.003)
Calls	5.793	(0.005)	6.849	(0.002)	7.323	(0.000)	7.731	(0.003)
	CANCER							
P	1064.4	(75.1)	1051.2	(9.5)	1073.0	(10.7)	1120.4	(13.1)
TP	546.1	(19.1)	488.5	(8.0)	945.8	(14.5)	1000.4	(11.9)
R	500.1	(18.5)	561.5	(6.0)	112.9	(13.2)	78.2	(10.7)
FP	18.2	(62.7)	1.3	(3.4)	14.3	(7.7)	41.8	(11.7)
SHD	595.1	(68.8)	635.8	(7.1)	191.5	(13.5)	164.4	(13.8)
JI	0.333	(0.019)	0.292	(0.004)	0.757	(0.018)	0.805	(0.016)
Calls	5.876	(0.026)	7.071	(0.004)	7.360	(0.001)	8.074	(0.004)
	ASIA							
P	805.6	(78.6)	774.6	(18.7)	906.8	(46.1)	712.9	(28.9)
TP	451.5	(19.3)	425.8	(15.2)	503.3	(62.6)	384.0	(31.1)
R	333.4	(49.7)	348.2	(17.7)	342.2	(53.9)	299.1	(25.2)
FP	20.7	(40.8)	0.6	(3.8)	61.2	(34.6)	29.8	(13.0)
SHD	812.2	(52.8)	817.8	(15.2)	800.8	(37.1)	888.7	(25.3)
JI	0.284	(0.023)	0.261	(0.009)	0.306	(0.042)	0.245	(0.022)
Calls	6.137	(0.040)	6.848	(0.002)	7.323	(0.000)	7.812	(0.003)
	SURVEY							
P	1290.6	(61.9)	1291.1	(13.9)	1307.2	(4.8)	1346.2	(20.5)
TP	700.2	(31.7)	688.5	(15.2)	1156.9	(14.3)	1211.8	(14.4)
R	578.2	(20.0)	602.3	(11.7)	146.9	(13.3)	117.7	(13.1)
FP	12.2	(34.2)	0.3	(1.2)	3.4	(2.0)	16.7	(18.3)
SHD	659.0	(44.5)	658.7	(14.9)	193.5	(14.5)	151.9	(16.4)
JI	0.362	(0.018)	0.360	(0.007)	0.773	(0.016)	0.818	(0.015)
Calls	5.837	(0.010)	6.851	(0.002)	7.357	(0.001)	7.827	(0.003)

Table A.1: (continued)

	pPC		PC		MMPC		HPC	
	ANDES							
P	1790.5	(34.3)	1770.4	(38.6)	1557.5	(19.6)	1773.2	(22.6)
TP	1448.0	(19.0)	1401.9	(29.8)	1370.2	(12.8)	1643.3	(13.5)
R	312.9	(26.2)	359.5	(43.2)	149.5	(8.2)	118.1	(7.1)
FP	29.5	(5.3)	9.1	(3.6)	37.7	(11.4)	11.8	(11.3)
SHD	826.5	(18.8)	852.2	(30.7)	912.5	(12.0)	613.5	(11.0)
JI	0.560	(0.010)	0.505	(0.010)	0.563	(0.006)	0.692	(0.006)
Calls	6.028	(0.019)	7.133	(0.003)	7.572	(0.004)	8.255	(0.004)
	WIN95PTS							
P	1368.4	(42.7)	1297.6	(33.4)	1025.6	(17.3)	1174.5	(13.7)
TP	1035.0	(11.8)	992.5	(16.2)	731.0	(14.9)	922.6	(10.7)
R	269.3	(20.0)	277.1	(14.3)	273.7	(12.2)	236.1	(9.8)
FP	64.1	(22.9)	28.0	(11.9)	20.9	(7.3)	15.9	(5.7)
SHD	1213.1	(23.7)	1219.5	(12.7)	1474.0	(14.7)	1277.3	(11.1)
JI	0.411	(0.008)	0.398	(0.006)	0.295	(0.007)	0.379	(0.005)
Calls	5.927	(0.005)	6.964	(0.002)	7.417	(0.001)	7.935	(0.003)
	CHILD							
P	1299.4	(2.8)	1303.3	(3.8)	977.2	(17.5)	1310.9	(9.0)
TP	956.3	(20.9)	939.5	(22.3)	183.2	(33.5)	1092.4	(42.3)
R	342.1	(21.1)	361.9	(22.6)	777.1	(32.6)	207.5	(41.1)
FP	1.1	(1.8)	1.9	(2.9)	17.0	(10.5)	11.1	(10.0)
SHD	353.8	(20.9)	371.3	(22.8)	1142.8	(30.0)	227.6	(50.6)
JI	0.579	(0.020)	0.548	(0.019)	0.087	(0.017)	0.717	(0.049)
Calls	5.869	(0.004)	6.840	(0.001)	7.340	(0.001)	7.700	(0.001)
	ALARM							
P	1267.4	(22.3)	1287.4	(14.3)	1011.8	(21.0)	1403.3	(16.4)
TP	956.2	(12.8)	962.1	(11.1)	450.4	(30.7)	935.5	(12.0)
R	309.2	(11.3)	325.0	(10.9)	549.1	(26.0)	451.2	(15.6)
FP	2.0	(4.8)	0.2	(0.8)	12.3	(4.1)	16.7	(5.4)
SHD	734.8	(11.8)	727.1	(10.8)	1250.8	(30.3)	770.2	(14.4)
JI	0.478	(0.006)	0.473	(0.007)	0.200	(0.015)	0.434	(0.008)
Calls	5.921	(0.007)	6.914	(0.001)	7.383	(0.001)	7.815	(0.002)

Table A.1: (continued)

	pPC		PC		MMPC		HPC	
	MIX							
P	1476.3	(25.2)	1483.1	(25.1)	1167.3	(14.5)	1567.7	(16.6)
TP	1184.8	(13.9)	1166.2	(16.9)	904.9	(16.5)	1334.8	(10.0)
R	285.5	(22.7)	313.5	(24.8)	244.8	(13.5)	223.7	(9.1)
FP	6.0	(3.6)	3.4	(2.8)	17.6	(6.2)	9.2	(6.6)
SHD	698.3	(14.0)	714.3	(16.5)	989.7	(18.0)	551.4	(10.5)
JI	0.546	(0.010)	0.506	(0.009)	0.423	(0.010)	0.633	(0.006)
Calls	6.342	(0.015)	7.249	(0.006)	7.427	(0.006)	8.072	(0.007)
	SACHS							
P	1791.2	(6.8)	1798.6	(3.4)	1372.0	(8.0)	1809.4	(2.6)
TP	1283.3	(38.0)	1241.7	(37.9)	656.0	(25.3)	1525.3	(9.9)
R	508.0	(39.1)	556.9	(38.6)	714.6	(25.3)	283.7	(10.0)
FP	0.0	(0.1)	0.0	(0.0)	1.3	(1.2)	0.4	(1.4)
SHD	538.8	(38.0)	580.3	(37.9)	1167.3	(25.4)	297.1	(10.0)
JI	0.551	(0.025)	0.508	(0.022)	0.259	(0.013)	0.724	(0.008)
Calls	5.916	(0.010)	6.912	(0.001)	7.349	(0.001)	7.916	(0.002)
	PIGS							
P	2171.2	(1.9)	2172.3	(1.2)	1533.1	(15.0)	2192.7	(10.6)
TP	2003.2	(9.5)	2027.1	(11.2)	1295.9	(26.2)	2158.1	(3.4)
R	167.9	(9.1)	145.2	(11.2)	213.6	(16.5)	14.6	(2.0)
FP	0.1	(0.3)	0.0	(0.0)	23.7	(7.0)	20.0	(10.3)
SHD	178.9	(9.5)	154.9	(11.2)	909.8	(28.0)	43.9	(10.4)
JI	0.852	(0.007)	0.867	(0.009)	0.536	(0.014)	0.974	(0.005)
Calls	6.567	(0.010)	7.594	(0.005)	7.663	(0.005)	8.407	(0.006)
	HEPAR2							
P	1286.6	(7.5)	1283.6	(8.0)	1015.3	(45.9)	1687.4	(54.7)
TP	636.3	(18.3)	598.9	(29.6)	636.9	(13.9)	1073.6	(20.6)
R	650.3	(21.0)	684.7	(34.7)	310.1	(10.8)	589.8	(15.7)
FP	0.0	(0.1)	0.0	(0.1)	68.3	(38.9)	24.0	(25.6)
SHD	1440.7	(18.2)	1478.2	(29.6)	1508.4	(32.8)	1027.3	(13.7)
JI	0.233	(0.008)	0.199	(0.005)	0.259	(0.006)	0.399	(0.005)
Calls	6.066	(0.006)	7.058	(0.002)	7.395	(0.001)	7.984	(0.003)

