Title

Local Causal Structure Learning with the Coordinated Multi-Neighborhood Learning Algorithm

Author

Smith, Stephen

Publication Date

2023

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Local Causal Structure Learning with the Coordinated Multi-Neighborhood Learning Algorithm

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

by

Stephen Vincent Smith

2023

ABSTRACT OF THE DISSERTATION

Local Causal Structure Learning
with the Coordinated Multi-Neighborhood Learning Algorithm

by

Stephen Vincent Smith
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2023
Professor Qing Zhou, Chair

Learning the structure of causal directed acyclic graphs is useful in many areas of machine learning and artificial intelligence, with applications in fields such as robotics, economics, and genomics. However, in the high-dimensional setting, it is challenging to obtain good empirical and theoretical results without strong and often restrictive assumptions. Additionally, it is questionable whether all of the variables purported to be included in the network are appropriate. It is of interest then to restrict consideration to a subset of the variables for relevant and reliable inferences. In fact, researchers in various disciplines can usually select a set of target nodes in the network for causal discovery. This dissertation develops a new constraint-based method for estimating the local structure around user-specified target nodes, employing rules from the Fast Causal Inference algorithm to coordinate structure learning between neighborhoods. Our method facilitates causal discovery without learning the entire DAG structure. We establish consistency results for our algorithm with respect to the local neighborhood structure of the target nodes in the true CPDAG. Empirical experimental results show that our algorithm is more accurate in learning the neighborhood structures with much

less computational cost than standard methods that estimate the entire DAG. An R package implementing our algorithm may be accessed at `http://github.com/stephenvsmith/CML`.

The dissertation of Stephen Vincent Smith is approved.

Chad J Hazlett

Jingyi Li

Yingnian Wu

Qing Zhou, Committee Chair

University of California, Los Angeles

2023

To Dad, Mom, Nathan, and Oma

for your gracious, sacrificial love.

And in loving memory of

Theoren Smith, Kitty Smith, and Vincent Daley.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

All numerical results in 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.

The work in Chapters 3 and 4 is a version of Smith and Zhou (Manuscript in preparation for submission). Dr. Qing Zhou conceived the idea for the paper, helped to guide the research process, and provided editorial direction.

Names and words on a page seem inadequate to convey the extent of my gratitude, and as I write down these brief words of thanks, please know that my heart overflows with thoughts and memories of the words of encouragement, fits of laughter, and cherished times together which have enabled me to reach this point. We know who we are by those who have formed us, and I am thankful for the formative role each of you have played in my life.

I want to begin by thanking my extended family, including all of my cousins who gave their support and love in manifold ways throughout my life. It is also my pleasure to thank all my exceptionally loving and generous uncles and aunts, including Aunt Kathy, Aunt Debbie, Aunt JJ, Uncle Bruce, and Uncle Dave. Uncle Trey is an inspiration to me and to all who knew him, and I treasure the memories we had together. His loss has left a gaping hole for all of us, and I wish as much as anything that I could share this moment with him. To me, he is the first "Dr. Smith" to come to mind, and so he will remain. Uncle Mark, thank you for being a constant support and a boundless supply of joy and humor. The car was an incredible gift to help me through graduate school, and the trips to the lake are always a welcome time of laughter and respite. Aunt Susan, you are one of the pillars in my life that I can always count on for support and love. Your earnest encouragement means the world to me. Aunt Sandra, thank you for all of your help during the early years of my education and for all the love you have shown me. I hope we will be able to celebrate this together. Uncle Robert and Aunt Stefanie, for you two must always be considered together, I cannot thank

you enough. You have been able to identify with my struggles in a uniquely empathetic way and have been solid guides throughout my education. More than that, you have encouraged me in Christ since I was born, and you have loved me every step of the way since then. It is a gracious gift for which I give my love and thanks.

Graduate school quickly teaches you that learning is a collaborative project, and I am glad that I had the opportunity to work with the wonderful faculty and students in the UCLA Department of Statistics. Dr. Jireh Huang, Wesley Cheng, Dr. Conor Kresin, and Dr. Ami Wulf were outstanding colleagues during the first few years of coursework. I learned so much more and with greater effectiveness because of your help. Thank you. I would also like to thank all the members of the Zhou research group, especially Drs. Jireh Huang, Gabriel Ruiz, Dale Kim, and Hangjian Li, who kindly gave their time and help to me on numerous occasions and were a joy to talk things over with as friends. Your research and breadth of learning always pushed me and served as a high bar to strive after. Dr. Gabriel Ruiz deserves special thanks and acknowledgement for helping me obtain the gene expression dataset, as does Dr. Jireh Huang for providing me with code for data generation. I would also like to thank Dr. Yuhao Yin and Jiayi Li for being kind and friendly officemates.

I am exceedingly grateful to the faculty for being exceptional scholars as well as nurturing educators and mentors. It was a joy to have the opportunity to work with many of you; thank you. Professor Mike Tsiang is exemplary in many ways, and I am especially grateful for all those little opportunities we have to catch up in the hallway or whenever we cross paths. Thank you for always checking on how I am doing and for encouraging me whenever you have the chance. It was also a great pleasure to work with Professor Chad Hazlett, who taught me in the classroom and side-by-side in our infant mortality risk modeling research. I learned a great deal more from these experiences than I could have anticipated due to your patient guidance, insightful questions, and masterful skill in academic writing. Thank you for all the work you have done with me and for the considerable investment of time you made to bring that project along. Professor Christou, where do I begin? You are one of the chief

reasons I am in graduate school, and you consistently rekindle my love for learning. One of the highlights of my life has been to work as the TA for Professor Nicolas Christou. Among other things it has given me the singular pleasure of having your mentoring and friendship. Thank you for investing in me. Thank you for believing in me. Thank you for helping me and for being an inspiration to me. I also want to thank the members of my committee, Drs. Chad Hazlett, Jessica Li, Yingnian Wu, and Qing Zhou. You are outstanding teachers and researchers, and I am grateful for your feedback and time.

Living together with other friends is a risky plunge into many unknowns, but I am exceedingly glad to thank all whom I have lived with for making our apartment a place of refuge and joy. Memories of our time together warm my heart and bring a smile to my face. Thank you. I would also like to extend special thanks to Zachary Wong, who had to deal with me the longest and did so with grace. Zach, you are an amazing friend, and I miss all of our talks and the various opportunities you gave me to learn from you. Your thirst for knowledge and your work ethic are inspiring. I look forward to celebrating more of your academic achievements. I would also like to thank Justin Su and Kyle DeGuzman. Though we never actually lived together, it felt like we did because you gave me so much of your time. Your friendship means so much to me, and you helped me more than you will ever know by being there for me all the times I needed it. Thank you.

A great deal of thanks belongs to my church family over the years. Hani Sefain was a consistent encouragement during the early years of graduate school. Ryan Bronkar has been a spiritual and academic mentor since high school, and I continue to look up to him as an example in learning and service. Thank you for always looking out for me. Todd Sorrell graciously reviewed a draft of this manuscript and made helpful suggestions, which is only a small taste of the many ways he serves others. You are another person I look up to, and I do so with gratitude for your encouragement and investment in me over many years. More recently, Victor Dorado has been a faithful mentor, and I am so thankful for your generous kindness and your ceaseless prayer. There are so many who have prayed for

me and have taken an interest in me and my work, and I want to thank all of you. I feel woefully inadequate to thank all of my brothers and sisters from Grace on Campus, for whom I have great affection. By God's grace, I am who I am today because of how God used you in my life. Thank you. Most of all, thank you, Jireh Huang. You spent far more time with me than you ever bargained for, and you have been a constant source of camaraderie and brotherhood. "Thanks" is simply not enough, so I will have to give you another hug when I see you. I appreciate you in uncountable ways, especially for your abounding encouragement and joy in the Lord. Thank you.

As academic advising goes, I can think of no better example than Dr. Qing Zhou, who conceived and shepherded this work for the last few years. Thank you for keeping me on track, giving great ideas, and for patiently encouraging and helping me each step of the way. Much of the writing in this work is sharper due to your edits and clarifications. It has been an honor to work with you. Thank you.

My greatest thanks belongs to my family. It is simply impossible to overstate how much you have shaped and helped me. Your boundless energy and love make life such a delight. Your sacrifices, investment, and all other forms of support are the reason I have made it thus far. Through highs and lows, you are there. I know I can always count on you—with a word of encouragement when I am feeling low or with celebratory joy for a triumph. You have put me ahead of yourselves so often, and I am humbled by your example. Thank you, Dad, Mom, and Nathan. I love you and dedicate this work to you. Just as much may be said for my grandparents, Vince and Berta Daley and Theoren and Kitty Smith. Oma, thank you for embodying service and for your love and prayers. I love you, and also dedicate this work to you. I also dedicate this work to the memory of Grandpa, Grandma, and Opa. They are dearly missed, and I wish I could share this moment with them. But, I am also thankful for the cherished memories I have in my heart. Thank you.

Ultimate thanks belongs to the triune God, maker of heaven and earth, from whom, through whom, and to whom are all things, and glory forever. The study of Statistics is

humbling, for it is essential to the discipline to acknowledge our finitude. We can only grasp at understanding the ways of the universe, but God knows them all, as well as our thoughts and cares even as he cares for us. My graduate education was indeed humbling and brought me to my knees on many occasions. Yet, God is faithful. It is a joy to study and seek to understand some of his providential works. *Fides quaerens intellectum.*

VITA

2017	B.S. Applied Mathematics, University of California, Los Angeles.
2017–2022	Teaching Assistant, Department of Statistics, University of California, Los Angeles.
2021–2022	Teaching Assistant Coordinator, Department of Statistics, University of California, Los Angeles.
2021–present	PhD candidate, Department of Statistics, University of California, Los Angeles.

PUBLICATIONS

C. Hazlett, A. P. Ramos, and S. Smith. Better individual-level risk models can improve the targeting and life-saving potential of early-mortality interventions. *Scientific Reports*, accepted for publication.

S. Smith and Q. Zhou. Coordinated multi-neighborhood learning on a directed acyclic graph. Manuscript in preparation for submission.

CHAPTER 1

Introduction

1.1 Preliminaries

In this chapter, we will consider some of the definitions and algorithms integral to coordinated multi-neighborhood learning. Though the methods we discuss here are global in scope, the principles contained therein are broadly applicable, including for the local setting. We will also explore why these methods are insufficient or improperly posed for problems relevant to researchers where a local approach is preferable. Specifically, this chapter will focus on the PC and FCI algorithms, since they provide constraint-based approaches for skeleton recovery and sound orientation rules under different sets of assumptions, such as the presence or absence of latent nodes from the underlying network, which is relevant to the multi-neighborhood setting where many nodes are intentionally removed from consideration.

1.1.1 Definitions

A graph $G = (V, E)$ consists of a set of nodes, or vertices, $V = \{V_1, \ldots, V_p\}$ and a set of edges $E \subseteq V \times V$ with ordered pairs of distinct nodes. The nodes in the graph correspond to the elements of a vector of random variables $\mathbf{X} = (X_1, X_2, \ldots, X_p)$, and for convenience we also write the set of nodes as $V = [p]$, where $[p] := \{1, \ldots, p\}$. Given a subset of nodes $N \subset V$, the induced subgraph of G over N is defined as $G_N = (N, E_N)$, where $E_N \subseteq E$ only contains edges between nodes in N. For $i, j \in V$, if $(i, j), (j, i) \in E$, then $i - j$ is an undirected edge in G. A directed edge $i \to j$ is in G if $(i, j) \in E$ but $(j, i) \notin E$. If $(i, j) \in E$ or $(j, i) \in E$,

then i and j are adjacent in G.

A path in G is a sequence of nodes $\pi = \langle V_1, V_2, \ldots, V_q \rangle$ such that, for $1 \leq i \leq q-1$, V_i and V_{i+1} are adjacent. If each edge is oriented as $V_i \to V_{i+1}$ on π, then this is called a directed path from V_1 to V_q. If there exists a directed path from V_i to V_j as well as a directed path from V_j to V_i in G, then G contains a cycle.

A Directed Acyclic Graph (DAG) is a graph where all edges are directed and there are no cycles. If there is a directed edge $i \to j$ in graph G, then i is a parent of j and j is a child of i. The parent set of node i is denoted $pa_G(i)$, and the child set is denoted $ch_G(i)$. The adjacency set of i in G is the set of all nodes directly connected to i by an edge in G, and is denoted $adj_G(i)$. A v-structure in a graph is an ordered triple of nodes (i, j, k) such that G contains the directed edges $i \to k$ and $j \to k$, where i and j are not adjacent. A spouse of i in G is a non-adjacent node which shares at least one child with i. The set of spouses of i in G is denoted $sp_G(i)$. If G contains the v-structure (i, j, k), then $i \in sp_G(j)$ and $j \in sp_G(i)$.

Given a DAG G, the causal relations among $\mathbf{X}$ are modeled via a structural equation model (SEM),

$$X_i = f_i(\mathbf{X}_{pa_G(i)}, \varepsilon_i), \quad i = 1, \ldots, p, \tag{1.1}$$

where f_i are deterministic functions, $\mathbf{X}_{pa_G(i)}$ are subvectors of $\mathbf{X}$ containing only the variables corresponding to the nodes belonging to the parent set, and ε_i are independent background variables. This implies that the joint distribution $P(\mathbf{X})$ is Markov with respect to G, meaning the probability distribution admits the factorization

$$P(X_1, \ldots, X_p) = \prod_{i \in V} P(X_i \mid \mathbf{X}_{pa_G(i)}), \tag{1.2}$$

according to G, since each node is conditionally independent of its nondescendants given its parents (Pearl, 2009). We say two variables X and Y are conditionally independent given $\mathbf{Z}$, if and only if $P(X = x, Y = y \mid \mathbf{Z} = \mathbf{z}) = P(X = x \mid \mathbf{Z} = \mathbf{z}) P(Y = y \mid \mathbf{Z} = \mathbf{z})$ for all values $x, y, \mathbf{z}$ such that $P(\mathbf{Z} = \mathbf{z}) > 0$, where $\mid$ represents conditioning. Following Dawid (1979), for non-adjacent nodes i, j and set of nodes S, we denote conditional independence (CI) between

their associated variables by $X_i \perp\!\!\!\perp X_j \mid \mathbf{X}_S$. Here, S is a separating set of nodes i and j in G, which may be identified using the d-separation graphical criterion for reading probabilistic CI relations from the graph.

Definition 1 (d-separation)**.** A path π is said to be d-separated by a set of nodes $S \subset V$ if and only if

1. π contains a sequence $i \to k \to j$ or $i \leftarrow k \to j$ such that $k \in S$, or
2. π contains a collider $i \to k \leftarrow j$ such that $k \notin S$ and there exists no node $m \in S$ such that there is a directed path from k to m (i.e., no descendant of k is in S).

A set $S \subset V$ d-separates $A, B \subset V$, $A \cap B = \emptyset$, if and only if S blocks every path from a node in A to a node in B (Spirtes et al., 2000; Pearl, 2009).

Along with the Markov conditions governing the relationship between the graph and the probability distribution, many methods in the literature require an additional faithfulness assumption to prove that their algorithms are sound and complete, since this ensures a tighter relationship between the probability distribution and the structural features of the graph (Pellet and Elisseeff, 2008).

Definition 2 (Faithfulness)**.** A probability distribution P is said to be faithful to a DAG G if, for any $i, j \in V$ with $i \neq j$ and any set $S \subseteq V \setminus \{i, j\}$,

$$X_i \perp\!\!\!\perp X_j \mid \mathbf{X}_S \iff i \text{ and } j \text{ are } d\text{-separated by } S \text{ in } G.$$

That is, P is faithful with respect to G when there is a one-to-one correspondence between the CI relations of the distribution P and the d-separations in G (Kalisch and Bühlmann, 2007). Some refer to such a graph G as a perfect map of P (Pellet and Elisseeff, 2008).

The skeleton of a DAG, denoted $\text{skel}(G) = (V_{\text{skel}(G)}, E_{\text{skel}(G)})$, removes all directionality from the graph such that if $(i, j) \in E$, then $(i, j), (j, i) \in E_{\text{skel}(G)}$. According to the pairwise

Markov condition, non-adjacencies in a graph reveal CI relations between pairs of variables conditioned on some subset of the remaining nodes. It is for this reason that the skeleton of a DAG is an important object for structure recovery, though in practice this often is only a first step before edge orientation. A partially directed acyclic graph (PDAG) is a graph where some edges are directed and some are undirected, and no cycle may be traced following the directed edges and either direction of the undirected edges.

For any probability distribution P, there is an equivalence class of DAGs such that P is faithful with respect to each member of the class. Two DAGs are equivalent if and only if they have the same skeleton and v-structures (Verma and Pearl, 1990; Kalisch and Bühlmann, 2007). On this basis, we can characterize a set of graphs which are equivalent to the DAG by identifying adjacency sets for each node and the graph's v-structures, along with any other edges compelled by the acyclic constraint or to avoid adding any new v-structures. Furthermore, the equivalence class may be uniquely represented by a completed PDAG (CPDAG). In the literature, PDAGs are also called patterns (Spirtes et al., 2000), and CPDAGs may also be called maximally oriented graphs (Meek, 1995) or essential graphs (Andersson et al., 1997).

Definition 3 (CPDAG)**.** A PDAG is complete and represents an equivalence class of DAGs if

1. every directed edge in the CPDAG exists in every DAG belonging to the equivalence class, and

2. for every undirected edge $i - j$ in the CPDAG, there exists a DAG with $i \to j$ and a DAG with $i \leftarrow j$ in the equivalence class.

The features of the CPDAG encode the features commonly held by all members of a DAG equivalence class.

1.2 DAG structure learning

With respect to causal inference, the gold standard is learning with randomized and controlled experimental data. However, there are numerous types of constraints—financial, ethical, etc.—which render these data infeasible to obtain. Though more difficult, learning with observational data is more common and largely free from the aforementioned constraints. One paradigm of causal inference with observational data makes use of graphical models as fundamental objects of learning, exploiting their facility for effectively representing the probabilistic information of high-dimensional datasets and the graph theoretic tools which may be applied to them for efficient inferential procedures.