Table A.1: (continued)

	pPC		PC		MMPC		HPC	
	INSURANCE							
P	1434.8	(10.5)	1438.5	(6.3)	991.2	(13.9)	1609.4	(39.5)
TP	1058.1	(19.3)	1050.7	(19.2)	495.0	(40.0)	883.2	(17.5)
R	375.6	(20.5)	387.8	(20.5)	473.4	(33.0)	696.0	(23.2)
FP	1.1	(0.5)	0.0	(0.0)	22.8	(4.6)	30.1	(16.1)
SHD	1063.0	(19.2)	1069.3	(19.2)	1647.8	(41.0)	1266.9	(17.7)
JI	0.424	(0.011)	0.404	(0.011)	0.189	(0.018)	0.310	(0.007)
Calls	5.983	(0.006)	6.978	(0.001)	7.379	(0.001)	7.962	(0.002)
	HAILFINDER							
P	927.1	(23.8)	911.1	(5.6)	830.5	(13.0)	1113.5	(19.1)
TP	653.7	(10.1)	643.1	(14.2)	468.1	(12.8)	651.1	(15.5)
R	218.8	(11.2)	257.1	(16.0)	316.2	(12.1)	461.9	(13.4)
FP	54.6	(17.9)	10.9	(3.0)	46.1	(11.5)	0.4	(1.1)
SHD	942.0	(20.8)	908.8	(14.6)	1119.0	(16.5)	890.4	(15.4)
JI	0.360	(0.009)	0.346	(0.008)	0.246	(0.008)	0.325	(0.008)
Calls	5.849	(0.003)	6.831	(0.001)	7.357	(0.001)	7.621	(0.001)
	WATER							
P	1338.6	(44.0)	1328.6	(62.8)	1313.6	(13.1)	1481.5	(46.6)
TP	836.5	(44.2)	776.7	(65.4)	797.7	(19.1)	842.8	(16.6)
R	498.4	(24.1)	551.0	(21.7)	513.4	(19.7)	616.3	(19.4)
FP	3.6	(4.2)	0.8	(1.7)	2.5	(1.7)	22.4	(24.1)
SHD	1470.0	(42.1)	1527.1	(64.6)	1507.8	(19.0)	1482.6	(20.6)
JI	0.298	(0.017)	0.291	(0.010)	0.283	(0.008)	0.287	(0.006)
Calls	5.840	(0.006)	6.849	(0.001)	7.349	(0.000)	7.785	(0.002)
	MUNIN1							
P	529.8	(6.8)	529.4	(13.0)	267.2	(10.0)	713.6	(14.5)
TP	182.2	(5.5)	191.0	(4.2)	8.3	(3.4)	25.1	(16.3)
R	307.4	(7.7)	309.7	(8.9)	216.0	(10.0)	636.2	(18.9)
FP	40.3	(2.6)	28.7	(5.1)	42.9	(6.1)	52.3	(6.4)
SHD	1634.1	(5.3)	1613.7	(5.2)	1810.6	(7.5)	1803.2	(17.1)
JI	0.086	(0.003)	0.090	(0.002)	0.004	(0.002)	0.010	(0.007)
Calls	6.484	(0.005)	7.609	(0.005)	7.562	(0.008)	8.343	(0.003)

Table A.1: (continued)

	pPC		PC		MMPC		HPC	
	PATHFINDER							
P	303.5	(5.9)	295.4	(5.5)	330.6	(11.0)	1185.3	(28.6)
TP	114.4	(7.2)	112.4	(6.4)	143.5	(5.7)	223.7	(51.1)
R	182.6	(6.7)	178.5	(6.2)	170.3	(8.4)	916.9	(58.3)
FP	6.5	(3.1)	4.6	(2.5)	16.9	(4.2)	44.7	(7.5)
SHD	1860.1	(7.2)	1860.2	(6.4)	1841.4	(6.7)	1789.0	(51.9)
JI	0.053	(0.003)	0.053	(0.003)	0.067	(0.003)	0.077	(0.023)
Calls	6.567	(0.009)	7.560	(0.007)	7.429	(0.001)	7.847	(0.002)
	DIABETES							
P	1030.5	(9.2)	1057.7	(4.5)	700.4	(11.0)	1495.9	(15.2)
TP	551.4	(13.8)	644.4	(7.1)	171.9	(11.7)	638.0	(18.9)
R	458.9	(9.7)	404.7	(6.8)	465.0	(13.5)	673.6	(16.3)
FP	20.2	(2.9)	8.6	(0.6)	63.5	(7.0)	184.2	(7.1)
SHD	1503.8	(14.5)	1399.2	(7.2)	1926.6	(15.5)	1581.2	(20.7)
JI	0.219	(0.006)	0.263	(0.004)	0.067	(0.005)	0.221	(0.008)
Calls	6.090	(0.003)	7.154	(0.001)	7.552	(0.001)	7.964	(0.001)
	MILDEW							
P	1048.7	(18.7)	1032.2	(19.5)	783.3	(14.8)	1493.0	(9.1)
TP	678.7	(20.7)	636.3	(23.6)	163.7	(82.2)	686.0	(12.0)
R	363.5	(13.9)	395.9	(12.4)	613.5	(76.9)	806.0	(13.6)
FP	6.6	(1.9)	0.0	(0.0)	6.2	(2.4)	0.9	(1.3)
SHD	1240.9	(20.0)	1276.7	(23.6)	1755.6	(82.3)	1227.9	(12.0)
JI	0.297	(0.010)	0.313	(0.007)	0.066	(0.034)	0.252	(0.005)
Calls	5.828	(0.008)	6.839	(0.000)	7.343	(0.000)	7.745	(0.001)
	BARLEY							
P	733.9	(10.9)	738.8	(7.6)	585.2	(10.6)	1322.6	(10.8)
TP	385.4	(10.9)	390.9	(8.4)	140.0	(9.7)	671.6	(16.1)
R	328.6	(9.4)	328.9	(7.8)	421.7	(13.7)	580.4	(16.6)
FP	19.9	(1.7)	19.1	(1.2)	23.5	(5.1)	70.6	(5.4)
SHD	1735.6	(10.7)	1729.2	(8.0)	1984.6	(10.4)	1500.0	(17.1)
JI	0.157	(0.005)	0.160	(0.004)	0.055	(0.004)	0.244	(0.007)
Calls	5.944	(0.005)	6.930	(0.001)	7.341	(0.000)	7.930	(0.001)