With relatively easier access to observational data, the challenge of DAG structure learning for causal inference comes from the size of the DAG space. The number of possible DAGs scales super-exponentially with the dimension of the data, thus making DAG learning an NP-hard problem (Chickering et al., 2004, 2012). Consequently, the initial task of structure learning is to find a way to constrain or efficiently navigate the DAG space toward an optimal solution. There are generally three approaches to accomplish this: score-based, constraint-based, and hybrid learning.

Score-based algorithms use some measurable criterion to efficiently search the space of DAGs for an optimal solution. Unlike constraint-based algorithms, which tend to propagate their errors through the rest of the graph, score-based methods are more resilient to such mistakes because they usually are contained to the local structure where they are made (Bernstein et al., 2019). For a scoring measure to be useful for a search, it is critical that it gives the same score to all networks which are equivalent. That is, since some network structures represent the same set of distributions and are indistinguishable with respect to observational data, a proper score should also refrain from distinguishing these structures (Chickering, 2002). Many score-based methods are implemented in R, such as the methods in the **sparsebn** package (Aragam et al., 2019), which use a regularized maximum likelihood estimate for

applications with high-dimensional datasets. The ℓ_1 penalty ensures a parsimonious solution and computer tractability using coordinate descent algorithms for both continuous and discrete data (Fu and Zhou, 2013; Aragam and Zhou, 2015; Gu et al., 2019).

For discrete data, some early attempts at the score-based approach include those of Bouckaert (1993), who used minimum description length, and Chickering and Heckerman (1997), who used BIC as their measure. However, due to the large size of the search space, these scores are inefficient for high-dimensional graphs, especially because these measures do not allow the use of gradient-based optimization procedures. Ensuing attempts sought to correct this by using an augmented Lagrangian formulation for the score, relaxing the discrete constraint to allow for continuous optimization. One such example may be found in the work of Ng et al. (2019), where the algorithm uses a penalized MSE as the score. These algorithms, however, are very computationally expensive and require significant storage overhead.

A critical advance came with the introduction of operators which can be scored locally. For these scores, any change in a DAG's score from a single alteration to the graphical structure can be computed by the change in score for a subset of nodes local to where the change took place, thus allowing for a faster greedy search (Chickering, 2002). These algorithms traverse the DAG space with single edge alterations, using well-defined rules for retaining or modifying the graph structure according to the changes in score.

The challenge with score-based algorithms is that the structure can, in a single iteration, change such that established results from a previous state of the DAG can be reversed if a DAG with a higher score is found later, which raises concerns about the robustness of these methods. Moreover, in the case of missing variables, it is unclear how a search should be structured or what kind of score could be used to determine the relative fitness of one graphical structure compared to another.

On the other hand, constraint-based algorithms such as the PC algorithm (named after its authors, Peter Spirtes and Clark Glymour) attempt to efficiently organize CI inferences from data to search for structures consistent with that information (Spirtes et al., 2000). The

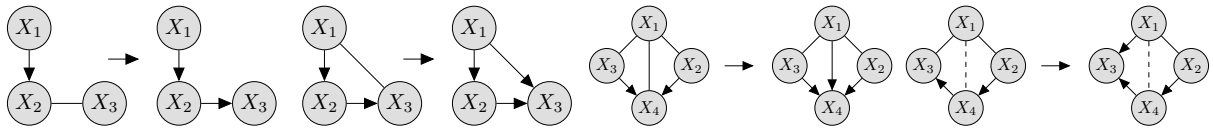
CI results serve as constraints on the space of DAGs considered for the estimated equivalence class, and the algorithm uses these constraints along with graphical techniques to recover as much of the CPDAG as possible. Other constraint-based algorithms include the IC (Verma and Pearl, 1990), the FCI (Spirtes et al., 2000), and the graphical lasso (Friedman et al., 2007).

Hybrid methods include the Max-Min Hill-Climbing (MMHC) algorithm, which applies a constraint-based procedure for local learning followed by a score-based hill-climbing procedure for combining the local neighborhoods to recover the global structure (Tsamardinos et al., 2006). Another method, Greedy Fast Causal Inference (GFCI), combines score-based solutions with principles from the FCI algorithm to retain asymptotic guarantees of correctness even while relaxing some of the restrictive assumptions usually made by score-based methods (Ogarrio et al., 2016).

Due to their relevance to the Coordinated Multi-Neighborhood Learning algorithm, we will discuss the PC and FCI algorithms in greater detail.

1.3 PC algorithm

Even with an infinite amount of observational data, we are unable to identify the entire causal structure of a DAG through an inferential procedure since the CI information of the distribution encoded in the correlation pattern of the observational data is compatible with multiple DAGs. That is, the causal structure is underdetermined. The equivalence class represents the upper limit of what can be recovered from a structure learning algorithm using sample data, and thus the CPDAG serves as the objective of structure learning for problems under standard assumptions. The PC algorithm is a constraint-based algorithm which employs an efficient method to iteratively remove edges from a complete, undirected graph using separating sets of increasing cardinality to obtain the graph skeleton, then identifying v-structures and using Meek's rules, illustrated in Figure 1.1, to complete the orientation



(a) Rule 1 (b) Rule 2 (c) Rule 3 (d) Rule 4

Figure 1.1: Illustration of Meek's rules. The dashed line in Rule 4 can be any kind of edge.

of the PDAG (Meek, 1995). In the population version, the algorithm returns the CPDAG, and as such is a sound and complete method for DAG structure learning. The algorithm uses inferential reasoning only in the recovery of the skeleton before using deterministic rules for orienting the remaining edges. The skeleton recovery algorithm pseudocode is given by Algorithm 1. For the population version of the algorithm, we use a CI oracle in Line 8, but for the sample version we replace the CI oracle with an appropriately chosen CI test, such as testing on Fisher's z-transformation of the partial correlation for Gaussian data. The separating sets for conditionally independent variables are stored in Line 9 in order to identify v-structures during edge orientation. The edge orientation procedure is described in Algorithm 2, beginning with v-structure identification and followed by recursive application of Meek's rules to compel the remaining edges which characterize the equivalence class.

Due to its efficiency in ordering CI tests and its positive theoretical results, the PC algorithm is often used as a benchmark, especially for constraint-based algorithms. However, the PC algorithm runs in exponential time complexity in the worst case, and in practical settings it usually cannot handle more than a hundred variables well (Spirtes et al., 2000). Still, despite these disadvantages, it is a popular algorithm and widely used in the high-dimensional sparse setting due to its positive theoretical guarantees such as uniform consistency where the network size is able to grow quickly relative to the number of sample observations (Kalisch and Bühlmann, 2007).

Algorithm 1 Population PC skeleton recovery

Input: vertex set V, CI oracle
Form a complete undirected graph $C = (V, E_C)$
$\ell = 0$
while $\exists$ at least one pair of adjacent nodes (i, j) with $|adj_C(i) \setminus \{j\}| \geq \ell$ **do**
    for each adjacent pair of nodes (i, j) s.t. $|adj_C(i) \setminus \{j\}| \geq \ell$ **do**
        while $(i, j), (j, i) \in E_C$ and there remains a potential separating set of size ℓ **do**
            Choose new set $\mathbf{k} \subseteq adj_C(i) \setminus \{j\}$ with $|\mathbf{k}| = \ell$
            if $X_i \perp\!\!\!\perp X_j \mid \mathbf{X_k}$ **then**
                Delete edge $i - j$ from C; Save $\mathbf{k}$ in S_{ij}, S_{ji}
            end if
        end while
    end for
    $\ell = \ell + 1$
end while
Output: estimated skeleton C and separation sets S

Algorithm 2 PC algorithm edge orientation

Input: undirected graph C, separation sets S
for pairs of nonadjacent nodes (i, j) with common neighbor k **do**
    if $k \notin S_{ij}$ **then**
        Replace $i - k - j$ in C with $i \to k \leftarrow j$
    end if
end for
Use Meek's Rules (see Figure 1.1) for further edge orientation
Output: CPDAG C

1.3.1 Theoretical results

The PC algorithm is asymptotically consistent for the skeleton of sparse DAGs even in the case where $p \gg n$, where n is the number of observations in the sample dataset. This is demonstrated in (Kalisch and Bühlmann, 2007) under the following assumptions.

Assumption 4 (PC consistency assumptions)**.**

(A0) The data $\mathbf{X}_1, \ldots, \mathbf{X}_n \in \mathbb{R}^{p_n}$ are i.i.d. random vectors from distribution P_n, where dimension p_n is allowed to grow with the sample size.

(A1) P_n is multivariate Gaussian and faithful to G_n for all n.

(A2) The dimension $p_n = O(n^a)$ for some $0 \leq a < \infty$.

(A3) The maximal number of neighbors in G_n is denoted $q_n = \max_{1 \leq i \leq p_n} |adj_G(i)|$, with $q_n = O(n^{1-b})$ for some $0 < b \leq 1$.

(A4) For the partial correlations between $\mathbf{X}_i$ and $\mathbf{X}_j$ given $\mathbf{X}_\mathbf{k}$ for some set $\mathbf{k} \subseteq [p_n] \setminus \{i, j\}$, denoted $\rho_{i,j|\mathbf{k}}$, the absolute values have upper and lower bounds

$$\inf_{i,j,\mathbf{k}} \{|\rho_{i,j|\mathbf{k}}| : \rho_{i,j|\mathbf{k}} \neq 0\} \geq c_n,$$

$$\sup_{i,j,\mathbf{k}} |\rho_{i,j|\mathbf{k}}| \leq M < 1,$$

where $c_n^{-1} = O(n^d)$ for some $0 < d < b/2$ and $0 < b \leq 1$ as in (A3).

Assumption (A1) is a standard assumption constraining the class of probability distributions we consider, while (A2) allows for high-dimensionality at a polynomial growth rate with respect to the sample size. Assumption (A3) is a sparseness assumption and (A4) is a regularity condition.

Recall that all inference is completed during the recovery of G_{skel}. If there are no errors from the CI tests, then edge orientation will never fail (Meek, 1995). This is stronger than saying the skeleton must be correct, because the proper orientation of the v-structures depends

on correctly identifying separating sets for non-adjacent nodes. Using the assumptions given above, Kalisch and Bühlmann (2007) prove skeleton estimation consistency. Since Meek's rules are sound and complete, we obtain asymptotic consistency for the CPDAG as well. The sparsity assumption is crucial for statistical consistency and for computational feasability, where the latter may be demonstrated by considering the number of potential separating sets which could be considered in Line 6 of Algorithm 1.

1.3.2 Variations

There are numerous alterations to the PC algorithm attempting to address its shortcomings or to make it suitable for different problems. Colombo and Maathuis (2014) pay special attention to removing the order-dependence of the PC algorithm, or the consequence of the structuring of the CI tests which makes the output of the algorithm depend on the order in which the variables are considered. In the high-dimensional setting, this can lead to highly variable results and additionally uncertain conclusions when using sample data. The PC algorithm skeleton estimation procedure and the concomitant identification of separating sets depend on the order in which node pairs are considered for CI tests. Because the candidate nodes for a potential separating set are selected from the superset of the output's skeleton at each step of the algorithm (see Line 7 of Algorithm 1), which is dynamically altered by CI results, earlier CI tests will affect subsequent tests and the potential separating sets used for those tests. As a result, the authors propose the PC-stable algorithm, which retains adjacency relations longer than in the original version after edges have been removed, thus alleviating the issue of order-dependence. For potential v-structures, there are supplementary rules taken from the conservative PC algorithm in the work of Ramsey et al. (2006) designed to handle potential orientations while maintaining order-independence. The authors show that this algorithm is sound and complete, and the consistency results from the work of Kalisch and Bühlmann (2007) continue to apply.

In the work of Sondhi and Shojaie (2019), the reduced PC (rPC) algorithm is designed

for biological systems with low average degree for most nodes, but with very high degree for certain hub nodes. Additionally, to reduce the number of CI tests, Cai et al. (2022) proposed the Pre-Processing Plus PC (P3PC) algorithm, which as its name suggests adds an initial pre-processing step using large conditioning sets selected at random to remove some of the edges from the original complete graph. Ha et al. (2016) introduced the PenPC algorithm for estimating the skeleton in a two-step procedure, first by using an adapted Markov blanket estimation method for each node followed by a modified PC-stable algorithm to remove false positive edges. Also, Spirtes et al. (2000) proposed a modified PC algorithm for the case where latent variables are present, which served as a prototype for the FCI algorithm. Related to Markov blanket estimation, there is a variation of the IAMB algorithm including a PC step after the initial growth phase to remove false positives with greater efficiency (Tsamardinos et al., 2003b). Recently, Huang and Zhou (2022) introduced the partitioned PC (pPC) algorithm, which improves the PC algorithm by using a p-value threshold to mitigate issues related to multiple testing and by providing additional capacity for parallel processing.

Especially relevant to our discussion is a slight variation of the PC algorithm which places a stopping condition on the size of potential separating sets. While Algorithm 1 will continue until the size of ℓ exceeds the maximum size of any adjacency set in the estimated graph, this "anytime" version of the PC algorithm will stop once ℓ reaches a specified value, usually denoted m or ℓ_{max}. Though this version is useful primarily for theoretical analysis, it also appears in practical implementation as well.

1.4 Ancestral graphs

In many cases, we cannot expect to be able to measure every variable in the true DAG, nor can we expect our dataset to be free from selection bias. In Spirtes et al. (2000), the authors show that even correct causal inference algorithms fail when the observational data samples are selected due to some of the variables under study. Cooper (1995) provided a similar result

which shows that analogous errors occur when the DAG under consideration includes latent or unmeasured variables. In this setting, the correlation structure among observed nodes can be misleading due to latent confounders, leading to an observed association that is incorrect (Spirtes et al., 2000). Consequently, structure learning methods confronting the assumption of latent and selection variables must account for these challenges by remaining judicious in their inferential reasoning and by using more flexible graphical structures to encode causal relations.

A mixed graph $\mathcal{G}$ can possess three kinds of edges: directed ($\rightarrow$), bi-directed ($\leftrightarrow$), and undirected ($-$), and at most one edge between each pair of nodes. The two kinds of ends of an edge, arrowheads ($>$) and tails (both ends of an undirected edge), are called marks or orientations. An edge is into (out of) a node if the edge mark at the node is an arrowhead (a tail).

We call i an ancestor of j and j a descendant of i if $i = j$ or there is a directed path from i to j. The set of ancestors of j in mixed graph $\mathcal{G}$ is denoted $an_{\mathcal{G}}(j)$. If for any pair of nodes i and j in $\mathcal{G}$, $i \rightarrow j$ and $j \in an_{\mathcal{G}}(i)$, then we can say there is a directed cycle in $\mathcal{G}$. Similarly, if $i \leftrightarrow j$ and $j \in an_{\mathcal{G}}(i)$, then we can say there is an almost directed cycle in $\mathcal{G}$. A node i is a collider on path π if two edges incident to i on π are both into i.

Ancestral graphs are a class of mixed graphs which can be used to encode certain causal and CI features of a distribution.

Definition 5 (Ancestral graph)**.** A mixed graph is ancestral if

1. there is no directed cycle;
2. there is no almost directed cycle;
3. for any undirected edge $i - j$, i and j have no parents or spouses.

Just as with DAGs, ancestral graphs can carry CI information from P which may be read using a graphical criterion. With respect to causal representation, however, the interpretations

of the markings for a causal ancestral graph differ from those of causal DAG markings. Edge $i \to j$ in $\mathcal{G}$ implies that i is a cause of j or a selection variable, but in this setting we refer to a cause i of node j as corresponding to a directed path from i to j in the underlying causal structure, which may include latent variables not present in the ancestral graph. The same edge $i \to j$ also implies $j \notin an_{\mathcal{G}}(i)$, which simply means j is not a cause of i or of a selection variable. Nodes i and j are siblings if $i \leftrightarrow j$, which implies that neither variable is the cause of the other or of a selection variable. The previous assertions considered together demonstrate that arrowheads provide negative qualitative causal information, since they imply the node into which the arrowhead directs does not cause the adjacent node. Assuming the maximality of the ancestral graph, a bidirected edge also implies that the adjacent nodes share a latent common cause, also known as a latent confounder. Undirected edges in the causal ancestral graph point to the effect of selection bias on the adjacent nodes. More precisely, an undirected edge implies that both of the nodes are either causes of the other or of some selection variable. Due to the acyclicity constraint of the underlying DAG, this is equivalent to saying that each node is the cause of some selection variable (Zhang, 2008a). Taken together, these statements allow us to conclude that tails provide positive qualitative causal information since they establish a causal relationship from the node which the edge is out of to the adjacent node. In the case where selection bias is absent, the causal conclusions are more straightforward since we may remove the further interpretive condition related to selection variables for the aforementioned types of edges in an ancestral graph. Indeed, it is this restraint and nuance in interpretation which makes ancestral graphs particularly useful, since we cannot be too ambitious in causal and probabilistic reasoning when a graph contains latent and selection variables.

We may further clarify the relationship between CI information in P and the structure of the ancestral graph by defining m-separation, the graphical criterion governing CI interpretations of ancestral graphs.

Definition 6 (m-separation)**.** In a mixed graph, a path π between nodes i and j is m-

connecting given a set of nodes S if

1. every non-collider on π is a not a member of S, and

2. every collider on π has a descendant in S.

If there is no m-connecting path between i and j given S, then i and j are m-separated by S.

Similar to d-separation for DAGs, m-separation in a mixed graph implies CI among observed variables via the global Markov property (Richardson and Spirtes, 2002). This allows us to conceptually integrate CI information from the distribution and the structural features of the ancestral graph. Under faithfulness, we can use CI information to constrain the space of ancestral graphs for structure learning and set edge orientations according to sound rules to obtain an equivalence structure, which is what we recover from the FCI algorithm. However, it is important to note that ancestral graphs, unlike DAGs, may have pairs of non-adjacent nodes which are not m-separated by any of the observed nodes, thus breaking the pairwise Markov property.

Definition 7 (Inducing paths)**.** A path is said to be inducing relative to set S if every node not in S (excluding the endpoints) is a collider on the path and every collider is an ancestor of a path endpoint. If S is empty, we simply call it an inducing path.