Table A.2: Detailed results comparing pHGS against HC, MMHC, and H2PC

	pHGS		HC		MMHC		H2PC	
	EARTHQUAKE							
P	911.3	(6.8)	1210.6	(9.6)	874.7	(26.7)	865.3	(21.3)
TP	860.7	(4.7)	588.1	(27.6)	657.2	(49.4)	693.8	(59.2)
R	21.9	(3.8)	313.4	(26.5)	208.2	(54.4)	161.1	(61.5)
FP	28.7	(5.9)	309.1	(10.7)	9.3	(15.6)	10.4	(14.4)
SHD	271.1	(7.7)	824.0	(33.0)	455.1	(56.0)	419.6	(63.2)
JI	0.746	(0.007)	0.341	(0.022)	0.500	(0.060)	0.548	(0.073)
Calls	5.724	(0.000)	6.378	(0.038)	6.321	(0.000)	6.732	(0.003)
	CANCER							
P	1124.1	(7.4)	1306.4	(10.0)	1107.0	(39.1)	1094.6	(16.4)
TP	947.3	(20.7)	716.7	(29.4)	808.9	(33.8)	822.2	(34.7)
R	126.5	(20.7)	364.3	(28.8)	255.9	(32.0)	252.3	(34.2)
FP	50.2	(7.2)	225.5	(10.4)	42.2	(33.0)	20.0	(12.2)
SHD	225.9	(23.4)	631.8	(36.9)	356.3	(39.3)	320.8	(34.6)
JI	0.729	(0.028)	0.419	(0.026)	0.570	(0.036)	0.590	(0.039)
Calls	5.739	(0.001)	6.444	(0.048)	6.326	(0.000)	7.074	(0.004)
	ASIA							
P	876.3	(6.5)	1432.0	(14.5)	852.3	(39.7)	702.0	(21.4)
TP	466.3	(14.0)	653.3	(25.3)	424.4	(20.0)	356.2	(15.6)
R	383.7	(13.7)	446.1	(24.6)	407.7	(21.2)	326.0	(15.0)
FP	26.3	(4.7)	332.5	(17.9)	20.2	(14.3)	19.8	(8.3)
SHD	803.1	(14.2)	922.2	(38.8)	838.9	(17.3)	906.6	(14.9)
JI	0.282	(0.011)	0.323	(0.018)	0.254	(0.012)	0.224	(0.011)
Calls	5.745	(0.002)	6.510	(0.048)	6.323	(0.000)	6.812	(0.003)
	SURVEY							
P	1329.2	(4.0)	1432.2	(7.1)	1327.6	(7.3)	1341.7	(5.6)
TP	1177.5	(24.7)	1093.4	(29.7)	1124.7	(27.8)	1160.7	(25.9)
R	145.2	(23.3)	241.2	(28.2)	190.1	(27.1)	171.6	(25.0)
FP	6.5	(2.3)	97.5	(8.4)	12.8	(5.1)	9.4	(4.0)
SHD	175.9	(24.9)	351.1	(36.1)	235.1	(28.0)	195.8	(25.6)
JI	0.786	(0.029)	0.649	(0.031)	0.726	(0.030)	0.760	(0.029)
Calls	5.742	(0.001)	6.506	(0.046)	6.327	(0.000)	6.823	(0.003)

Table A.2: (continued)

	pHGS		HC		MMHC		H2PC	
	ANDES							
P	1853.6	(8.4)	2572.1	(35.8)	1549.8	(10.7)	1770.7	(16.9)
TP	1751.8	(10.5)	1823.7	(41.2)	1337.3	(21.4)	1633.3	(24.0)
R	57.8	(5.6)	270.2	(32.6)	185.2	(18.7)	130.5	(19.2)
FP	44.0	(5.7)	478.3	(39.7)	27.3	(5.3)	6.9	(5.1)
SHD	537.2	(12.4)	899.6	(77.8)	935.0	(21.2)	618.6	(23.3)
JI	0.746	(0.006)	0.610	(0.027)	0.544	(0.012)	0.686	(0.015)
Calls	5.924	(0.004)	6.776	(0.047)	6.442	(0.000)	7.201	(0.004)
	WIN95PTS							
P	1308.6	(7.5)	2334.8	(24.8)	1024.0	(13.9)	1167.4	(12.6)
TP	1111.9	(8.1)	1028.4	(36.9)	649.9	(19.3)	832.0	(21.0)
R	158.2	(5.8)	501.5	(24.7)	358.2	(18.4)	323.8	(18.8)
FP	38.5	(4.1)	804.8	(36.6)	15.9	(4.4)	11.6	(4.3)
SHD	1110.6	(9.5)	1960.4	(71.0)	1550.0	(19.2)	1363.6	(20.9)
JI	0.467	(0.004)	0.295	(0.015)	0.254	(0.009)	0.330	(0.010)
Calls	5.864	(0.002)	6.627	(0.033)	6.402	(0.001)	6.926	(0.003)
	CHILD							
P	1298.8	(1.1)	1438.7	(11.6)	969.6	(12.4)	1298.0	(2.4)
TP	1198.1	(35.1)	851.6	(39.4)	364.8	(21.8)	905.3	(27.7)
R	100.7	(34.9)	449.2	(38.9)	597.8	(17.3)	391.5	(27.5)
FP	0.1	(0.3)	137.9	(12.2)	7.0	(3.3)	1.1	(2.0)
SHD	111.0	(35.2)	595.4	(45.2)	951.2	(22.1)	404.8	(27.9)
JI	0.851	(0.046)	0.450	(0.032)	0.191	(0.013)	0.532	(0.025)
Calls	5.798	(0.001)	6.486	(0.046)	6.338	(0.001)	6.698	(0.001)
	ALARM							
P	1282.9	(4.8)	1828.8	(15.5)	1020.6	(16.4)	1399.9	(11.4)
TP	1010.1	(5.5)	932.5	(32.0)	606.5	(21.2)	1020.0	(28.5)
R	269.9	(3.9)	511.8	(28.3)	403.2	(17.8)	371.8	(24.3)
FP	3.0	(1.5)	384.6	(19.6)	10.9	(3.8)	8.1	(3.4)
SHD	681.9	(5.5)	1141.1	(46.1)	1093.5	(22.2)	677.1	(28.3)
JI	0.515	(0.003)	0.361	(0.018)	0.288	(0.012)	0.493	(0.019)
Calls	5.837	(0.002)	6.577	(0.047)	6.382	(0.001)	6.812	(0.002)