The lack of causal sufficiency and the presence of inducing paths for non-adjacent nodes in ancestral graphs calls for the notion of maximality to further specify the graphical object of interest for structure learning.

Definition 8 (Maximal ancestral graph)**.** An ancestral graph is said to be maximal if there is no inducing path between any two non-adjacent nodes. Accordingly, every pair of non-adjacent nodes in a maximal ancestral graph (MAG) is m-separated by some subset of nodes.

It may be shown that, using a finite number of steps, any ancestral graph may be converted to a structure where the pairwise Markov property holds, and such a structure is a MAG. Similar to DAGs, multiple MAGs may encode the same set of m-separations and form an equivalence class, which is represented by a partial ancestral graph (PAG). A PAG has three possible kinds of marks, adding the circle ($\circ$) along with the tail and the arrowhead. Each circle mark corresponds to a variant mark and each non-circle mark is invariant in the equivalence class of an MAG (Zhang, 2008b).

The PAG can contain valuable information about the causal relationships and the CI relations of the marginal distribution over the observed variables (Spirtes et al., 2000). In order for a PAG $\mathcal{P}$ to represent the equivalence class of the MAG $\mathcal{G}$, $\mathcal{P}$ must have the same set of adjacencies as each member of the equivalence class, and every non-circle mark in $\mathcal{P}$ must correspond to an invariant mark common to every MAG in the equivalence class. Such a PAG is called the maximally informative PAG for the equivalence class of $\mathcal{G}$. This structure corresponds to the CPDAG with respect to DAG structure recovery, and serves as the object of interest for structure learning under the assumption of latent and selection variables in the underlying causal graph. In the literature, some use a similar but less informative object of interest referred to as the partially oriented inducing path graph (POIPG) (Spirtes et al., 2000; Zhang, 2008b).

1.5 FCI algorithm

The Fast Causal Inference (FCI) algorithm is a constraint-based algorithm designed for causal discovery in the presence of latent and selection variables (Spirtes et al., 2000). The algorithm was designed according to the sufficient conditions for sound causal paths in the presence of latent and selection variables (Spirtes et al., 1995). By convention, we partition the nodes of the true DAG as $V = O \cup L \cup S$, where O is the set of observed variables, L is the set of latent variables, and S is the set of selection variables. It is similar to the PC algorithm in

that it consists of skeleton learning using CI tests followed by edge orientation using a set of deterministic rules. The FCI algorithm modifies the PC algorithm to account for removing the causal sufficiency assumption. In fact, it was originally conceived as an improvement to the modified PC algorithm, since the structure of its tests are better suited to the richer syntax of ancestral graphs (Spirtes et al., 2000).

The FCI algorithm is efficient in how it organizes CI information to avoid repetitive testing, beginning with the same hierarchical testing strategy as found in Algorithm 1. Similar to the PC, the FCI algorithm begins with a complete graph and iteratively removes edges based on CI tests conditioned on potential separating sets of increasing cardinality. However, unlike the PC, the graph produced by this step is only a preliminary skeleton, a superset of the skeleton for the PAG we are recovering (Chen et al., 2023). This is due to the potential for latent variables which, unobserved, do not separate pairs of nodes which should be non-adjacent given the structure of the underlying causal graph. This is a critical difference between the two algorithms, because the PC is able to reduce the number of required CI tests since, for any pair of non-adjacent nodes i and j in G, they will be separated by either $pa_G(i)$ or $pa_G(j)$. In the case of the FCI algorithm, however, we cannot guarantee $pa_G(i) \subseteq O \setminus \{i\}$ or $pa_G(j) \subseteq O \setminus \{j\}$.

To supplement the algorithm and ensure that it is sound in skeleton recovery, the authors provided an additional step after v-structure orientation. For any pair of adjacent nodes i and j, the d-separation set d-Sep(i, j) contains any node k belonging to a path between i and j on which every node except the endpoints is a collider and is an ancestor of either i or j. For each pair of nodes which are adjacent at this stage, the algorithm will search for the d-separation set. That is, the algorithm searches for nodes belonging to inducing paths between pairs of adjacent nodes, because a subset of this set will be sufficient to remove any remaining edges which are incorrect. However, since we have incomplete edge orientation at this stage of the algorithm, we relax the condition for inducing paths and consider only a necessary condition for membership in the d-separation sets, using this criterion to form

Algorithm 3 Population FCI

1: **Input:** observed vertex set O, CI oracle
2: Form a complete undirected graph $C = (O, E_C)$
3: $\ell = 0$
4: **while** $\exists$ at least one pair of adjacent nodes (i, j) with $|adj_C(i) \setminus \{j\}| \geq \ell$ **do**
5: **for** each adjacent pair of nodes (i, j) s.t. $|adj_C(i) \setminus \{j\}| \geq \ell$ **do**
6: **while** $(i, j), (j, i) \in E_C$ and there remain potential separating sets of size ℓ **do**
7: Choose new set $\mathbf{k} \subseteq adj_C(i) \setminus \{j\}$ with $|\mathbf{k}| = \ell$
8: **if** $X_i \perp\!\!\!\perp X_j \mid \mathbf{X_k}$ **then**
9: Delete edge between i and j from C; Save $\mathbf{k}$ in S_{ij}, S_{ji}
10: **end if**
11: **end while**
12: **end for**
13: $\ell = \ell + 1$
14: **end while**
15: Set each undirected edge $i - j$ as $i \circ\!\!-\!\!\circ j$ and denote the new graph by $\mathcal{P}$
16: Find v-structures using the same procedure as lines 2 to 6 of Algorithm 2
17: **for** every $i \circ\!\!-\!\!\circ j$ in $\mathcal{P}$ **do**
18: If there exists $\mathbf{k} \subseteq$ p-d-Sep$_{\mathcal{P}}(i, j)$ or $\mathbf{k} \subseteq$ p-d-Sep$_{\mathcal{P}}(j, i)$ such that $X_i \perp\!\!\!\perp X_j \mid \mathbf{X_k}$, then remove $i \circ\!\!-\!\!\circ j$ from $\mathcal{P}$ and store $\mathbf{k}$ in S_{ij}, S_{ji}
19: **end for**
20: Repeatedly apply $\mathcal{R}_1$-$\mathcal{R}_{10}$ until none apply
21: **Output:** estimated PAG $\mathcal{P}$ and separation sets S

supersets of the d-separation sets called the possible d-separation (p-d-Sep) sets over which to search for remaining separating sets (Spirtes et al., 2000).

After the skeleton recovery procedure and v-structure orientation ($\mathcal{R}_0$), the first three rules of the FCI's edge orientation procedure ($\mathcal{R}_1$-$\mathcal{R}_3$) are essentially the same as Meek's first

three rules. The rule $\mathcal{R}_4$ is unique to MAGs with bi-directed edges. This set of rules ($\mathcal{R}_0$-$\mathcal{R}_4$) was originally published with the algorithm and renders the method sound, but not complete. That is, the rules could not, in principle, identify all non-variant marks in the equivalence class of the MAG, which represents all causal information that is not underdetermined for the observed variables. The rules are, however, arrowhead complete. Zhang (2008a) augmented the rule set with $\mathcal{R}_5$-$\mathcal{R}_{10}$, which are necessary to pick up all of the invariant tails such that the rule set is complete. Consequently, the output of the algorithm with a CI oracle (i.e., the population version) is the PAG representing the equivalence class of the true MAG. It is worthy of note that $\mathcal{R}_5$-$\mathcal{R}_7$ are only relevant where selection variables are present, and may be safely ignored in the case where selection variables are absent, which is the case for the multi-neighborhood work presented in Chapter 3.

1.5.1 Theoretical results

In high-dimensional settings, the FCI algorithm is consistent but computationally expensive (Colombo et al., 2012). In the worst case, the algorithm runs in exponential time in the number of variables, even when the maximum number of adjacent nodes is held fixed (Spirtes et al., 2000). Of particular note is the second stage of skeleton recovery, since the p-d-Sep sets can be very large, requiring extensive searches for potential separating sets (Chen et al., 2023). Moreover, the output is generally less informative when selection bias is present due to the addition of undirected edges in the output (Spirtes et al., 1995). Assuming the reliability of CI testing, the causal Markov condition, and causal faithfulness, the FCI algorithm is sound with respect to the MAG equivalence class, constructing a PAG representing all invariant common features (Spirtes et al., 1995; Zhang, 2008a). Faithfulness, however, is a restrictive assumption, as shown in the work of Uhler et al. (2013).

Moreover, the work of Zhang (2008a) provides completeness results. Assuming that we correctly recover the skeleton with accurate CI tests, then the orientation rules are provably correct and obtain all of the invariant tails and arrowheads in the MAG equivalence

class. Using probabilistic and causal facts extracted from the data, the FCI algorithm, using its augmented rule set, will recover the maximally oriented PAG under the causal Markov and faithfulness assumptions. In summary, the orientation rules provide a complete characterization of the invariant marks in the MAG equivalence class.

1.5.2 Variations

Some recent efforts have been made to improve the computational efficiency of the FCI algorithm so that it is more practically useful (Colombo et al., 2012; Chen et al., 2023).

The "anytime" version of the FCI, like that of the PC, learns a skeleton only guaranteed to be a superset of the ground truth skeleton even with a CI oracle, thereby making it less informative than the FCI output as the price for the reduction in complexity (Spirtes, 2001). This is due to adding a stopping condition on CI tests for potential separating sets over a certain size. Despite its limitations, it is proven to be correct in the large sample limit (Spirtes, 2001).

In the work of Colombo et al. (2012), the authors present the Really Fast Causal Inference (RFCI) algorithm, which proposes multiple strategies for reducing the size of p-d-Sep sets in the second phase of skeleton recovery, such as by intersecting the sets with bi-connected components or by using conservative ordering rules. The RFCI output is also slightly less informative than that of the FCI, but it is consistent and requires a weaker sparsity assumption due to its lower complexity. The authors also modify the "anytime" algorithm with the Adaptive Anytime FCI (AAFCI) algorithm, which adaptively changes the maximum potential separating set taken as a subset of the p-d-Sep sets during the second phase of skeleton recovery.

The local FCI (lFCI) algorithm is a recent attempt to deal with the presence of highly connected hub nodes which, due to breaking the sparsity constraint, are very challenging to deal with under the standard FCI procedure due to the computational requirements (Chen

et al., 2023). These types of graphs appear regularly in practical networks such as in biological models. To deal with this, the lFCI algorithm only considers CI tests with separating sets composed of variables that are within a short distance to the node pair. In doing so, the algorithm only requires a local-separation property, a reasonable assumption for a search strategy based on the subgraph local to the pairs of nodes.

1.6 Conclusion

In this chapter we have introduced essential terminology and reviewed some of the literature concerning constraint-based DAG structure learning, considering two popular algorithms which are sound and complete for their respective tasks. The PC algorithm is important because it serves as a standard against which most constraint-based algorithms are compared. Moreover, its skeleton recovery procedure will serve as the basis for the first phase of skeleton recovery for the method we develop in Chapter 3. With respect to the local learning problem, the FCI algorithm is also of particular importance because it approaches structure learning without the assumption of causal sufficiency. Due to our interest in local learning, which is of a more narrow focus, certain variables in the true DAG will be removed from consideration. Therefore, as we coordinate learning between multiple local neighborhoods, it is incumbent upon us to organize the CI information in a manner similar to the FCI algorithm, including using a subset of the FCI orientation rules.

Before we proceed to a proper introduction of the Coordinated Multi-Neighborhood Learning algorithm in Chapter 3, we will first survey existing local learning algorithms in Chapter 2 since these methods may be used to provide estimates of the target neighborhood sets as the initial step of the algorithm. Following our presentation of the algorithm, we will demonstrate its empirical benefits and unique contributions using synthetic and real-world data in Chapter 4. Then, since this work is intended to aid researchers answer causal questions in their respective fields, we provide implementation details and sample code for our R package

in Chapter 5. Finally, we conclude with some final observations and future research directions in Chapter 6.

CHAPTER 2

Local Learning Methods

2.1 Introduction

Researchers in a variety of fields working with high-dimensional datasets seek to reduce the size of large feature sets while retaining predictive power or preserving causal relationships (Fu and Desmarais, 2010). Datasets in areas where such needs are common include gene expression array studies, text analysis, image classification, business analytics, and many others (Aliferis et al., 2010a). To address this problem, researchers look to leverage various local learning methods for graphical modeling. In particular, a substantial amount of work has been dedicated to estimating Markov blankets or the parent-child set, especially as such methods relate to the feature selection problem (Aliferis et al., 2010a; Khan et al., 2023). Some of these algorithms were developed to improve predictive models, since an accurate understanding of the causal mechanisms of the data generating process should improve the accuracy of predictions, particularly when the data undergoes an intervention of some kind (Yu et al., 2020). Conceptually, variable or feature selection for predictive modeling aims to select a subset of available variables for a classification or regression task, thereby minimizing the problems of overfitting and computational overhead without sacrificing predictive power (Aliferis et al., 2010a; Acid et al., 2013). In fact, it has been shown that the Markov blanket is the theoretically optimal feature set for prediction in the case of a faithful distribution (Koller and Sahami, 1996; Kohavi and John, 1997; Pellet and Elisseeff, 2008).

Markov blanket learning is useful beyond merely feature selection and prediction. Indeed,

Markov blanket learning is an important task for causal discovery, whether exclusively for local learning or as a step toward learning the entire DAG structure (Margaritis and Thrun, 1999). However, many recovery algorithms proceed no further than estimating the Markov blanket set without distinguishing between parents, children, and spouses, though such distinctions are often beneficial, such as in causal effect estimation. Though this is a frequent shortcoming, there are some local methods which attempt to distinguish between the members of the recovered set (Yu et al., 2020; Pellet and Elisseeff, 2008; Fang et al., 2022). In addition, it should be noted that the use of non-causal feature selection algorithms for causal discovery is not a sound approach, since these methods have a tendency to use highly predictive nodes from all over the network (Aliferis et al., 2010a). This remark is reflected in the sections that follow, where we will constrain our consideration to those algorithms that are appropriate for causal discovery in the local setting.

2.2 Definitions

In the literature, the definition of the Markov blanket is sometimes broader than the one which will be used in this work, and what we refer to as the Markov blanket is there labeled the Markov boundary. In the broader conception, the Markov blanket for node i is any set which renders X_i conditionally independent of the remaining variables represented in the DAG. Following Margaritis and Thrun (1999) and others, we define the Markov blanket more narrowly as follows.

Definition 9 (Markov blanket)**.** The Markov blanket (Mb) of node i, $mb_G(i) \subseteq V \setminus \{i\}$, is the minimal set for which X_i is rendered conditionally independent of the variables corresponding to the graph's remaining nodes. That is, none of the proper subsets of the Mb render X_i conditionally independent of the remaining variables represented in the DAG.

Under the faithfulness assumption, we can identify the Mb as the union of a node's parents, children, and spouses, written as $mb_G(i) = pa_G(i) \cup ch_G(i) \cup sp_G(i)$ (Tsamardinos

and Aliferis, 2003; Tsamardinos et al., 2003a). One may think of the Mb as containing all information sufficient for the conditional distribution of node t, such that we can write $P(X_t \mid \mathbf{X}_{V\setminus\{t\}}) = P(X_t \mid \mathbf{X}_{mb_G(t)})$ (Aliferis et al., 2010a). In the context of Mb learning and local learning more generally, we define the target node as the node whose Mb we are recovering. For the remainder of this work, we use Mb learning, Mb estimation, and Mb recovery synonymously to refer to a method which aims to identify the Mb of a particular target node or the Mbs of a set of target nodes. In addition, some of the algorithms we discuss recover a subset of the Mb, namely the parent-child (P/C) set composed of the union of the target's parents and children.

2.3 Markov blanket learning approaches

As mentioned previously, Mb estimation is well-developed in the literature, especially as it relates to the problem of feature selection since the procedure removes irrelevant variables and improves the generalizability of predictive models (Fu and Desmarais, 2010; Khan et al., 2023). For the feature selection problem, recovering the Mb is intuitively useful because it ties together prediction and causality, especially as this improves interpretability and the robustness of the predictive model (Ling et al., 2022a). To illustrate this, one could consider a regression task where feature selection may be applied to reduce an excessively large feature set and improve the interpretability of the model. Even without *a priori* knowledge of the structural equation model, selecting only the most causally relevant features should provide predictive benefits, especially since conditioning on these variables theoretically makes the target independent of the rest of the feature set. This roughly corresponds to the concepts "relevance" and "optimal feature subset" proposed by Kohavi and John (1997) for feature selection. Additionally, many researchers may wish to pursue prediction in an experimental context such that parts of the underlying network undergo interventions. Such decisions make causal discovery of greater importance than merely identifying features for prediction since

the causal properties of the Mb will assist with prediction tasks involving experimental data.

When considering Mb estimation algorithms from the perspective of feature selection, algorithms may usually be categorized as wrappers or filters (Aliferis et al., 2010a). Wrapper algorithms use a heuristic search across valid variable subsets, comparing candidate sets on the basis of performance with respect to the classification or regression task. Filter algorithms are task agnostic, instead applying statistical criteria to identify the most relevant features to the target variable being modeled. Though there are some variations on these general categories, these are the primary sets of algorithms for consideration.

In our case, since we are not interested in using the output of the algorithm for prediction, we prefer to use filtering techniques for Mb recovery. Most of these methods are characterized by some heuristic used to identify or remove features sequentially (Zhang et al., 2010). As noted in the work of Aliferis and Tsamardinos (2003), Mb learning algorithms should have well-defined properties with minimal assumption requirements to guarantee soundness, good performance in practical application, and scalability with respect to running time. However, as with other algorithms, choosing between methods entails choosing between various trade-offs. Some algorithms are efficient in structuring the CI tests but are data inefficient because the CI tests potentially require large conditioning sets. On the other hand, while some methods can avoid this problem, doing so requires additional tests to ensure soundness, thus leading to a higher false discovery rate in practice due to the multiple testing problem (Borboudakis and Tsamardinos, 2019). A large number of statistical tests for selection using the same data, often repeating CI queries which only vary by conditioning sets, entails that the test statistics do not actually follow the claimed distribution, and thus the corresponding p-values are too small (Hastie et al., 2009). Intuitively, multiple testing often leads to spurious relations between variables simply because these pairs are given arbitrarily many chances to do so. Though certain choices may mitigate the extent of the problem, multiple testing is a concern for any local algorithm, particularly for high-dimensional or dense graphs (Aliferis et al., 2010a).

Mb learning consists of some score-based algorithms and the more common constraint-based algorithms. Most of the state-of-the-art algorithms are constraint-based, but there are some score-based algorithms which are worth considering. Similar to the measures used for DAG learning, scoring functions should be decomposable, computing the global score by aggregating local scores. Examples include K2, BDe, BDeu, BIC, AIC, and MIT. The BDeu score, for instance, is a metric which uses a Bayesian paradigm, assigning a uniform prior over the parameters for each configuration of a node and its parents in potential networks such that a MAP network may be selected according to the input data (Scutari, 2018). The Mutual Information Test (MIT) measures the mutual information between variables and their parents in the network using a statistic which includes a penalizing term to take model complexity and the input data into account (de Campos, 2006). This could be conceived of as a penalized Kullback-Leibler divergence between the joint probability distribution of the candidate network and the available dataset. Unlike BDeu, MIT is a score based on information theory, which highlights the different considerations involved in choosing a metric.

Acid et al. (2013) developed DMB and RPDMB as score-based algorithms for local structure learning. The former algorithm searches across the space of class-focused DAGs (C-DAGs), a structure which only permits edges linked to the target node or its spouses in candidate networks. Because this space still grows exponentially even with these constraints, the DMB algorithm conducts a heuristic search across the C-DAG space using an appropriate score. The operators used in the search to add or remove edges ensure that each iteration produces another C-DAG. The second algorithm works similarly, but for Restricted PDAGs (RPDAGs) and C-RPDAGs, which correspond to DAGs and C-DAGs in the previous discussion but in the context of a class of PDAGs used in other score-based learning algorithms. The RPDMB algorithm provides rules for a heuristic local search, iterating from one structure to another based on score improvements until the final Mb is obtained. Both algorithms begin with an empty graph and add edges until the scoring function no longer improves. However, these algorithms produce a high number of false positives due to the expansive search space,

which can be alleviated but only at the cost of greater efficiency. Though these and other score-based methods may be useful for analysis and comparison, we will primarily direct our focus to constraint-based learning algorithms in the following sections.

In addition to the various approaches for Mb or P/C recovery, some Mb learning algorithms are designed not merely for local structure learning, but as the primary component for learning the global network with greater efficiency. Learning the Mb as part of a divide-and-conquer algorithm for estimating the identifiable structure of the entire network carries significant benefits, since such a strategy combines quality performance with improved scalability compared to other global learning methods, even in datasets with thousands of variables (Aliferis and Tsamardinos, 2003; Meinshausen and Bühlmann, 2006). We will consider this extension when we discuss the Grow-Shrink algorithm, because this approach combines estimated local neighborhoods from different parts of the network and thus provides insight into a potential strategy for addressing the multi-neighborhood learning problem.

2.4 Nodewise regression

A naïve approach to the Mb estimation problem is to regress the target node against the remaining features, which seems to run afoul of the principle previously stated to avoid employing a prediction-based feature selection procedure for causal discovery. Though this procedure is more formally aligned with a wrapper algorithm, such an approach does in fact contain some similarities with the other filtering algorithms we consider. In fact, this approach is also called an embedded method, which combines the filter selection stage with the learning step for optimizing an objection function (Guo et al., 2022). The basic idea of this approach is to identify a parsimonious model from which we can form the Mb by selecting features with non-zero or statistically significant coefficients. Dobra and West (2004) used a similar idea in their global learning method by chaining together regression models to form a structure by employing a stochastic method which maximizes a score for a DAG

structure.

For this approach, including a lasso penalty is particularly useful since the optimal model will be sparse due to the properties of the ℓ_1 norm. In practice, however, this is too slow to be feasible for a high-dimensional dataset. Algorithms such as the Least Absolute Shrinkage and Selection Operator (LASSO) (Tibshirani, 1996), Least-angle Regression (LARS) (Efron et al., 2004), and Orthogonal Matching Pursuit (OMP) (Pati et al., 1993) are different types of penalized regression. Moreover, they may be formally compared to other Mb learning algorithms, which are generally characterized by forward and backward steps in defining the candidate Mb set. A forward step usually uses some statistical criterion to add a feature to the candidate set, while a backward step prunes the feature set by removing false positives often as a result of a CI test conditioned on the other variables in the candidate set. Borboudakis and Tsamardinos (2019) provide a brief high-level overview of the similarities these embedded algorithms share with other filtering algorithms.

2.5 Grow-Shrink algorithm

Other local learning algorithms directly search the feature space to obtain a candidate set of variables relevant to the target according to statistical criteria and without consideration of the regression or classification task. Koller and Sahami (1996) were the first to use the Mb concept to learn a feature subset for a classifier. They proposed a greedy algorithm for backward feature selection using cross-entropy, eliminating irrelevant features one at a time from the full feature set. Singh et al. (1996) followed a similar approach but used forward feature selection instead. Sierra and Larrañaga (1998) used a wrapper method within a score-based algorithm for learning a network. Their search method is a genetic algorithm in the space of DAGs, where all features must belong to the Mb of the class variable, and each candidate network is evaluated by the accuracy of the classifier. However, these were early attempts with significant deficiencies preventing their widespread use (Aliferis et al., 2003).

The Grow-Shrink (G-S) algorithm is one of the fundamental algorithms for Mb recovery and made significant improvements upon these earlier attempts, though it is not without its own limitations (Aliferis and Tsamardinos, 2003). When introduced by Margaritis and Thrun (1999), it was the first sound Mb estimation algorithm to be published, using a novel structuring of independence tests and mutual information calculations for local discovery (Gao and Ji, 2017). As indicated by its name, the algorithm is broken into two phases: a growth phase and a shrinking phase. Algorithm 4 presents the structure of the algorithm, which also serves as the basic template for many other algorithms following the same two-phase procedure, also known as forward and backward selection. During the growth phase, variables are added to a candidate Mb set based on sufficient dependence on the target node, conditioned on the candidate set of the current iteration, until there are no more remaining variables with sufficient conditional dependence. Then, during the shrinking phase, nodes are removed from the candidate set based on CI results conditioned on the remaining nodes of the candidate set, until there are none remaining which we can conclude are conditionally independent. During the growth phase, it is possible to mistakenly add false positives to the candidate set, which is the theoretical basis for adding the shrinking phase to remove them.

This algorithm is $O(p)$ with respect to CI tests (Margaritis and Thrun, 1999). In practice, variables are often ordered by similarity with the target before the growth phase in lines 3 to 5 for better efficiency and to reduce the number of tests required in the shrink phase of lines 6 to 8. This will speed up runtime and improve data efficiency (Gao and Ji, 2017). However, because this is a weak heuristic, the candidate set will grow more than necessary, which exacerbates the problem of data inefficiency since the required sample size grows exponentially with respect to the candidate size in order to obtain reliable CI test results. The larger sample requirement follows from the degrees of freedom in the CI test, which are exponential in the size of the conditioning set, and some of the tests will condition on the entire Mb candidate set due to the structure of the algorithm (Peña et al., 2005; Aliferis and Tsamardinos, 2003). This is a problem for smaller datasets which cannot compensate for the

loss of power.

Algorithm 4 Basic G-S Mb estimation

Input: target t
$\widehat{mb}(t) \leftarrow \emptyset$
while $\exists\, i \in V \setminus (\widehat{mb}(t) \cup \{t\})$ s.t. $X_i \not\perp\!\!\!\perp X_t \mid \mathbf{X}_{\widehat{mb}(t)}$ **do**
    $S \leftarrow S \cup \{i\}$
end while
while $\exists\, i \in \widehat{mb}(t)$ s.t. $X_i \perp\!\!\!\perp X_t \mid \mathbf{X}_{\widehat{mb}(t)\setminus\{i\}}$ **do**
    $\widehat{mb}(t) \leftarrow \widehat{mb}(t) \setminus \{i\}$
end while
Output: $\widehat{mb}(t)$

This algorithm may also be extended to estimate the global structure. The steps for the complete algorithm are found in Algorithm 5, along with orientation rules in Algorithm 6. In the plain version, we begin by computing the Mb for all nodes in the graph. Then, adjacencies within each Mb are determined using a CI test conditioned on the rest of the Mb, where dependence entails adjacency and independence entails non-adjacency, thus preserving the pairwise Markov property. This step produces an estimate of the skeleton, at which point the algorithm begins orienting some of the edges by first identifying spouses. Between two nodes in the same Mb, we test all possible subsets of the smaller of the Mbs of each node as potential separating sets. Edges are then oriented based on a dependence rule, which exploits the fact that two variables with a common descendant become dependent when conditioning on a set that includes any such descendant. For a pair of adjacent nodes i and j, we can determine whether j is a parent of i if there exists another node k adjacent to i and non-adjacent to j such that any attempt to produce a CI relation between j and k by conditioning on a subset of the Mb of j which includes i fails, assuming $|Mb(j)| < |Mb(k)|$. If there is a v-structure (j, k, i) in the true DAG, then there should be no such subset, because there is a permanent dependency path when conditioning on i. After this step, all cycles are

removed from the graph by deleting edges that are involved in the most cycles until there are no more cycles, then reinserting those edges with reversed orientation. In addition, if there is a directed path from i to j and i and j are adjacent, then direct $i \to j$ to preserve acyclicity. Though these rules are fewer and Meek's rules are complete, we argue that the application of the global G-S extension steps to the multi-neighborhood problem is roughly equivalent to applying the PC algorithm to each neighborhood individually after estimating the Mb of each target node.

However, while this extension is valuable for us to consider as a possible approach to the multi-neighborhood problem, it is limited in its utility. First, the orientation rules are limited to identifying v-structures or preventing cycles. In the multi-neighborhood setting, only the v-structure orientation rules are likely to be of use. Second, even with the addition of Meek's rules, this algorithm does not provide guidance for coordinating learning across multiple neighborhoods. That is, the algorithm does not provide any means to take graph topology into account without estimating the global structure. The global extension of G-S provides a good first attempt toward addressing the multi-neighborhood problem by learning the Mbs of multiple target nodes as well as the skeletons of the neighborhoods. However, the inability to coordinate learning across the neighborhoods without estimating the entire DAG structure is a significant liability.

Algorithm 5 G-S complete DAG structure learning

Set $G = (V, \emptyset)$
for all $i \in V$ **do**
    Compute $\widehat{mb}(i)$ using Algorithm 4
end for
for all $i \in V$ and $j \in \widehat{mb}(i)$ **do**
    $adj_G(i) \leftarrow adj_G(i) \cup \{j\}$; $adj_G(j) \leftarrow adj_G(j) \cup \{i\}$
    Let R be the smaller of $\widehat{mb}(i) \setminus \{j\}$ and $\widehat{mb}(j) \setminus \{i\}$
    for all $S \subseteq R$ **do**
        if $X_i \perp\!\!\!\perp X_j \mid \mathbf{X}_S$ **then**
            $adj_G(i) \leftarrow adj_G(i) \setminus \{j\}$; $adj_G(j) \leftarrow adj_G(j) \setminus \{i\}$
        end if
    end for
end for
for all $i \in V$ and $j \in adj_G(i)$ **do**
    for $k \in adj_G(i) \setminus (adj_G(j) \cup \{j\})$ **do**
        Orient $j \rightarrow i$ and $k \rightarrow i$
        Let Q be the smaller of $\widehat{mb}(j) \setminus \{k\}$ and $\widehat{mb}(k) \setminus \{j\}$
        for all $S \subseteq (Q \setminus \{i\})$ **do**
            if $X_j \perp\!\!\!\perp X_k \mid \mathbf{X}_{S \cup \{i\}}$ **then**
                Orient $j - i$ and $k - i$
            end if
        end for
    end for
end for
Complete edge orientation of G using Algorithm 6
Output: G

Algorithm 6 G-S edge orientation

Input: Partially directed graph $G = (V, E_G)$
$R \leftarrow \emptyset$; multiset $C \leftarrow \emptyset$
for each directed path $\pi = \langle v_1, v_2, \ldots, v_q, v_1 \rangle$ in G **do**
    for every $k \in [q-1]$ **do**
        $C \leftarrow C \cup \{(v_k, v_{k+1})\}$
    end for
    $C \leftarrow C \cup \{(v_q, v_1)\}$
end for
while there exists cycles in G **do**
    Find (i, j) with multiplicity m_C, currently the largest in C
    $E_G \leftarrow E_G \setminus \{(i, j)\}$; $R \leftarrow R \cup \{(i, j)\}$
    Remove all instances of (i, j) from C: $C \leftarrow C \setminus \{(i, j)^{m_C}\}$
end while
for all $(i, j) \in R$ **do**
    $E_G \leftarrow E_G \cup \{(j, i)\}$
end for
for all i and j s.t. $(i, j), (j, i) \in E_G$ **do**
    if $\exists$ a directed path from i to j **then**
        Orient $i \rightarrow j$: $E_G \leftarrow E_G \setminus \{(j, i)\}$
    end if
end for

With respect to the G-S algorithm for a single target node, though the algorithm is sound, it also has deficiencies and shortcomings. The G-S algorithm works best when the largest Mb is small, though in practice this is not always the case. Additionally, G-S is not reliable in recovering the correct Mb with small datasets because of its inefficient heuristic. Rather than dynamically ordering variables for inclusion based on target similarity conditioned on

the updated candidate set during forward selection, G-S uses a static ordering from the first step of the growth phase based on the strength of association of each node with the target conditioned on the empty set (Aliferis and Tsamardinos, 2003). Since spouses are only weakly associated with the target node when conditioned on the empty set, these nodes will not be admitted to the candidate set until late in the growth phase. This permits more false positives to enter the candidate set and makes some CI results unreliable much sooner.

Consequently, while G-S serves as a helpful starting point for our consideration as a sound algorithm for Mb recovery, we must look elsewhere for further improvements to address the inherent deficiencies in the method.

2.6 IAMB family

In the work of Tsamardinos et al. (2003b), the G-S algorithm is modified to produce the Incremental Association Markov Blanket (IAMB) algorithm, which uses an improved heuristic and an optional post-processing step to improve the scalability of the algorithm, as demonstrated empirically with the Thrombin dataset and with other simulated datasets (Aliferis et al., 2010a). The IAMB heuristic dynamically orders the variables according to conditional dependence each time the Mb candidate set changes, thus reducing the number of false positives in the growth phase and improving overall accuracy (Gao and Ji, 2017). Additionally, this method is better suited than the G-S algorithm for working with smaller datasets where the size of the neighborhood is large, since the latter suffers from data efficiency problems due to larger conditioning sets (Tsamardinos et al., 2003b). For constraint-based Mb recovery algorithms, IAMB is frequently used as a standard of comparison.

However, IAMB does not come without its shortcomings. Though it does improve upon the G-S algorithm, IAMB can still be data inefficient, requiring a sample size at least exponential in the size of the Mb. IAMB may require fewer tests than other methods, but since these tests potentially condition on larger sets, we encounter the similar sample inefficiencies as the

G-S algorithm (Băncioiu and Brad, 2022). These results were confirmed in the work of Fu and Desmarais (2008), where various Mb algorithms were compared using a CI oracle instead of a sample CI test. In their findings, IAMB performs well with respect to reducing CI tests, but fails to produce similar improvements in data efficiency.

Further modifications were made to the IAMB algorithm, and these algorithms may be jointly classified as members of the IAMB family of algorithms. In the initial paper, Tsamardinos et al. (2003b) proposed some variants of the original algorithm: IAMBnPC, interIAMB, and interIAMBnPC. These variants introduce combinations of two new concepts to address sample inefficiency: an interleaving principle and a post-processing step with the PC algorithm. The interleaving principle combines the forward and backward steps in each forward iteration to keep the candidate Mb as small as possible. The post-processing step exchanges the backward phase of the algorithm for the PC algorithm over the candidate Mb, since the PC uses an efficient separating set search. While interIAMBnPC uses both concepts, interIAMB and IAMBnPC use the interleaving principle and the post-processing step, respectively. In addition to these, Peña et al. (2007) introduced the KIAMB algorithm, which modifies the IAMB algorithm by using an improved heuristic that allows for a tradeoff between greediness and randomness in its search. Other algorithms in the IAMB family include FastIAMB (Yaramakala and Margaritis, 2005) and λ-IAMB (Zhang et al., 2010).

2.7 HITON family

In the work of Aliferis et al. (2003), the authors present HITON, named for its similarity to the Greek word for "blanket," as a sound and data efficient Mb learning algorithm which has been used to diagnose melanoma (Sboner and Aliferis, 2005), to identify biomarkers for cancer diagnosis (Statnikov et al., 2005), and for other applications (Aliferis et al., 2010a). Unlike G-S and the IAMB family, the HITON algorithm breaks the problem of Mb estimation into sequentially recovering the sets of a partition of the Mb, first finding the P/C set and

then finding the spouses of the target node. The P/C identification step is similar to the growth phase of G-S, since it adds variables to the candidate Mb set based on similarity with the target variable according to an appropriate dependence measure. However, HITON also employs an interleaving strategy in the P/C recovery step. As with interIAMB, the interleaving strategy includes both inclusion and exclusion at each iteration of the growth phase until a stopping criterion is reached. During each iteration of P/C recovery, HITON removes nodes from the candidate set and from further consideration if there is a CI relation between the node and the target conditioned on the rest of the candidate set. This subroutine, concerned with P/C discovery, is denoted HITON-PC. After HITON-PC is applied to the target node to obtain the candidate P/C set, it is sequentially applied to each of the members of the candidate P/C set to obtain second-order candidate P/C sets. The initial candidate Mb set for the remaining steps of the HITON algorithm, denoted HITON-MB, is the union of the candidate P/C set and the second-order P/C sets. From there, a series of CI tests are conducted to remove false positives using conditioning sets of increasing size until conditional dependence is affirmed, in which case the node is retained in the candidate Mb set, or until a CI relation is found, leading to removal of the node from the candidate set.