Table A.2: (continued)

	pHGS		HC		MMHC		H2PC	
	MIX							
P	1547.8	(6.3)	1937.4	(22.7)	1171.8	(11.6)	1563.5	(10.8)
TP	1440.1	(11.5)	1473.2	(32.5)	851.8	(22.5)	1410.0	(19.0)
R	94.2	(9.0)	231.7	(28.0)	303.2	(19.2)	150.2	(19.0)
FP	13.5	(3.4)	232.5	(26.7)	16.8	(5.7)	3.3	(2.6)
SHD	450.4	(12.3)	636.3	(56.2)	1042.0	(23.8)	470.3	(19.0)
JI	0.726	(0.009)	0.630	(0.027)	0.388	(0.013)	0.695	(0.015)
Calls	6.223	(0.008)	6.631	(0.051)	6.397	(0.005)	6.950	(0.003)
	SACHS							
P	1797.5	(2.5)	1944.2	(10.7)	1373.0	(8.2)	1779.1	(6.5)
TP	1615.7	(40.7)	911.6	(34.9)	690.9	(28.4)	1061.1	(43.7)
R	181.9	(40.4)	855.0	(32.3)	680.8	(28.6)	718.0	(41.1)
FP	0.0	(0.0)	177.6	(12.6)	1.3	(1.2)	0.0	(0.2)
SHD	206.3	(40.7)	1088.0	(42.5)	1132.4	(28.5)	760.9	(43.7)
JI	0.807	(0.036)	0.320	(0.017)	0.276	(0.014)	0.418	(0.024)
Calls	5.847	(0.002)	6.577	(0.038)	6.346	(0.000)	6.905	(0.001)
	PIGS							
P	2170.6	(1.3)	2298.6	(27.5)	1533.8	(15.3)	2172.9	(2.1)
TP	2166.0	(1.7)	2026.6	(39.3)	1152.9	(35.4)	2166.2	(5.5)
R	4.6	(0.6)	152.5	(38.7)	357.2	(28.5)	6.7	(4.6)
FP	0.0	(0.0)	119.5	(28.0)	23.6	(7.0)	0.0	(0.0)
SHD	16.0	(1.7)	275.0	(66.1)	1052.7	(38.0)	15.8	(5.5)
JI	0.991	(0.001)	0.827	(0.039)	0.450	(0.019)	0.990	(0.004)
Calls	6.523	(0.009)	6.856	(0.052)	6.650	(0.005)	7.228	(0.003)
	HEPAR2							
P	1424.5	(7.7)	1781.5	(12.7)	988.3	(27.9)	1576.6	(15.9)
TP	1245.9	(15.6)	1335.3	(29.9)	606.1	(25.3)	1347.6	(32.6)
R	158.5	(14.2)	296.4	(26.4)	336.4	(23.9)	219.4	(30.5)
FP	20.1	(5.2)	149.8	(14.6)	45.7	(20.1)	9.6	(4.1)
SHD	851.2	(16.8)	891.6	(41.5)	1516.5	(28.0)	739.0	(32.1)
JI	0.552	(0.010)	0.529	(0.020)	0.247	(0.012)	0.585	(0.021)
Calls	5.978	(0.003)	6.517	(0.028)	6.377	(0.000)	6.912	(0.002)

Table A.2: (continued)

	pHGS		HC		MMHC		H2PC	
	INSURANCE							
P	1512.0	(4.8)	2003.5	(9.9)	995.0	(13.2)	1536.9	(20.4)
TP	1375.2	(33.5)	1108.5	(35.6)	545.4	(22.8)	1051.4	(26.8)
R	133.8	(32.2)	582.6	(31.5)	426.8	(18.6)	469.7	(25.4)
FP	3.0	(1.3)	312.4	(15.4)	22.8	(4.6)	15.8	(6.3)
SHD	747.8	(33.6)	1323.9	(45.1)	1597.5	(24.4)	1084.4	(28.3)
JI	0.610	(0.023)	0.368	(0.016)	0.212	(0.010)	0.404	(0.014)
Calls	5.910	(0.002)	6.584	(0.046)	6.376	(0.001)	6.958	(0.002)
	HAILFINDER							
P	986.8	(8.9)	1574.8	(15.0)	838.9	(13.1)	1101.1	(18.2)
TP	783.3	(7.9)	974.6	(55.0)	518.7	(22.5)	825.3	(34.9)
R	117.1	(5.0)	328.6	(48.8)	273.1	(19.1)	275.5	(31.2)
FP	86.4	(6.7)	271.6	(21.0)	47.1	(11.5)	0.3	(0.6)
SHD	844.1	(9.7)	838.0	(71.8)	1069.4	(23.9)	715.9	(34.9)
JI	0.449	(0.006)	0.456	(0.040)	0.279	(0.014)	0.455	(0.026)
Calls	5.790	(0.004)	6.524	(0.050)	6.354	(0.001)	6.617	(0.001)
	WATER							
P	1321.1	(7.6)	1842.9	(15.6)	1312.7	(9.7)	1449.6	(9.9)
TP	1032.5	(25.9)	786.0	(40.0)	763.9	(29.8)	926.1	(36.4)
R	288.3	(24.0)	632.6	(30.7)	547.4	(28.4)	517.4	(33.0)
FP	0.3	(0.5)	424.3	(22.8)	1.4	(1.3)	6.1	(2.6)
SHD	1270.7	(26.0)	1941.4	(58.3)	1540.4	(29.9)	1383.0	(36.2)
JI	0.399	(0.014)	0.234	(0.015)	0.268	(0.013)	0.328	(0.017)
Calls	5.779	(0.002)	6.553	(0.036)	6.344	(0.000)	6.779	(0.001)
	MUNIN1							
P	518.2	(5.4)	1692.7	(10.3)	267.8	(10.0)	710.7	(14.1)
TP	120.8	(3.0)	717.4	(23.4)	14.3	(4.5)	217.8	(19.7)
R	364.2	(4.9)	318.1	(17.1)	210.8	(9.8)	444.4	(17.7)
FP	33.3	(2.2)	657.2	(17.9)	42.7	(6.3)	48.5	(5.7)
SHD	1688.5	(3.9)	1715.8	(37.8)	1804.4	(8.3)	1606.6	(20.5)
JI	0.056	(0.001)	0.261	(0.011)	0.007	(0.002)	0.096	(0.009)
Calls	6.437	(0.005)	6.449	(0.020)	6.562	(0.008)	7.343	(0.003)

Table A.2: (continued)

	pHGS		HC		MMHC		H2PC	
	PATHFINDER							
P	290.6	(3.7)	1430.4	(7.6)	330.9	(7.4)	1107.7	(17.9)
TP	151.6	(5.1)	894.3	(41.1)	172.4	(7.7)	787.1	(55.0)
R	135.6	(4.5)	348.0	(41.8)	142.8	(6.7)	274.2	(54.9)
FP	3.4	(1.9)	188.2	(7.5)	15.7	(3.1)	46.4	(5.8)
SHD	1819.8	(5.5)	1261.8	(42.0)	1811.3	(8.5)	1227.2	(54.9)
JI	0.072	(0.003)	0.358	(0.022)	0.081	(0.004)	0.345	(0.032)
Calls	6.531	(0.009)	6.437	(0.029)	6.429	(0.001)	6.844	(0.002)
	DIABETES							
P	1053.3	(9.3)	2209.5	(14.2)	702.5	(12.0)	1422.5	(12.4)
TP	508.6	(14.3)	770.0	(42.0)	161.8	(11.2)	671.2	(33.4)
R	523.0	(8.7)	741.0	(31.1)	477.7	(14.7)	602.1	(28.5)
FP	21.7	(3.2)	698.5	(22.9)	63.1	(6.8)	149.1	(6.5)
SHD	1548.1	(14.1)	1963.5	(59.3)	1936.3	(13.6)	1512.8	(36.0)
JI	0.197	(0.006)	0.222	(0.015)	0.063	(0.005)	0.241	(0.015)
Calls	6.040	(0.002)	6.650	(0.029)	6.552	(0.001)	6.964	(0.001)
	MILDEW							
P	1094.8	(7.8)	1599.2	(11.1)	778.1	(11.7)	1314.7	(15.2)
TP	729.1	(15.7)	958.0	(31.6)	257.7	(17.6)	986.4	(37.8)
R	365.4	(11.6)	399.3	(21.9)	515.3	(15.8)	327.9	(26.2)
FP	0.2	(0.5)	241.9	(11.3)	5.1	(2.4)	0.4	(0.8)
SHD	1184.2	(15.7)	1196.9	(40.9)	1660.4	(18.1)	927.0	(37.9)
JI	0.320	(0.008)	0.375	(0.017)	0.106	(0.008)	0.440	(0.022)
Calls	5.772	(0.002)	6.529	(0.049)	6.343	(0.000)	6.744	(0.001)
	BARLEY							
P	758.6	(10.1)	1603.1	(7.5)	587.9	(10.9)	1170.5	(9.9)
TP	404.6	(12.6)	763.1	(35.1)	171.7	(13.8)	732.5	(35.9)
R	332.6	(8.0)	468.2	(23.5)	393.5	(13.8)	400.8	(30.1)
FP	21.3	(1.8)	371.9	(16.3)	22.7	(5.1)	37.2	(5.7)
SHD	1717.7	(12.0)	1709.8	(50.2)	1952.0	(14.3)	1405.7	(38.0)
JI	0.165	(0.005)	0.260	(0.015)	0.068	(0.006)	0.289	(0.018)
Calls	5.869	(0.002)	6.507	(0.050)	6.342	(0.000)	6.929	(0.001)