Due to the interleaving principle and the structure of the algorithm, HITON provides a framework for scalable, data efficient Mb learning algorithms. However, in the work of Peña et al. (2007), the authors identify a flaw in the soundness proof, meaning that HITON does not produce the correct Mb, assuming a CI oracle, in all cases. It should be noted, however, that HITON does include a post-processing wrapper algorithm which may, in principle, remove these false positives (Aliferis and Tsamardinos, 2003). To address this issue, HITON requires a "symmetry correction" such as the one proposed in the work of Tsamardinos et al. (2006), where a node is only included in the output if its own HITON output includes the target node.

Just as in the case with the IAMB algorithm, there are other algorithms structurally similar to HITON which may be classified as part of the HITON family of algorithms.

One such algorithm is PCMB, proposed by Peña et al. (2007), which corrects the errors of HITON and MMMB, another algorithm with the same formal structure and symmetry correction requirement. Just as in HITON, the PCMB algorithm finds the Mb by separating its search into two tasks: first estimating the P/C set, then searching for the spouses of the target node. This algorithm uses a min-max heuristic and an interleaving principle during the growth phase, followed by a shrinking phase where false positives are removed. This algorithm is more data efficient than IAMB because it takes graph topology into account, leading to smaller conditioning sets for the CI tests, a result empirically demonstrated in its improved accuracy (Peña et al., 2007; Băncioiu and Brad, 2022). Fu and Desmarais (2008) also proposed the Iterative Parent-Child Based Search of Markov Blanket (IPC-MB), which improves the efficiency of PCMB by ensuring that the first CI tests are performed with the smallest conditioning sets, thereby removing false positives with data efficiency and leading to an increase in accuracy (Fu, 2010; Băncioiu and Brad, 2022).

2.8 Recent algorithms

In addition to these popular families of algorithms, this section will cover other recent methods.

Niinimäki and Parviainen (2012) presented a score-based learning algorithm (SLL) which provides a soundness guarantee when the sample size approaches infinity. This is another two-phased method with structural similarities to HITON, first identifying the P/C set before identifying the rest of the spouses in the Mb. However, instead of using independence tests to incrementally define the Mb, SLL instead obtains the Mb from optimal networks according to the selected scoring metric. Though this method was accurate in the empirical study, it is costly with respect to runtime.

Ling et al. (2022a) proposed two online local learning algorithms, the Online SimulTaneous (O-ST) and the Online Divide-and-Conquer (O-DC) Mb learning algorithms. While the

O-ST algorithm learns the entire Mb simultaneously, using tests conditioned on the current candidate set to accept or reject nodes, the O-DC follows the HITON family by learning the P/C and spouse sets sequentially. In most cases, the authors argue, it is more likely that researchers will not be able to obtain the entire feature space in advance. This means that variable sets arrive sequentially in time, providing a practical justification for their online methods.

In Kaufmann et al. (2016), the authors introduced a Bayesian algorithm for learning the Mbs of multiple targets in set T simultaneously without estimating the entire network, using the blockwise decoupling in the factorization of the posterior distribution such that the subgraph over the Mbs is conditionally independent of the rest of the network. This algorithm is primarily for Gaussian data, employing a block Gibbs sampler for the posterior distribution of the covariance matrix, assumed to follow the Wishart distribution, but with a compound prior distribution to ensure sparsity. The authors prove that the resulting posterior distribution of the Mb for a set of target nodes has an analytic form, independent of a large portion of the network. This method may also be extended to include other kinds of data, including discrete data and data which includes missing values. However, it is a very computationally expensive algorithm where sampling from the posterior distribution is $O(|T|q^3)$, where $q = p - |T|$.

Gao and Ji (2017) proposed the simultaneous Markov blanket (STMB) algorithm, an efficient, topology-based Mb discovery algorithm which removes the symmetry correction step required for algorithms such as HITON while following a similar divide-and-conquer strategy. The algorithm avoids the computational cost of symmetry correction by exploiting a coexistence property between spouses and descendants of the target node, which allows the algorithm to properly identify false positives in the P/C set. The method is sound and complete while reducing the number of tests required.

To improve computation time for the feature selection problem, Borboudakis and Tsamardinos (2019) suggested an algorithm which uses multiple rounds of forward-backward selection

with early dropping. Early dropping is a heuristic which speeds up forward selection without sacrificing the quality of the estimated set by filtering out variables independent of the target node conditioned on the iteration's current candidate set. This is similar to the interleaving principle, but is applied to nodes which may potentially be added to the candidate set. Rather than only running the algorithm once, which would produce false negatives, the authors propose running the algorithm up to K times, initializing each succeeding round of the algorithm with the output set from the prior run. They call this method the Forward-Backward selection with Early Dropping (FBED^K) algorithm. The authors note that usually only a small value of K is actually necessary. Early dropping allows the FBED^K family of algorithms to mitigate the inefficiency and multiple testing problems by reducing the pool of potential variables at each step. Along with competitive empirical performance, the authors prove that, if the distribution may be faithfully represented by a causal graph, FBED^K will identify the Mb of the target.

In Guo et al. (2022), the authors designed their algorithm, Error-Aware Markov Blanket (EAMB) learning, to deal with the problem of multiple testing. They develop two novel subroutines, beginning with the Efficiently Simultaneous MB (ESMB) algorithm, which speeds up computational efficiency of EAMB by using a "double-shrinking" strategy to reduce the sizes of the conditioning set for the CI tests as well as the feature set pool simultaneously. The second method, the Selectively Recover MB (SRMB) algorithm, uses a strategy to efficiently identify the Mb among features discarded due to unreliable CI tests. During this second step, the algorithm recovers missing spouses using a relaxed rule for symmetry correction. This algorithm maintains a complexity comparable to other state-of-the-art algorithms while improving data efficiency.

Finally, some recent work does not contribute new algorithms, but they do improve the efficiency of existing methods. In the work of Srivastava et al. (2020), the authors propose a parallel framework which allows for an efficient parallel version of any blanket learning algorithm for a local-to-global approach, which can work for algorithms such as G-S, IAMB,

and interIAMB.

2.9 MMPC algorithm

Due to its previous implementation in R, we will take a closer look at the Max-Min Parent Child (MMPC) algorithm because we use this algorithm in the empirical study of the multi-neighborhood learning problem in Chapter 4. The MMPC algorithm is one of the subroutines for the Max-Min Markov Blanket (MMMB) algorithm, a member of the HITON family. The MMPC algorithm learns the P/C set of a target node using a max-min heuristic and a similarity metric, provided in Algorithm 7, forming the P/C set using a grow-shrink method followed by a symmetry correction step to avoid the same error we find in the HITON algorithm (Tsamardinos et al., 2006; Peña et al., 2007). The heuristic selects the variable that maximizes the minimum association with the target node relative to a subset of the current candidate P/C set. Intuitively, the selected variable is included in the candidate P/C set because it remains highly associated with the target node despite the "best effort" to make the variable conditionally independent of the target node.

Algorithm 7 Max-Min heuristic

1: **Input:** target t, candidate parent-child set $CPC \subseteq V \setminus \{t\}$

2: $assocF = \max_{i \in V} MinAssoc(i; t \mid CPC)$

3: $F = \arg\max_{i \in V} MinAssoc(i; t \mid CPC)$

4: **Output:** $\langle F, Fassoc \rangle$

Upon completion of the process outlined by the pseudocode in Algorithm 8, the algorithm outputs the estimated P/C set for the target node (Tsamardinos et al., 2006). Lines 2 to 8 describe a greedy procedure to construct the candidate P/C set, which may lead to the addition of false positives. Therefore, in lines 9 to 13 we search for any subset of the candidate P/C set to identify and remove false positive nodes.

The pseudocode for the entire procedure may be found in Algorithm 9, which also

Algorithm 8 $\overline{MMPC}$

Input: target t
$CPC = \emptyset$
while CPC is still changing **do**
    $\langle F, assocF \rangle = MaxMinHeuristic(t, CPC)$
    if $assocF \neq 0$ **then**
        $CPC \leftarrow CPC \cup F$
    end if
end while
for $i \in CPC$ **do**
    if $\exists S \subseteq CPC$ s.t. $X_i \perp\!\!\!\perp X_t \mid \mathbf{X}_S$ **then**
        $CPC \leftarrow CPC \setminus \{i\}$
    end if
end for
Output: candidate parent-child set CPC

includes the symmetry correction step to remove any remaining false positives and ensure the correctness of the algorithm.

The MMPC algorithm improves the data efficiency of previous algorithms because, unlike G-S and IAMB, the sample size requirements depend on the local topology instead of the conditioning set (Gao and Ji, 2017). However, as Peña et al. (2007) note, the output of MMPC must be further processed even after the symmetry correction, because the candidate P/C set may contain some descendants of the target node other than its children. Notwithstanding these issues, the MMPC and MMMB algorithms are still frequently used in practice and as a benchmark for newer methods. In fact, MMMB serves as the underlying method for the global Max-Min Hill Climbing (MMHC) algorithm, a hybrid method which conducts a score-based search over the space of DAGs using the constraints provided by the Mbs recovered from repeated application of MMPC over all the nodes. In the work of Tsamardinos et al. (2006),

Algorithm 9 $MMPC$

Input: target t
$CPC = \overline{MMPC}(t)$
for $i \in CPC$ **do**
    if $t \notin \overline{MMPC}(i)$ **then**
        $CPC \leftarrow CPC \setminus \{i\}$
    end if
end for
$\widehat{P/C}(t) \leftarrow CPC$
Output: $\widehat{P/C}(t)$, the estimated P/C set for node t

the authors find that this method is both scalable and attains a high standard of quality compared to other global network learning algorithms (Aliferis et al., 2010a). However, due to the nature of score-based algorithms, there is no clear framework for applying the principles of MMHC to the multi-neighborhood problem without estimating the global structure.

2.10 Conclusion

The multi-neighborhood problem is fundamentally local in scope, and as such any suitable approach depends on efficient and sound methods for recovering the Mbs of target nodes. In this chapter, we considered different strategies and families of algorithms, any of which may, in principle, be used for the first stage of the algorithm we introduce in the next chapter to filter out irrelevant variables. In choosing such a method, it is of primary concern to address the potential problems of overfitting due to multiple testing and data inefficiency, especially for applications with smaller datasets. Moreover, we also observed that some algorithms provide a framework for extending Mb recovery algorithms to estimate the global structure of the network. However, none of these approaches are easily or efficiently adaptable to learning the graphical structure in the multi-neighborhood setting without estimating the

global structure. We will address such an extension that avoids the inefficiency of global structure learning in the next chapter.

CHAPTER 3

Coordinated Multi-Neighborhood Learning Algorithm

3.1 Preliminaries

In the previous chapters, we laid the groundwork for the Coordinated Multi-Neighborhood Learning algorithm by defining terms and presenting algorithms which contain principles we will be able to draw from and apply to our method. In this chapter, we will introduce and motivate the multi-neighborhood problem, present the algorithm procedure, and examine some of the algorithm's theoretical properties and practical benefits.

3.2 Motivation

In recent years there has been increased development in structure learning algorithms for directed acyclic graphs (DAGs) (Heinze-Deml et al., 2018; Vowels et al., 2022; Kaddour et al., 2022). However, these algorithms are limited by strong assumptions and intractable practical requirements which render them unreasonable or too restrictive for use in many applied settings. Similarly, in empirical settings we observe the rapid deterioration of the speed and accuracy of most algorithms as the size of the network increases even moderately (Gu and Zhou, 2020). Moreover, in fields such as genomics, researchers are often interested in causal discovery for only a few nodes in order to estimate their causal effects on other downstream nodes. This can be particularly challenging, especially since datasets frequently have many features and few observations (Friedman et al., 2000). In situations such as these, global causal discovery methods suffer from the problems previously mentioned.

In this work, we distinguish between local and global algorithms based on the proportion of network variables included in the algorithm and its output. Global algorithms aim to estimate features from the entire graph among all nodes, while local algorithms are limited to estimation on a strict subgraph. It would be advantageous for a suitable algorithm to leverage local knowledge of the causal structure to adequately and reliably answer causal inquiries at a level at least commensurate with the performance of global algorithms. Referring again to the field of genomics, for example, it may be of interest to consider only a few target genes in a gene regulatory network and identify the causal effects of these on other genes of interest. Indeed, many experiments are limited to only a few such nodes due to various constraints (Michailidis and d'Alché Buc, 2013). Rather than focusing on estimating the entire causal structure from data to answer causal questions related to only a select portion of the features, we take a different approach. We aim to learn only the identifiable structure on a subgraph sufficient for estimating causal effects of interest. Our local approach reflects a priority to estimate the causal effects most relevant to researchers and pursue causal discovery accordingly.

The basis for our algorithm rests on the observation that, in order to estimate causal effects, we only need to estimate the local structure around the node whose causal effects on other nodes we are seeking to estimate. Let X be a target node in a DAG G. If the parent set of X is given, then one can calculate its causal effect on any other variable in the DAG by the so-called back-door, or parent set, adjustment (Pearl, 2009). We now define Z as the parent set of X in the causal graph. Our goal is to estimate the causal effect of X on another node Y which, using Pearl's atomic intervention notation, may be written as $p(y \mid do(X = x))$. This can be calculated using the parent set adjustment as

$$p(y \mid do(X = x)) = \int_z p(y \mid do(x), z)p(z \mid do(x))dz \tag{3.1}$$

$$= \int_z p(y \mid x, z)p(z)dz. \tag{3.2}$$

From the last statement, we conclude that, in order to estimate the causal effect of X on Y

using exclusively observational data, we only require knowledge of the parent set of X.

On this basis, we reason that local learning algorithms should be primarily focused on estimating the neighborhood structure around the specified target nodes, particularly to identify parents for the purpose of estimating causal effects. Indeed, it is this concept which serves as the basis for the IDA and joint-IDA algorithms in the estimation of causal effects, both of which assume parent sets are given in order to proceed with their estimation procedures (Maathuis et al., 2009; Nandy et al., 2017).

However, apart from a few exceptions, current methods for learning parent sets suffer from inefficiencies inherent in global causal discovery algorithms or from a failure to adequately distinguish between parents, children, and spouses in local learning algorithms, such as some Mb learning algorithms (Aliferis et al., 2010a,b; Gao and Ji, 2017). Our method is designed to correct these shortcomings by maintaining the efficiency of a local learning approach while attempting to orient as many identifiable edges in the subgraph of the DAG as possible, thereby limiting the size of a possible parent set. This motivates us to pursue coordinated local learning, since by learning the structure of the neighborhoods simultaneously, we can orient more edges than we could by examining individual neighborhoods. Additionally, local structure learning provides theoretical and practical advantages by relaxing global assumptions and substantially reducing computational complexity and runtime, as will be demonstrated by our empirical results in Chapter 4. In reality, one can rarely be assured of the propriety of including as many variables as are present in a high-dimensional graph, nor is it always safe to make faithfulness or other assumptions on the entire network. The principles of our algorithm are founded on more modest claims focused on a subgraph of only the most relevant nodes. To summarize, we will show that coordinated multi-neighborhood learning is, especially in comparison to global algorithms, a scalable and efficient approach with a concomitant reduction in the size of the set of possible parents for greater precision in causal effect estimation.

While there are existing methods for estimating the neighborhood and the graphical

structure around a single target node, such as those we cover in Chapter 2, there are, to our knowledge, no existing methods to coordinate structure learning around multiple neighborhoods of interest from observational data without estimating the global structure. Applying existing methods to each neighborhood individually is sound for identifying members of each neighborhood provided the underlying algorithm is correct. However, these methods are limited in their ability to orient edges within single neighborhoods due to the inevitable loss of structural information. Moreover, it would be particularly challenging to identify any topological ordering between the neighborhoods. Causal discovery in such a scenario is limited if we restrict ourselves to considering one neighborhood at a time rather than finding a way to coordinate structure learning over multiple neighborhoods.

3.2.1 Background

For the local learning problem in this work, we intend to learn the structure of local neighborhoods, coordinating the results such that causal information, as encoded in edge orientation, can, in principle, be passed from one neighborhood to another. Unlike many other algorithms, our algorithm treats some of the variables as latent since we refrain from estimating the entire graph structure. This restriction requires a different class of graphs to accommodate latent variables while retaining the capacity for encoding causal information. For this purpose, we use ancestral graphs due to their facility for conveniently representing causal information inferred from data on observed nodes in the presence of latent variables (Richardson and Spirtes, 2002; Zhang, 2008a). Because ancestral graphs can represent true CI and causal relations among observed variables in the presence of latent variables, they are perfectly suited for our problem.

For clarity, it must be acknowledged that none of the nodes are latent in a proper sense. That is, each of the variables may, in principle, be invoked and potentially included in the algorithm or in the final output. However, after being filtered out by the Mb learning algorithm, many nodes will not be included in the final output. Thus, it is more accurate to

say that these nodes are graphically latent. While these nodes are included in the underlying DAG and may even be used during the execution of the algorithm as potential members of a separating set, they will otherwise be treated as latent variables in the final output. We further assume the absence of any nodes which may properly be called latent, thus ensuring causal sufficiency within each neighborhood.

In this work, we only consider two kinds of edges: directed ($\rightarrow$) and bi-directed ($\leftrightarrow$). Since we assume the absence of selection bias, there are no undirected edges in the ancestral graphs we consider in this chapter. In the literature these graphs are sometimes referred to as directed maximal ancestral graphs (DMAGs), which are special cases of the MAG defined in Definition 8, differing only in that they are graphical representations accommodating a marginal distribution over observed variables under the assumption of potential latent confounders but not selection variables (Borboudakis et al., 2012).

The method we are proposing fits within the class of constraint-based structure learning algorithms such as the PC and FCI algorithms, using both CI tests and deterministic rules to recover a ground truth graph. The novelty of the algorithm we are proposing is twofold: it offers coordinated local learning across multiple neighborhoods with theoretical guarantees, and it lowers computational cost by not estimating the entire graph structure. The algorithm is well-motivated by causal inference problems, since the local structure is sufficient for causal effect estimation by back-door adjustment. Some of the components of the algorithm, especially the specific Mb estimation algorithms, may be substituted and are built upon other work. Though it relies on other algorithms for a pre-processing step, the method provides a novel, general framework for approaching coordinated local learning across multiple neighborhoods, which allows us to orient more edges and improve subsequent causal effect estimation attempts. In contrast, most other existing local algorithms consider one neighborhood at a time without any coordination, which limits their ability to distinguish between parents and children, an essential task for causal effect estimation.

One may instead consider naïvely estimating the structure of the entire DAG with a

global algorithm such as those discussed in Chapter 1. This approach has the advantage of coordinating learning across target neighborhoods and does a better job distinguishing between parents and children. However, global algorithms are often computationally expensive and prone to more errors, often propagated by nodes outside of the specific set of interest.

There are some attempts in the literature which pursue causal effect estimation using local algorithms for structure learning. Gupta et al. (2023) run the PC algorithm locally to discover neighborhood structures around nodes using an algorithm called Sequential Discovery and Local Discovery with Eager Collider Checks (LDECC). The collider checks entail searching for separated nodes which become dependent when the target is added to the separating set. The algorithm then orients the smallest subset that d-separate the target from the two nodes as parents. They prove that, with a CI oracle, the estimated average treatment effect (ATE) using this method is equal to the true ATE. Fang et al. (2022) proposed a local approach for identifying causes of a target using a novel graphical condition to check the existence of a causal path between a variable and the target in every Markov equivalent DAG. Their work provides the basis for their algorithm, the local ITC, which takes a target and its neighbors, estimated by a local algorithm, as input and conducts a series of CI tests to determine causal relationships with the target node. The authors also provide a global extension of the algorithm, using the estimated CPDAG as input. One of the principal benefits of this algorithm is that it can increase the accuracy of a causal effect estimation algorithm by definitively ruling out nodes which are definitively non-causes with respect to the target node. While both of these methods share some similarities with ours and may point in fruitful future research directions, they do not provide guidance for coordinated learning for multiple target nodes.

3.2.2 Contribution

In this chapter, we develop a method to address the lacuna in the literature for coordinated learning across multiple neighborhoods. The Coordinated Multi-Neighborhood Learning

(CML) algorithm is designed to maximize causal structure learning in targeted neighborhoods with efficiency and scalability. We do not need to estimate the entire graphical structure, but instead we limit our attention to only the relevant subgraph, which by definition classifies CML as a local algorithm. The algorithm first identifies target neighborhoods using existing Mb estimation methods. We then develop a two-stage constraint-based algorithm. The first stage consists of two phases, where the first constructs the skeleton of a maximal ancestral graph (MAG) over the union of target neighborhoods, maintaining ancestral relationships connecting the distinct neighborhoods. The second phase further prunes edges within each neighborhood after additional CI tests. Both phases of skeleton recovery follow the hierarchical CI test ordering from the PC algorithm for efficient constraint-based design. The last stage involves applying a subset of the complete FCI rules to simultaneously orient edges in all neighborhoods.

The outline for this chapter is as follows. We will introduce CML and present its important features in Section 3.3, as well as a brief discussion of a special case of the algorithm where the target set only includes one node. Then, we will produce some theoretical results in Sections 3.4 and 3.5, including proofs of consistency and discussions of computational complexity. We conclude by reviewing the literature of causal effect estimation in Section 3.6, demonstrating how our algorithm facilitates existing procedures by providing possible parent sets as inputs. We refrain from discussing empirical results until Chapter 4.

3.3 CML algorithm

In many research scenarios, we can safely assume sufficient background knowledge to identify a set of target nodes T. These nodes will be designated by the user to learn their local structures simultaneously, especially to obtain the parent set of each node for causal effect estimation on other nodes downstream using the back-door adjustment. Our method may also partially identify the topological relationship among the target neighborhoods in the

underlying DAG, though this global graph structure is never estimated.

3.3.1 Definitions

We define the set of first-order neighbors of a node i, denoted N_i^1, to be its Mb as given in Definition 9. We call $NB_i := N_i^1 \cup \{i\}$ the neighborhood of i. The second-order neighbor set of node i is the union of the Mbs for each node in its neighborhood, except for nodes already in NB_i, denoted $N_i^2 = \cup_{j \in N_i^1} N_j^1 \setminus NB_i$. For a set of nodes T, the union of their neighborhoods is denoted $NB_T = \cup_{t \in T} NB_t$.

3.3.2 Algorithm details

For the population version of the CML algorithm, we assume perfect knowledge of the CI relations, or a CI oracle, which will be replaced with appropriate CI tests for the sample version. We further assume that we are provided with first- and second-order neighbors of each target node, N_t^1 and N_t^2, respectively, for $t \in T$. For the sample version of CML, we use existing Mb learning algorithms to estimate the neighbor sets. Following similar notation as the FCI algorithm, which partitions the node set $V = O \cup L \cup S$, where O is the observed nodes, L the set of latent nodes, and S the set of selection variables, we define $O = NB_T$, the graphically latent nodes $L = V \setminus NB_T$, and $S = \emptyset$.

Algorithm 10 outlines the steps of the CML algorithm. For skeleton recovery (lines 1 to 12), we begin with a complete graph over NB_T and recursively delete edges based on the CI oracle, or CI tests in the sample version. However, in order to ensure that edges between the neighborhoods are properly maintained for coordinating orientations, this stage takes place in two successive phases. The first phase (lines 2 to 5), the union skeleton recovery phase, is equivalent to the skeleton learning in the FCI algorithm over $O = NB_T$. Thus, only subsets of NB_T are possible separation sets. Edges between nodes in two different neighborhoods may be preserved because they are not separated by any subset of NB_T.

Algorithm 10 Coordinated Multi-Neighborhood Learning

1: $O \leftarrow NB_T$; $E \leftarrow$ edge set of complete, undirected graph over NB_T
2: **for** $(i, j) \in E$ **do**
3: Search for separating set $S_{ij} \subset O$ such that $X_i \perp\!\!\!\perp X_j \mid S_{ij}$
4: If S_{ij} is found, then update $E \leftarrow E \setminus \{(i, j), (j, i)\}$
5: **end for**
6: $E_t \leftarrow \{(i, j) \in E : i, j \in NB_t\}$ for all $t \in T$
7: **for** $t \in T$ **do**
8: **for** $(i, j) \in E_t$ **do**
9: Search for $S_{ij} \subset N_i^1 \cup N_j^1$ such that $X_i \perp\!\!\!\perp X_j \mid S_{ij}$
10: If S_{ij} is found, then update $E \leftarrow E \setminus \{(i, j), (j, i)\}$
11: **end for**
12: **end for**
13: Replace each edge with $\circ\!\!-\!\!\circ$
14: Apply $\mathcal{R}_0$ of the FCI algorithm to identify v-structures based on E and S_{ij}
15: Apply FCI rules $\mathcal{R}_1$ to $\mathcal{R}_4$ and $\mathcal{R}_8$ to $\mathcal{R}_{10}$ until none of them apply
16: Modify edge marks within each single neighborhood with rule $\mathcal{R}_N$

Consider the two neighborhoods $NB_3 = \{1, 2, 3, 4, 5\}$ and $NB_8 = \{7, 8, 9, 10\}$ in Figure 3.1a. No subsets of $NB_3 \cup NB_8$ m-separate X_4 and X_9, and thus the edge $(4, 9)$ will remain in the skeleton in Figure 3.1b, and similarly the edge $(2, 9)$. Note that the two edges correspond to inducing paths relative to $L = \{6, 11, 12, 13\}$. Such between-neighborhood edges will facilitate coordinated orientation in the second stage of our algorithm.

After this phase, there may be extraneous edges present within each target neighborhood, such as the edge $(1, 2)$ in Figure 3.1b. During the second phase (lines 6 to 12), the local skeletons recovery phase, we narrow our focus to one target neighborhood at a time, each considered separately. Now we make use of the second-order neighbors and search for separating sets in $N_i^1 \cup N_j^1$ for $i, j \in NB_t$ to further delete edges within the same neighborhood.

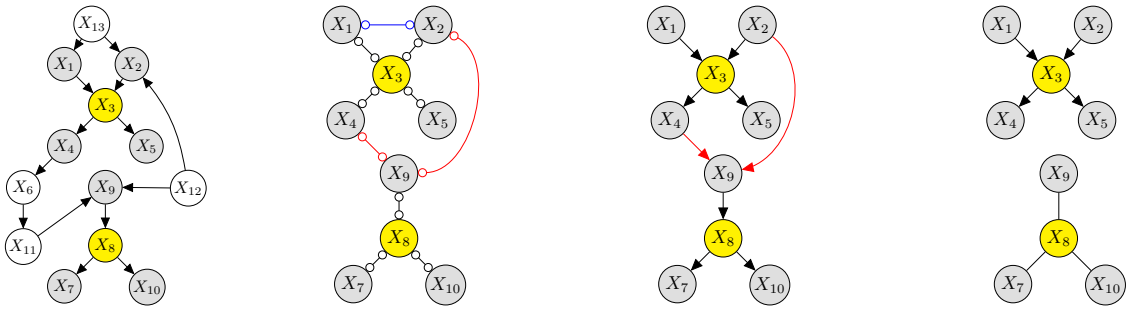


Figure 3.1: An illustration of the Coordinated Multi-Neighborhood Learning algorithm. (a) The neighborhoods of two target nodes. The highlighted nodes $\{X_3, X_8\}$ are the specified target nodes, the gray nodes are members of the Mb of one of the target nodes, and the white nodes are second-order neighbors. (b) Graph after the first phase of skeleton recovery. Edges in red are between-neighborhood edges and edges in black are within-neighborhood edges. The edge in blue is removed during the second phase of skeleton recovery. (c) Output of the CML algorithm. (d) Output of the Single Neighborhood Learning algorithm.

Consider $N_1^1 = \{2, 3, 13\}$ and $N_2^1 = \{1, 3, 13\}$ in Figure 3.1a in order to remove the extraneous edge $(1, 2)$. Though nodes 1 and 2 could not be separated by a subset of NB_3 in the first phase, they are separated by node 13 in the second phase since $13 \in N_3^2$. Therefore, the edge $(1, 2)$ is removed in Figure 3.1c.

By construction every edge between two nodes not in the same neighborhood will remain connected during the second phase of skeleton recovery. This preserves the potential to coordinate learning between neighborhoods and ensures that local learning within individual neighborhoods is maximally informative. For the example we are following in Figure 3.1, we obtain the skeleton of the graph in Figure 3.1c after line 12 of our algorithm, which still retains between-neighborhood edges $(4, 9)$ and $(2, 9)$.

After the skeleton recovery stage, we identify v-structures using the stored separation sets (line 14) and then apply the relevant FCI rules (line 15) discussed in Section 1.5. The

advantage of our method is illustrated in Figure 3.1. Figure 3.1c shows the CML output, in which all edges in the two neighborhoods NB_3 and NB_8 have been oriented and the two between-neighborhood edges are shown in red. If we were to apply our algorithm to each of the two neighborhoods separately, then the output graphs would be the two shown in Figure 3.1d. Because the induced subgraph over NB_8 does not contain a v-structure, none of the four edges are oriented.

After line 15, there may be four types of edges ($\leftrightarrow$, $\rightarrow$, $\circ\!\!\rightarrow$, $\circ\!\!-\!\!\circ$) in the estimated PAG. However, with knowledge of the first- and second-order neighbors, there should be no bidirected edges between two nodes in the same neighborhood. Consequently, we apply an additional set of rules ($\mathcal{R}_N$) to simplify the edge marks in a neighborhood.

$\mathcal{R}_N$: For nodes i and j in the same target neighborhood, convert $i \circ\!\!-\!\!\circ j$ to an undirected edge $i - j$, and convert $i \circ\!\!\rightarrow j$ to a directed edge $i \rightarrow j$.

The soundness of $\mathcal{R}_N$ follows from the interpretation of the edge marks in MAGs and the fact that there are no bidirected edges between pairs of nodes in the same neighborhood. If $i \circ\!\!\rightarrow j$ is within a neighborhood, then the possible orientations are $i \rightarrow j$ or $i \leftrightarrow j$, but the latter (bidirected) is excluded, and thus the orientation must be $i \rightarrow j$. This is because we assume knowledge of N_t^1 and N_t^2 and the absence of latent confounders for each neighborhood, which prevents there being any inducing path between the two nodes. A similar line of reasoning applies to converting $i \circ\!\!-\!\!\circ j$ to an undirected edge, since this denotes uncertainty regarding causal direction while denying the possibility of a bidirected edge between nodes belonging to the same neighborhood.

Remark 10. For a finite sample, a bidirected edge could appear within a neighborhood after we apply the FCI rules in line 15. In such a case, we also convert it to an undirected edge to resolve this conflict in a practical way. This is similar to the situation of conflicting v-structures in learning DAGs.

Remark 11. For each target node $t \in T$, the output graph from CML provides a set of parents $\widehat{pa}(t)$ and possible parents $\widehat{ppa}(t)$ (nodes connected to t by an undirected edge) in

the neighborhood NB_t. Once we have these estimates of the target neighborhoods, we can use existing causal effect estimation procedures such as recursive regressions for causal effects (RRC) or modifying Cholesky decompositions (MCD) of the covariance matrix to infer causal effects of interest (Nandy et al., 2017). We discuss this in more detail in Section 3.6.

3.3.3 Single Neighborhood Learning algorithm

As a special case, and for the purpose of illustrating the advantages of our coordinated method, we consider our algorithm design in the case where there is only one target node. Since we only consider one target node and its neighborhood at a time, there is no requirement for ancestral graphs and the rules from the FCI algorithm, which makes this method a modified version of the PC algorithm applied to a local neighborhood. The Single Neighborhood Learning algorithm (SNL) is described in the pseudocode provided below in Algorithm 11.

The skeleton recovery stage of SNL is only a single phase (lines 2 to 8), which corresponds to the local skeletons recovery phase of CML (Algorithm 10, lines 7 to 12). After skeleton recovery, we orient v-structures and apply Meek's rules, since we no longer require the extra rules provided by the FCI algorithm.

In the case where there are multiple target nodes, the algorithm is applied serially to each target and its neighborhood. Because there is no coordination between the neighborhoods, we expect there to be fewer directed edges than we find in the CML output. As we mentioned in Chapter 2, this is similar to the global extension of the G-S algorithm, since we apply CI tests and a set of deterministic rules over a set of nodes selected by the Mb learning algorithm to obtain the neighborhood structure and orient some edges. However, SNL uses Meek's rules rather than the attenuated G-S rule set, which is usually only applicable to the global structure, for the subgraph over NB_T.

The output of this algorithm given a CI oracle is a PDAG with perfect recovery of the underlying skeleton and v-structures of the induced subgraph G_t over NB_t. Figure 3.1d

presents the output of SNL over the target neighborhoods for the graph in Figure 3.1a. While Meek's rules allow us to properly orient all edges in NB_3, we are unable to orient any of the edges in NB_8.

Algorithm 11 Single Neighborhood Learning

1: $O \leftarrow NB_t$; $E_t \leftarrow$ edge set of complete, undirected graph over NB_t
2: **for** $(i, j) \in E_t$ **do**
3: Search for subset $S_{ij} \subset N_t^1 \cup N_t^2$ such that $X_i \perp\!\!\!\perp X_j \mid S_{ij}$
4: **if** a set S_{ij} is found such that $X_i \perp\!\!\!\perp X_j \mid S_{ij}$ **then**
5: Delete edge (i, j) from E_t
6: Store S_{ij}
7: **end if**
8: **end for**
9: Identify v-structures based on E_t and S
10: Apply Meek's Rules to further orient edges in G_t

3.3.4 Discussion

Although built upon components of some existing algorithms, Algorithm 10 is far from a straightforward extension. The first for-loop (lines 2 to 5) considers multiple neighborhoods as a whole so that later edge orientation (lines 14 to 16) can be done in a coordinated manner. That is, orientation of one neighborhood may help orientation of another neighborhood through edges between the two neighborhood (see Figure 3.1c for an illustration). This step is similar to the skeleton step of the FCI algorithm. In the second for-loop (lines 7 to 12), we further delete edges within each neighborhood to reduce the uncertainty in parent identification, making use of the estimated Mbs of the nodes. This is similar to the skeleton learning of the PC algorithm.

From the preceding reasoning, we may conclude that CML is not a simple extension of

existing work. The skeleton estimation procedure of our algorithm is very different from that of the FCI, and we also add a few additional rules (line 16) for edge orientation to incorporate our knowledge of the target neighborhoods in our graph interpretation. A simple modified PC algorithm for local learning is best embodied in the serial use of the SNL algorithm (Algorithm 11), which cannot coordinate the results between multiple neighborhoods for further edge orientation. This is illustrated in Figure 3.1d and consistent with our empirical results in Chapter 4.

3.4 Theoretical analysis

To perform theoretical analysis of Algorithm 10, we start by defining the ground truth graph for the multi-neighborhood learning problem. Let $\mathcal{G} = \mathcal{G}(V)$ be a DAG over vertex set $V = [p]$, $T \subset V$ be a set of target nodes, and $N = NB_T$ be the union of the neighborhoods of $t \in T$. Then, $B = N \times N - \cup_{t \in T} NB_t \times NB_t$ is the set of node pairs that do not belong to any common neighborhood, which is referred to as between-neighborhood pairs. Denote by $\mathcal{G}_N$ the induced subgraph of $\mathcal{G}$ over N. For each $(i, j) \in B$, if there is an inducing path between them relative to $L = V - N$ in $\mathcal{G}$, add an edge between i and j to $\mathcal{G}_N$ with the following orientation rules: (i) orient as $i \to j$ if $i \in \text{an}_{\mathcal{G}}(j)$; orient as $j \to i$ if $j \in \text{an}_{\mathcal{G}}(i)$; (iii) otherwise, orient as $i \leftrightarrow j$. Denote the resulting graph as $\mathcal{G}_N^*$. As an example, if $\mathcal{G}$ is the DAG in Figure 3.1a with $T = \{3, 8\}$, then $\mathcal{G}_N^*$ is the graph in Figure 3.1c.