Table A.3: Detailed results comparing pHGS against GES and FGES

	pHGS		GES		FGES	
	EARTHQUAKE					
P	911.3	(6.8)	1099.5	(7.9)	1082.0	(6.9)
TP	860.7	(4.7)	680.3	(18.8)	686.8	(19.0)
R	21.9	(3.8)	227.5	(17.5)	220.1	(17.6)
FP	28.7	(5.9)	191.7	(9.0)	175.1	(8.6)
SHD	271.1	(7.7)	614.4	(26.5)	591.2	(26.8)
JI	0.746	(0.007)	0.447	(0.020)	0.459	(0.020)
Calls	5.724	(0.000)	6.395	(0.046)	5.735	(0.002)
	CANCER					
P	1124.1	(7.4)	1203.3	(9.2)	1179.2	(8.1)
TP	947.3	(20.7)	949.5	(15.5)	947.0	(16.3)
R	126.5	(20.7)	135.0	(15.0)	131.3	(15.9)
FP	50.2	(7.2)	118.8	(9.4)	101.0	(8.2)
SHD	225.9	(23.4)	292.4	(21.7)	276.9	(21.5)
JI	0.729	(0.028)	0.690	(0.021)	0.699	(0.022)
Calls	5.739	(0.001)	6.326	(0.038)	5.724	(0.001)
	ASIA					
P	876.3	(6.5)	1199.7	(6.1)	1176.4	(5.8)
TP	466.3	(14.0)	947.9	(17.9)	941.1	(18.8)
R	383.7	(13.7)	177.4	(16.9)	175.6	(18.0)
FP	26.3	(4.7)	74.4	(7.8)	59.7	(6.8)
SHD	803.1	(14.2)	369.6	(22.2)	361.7	(22.5)
JI	0.282	(0.011)	0.634	(0.020)	0.637	(0.021)
Calls	5.745	(0.002)	6.374	(0.052)	5.751	(0.001)
	SURVEY					
P	1329.2	(4.0)	1431.0	(7.8)	1418.7	(7.9)
TP	1177.5	(24.7)	992.4	(23.5)	987.3	(24.0)
R	145.2	(23.3)	340.5	(22.2)	340.6	(22.9)
FP	6.5	(2.3)	98.2	(7.6)	90.7	(7.6)
SHD	175.9	(24.9)	452.8	(27.5)	450.4	(28.2)
JI	0.786	(0.029)	0.556	(0.021)	0.556	(0.022)
Calls	5.742	(0.001)	6.237	(0.013)	5.722	(0.000)

Table A.3: (continued)

	pHGS		GES		FGES	
	ANDES					
P	1853.6	(8.4)	2398.4	(27.0)	2341.8	(27.4)
TP	1751.8	(10.5)	2009.2	(17.0)	1934.6	(16.1)
R	57.8	(5.6)	83.8	(11.1)	94.3	(10.7)
FP	44.0	(5.7)	305.4	(28.1)	312.9	(28.2)
SHD	537.2	(12.4)	541.1	(41.5)	623.4	(40.8)
JI	0.746	(0.006)	0.763	(0.016)	0.730	(0.015)
Calls	5.924	(0.004)	6.438	(0.010)	5.918	(0.002)
	WIN95PTS					
P	1308.6	(7.5)	2023.0	(13.4)	1973.0	(12.1)
TP	1111.9	(8.1)	1403.3	(19.1)	1366.5	(21.0)
R	158.2	(5.8)	225.3	(10.7)	224.1	(10.9)
FP	38.5	(4.1)	394.4	(17.4)	382.4	(18.6)
SHD	1110.6	(9.5)	1175.1	(33.8)	1199.9	(37.5)
JI	0.467	(0.004)	0.501	(0.010)	0.490	(0.012)
Calls	5.864	(0.002)	6.327	(0.008)	5.893	(0.001)
	CHILD					
P	1298.8	(1.1)	1312.0	(1.8)	1304.6	(5.8)
TP	1198.1	(35.1)	1239.0	(11.5)	1200.5	(67.8)
R	100.7	(34.9)	66.7	(11.6)	99.4	(62.7)
FP	0.1	(0.3)	6.3	(1.7)	4.7	(1.3)
SHD	111.0	(35.2)	76.3	(12.0)	113.2	(67.5)
JI	0.851	(0.046)	0.897	(0.016)	0.853	(0.079)
Calls	5.798	(0.001)	6.277	(0.052)	5.785	(0.001)
	ALARM					
P	1282.9	(4.8)	1649.4	(9.9)	1631.2	(9.4)
TP	1010.1	(5.5)	1322.2	(18.4)	1313.2	(17.9)
R	269.9	(3.9)	203.8	(15.3)	203.5	(15.1)
FP	3.0	(1.5)	123.4	(9.5)	114.5	(8.7)
SHD	681.9	(5.5)	490.2	(25.5)	490.3	(24.0)
JI	0.515	(0.003)	0.656	(0.016)	0.654	(0.015)
Calls	5.837	(0.002)	6.306	(0.016)	5.862	(0.001)

Table A.3: (continued)

	pHGS		GES		FGES	
	MIX					
P	1547.8	(6.3)	1850.7	(14.8)	1829.6	(15.0)
TP	1440.1	(11.5)	1592.2	(14.9)	1567.6	(13.4)
R	94.2	(9.0)	126.0	(10.6)	130.2	(9.8)
FP	13.5	(3.4)	132.5	(17.1)	131.8	(15.5)
SHD	450.4	(12.3)	417.4	(28.7)	441.3	(24.6)
JI	0.726	(0.009)	0.746	(0.014)	0.733	(0.012)
Calls	6.223	(0.008)	6.317	(0.012)	5.907	(0.002)
	SACHS					
P	1797.5	(2.5)	1864.4	(6.8)	1862.3	(7.0)
TP	1615.7	(40.7)	1544.3	(47.4)	1537.8	(47.1)
R	181.9	(40.4)	264.3	(46.6)	270.0	(46.5)
FP	0.0	(0.0)	55.8	(7.3)	54.4	(7.2)
SHD	206.3	(40.7)	333.6	(52.5)	338.6	(51.7)
JI	0.807	(0.036)	0.722	(0.040)	0.717	(0.039)
Calls	5.847	(0.002)	6.451	(0.028)	5.855	(0.005)
	PIGS					
P	2170.6	(1.3)	2280.2	(15.7)	2280.6	(16.0)
TP	2166.0	(1.7)	2124.7	(7.5)	2122.2	(7.7)
R	4.6	(0.6)	49.7	(6.8)	51.2	(7.0)
FP	0.0	(0.0)	105.7	(15.2)	107.2	(15.5)
SHD	16.0	(1.7)	163.0	(21.8)	167.0	(22.3)
JI	0.991	(0.001)	0.909	(0.011)	0.907	(0.012)
Calls	6.523	(0.009)	6.441	(0.009)	6.139	(0.002)
	HEPAR2					
P	1424.5	(7.7)	1745.2	(12.5)	1685.5	(12.5)
TP	1245.9	(15.6)	1375.9	(19.2)	1325.1	(18.1)
R	158.5	(14.2)	239.2	(14.3)	243.1	(14.7)
FP	20.1	(5.2)	130.1	(13.3)	117.3	(12.0)
SHD	851.2	(16.8)	831.3	(28.9)	869.2	(26.8)
JI	0.552	(0.010)	0.563	(0.013)	0.544	(0.012)
Calls	5.978	(0.003)	6.317	(0.010)	5.822	(0.002)