Assumption 12. In the DAG $\mathcal{G}$, there is no inducing path relative to L between any two nodes in the same neighborhood NB_t, $t \in T$, such that some intermediate node not in L on the path is in $N \setminus NB_t$.

Lemma 13. *Under Assumption 12, the $\mathcal{G}_N^*$ defined by the above procedure is a MAG.*

Proof of Lemma 13. Since $\mathcal{G}_N$ is an induced subgraph of DAG $\mathcal{G}$, it follows that $\mathcal{G}_N$ is ancestral because there are no directed cycles on the graph since $\mathcal{G}_N$ is a DAG as well. For

the additional edges in $\mathcal{G}_N^*$, it follows from their construction that no directed or almost directed cycles will be introduced. For any $(i, j) \in B$ such that there is an inducing path relative to $L = V - N$, if we set $i \to j$, then it follows that $j \notin an_{\mathcal{G}}(i)$ and $j \notin an_{\mathcal{G}_N^*}(i)$. If we set $i \leftrightarrow j$, then $i \notin an_{\mathcal{G}}(j)$ and $j \notin an_{\mathcal{G}}(i)$ by construction, which also implies $i \notin an_{\mathcal{G}_N^*}(j)$ and $j \notin an_{\mathcal{G}_N^*}(i)$. Therefore, $\mathcal{G}_N^*$ is ancestral since it has neither directed nor almost directed cycles. (Richardson and Spirtes, 2002) proved that DAGs are maximal ancestral graphs, and therefore $\mathcal{G}_N$ is a maximal ancestral graph. After we add directed edges $i \to j$ for $(i, j) \in B$, we still have a DAG and thus preserve the maximality. Furthermore, we would retain maximality after bidirected edges are added between nodes in distinct neighborhoods. We prove the last assertion by contradiction. Assume there is an inducing path with non-adjacent endpoints $\pi_I = \langle \alpha, \beta, \gamma, \ldots, \epsilon, \omega \rangle \in \mathcal{G}_N^*$. The orientation of the edges on π_I is $\alpha * \!\!\rightarrow \beta \leftrightarrow \gamma \leftrightarrow \cdots \leftrightarrow \epsilon \leftarrow\!\!* \omega$, where $*$ is a wildcard which can represent either a tail or an arrowhead. By Assumption 12, α and ω must be in different target neighborhoods. It is easy to see that this path π_I corresponds to an inducing path relative to L in the original DAG $\mathcal{G}$, and thus by construction (α, ω) is an edge in $\mathcal{G}_N^*$. This leads to a contradiction. □

We hasten to note, however, that $\mathcal{G}_N^*$ is not the MAG obtained from $\mathcal{G}$ over N, which is demonstrated by the removal of the edge $(1, 2)$ in Figure 3.1b. In Figure 3.1a, we observe an inducing path $\langle X_1, X_{13}, X_2 \rangle$ relative to $\{X_{13}\}$. Since X_{13} is a second-order neighbor of target node X_3, it will be treated as graphically latent in our algorithm output. That is, though we use second-order neighbors as potential members of a separating set in the second phase of the skeleton recovery portion of our algorithm, we do not invoke the second-order neighbors in any other portion of structure recovery or in the output. On the other hand, in the MAG constructed by marginalizing X_{13}, a bidirected edge will be added between X_1 and X_2. In Figure 3.1a, $\mathcal{G}_N$ is the sub-DAG over $N = \{1, 2, 3, 4, 5, 7, 8, 9, 10\}$, the MAG over N would have skeleton as in Figure 3.1b, and $\mathcal{G}_N^*$ is in Figure 3.1c. In terms of the skeleton, $\mathcal{G}_N \subseteq \mathcal{G}_N^* \subseteq$ the MAG. In summary, $\mathcal{G}_N^*$ is in general a proper subgraph of, and thus sparser than, the MAG constructed by marginalizing L from the DAG $\mathcal{G}(V)$, which can have

additional edges in a neighborhood due to inducing paths.

As a consequence of Lemma 13, the Markov equivalence class of $\mathcal{G}_N^*$ is represented by a PAG, denoted $[\mathcal{G}_N^*]$. Given the CI oracle, which also can be used to perfectly recover the neighbor set, or Mb, of any node, we have the following result for the population version of our algorithm:

Theorem 14. *Suppose the joint distribution $P(X_1, \ldots, X_p)$ is faithful to $\mathcal{G}$. Given the CI oracle, the graph constructed by Algorithm 10 up to the completion of line 15 is the PAG $[\mathcal{G}_N^*]$.*

Proof of Theorem 14. We begin by showing that we recover the skeleton of $\mathcal{G}_N^*$ after the skeleton recovery stage of our algorithm. Since the distribution $P(X_1, \ldots, X_p)$ is faithful to $\mathcal{G}$, conditional independence of X_i and X_j given $\mathbf{X_k}$ is equivalent to m-separation of nodes i and j given set $\mathbf{k}$ for $i, j \in N$ and $\mathbf{k} \subseteq N$. Therefore, after line 5, all extraneous edges between neighborhoods are removed, and the edge set E corresponds to the skeleton of the true MAG over N, which is a supergraph of the skeleton of $\mathcal{G}_N^*$. For any $t \in T$ and $(i, j) \in NB_t$, $N_i^1 \cup N_j^1$ will be sufficient to remove edges between non-adjacent i, j in $\mathcal{G}$. Then, after having used second order neighbors within each neighborhood, we obtain the skeleton of $\mathcal{G}_N^*$ after line 12.

While the correctness of the FCI rules have been shown by Zhang (2008a), we must show that our use of the rules in the CML algorithm is valid. The rules only depend on the skeleton and whether a node $\gamma \in N$ is in a separating set S_{ij} ($\mathcal{R}_0$ and $\mathcal{R}_4$). For any separating set S_{ij} found in the skeleton recovery stage, let $S'_{ij} = S_{ij} \cap N$. That is, we remove any second order-neighbors from the separating set. In the application of the FCI rules requiring the separating set S_{ij}, using S'_{ij} instead will lead to the same orientation result since $\gamma \in S_{ij}$ if and only if $\gamma \in S'_{ij}$ for any $\gamma \in N$. On the other hand, the sets $\{S'_{ij}\}$ are all the separating sets for the sound and complete orientation of $[\mathcal{G}_N^*]$ by the FCI rules. This completes the proof. □

We can further establish structure learning consistency for the sample version of our algorithm when the CI oracle is replaced by consistent CI tests. Denote by $\widehat{G}_n$ the graph

constructed by Algorithm 10 up to the completion of line 15 given a sample of size n from $P(X_1, \ldots, X_p)$.

Theorem 15. *Suppose the joint distribution $P(X_1, \ldots, X_p)$ is faithful to $\mathcal{G}$ and we perform all CI checks in Algorithm 10 using consistent CI tests with significance level α_n. Then, there exists $\alpha_n \to 0$, such that $P(\widehat{G}_n = [\mathcal{G}_N^*]) \to 1$ as $n \to \infty$.*

Proof of Theorem 15. Based on Theorem 14, we just need to show that both Type I and Type II errors of the CI tests approach zero as $n \to \infty$.

Let $p_{n;i,j|\mathbf{k}}$ and $p^*_{n;i,j|\mathbf{k}}$ be the p-values for testing the independence between nodes i and j conditioned on set $\mathbf{k}$ using a dataset with n observations and i and j are and are not separated by $\mathbf{k}$ in $\mathcal{G}$, respectively. In the case where i and j are conditionally independent given $\mathbf{k}$, the Type I error is given by $P(p_{n;i,j|\mathbf{k}} \leq \alpha_n) = \alpha_n$.

For easy understanding of the proof, let us assume for now that the joint distribution P is Gaussian. Due to the faithfulness assumption, for any i and j not d-separated by $\mathbf{k}$ in $\mathcal{G}$ there exists a lower bound $\rho^* > 0$ for the magnitude of the partial correlation. The Type II error may be expressed as $P(p^*_{n;i,j|\mathbf{k}} > \alpha_n)$. In the case where we have Gaussian data, we use Fisher's z-transformation of sample partial correlation $\hat{\rho}_{i,j|\mathbf{k}}$. Let $Z^* = \frac{1}{2}\log\left(\frac{1+\rho^*}{1-\rho^*}\right)$. For all i and j not separated by $\mathbf{k}$, $Z^*_n + O(n^{-1/2}) \leq |Z(i,j;\mathbf{k})|$, where $Z(i,j;\mathbf{k}) = \frac{1}{2}\log\left(\frac{1+\hat{\rho}_{i,j|\mathbf{k}}}{1-\hat{\rho}_{i,j|\mathbf{k}}}\right)$. Therefore, we have

$$\begin{aligned}
P(p^*_{n;i,j|\mathbf{k}} > \alpha_n) &= P(2(1 - \Phi(|Z(i,j;\mathbf{k})|\sqrt{n - |\mathbf{k}| - 3})) > \alpha_n) \\
&= P(|Z(i,j;\mathbf{k})|\sqrt{n - |\mathbf{k}| - 3} < \Phi^{-1}(1 - \alpha_n/2)) \\
&\leq P(|Z^*_n|\sqrt{n - |\mathbf{k}| - 3} + O(1) \leq \Phi^{-1}(1 - \alpha_n/2)).
\end{aligned}$$

Note that $|Z^*_n|\sqrt{n-k-3}$ is on the order of $\sqrt{n}$. We can choose $\alpha_n \to 0$ but $\Phi^{-1}(1-\alpha_n/2) = o(\sqrt{n})$ as $n \to \infty$. Then, both the Type I and the Type II errors go to 0 in the limit.

Similar arguments can be used to prove the result for other distributions with a consistent CI test. □

With these results, we have shown that our algorithm is sound and complete. Using the sample version with consistent CI tests, we have consistency with respect to the equivalence class of the ground truth $\mathcal{G}_N^*$. These results for the CML algorithm will hold for any data distribution as long as a consistent CI test is used.

3.5 Computational complexity

One of the primary benefits of a local algorithm is its improved efficiency, which we seek to demonstrate. We begin by discussing the complexity of Mb recovery algorithms before the complexity of CML proper. Though not necessarily the best or most efficient, we first consider the G-S algorithm as somewhat representative, which uses $O(p)$ CI tests for learning the Mb of a single node. Thus, the total number of tests for finding the first- and second-order neighbors is $O(p|N|)$, where $|N|$ is the number of nodes in the union of the neighborhoods. In our numerical results, we choose to use MMPC for neighborhood estimation. Using this algorithm, the complexity for the neighborhood estimation stage is $O(p|N||\mathbf{PC}_{max}|^{\ell_{max}+1})$, where $\mathbf{PC}_{max}$ is the maximum neighborhood size over nodes in N and ℓ_{max} is the largest size of the tested conditioning sets. Typically, both ℓ_{max} and $|\mathbf{PC}_{max}|$ are small (say bounded by a constant), which is consistent with our aim of causal discovery for small neighborhoods in large graphs. In this case, we will find that the number of tests also reduces to $O(p|N|)$.

Let $f(k)$ and $g(k)$ be the respective computational complexities of the FCI and the PC algorithms on a k-node problem. Then the computational complexity of Algorithm 10 is bounded by $f(|N|) + \sum_{t\in T} g(|NB_t|)$. In the worst case, the computational complexity of the FCI and the PC algorithm is exponential in the number of nodes (Spirtes et al., 2000). Clearly, our local algorithm will achieve substantial computational savings when $|N| \ll p$ compared to applying the PC algorithm on all the p nodes. In this local setting, the Mb recovery algorithm will dominate the rest of the algorithm, though still with reduced complexity compared to a global algorithm. These conclusions will be further demonstrated with our

numerical comparisons in Chapter 4.

3.6 Causal effect estimation

Recall that one of the primary aims for our local method is to learn a portion of the underlying graphical structure sufficient for estimating the causal effects of the target nodes on other nodes in the graph. We now consider how to apply the output from CML to existing algorithms for causal effect estimation.

The intervention do-calculus when the DAG is absent (IDA) method (Maathuis et al., 2009) is used to estimate the total causal effect of a single target node t on another variable given an equivalence class for the underlying graph structure. From the equivalence class, this algorithm identifies valid parent sets using a local criterion to verify that no new v-structures are introduced by the proposed parent set. Let $\widehat{pa}(t)$ and $\widehat{ppa}(t)$ be the parent set and the possible parent set (i.e., adjacent nodes connected by an undirected edge) of the target node estimated by CML, respectively. We may use their algorithm to enumerate all candidate parent sets $\widehat{PA}(t) = \{\widehat{pa}^{(i)}(t) : \widehat{pa}(t) \subseteq \widehat{pa}^{(i)}(t) \subseteq \widehat{pa}(t) \cup \widehat{ppa}(t)\}$ for node t. Let j be the node on which we are estimating the causal effect of t. Then, for all $\widehat{pa}^{(i)}(t) \in \widehat{PA}(t)$, we compute the estimate of the causal effect of t on j, assuming the parent set of t is $\widehat{pa}^{(i)}(t)$, and include the result in the multiset Θ_t^j. In the case where we are considering Gaussian data, this simply comes from the regression of X_j on X_t and $\mathbf{X}_{\widehat{pa}^{(i)}(t)}$. The multiset Θ_t^j provides a range of possible values with which we can estimate the causal effect and provide bounds on the estimate.

However, the multi-neighborhood problem requires an algorithm which can accommodate multiple target nodes considered jointly. Nandy et al. (2017) proposed the joint IDA algorithm, a method which generalizes IDA for causal effect estimation such that researchers may estimate causal effects under multiple simultaneous interventions T, where $|T| > 1$. In order to make this extension, the combination of parent sets cannot be simply extracted from the estimated

graph using the same local criterion as in IDA. Each parent set has to be jointly valid in the equivalence class of the true graph. Additionally, rather than simply using regression with covariate adjustment as in IDA, the authors propose a new set of estimation methods, recursive regressions for causal effects (RRC) and modifying Cholesky decompositions of the covariance matrix (MCD). RRC uses a recursive formula to transform the results from local regressions into elements of the total joint effect. Since there will be multiple possible parent sets for each local regression, there will also be a multiset of total joint effects from which we can bound the effect we are attempting to estimate. In the second method (MCD), the Cholesky factorization of the covariance matrix is recursively calculated, rearranged for each target node and its parent set, and modified until the procedure is completed for each target node. Upon completion, the modified covariance matrix can be used to easily calculate the total joint effect for each target node on another node j. These procedures can be applied to each jointly valid possible parent set from DAGs which are members of the estimated equivalence class, and each estimated total joint effect may be inserted into the multiset, just as in IDA. See Sections 3.1 and 3.2 in the work of Nandy et al. (2017) for a more detailed description.

From the description of these algorithms, we can see that CML, in principal, learns a sufficient subset of the structure for causal effect estimation. Moreover, because we use coordinated learning, CML offers greater specificity for the possible parent sets of the target nodes, which in turn reduces uncertainty in the inferential procedure by reducing the size of the causal effect estimation multiset.

3.7 Extensions

To this point, we have only considered Gaussian datasets and Fisher's z-transformation of the partial correlation for CI testing. However, CML may also use different kinds of data and CI tests, which we discuss briefly in this section.

For continuous datasets, continuous conditional mutual information (Runge, 2018) is another option as a statistic for CI testing, as are kernel-based methods such as the Kernel-based Conditional Independence test (KCI-test), which is computationally efficient and easy to use (Zhang et al., 2011).

To generalize our algorithm for discrete datasets, we need only substitute an appropriate CI test for the data we are considering. The G^2 statistic, a log-likelihood ratio test which has a limiting chi-square distribution, is a typical choice (Aliferis et al., 2010a; Schlüter, 2012). Pearson's χ^2 test is another potentially suitable option. However, these tests are often impractical because they have high sample complexity (Marx and Vreeken, 2019). Conditional Mutual Information (CMI) is another test frequently used in practice, though it may lead to spurious dependencies and related practical issues depending on the threshold value (Zhang et al., 2010). Marx and Vreeken (2019) suggested a new test, the Stochastic complexity based Conditional Independence criterium (SCI), which deals with the problem of sample complexity due to limited available data by taking the size of the distribution into account within the test. Its empirical performance shows improvement in accuracy compared to alternative tests, especially for smaller sample sizes.

3.8 Conclusion

The method we present is novel and introduces a general framework which can coordinate learning even across disjoint neighborhoods, retaining the efficiency of a local method while increasing the capacity for encoding CI information in the output. The CML algorithm is perfectly suited for researchers who wish to ask causal queries for a small subset of target nodes without the computational cost of a global structure learning algorithm. Moreover, we demonstrate that, over the union of the target neighborhoods, CML is sound and complete for the novel ground truth graph we define, and also provides more structural information than if we were to consider each neighborhood individually and without coordinating the learning

between them. We accomplish this in the algorithm in multiple stages: using available Mb learning algorithms to obtain the local neighborhood nodes for each target, then recovering the skeleton using a hierarchical testing strategy from the FCI and PC algorithms which preserves between-neighborhood edges, and finally orienting as many edges as possible using the augmented rule set for the FCI algorithm. We also consider some of the primary benefits of our algorithm, such as the improvement in computational complexity where the size of the neighborhoods is small compared to the number of nodes in the entire network. In addition, we examine the potential for improved performance by using the output from CML for causal effect estimation, since it is optimally suited for existing estimation algorithms, given local learning constraints, by specifying a higher proportion of the targets' parent sets. In the next chapter, we will further demonstrate these conclusions with numerical comparisons of CML with other global and local algorithms using simulated and real-world data.

CHAPTER 4

Numerical Results and Applications

4.1 Preliminaries

Along with motivating and introducing CML in Chapter 3, we made specific claims about its advantages in comparison to other existing local and global algorithms. In the following analysis, we will empirically verify those claims using both simulated and real-world datasets.