Table A.3: (continued)

	pHGS		GES		FGES	
	INSURANCE					
P	1512.0	(4.8)	1899.5	(7.8)	1848.4	(7.9)
TP	1375.2	(33.5)	1198.9	(15.9)	1178.4	(15.7)
R	133.8	(32.2)	479.0	(15.0)	479.5	(14.7)
FP	3.0	(1.3)	221.6	(8.0)	190.4	(7.1)
SHD	747.8	(33.6)	1142.8	(22.2)	1131.9	(20.9)
JI	0.610	(0.023)	0.425	(0.009)	0.422	(0.008)
Calls	5.910	(0.002)	6.368	(0.006)	5.868	(0.002)
	HAILFINDER					
P	986.8	(8.9)	1493.8	(5.0)	1480.8	(5.2)
TP	783.3	(7.9)	1191.0	(29.4)	1192.9	(28.5)
R	117.1	(5.0)	143.6	(24.1)	135.1	(23.0)
FP	86.4	(6.7)	159.3	(12.0)	152.8	(13.0)
SHD	844.1	(9.7)	509.3	(37.6)	500.9	(37.3)
JI	0.449	(0.006)	0.646	(0.026)	0.653	(0.026)
Calls	5.790	(0.004)	6.267	(0.013)	5.819	(0.001)
	WATER					
P	1321.1	(7.6)	1672.4	(10.7)	1642.3	(10.6)
TP	1032.5	(25.9)	1048.8	(13.6)	1033.8	(13.9)
R	288.3	(24.0)	427.3	(10.2)	427.5	(10.4)
FP	0.3	(0.5)	196.3	(12.4)	181.0	(12.6)
SHD	1270.7	(26.0)	1450.5	(23.5)	1450.2	(24.3)
JI	0.399	(0.014)	0.358	(0.007)	0.355	(0.007)
Calls	5.779	(0.002)	6.250	(0.006)	5.783	(0.001)
	MUNIN1					
P	518.2	(5.4)	1629.6	(5.8)	1614.4	(5.8)
TP	120.8	(3.0)	890.5	(17.7)	891.3	(17.1)
R	364.2	(4.9)	215.7	(12.1)	216.0	(11.6)
FP	33.3	(2.2)	523.4	(13.3)	507.1	(12.5)
SHD	1688.5	(3.9)	1408.8	(29.0)	1391.8	(27.5)
JI	0.056	(0.001)	0.354	(0.010)	0.357	(0.009)
Calls	6.437	(0.005)	6.423	(0.004)	6.194	(0.003)

Table A.3: (continued)

	pHGS		GES		FGES	
	PATHFINDER					
P	290.6	(3.7)	1458.5	(5.1)	1458.5	(5.3)
TP	151.6	(5.1)	1017.9	(36.8)	1006.9	(62.9)
R	135.6	(4.5)	262.0	(32.8)	274.5	(63.5)
FP	3.4	(1.9)	178.7	(10.4)	177.1	(9.9)
SHD	1819.8	(5.5)	1128.8	(43.6)	1138.2	(64.2)
JI	0.072	(0.003)	0.423	(0.022)	0.417	(0.034)
Calls	6.531	(0.009)	6.560	(0.061)	6.328	(0.049)
	DIABETES					
P	1053.3	(9.3)	1988.2	(8.7)	1972.2	(8.7)
TP	508.6	(14.3)	1303.6	(37.7)	1305.0	(41.4)
R	523.0	(8.7)	399.1	(33.8)	395.2	(37.4)
FP	21.7	(3.2)	285.5	(11.2)	272.0	(11.1)
SHD	1548.1	(14.1)	1016.8	(46.0)	1002.0	(49.4)
JI	0.197	(0.006)	0.480	(0.021)	0.483	(0.023)
Calls	6.040	(0.002)	6.496	(0.011)	6.043	(0.001)
	MILDEW					
P	1094.8	(7.8)	1526.1	(10.5)	1502.7	(10.6)
TP	729.1	(15.7)	1241.5	(14.2)	1232.8	(14.2)
R	365.4	(11.6)	178.2	(7.0)	178.0	(7.2)
FP	0.2	(0.5)	106.4	(8.0)	91.9	(7.0)
SHD	1184.2	(15.7)	777.9	(19.5)	772.1	(17.8)
JI	0.320	(0.008)	0.565	(0.009)	0.565	(0.009)
Calls	5.772	(0.002)	6.138	(0.016)	5.752	(0.000)
	BARLEY					
P	758.6	(10.1)	1481.1	(8.5)	1462.7	(8.2)
TP	404.6	(12.6)	1120.8	(24.4)	1117.6	(24.7)
R	332.6	(8.0)	212.9	(14.8)	213.6	(15.3)
FP	21.3	(1.8)	147.4	(10.1)	131.5	(8.2)
SHD	1717.7	(12.0)	1127.7	(33.3)	1114.9	(31.8)
JI	0.165	(0.005)	0.455	(0.014)	0.457	(0.014)
Calls	5.869	(0.002)	6.171	(0.016)	5.782	(0.001)

Bibliography

Rajeev Agrawal. Sample mean based index policies by $O(\log n)$ regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4):1054–1078, 1995. https://doi.org/10.2307/1427934.

Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974. https://doi.org/10.1109/tac.1974.1100705.

Constantin F Aliferis, Ioannis Tsamardinos, and Alexander Statnikov. HITON: a novel Markov Blanket algorithm for optimal variable selection. In *AMIA Annual Symposium proceedings*, volume 2003, pages 21–25. AMIA Symposium, 2003. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1480117/.

Constantin F. Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D. 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(7):171–234, 2010. http://jmlr.org/papers/v11/aliferis10a.html.

Bryon Aragam, Jiaying Gu, and Qing Zhou. Learning Large-Scale Bayesian Networks with the sparsebn Package. *Journal of Statistical Software*, 91(11):1–38, 2019. https://doi.org/10.18637/jss.v091.i11.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine learning*, 47(2):235–256, 2002. https://doi.org/10.1023/A:1013689704352.

Donald A Berry and Bert Fristedt. Bandit problems: sequential allocation of experiments

(Monographs on statistics and applied probability). *London: Chapman and Hall*, 5(71-87): 7–7, 1985. https://link.springer.com/book/10.1007/978-94-015-3711-7.

Wray Buntine. Theory Refinement on Bayesian Networks. In Bruce D. D'Ambrosio, Philippe Smets, and Piero P. Bonissone, editors, *Uncertainty Proceedings 1991*, pages 52–60. Morgan Kaufmann, San Francisco (CA), 1991. ISBN 978-1-55860-203-8. https://doi.org/10.1016/B978-1-55860-203-8.50010-3.

David Maxwell Chickering. Learning Equivalence Classes of Bayesian-Network Structures. *Journal of Machine Learning Research*, 2(Feb):445–498, 2002a. https://www.jmlr.org/papers/v2/chickering02a.html.

David Maxwell Chickering. Optimal Structure Identification With Greedy Search. *Journal of Machine Learning Research*, 3(Nov):507–554, 2002b. https://jmlr.org/papers/v3/chickering02b.html.

David Maxwell Chickering, David Heckerman, and Christopher Meek. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research*, 5(Oct):1287–1330, 2004. https://www.jmlr.org/papers/v5/chickering04a.html.

Diego Colombo and Marloes H. Maathuis. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15(116):3921–3962, 2014. http://jmlr.org/papers/v15/colombo14a.html.

Gregory F. Cooper and Edward Herskovits. A Bayesian Method for Constructing Bayesian Belief Networks from Databases. In Bruce D. D'Ambrosio, Philippe Smets, and Piero P. Bonissone, editors, *Uncertainty Proceedings 1991*, pages 86–94. Morgan Kaufmann, San Francisco (CA), 1991. ISBN 978-1-55860-203-8. https://doi.org/10.1016/B978-1-55860-203-8.50015-2.

Noel Cressie and Timothy R. C. Read. Pearson's χ^2 and the Loglikelihood Ratio Statistic G^2: A Comparative Review. *International Statistical Review / Revue Internationale de Statistique*, 57(1):19–43, 1989. ISSN 03067734, 17515823. https://doi.org/10.2307/1403582.

Arnoud de Kroon, Joris Mooij, and Danielle Belgrave. Causal bandits without prior knowledge using separating sets. In *First Conference on Causal Learning and Reasoning*, 2022. https://openreview.net/forum?id=50eDSLScz7r.

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

Daniel Eaton and Kevin Murphy. Exact Bayesian structure learning from uncertain interventions. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 107–114, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR. https://proceedings.mlr.press/v2/eaton07a.html.

Nir Friedman and Daphne Koller. Being Bayesian About Network Structure. A Bayesian Approach to Structure Discovery in Bayesian Networks. *Machine learning*, 50(1):95–125, 2003. https://doi.org/10.1023/A:1020249912095.

Nir Friedman, Iftach Nachman, and Dana Peér. Learning Bayesian Network Structure from Massive Datasets: The "Sparse Candidate" Algorithm. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, page 206–215, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606149. https://dl.acm.org/doi/10.5555/2073796.2073820.

Fei Fu and Qing Zhou. Learning Sparse Causal Gaussian Networks With Experimental Intervention: Regularization and Coordinate Descent. *Journal of the American Statistical Association*, 108(501):288–300, 2013. https://doi.org/10.1080/01621459.2012.754359.

José A Gámez, Juan L Mateo, and José M Puerta. Learning Bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood. *Data Mining and Knowledge Discovery*, 22(1-2):106–148, 2011. https://doi.org/10.1007/s10618-010-0178-6.

Maxime Gasse, Alex Aussem, and Haytham Elghazel. A hybrid algorithm for Bayesian network structure learning with application to multi-label learning. *Expert Systems with Applications*, 41(15):6755–6772, 2014. ISSN 0957-4174. https://doi.org/10.1016/j.eswa.2014.04.032.

Kristjan Greenewald, Dmitriy Katz, Karthikeyan Shanmugam, Sara Magliacane, Murat Kocaoglu, Enric Boix Adsera, and Guy Bresler. Sample Efficient Active Learning of Causal Trees. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. https://proceedings.neurips.cc/paper/2019/file/5ee5605917626676f6a285fa4c10f7b0-Paper.pdf.

Jiaying Gu and Qing Zhou. Learning Big Gaussian Bayesian Networks: Partition, Estimation and Fusion. *Journal of Machine Learning Research*, 21(158):1–31, 2020. http://jmlr.org/papers/v21/19-318.html.

Jiaying Gu, Fei Fu, and Qing Zhou. Penalized estimation of directed acyclic graphs from discrete data. *Statistics and Computing*, 29(1):161–176, 2019. https://doi.org/10.1007/s11222-018-9801-y.

John A Hartigan. Consistency of Single Linkage for High-Density Clusters. *Journal of the American Statistical Association*, 76(374):388–394, 1981. https://doi.org/10.1080/01621459.1981.10477658.

David Heckerman, Dan Geiger, and David M Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995. https://doi.org/10.1007/BF00994016.

Jireh Huang and Qing Zhou. Partitioned hybrid learning of Bayesian network structures. *Machine Learning*, 2022a. https://doi.org/10.1007/s10994-022-06145-4.

Jireh Huang and Qing Zhou. Bayesian Causal Bandits with Backdoor Adjustment Prior. Manuscript submitted for review, 2022b.

Markus Kalisch and Peter Bühlmann. Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm. *Journal of Machine Learning Research*, 8(22):613–636, 2007. http://jmlr.org/papers/v8/kalisch07a.html.

Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H. Maathuis, and Peter Bühlmann. Causal Inference Using Graphical Models with the R Package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012. ISSN 1548-7660. https://doi.org/10.18637/jss.v047.i11.

Emilie Kaufmann, Olivier Cappe, and Aurelien Garivier. On Bayesian Upper Confidence Bounds for Bandit Problems. In Neil D. Lawrence and Mark Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 592–600, La Palma, Canary Islands, 21–23 Apr 2012. PMLR. https://proceedings.mlr.press/v22/kaufmann12.html.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009. ISBN 0262013193.

Alexander Kraskov, Harald Stögbauer, Ralph G Andrzejak, and Peter Grassberger. Hierarchical clustering using mutual information. *EPL (Europhysics Letters)*, 70(2):278–284, apr 2005. https://doi.org/10.1209/epl/i2004-10483-y.

Jack Kuipers and Giusi Moffa. Partition MCMC for Inference on Acyclic Digraphs. *Journal of the American Statistical Association*, 112(517):282–299, 2017. https://doi.org/10.1080/01621459.2015.1133426.

Jack Kuipers, Polina Suter, and Giusi Moffa. Efficient Sampling and Structure Learning of Bayesian Networks. *Journal of Computational and Graphical Statistics*, 0(0):1–12, 2022. https://doi.org/10.1080/10618600.2021.2020127.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985. ISSN 0196-8858. https://doi.org/10.1016/0196-8858(85)90002-8.

Finnian Lattimore, Tor Lattimore, and Mark D Reid. Causal Bandits: Learning Good Interventions via Causal Inference. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. https://proceedings.neurips.cc/paper/2016/file/b4288d9c0ec0a1841b3b3728321e7088-Paper.pdf.

Thuc Duy Le, Tao Hoang, Jiuyong Li, Lin Liu, Huawen Liu, and Shu Hu. A Fast PC Algorithm for High Dimensional Causal Discovery with Multi-Core PCs. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16:1483–1495, 2016. https://doi.org/10.1109/TCBB.2016.2591526.

Sanghack Lee and Elias Bareinboim. Structural causal bandits: Where to intervene? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. https://proceedings.neurips.cc/paper/2018/file/c0a271bc0ecb776a094786474322cb82-Paper.pdf.

Zhifa Liu, Brandon Malone, and Changhe Yuan. Empirical evaluation of scoring functions for Bayesian network model selection. In *BMC Bioinformatics*, volume 13, pages 1–16. Springer, 2012. https://doi.org/10.1186/1471-2105-13-S15-S14.

Yangyi Lu, Amirhossein Meisami, Ambuj Tewari, and William Yan. Regret Analysis of Bandit Problems with Causal Background Knowledge. In Jonas Peters and David Sontag, editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 141–150. PMLR, 03–06 Aug 2020. https://proceedings.mlr.press/v124/lu20a.html.

Yangyi Lu, Amirhossein Meisami, and Ambuj Tewari. Causal Bandits with Unknown Graph Structure. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 24817–24828. Curran Associates, Inc., 2021. https://proceedings.neurips.cc/paper/2021/file/d010396ca8abf6ead8cacc2c2f2f26c7-Paper.pdf.

Marloes H. Maathuis, Markus Kalisch, and Peter Bühlmann. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 37(6A):3133 – 3164, 2009. https://doi.org/10.1214/09-AOS685.

David Madigan, Jeremy York, and Denis Allard. Bayesian Graphical Models for Discrete Data. *International Statistical Review / Revue Internationale de Statistique*, 63(2):215–232, 1995. ISSN 03067734, 17515823. http://www.jstor.org/stable/1403615.

Anders L. Madsen, Frank Jensen, Antonio Salmerón, Helge Langseth, and Thomas D. Nielsen. A parallel algorithm for Bayesian network structure learning from large data sets. *Knowledge-Based Systems*, 117:46–55, 2017. ISSN 0950-7051. https://doi.org/10.1016/j.knosys.2016.07.031.

Aurghya Maiti, Vineet Nair, and Gaurav Sinha. Causal Bandits on General Graphs. *arXiv preprint arXiv:2107.02772*, 2021. https://arxiv.org/abs/2107.02772.

Dimitris Margaritis. *Learning Bayesian network model structure from data*. PhD thesis, Carnegie Mellon University School of Computer Science, 2003. https://apps.dtic.mil/sti/citations/ADA461103.

Christopher Meek. Causal Inference and Causal Explanation with Background Knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, page 403–410, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859. https://dl.acm.org/doi/10.5555/2074158.2074204.

Christopher Meek. *Graphical Models: Selecting causal and statistical models*. PhD thesis, Carnegie Mellon University School of Computer Science, 1997.

Vineet Nair, Vishakha Patil, and Gaurav Sinha. Budgeted and Non-Budgeted Causal Bandits. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2017–2025. PMLR, 13–15 Apr 2021. https://proceedings.mlr.press/v130/nair21a.html.

Preetam Nandy, Alain Hauser, and Marloes H. Maathuis. High-dimensional consistency in score-based and hybrid structure learning. *The Annals of Statistics*, 46(6A):3151–3183, 2018a. https://doi.org/10.1214/17-AOS1654.

Preetam Nandy, Alain Hauser, and Marloes H. Maathuis. High-dimensional consistency in score-based and hybrid structure learning. *The Annals of Statistics*, 46(6A):3151–3183, 2018b. https://doi.org/10.1214/17-AOS1654.

Richard E Neapolitan. *Learning Bayesian Networks*, volume 38. Pearson Prentice Hall, 2004.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 12 1995. ISSN 0006-3444. https://doi.org/10.1093/biomet/82.4.669.

Judea Pearl. *Causality*. Cambridge University Press, 2000.

Johan Pensar, Topi Talvitie, Antti Hyttinen, and Mikko Koivisto. A Bayesian Approach for Estimating Causal Effects from Observational Data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5395–5402, Apr. 2020. https://ojs.aaai.org/index.php/AAAI/article/view/5988.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021. https://www.R-project.org/.

Joseph D Ramsey. Scaling up Greedy Equivalence Search for Continuous Variables. *Computing Research Repository*, abs/1507.07749, 2015. http://arxiv.org/abs/1507.07749.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978. ISSN 0005-1098. https://doi.org/10.1016/0005-1098(78)90005-5.

Robert W Robinson. Counting unlabeled acyclic digraphs. In Charles H. C. Little, editor, *Combinatorial Mathematics V*, pages 28–43. Springer, Berlin, Heidelberg, 1977. ISBN 978-3-540-37020-8. https://doi.org/10.1007/BFb0069178.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson

Prentice Hall., third edition, 2009.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005. https://doi.org/10.1126/science.1105809.

Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2): 461–464, 1978. ISSN 00905364. https://doi.org/10.1214/aos/1176344136.

Marco Scutari. Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software*, 35(3):1—22, 2010. https://doi.org/10.18637/jss.v035.i03.

Marco Scutari. Bayesian Network Constraint-Based Structure Learning Algorithms: Parallel and Optimized Implementations in the bnlearn R Package. *Journal of Statistical Software*, 77(2):1—20, 2017. https://doi.org/10.18637/jss.v077.i02.

Jun Shao. *Mathematical Statistics*, pages 91–160. Springer, second edition, 2003.

Peter Spirtes. Introduction to Causal Inference. *Journal of Machine Learning Research*, 11 (54):1643–1662, 2010. http://jmlr.org/papers/v11/spirtes10a.html.

Peter Spirtes and Clark Glymour. An Algorithm for Fast Recovery of Sparse Causal Graphs. *Social Science Computer Review*, 9(1):62–72, 1991. https://doi.org/10.1177/089443939100900106.

Peter Spirtes, Clark Glymour, Richard Scheines, and David Heckerman. *Causation, Prediction, and Search*. MIT Press, second edition, 2000.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 12 1933. ISSN 0006-3444. https://doi.org/10.1093/biomet/25.3-4.285.

Ioannis Tsamardinos, Constantin F. Aliferis, and Alexander Statnikov. Time and Sample Efficient Discovery of Markov Blankets and Direct Causal Relations. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, page 673–678, New York, NY, USA, 2003a. Association for Computing Machinery. ISBN 1581137370. https://doi.org/10.1145/956750.956838.

Ioannis Tsamardinos, Constantin F Aliferis, and Alexander R Statnikov. Algorithms for Large Scale Markov Blanket Discovery. In *FLAIRS Conference*, pages 376–380, 2003b.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006a. https://doi.org/10.1007/s10994-006-6889-7.

Ioannis Tsamardinos, Alexander R Statnikov, Laura E Brown, and Constantin F Aliferis. Generating Realistic Large Bayesian Networks by Tiling. In *FLAIRS Conference*, pages 592–597, 2006b.

Thomas Verma and Judea Pearl. *Equivalence and Synthesis of Causal Models*. UCLA Computer Science Department, 1991.

Chirayu Wongchokprasitti. R-causal: R Wrapper for Tetrad Library, 2019. https://github.com/bd2kccd/r-causal.

Akihiro Yabe, Daisuke Hatano, Hanna Sumita, Shinji Ito, Naonori Kakimura, Takuro Fukunaga, and Ken-ichi Kawarabayashi. Causal Bandits with Propagating Inference. In Jen-

nifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5512–5520. PMLR, 10–15 Jul 2018. https://proceedings.mlr.press/v80/yabe18a.html.

Sandeep Yaramakala and Dimitris Margaritis. Speculative Markov blanket discovery for optimal feature selection. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, 2005. https://doi.org/10.1109/ICDM.2005.134.

Behrooz Zarebavani, Foad Jafarinejad, Matin Hashemi, and Saber Salehkaleybar. cuPC: CUDA-Based Parallel PC Algorithm for Causal Structure Learning on GPU. *IEEE Transactions on Parallel and Distributed Systems*, 31(3):530–542, 2020. https://doi.org/10.1109/TPDS.2019.2939126.