First, we seek to demonstrate that, for the multi-neighborhood problem, CML is competitive with other algorithms with respect to learning the local structures of the target neighborhoods. Next, we will consider whether CML achieves its primary objective by optimally identifying the parent sets of the target nodes with greater accuracy and specificity than the alternatives. We also will examine the efficiency of our method to ensure its improvement in comparison to global methods while remaining competitive with a local algorithm that does not coordinate learning between neighborhoods (SNL). Finally, we will conclude the empirical analysis of the simulated datasets by examining CML in greater depth and provide preliminary advice for practical usage.

Following the empirical analysis using synthetic datasets, we will continue our numerical study by applying the structure learning methods to a gene expression dataset. Here, since the size of the dataset is large, we expect to find a significant reduction in runtime for our local methods while the fitted models are largely comparable to those formed from the global output. Moreover, we anticipate the CML algorithm to orient more edges and identify more target parents than the SNL algorithm, thus demonstrating the value of coordination.

4.2 Experimental results

4.2.1 Parameter settings

The algorithm we have discussed is flexible such that it can be used for any data faithful to the underlying DAG. For our simulations, however, we assume a linear SEM with independent Gaussian errors (denoted ε_j) for $[X_j \mid \mathbf{X}_{pa_G(j)}]$:

$$X_j = \sum_{i \in pa_G(j)} \beta_{ij} X_i + \varepsilon_j, \quad j \in [p]. \tag{4.1}$$

For our analysis, we simulate data from 13 networks provided by the **bnlearn** network repository (Scutari, 2010). The SEM coefficients are drawn from the Unif$(0.4, 0.75)$ distribution, multiplied by a sign term with equal probability of being positive or negative. We also generate error terms from $N(0, \sigma_j^2)$ with σ_j drawn from Unif$(0.1, 0.5)$. With these parameters, we randomly generate datasets of size $n \in \{500, 1000, 10000\}$. To estimate Mbs and obtain first- and second-order neighbors for each target, we use the MMPC algorithm from the **MXM** package (Tsamardinos et al., 2003a). For both the Mb estimation and skeleton recovery steps, we use significance levels $(\alpha_{Mb}, \alpha_{skel}) \in \{0.01, 0.05, 0.1\}^2$, where α_{Mb} is used for the Mb recovery algorithm and α_{skel} is the significance level for the CI tests in CML skeleton recovery. These elements together form a unique parameter setting, which is composed of a network, dataset size, Mb estimation algorithm, and significance level pair. For each setting, three datasets are generated with unique randomly drawn coefficients and error variances.

For each network, we randomly select a set of target sets of varying cardinality ranging from two to four nodes from the entire vertex set. These target sets are used in each parameter setting for the network. After generating each dataset, we ran the global PC algorithm one time, using the implementation from the **pcalg** library (Kalisch et al., 2012). For additional comparison on each target set, we also ran SNL along with CML. Recall that we may consider SNL an augmented version of global G-S applied to the subgraph over the neighborhoods of

multiple targets, and for our purposes represents the extension of any single Mb algorithm to the multi-neighborhood problem. Both SNL and CML are applied on all datasets, and once for each target set. For our analysis, we use the MMPC algorithm to estimate the first- and second-order neighbors for each target node, applying the aforementioned additional CI tests to recover spouses belonging to the first-order neighbor sets. We also filter our results by the number of nodes in target neighborhoods and the number of estimated edges to ensure that we are only considering relatively sparse, smaller neighborhoods. In the analysis below, we only use target sets such that the total number of nodes under consideration, according to the subgraph of the ground truth CPDAG, is in the set $[8, 20]$, and the number of edges is in the set $[3, 20]$. We further filter the simulations to only include $\alpha_{Mb} = 0.01$ since this was the optimal choice for MMPC across different datasets according to F1 score.

Let G' be the CPDAG of DAG G, and G'_{NB_T} be the induced subgraph of G' over NB_T for a given target set T. For each estimated setting and target set, we compare the estimated graph to G'_{NB_T}, since this represents the maximal amount of information which is not underdetermined and in principle recoverable by a global structure learning algorithm, say the PC algorithm, to which we are comparing our local method. We consider this to be the ground truth against which we measure the performance of our algorithms. We measure the subgraph of the PC algorithm output over NB_T against the ground truth as well in order to compare our local method with a global algorithm.

The summary statistics in Table 4.1 provide details about the simulations used for the empirical study. For each parameter setting, we provide results from as many as three datasets, with some settings showing incomplete results due to excessive computation time requirements for some datasets. The last four columns represent the number of target sets of each cardinality we use in our analysis. For example, in the first row, we consider nine different settings (i.e., combinations of α_{skel} and n) for the `insurance` network, which entails that we produce 27 datasets. Given the neighborhood filtering criteria stated above, we consider six target sets with $|T| = 2$, yielding 162 total simulation results.

Table 4.1: The simulations we produce for our empirical analysis. Each network from the **bnlearn** repository is listed and sorted by increasing network size. The number of settings is the amount of combinations of significance levels for the CI tests and sizes of the datasets. The remaining columns represent the total number of simulations we produce for target sets of cardinality ranging from two to four for each network ($N_{sims;|T|=x}$).

Network	p	Num. Settings	$N_{sims;\|T\|=2}$	$N_{sims;\|T\|=3}$	$N_{sims;\|T\|=4}$
`insurance`	27	9	162	108	0
`mildew`	35	9	189	189	27
`alarm`	37	9	135	189	189
`barley`	48	9	135	135	54
`hepar2`	70	9	81	81	108
`arth150`	107	9	51	39	60
`andes`	223	9	81	162	27
`diabetes`	413	9	183	158	131
`pigs`	441	8	46	82	74
`link`	724	9	162	108	108
`munin2`	1003	9	96	154	131
`munin4`	1038	9	98	143	118
`munin3`	1041	7	118	110	117

4.2.2 Partial correlation tests

In the sample version of our algorithms, we calculate sample partial correlations to infer CI relationships. The sample partial correlation $\hat{\rho}_{i,j|\mathbf{k}}$ can be calculated in various ways, and in our implementation we use the well-known function of elements from the inverted submatrix of the covariance matrix over the variables corresponding to nodes in $\{i, j\} \cup \mathbf{k}$. Upon obtaining $\hat{\rho}_{i,j|\mathbf{k}}$, we can test whether or not nodes i and j are conditionally independent

by carrying out the following hypothesis test:

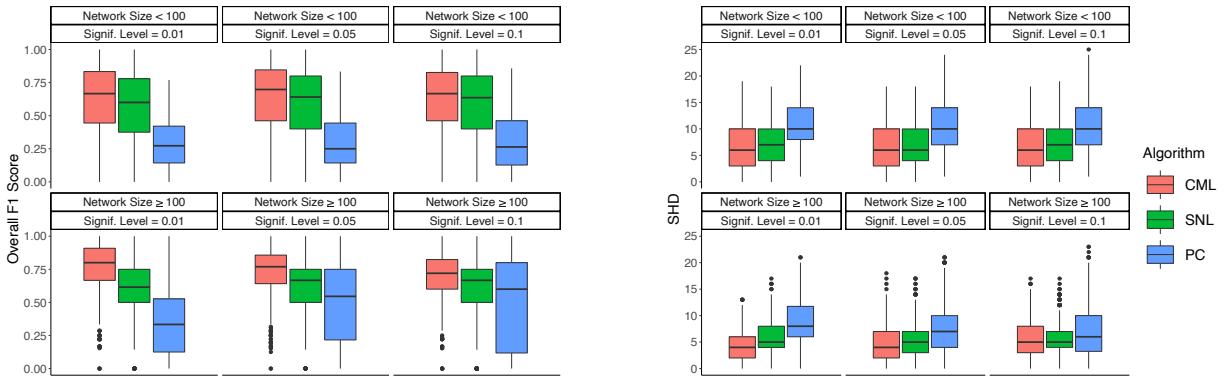
$$H_0 : \rho_{i,j|\mathbf{k}} = 0$$

$$H_a : \rho_{i,j|\mathbf{k}} \neq 0$$

To test this, we apply Fisher's z-transformation to obtain the sample statistic $Z(i, j; \mathbf{k}) = \frac{1}{2} \log \left(\frac{1+\hat{\rho}_{i,j|\mathbf{k}}}{1-\hat{\rho}_{i,j|\mathbf{k}}} \right)$, since $\sqrt{n - |\mathbf{k}| - 3}[Z(i, j; \mathbf{k})] \sim N(0, 1)$.

4.2.3 Adapting P/C algorithms for Mb learning

For the Mb recovery stage, one may use a P/C recovery algorithm, such as MMPC, along with additional CI tests to obtain the spouses belonging to each target's first-order neighbor set. In order to extend a P/C algorithm to identify spouses of target nodes, we require knowledge of the P/C sets for each node in the P/C set of the target node. This fits naturally with CML, since we must obtain second-order neighbors for the second phase of skeleton recovery. Then, there are various techniques to identify spouses from those additional P/C sets. One may simply identify spouses as those nodes in other P/C sets which are dependent on the target conditioned on the target's P/C set. This only requires one CI test for each potential spouse, and it is the strategy we use in our empirical study. Other methods involve running additional tests for a selection of proper subsets of the target's P/C set to be used as conditioning sets to verify if a potential spouse should still be retained in the estimated Mb set. This is a slightly different approach than that of the HITON-MB algorithm, which includes all second-order neighbors in the candidate Mb set before searching for separating sets to remove non-spousal nodes (Aliferis et al., 2003). The HITON search is more comprehensive, but we can afford to be more permissive here, since any false positives will be dealt with during skeleton recovery in CML. However, the second-order P/C sets are not sufficient to remove every possible kind of extraneous edge during the second phase of skeleton recovery, which entails that the local algorithms may lose some accuracy when using augmented P/C recovery algorithms, a small tradeoff for a slight reduction in complexity.



(a) (b)

Figure 4.1: Comparisons between the global and local algorithms with respect to accuracy. The distributions of (a) F1 scores and (b) Structural Hamming Distances (SHDs) across different combinations of network sizes and CI test significance levels.

4.2.4 Overall accuracy

The Overall F1 score measures how well the edge set of the estimated graph precisely conforms to that of the ground truth graph, and is given by $\text{F1} = \frac{2TP}{2TP+FP+FN+IO}$, where TP, FP, and FN are the number of true positives, false positives, and false negatives, respectively. The last term, IO, denotes the number of edges in the estimated graph which have incorrect orientation with respect to the ground truth graph. These errors are distinct from false positives and false negatives because the adjacency relations are still correct for edges with incorrect orientation. In Figure 4.1a, we compare our method to the global PC algorithm as well as to SNL applied to each target node. In terms of overall performance, the CML algorithm is superior to the other methods in both smaller (top panels, $p < 100$) and larger networks (lower panels, $p \geq 100$). For example, the median F1 score of the CML algorithm is 159% higher than that of the global PC algorithm, using significance level $\alpha_{skel} = 0.01$.

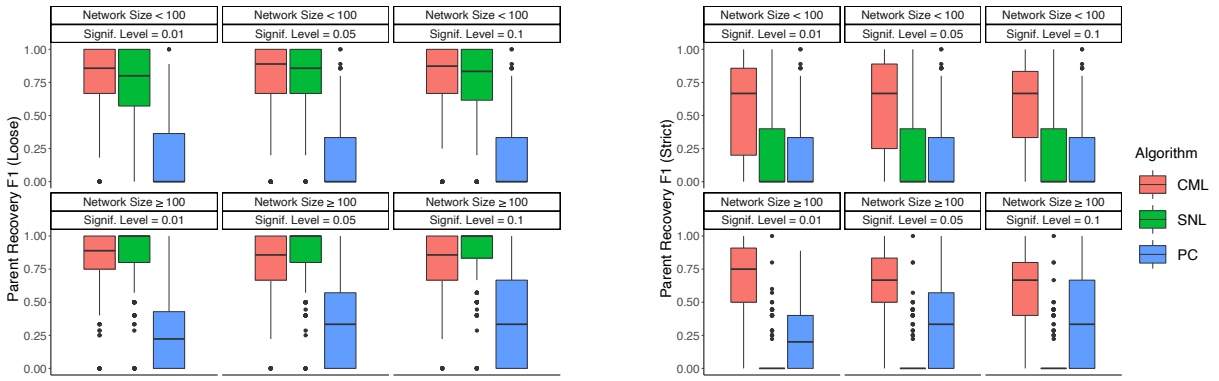
The same conclusion is drawn when using the Structural Hamming Distance (SHD) as the measure of structure learning accuracy, since the SHD measures the number of errors of the estimated graph compared to the ground truth. Figure 4.1b provides evidence of improvement

in structure learning for the local algorithms, since both CML and SNL typically have fewer errors.

Along with improvement in median performance, the local algorithms also show less variation in their performance than the global PC for larger networks, which may be observed by the smaller interquartile range in the results across the lower panels. The local algorithms, and CML in particular, show greater consistency and accuracy than the global algorithm for the multi-neighborhood problem when considering larger networks. This verifies our contention that the primary benefits of our local method will be primarily for the high-dimensional setting.

Both plots in Figure 4.1 provide summaries of accuracy measurements on the subgraph of the CPDAG over a narrowly considered set of nodes, namely NB_T from the true CPDAG. With respect to general local structure recovery, the evidence from this study clearly favors the use of local algorithms.

4.2.5 Parent recovery



(a) (b)

Figure 4.2: The distributions of the parent recovery accuracy F1 scores for different network size and significance level combinations. (a) The loose F1 score; (b) The strict F1 score.

One of the primary goals of our work is to identify as many parents of the target nodes

as possible, since this would reduce the uncertainty in parent set adjustment for estimating causal effects. Therefore, beyond structural conformity to the ground truth graph, it is of special interest to examine how well our algorithm performs in target node parent recovery. Here, the parents of a node are defined by directed edges into this node in the true CPDAG, since these are the identifiable parents with observational data alone.

In Figure 4.2, we compare the parent recovery accuracy (PRA) of the different algorithms using the F1 score of the estimated parent set with respect to the true parent set. The PRA F1 score is given by $\text{F1} = \frac{2TP}{2TP+FP+FN}$ where TP, FP, and FN are the number of true positives, false positives, and false negatives for the estimated parent set, respectively. The F1 score is considered under two different principles for counting. The loose version of the score, used in Figure 4.2a, counts an estimated undirected edge between a parent and the target node in the true CPDAG as a true positive, while the strict version, used in Figure 4.2b, counts such an edge as a false negative. Considering again the example in Figure 3.1, suppose we call the CPDAG of the DAG in Figure 3.1a G' and denote by G'_{NB_T} its induced subgraph over NB_T, where $T = \{3, 8\}$. The ground truth graph, G'_{NB_T}, will be equivalent to the graph in Figure 3.1c without the red edges. Suppose we are considering the SNL output, depicted in Figure 3.1d. For T, the identifiable parents to be recovered are $pa_{G'_{NB_T}}(3) \cup pa_{G'_{NB_T}}(8) = \{1, 2, 9\}$. The SNL output correctly identifies parent edges $(1, 3)$ and $(2, 3)$, along with placing an undirected edge between nodes 8 and 9. Under the loose version of the score, we count the undirected edge between nodes 8 and 9 as a true positive, thus assigning the SNL output a PRA F1 score of 1. Under the strict version of the score, however, the undirected edge does not count, and the SNL output receives a score of $2/3$. Distinguishing the results in two sets of scores allows us to consider how well an algorithm performs in finding possible parents (loose) as well as in providing greater specificity to the possible parent set with correctly defined invariant parent edges (strict).

In considering the results, we first note that the CML algorithm is consistently better than the PC algorithm in identifying parents of the target nodes for all settings. This shows

that the accumulation of errors in global learning will deteriorate structure learning for strictly local problems. Second, the differences between the two sets of plots in Figure 4.2 imply that the CML algorithm correctly orients far more directed edges than SNL. Consider as an example the results for $p \geq 100$ with significance level 0.01. In the loose PRA F1 score, the SNL algorithm narrowly outperforms the CML algorithm when comparing the reported percentiles. However, in the strict version of the score, the SNL algorithm performs significantly worse than the CML algorithm and even the PC algorithm. In fact, apart from a few isolated cases, all of the PRA F1 scores for the SNL algorithm are 0 under the strict definition. This is expected and confirms our discussion about Figure 3.1 since, in the SNL algorithm, the orientation of edges in one target neighborhood has no influence on the orientation of edges in another neighborhood when those neighborhoods do not share nodes. Consequently, the SNL algorithm fails to orient as many edges as the CML algorithm, which has the advantage of inferring additional orientations with ancestral information between neighborhoods.

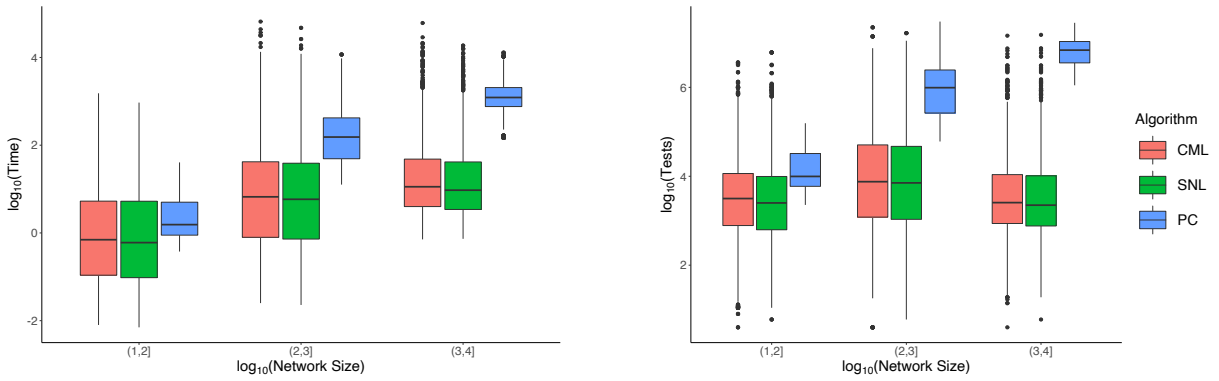
The results in Table 4.2 give further insight into the performance of the CML algorithm. The table provides the average parent recovery performance statistics across all target sets and parameter settings for all networks, which are organized in order of increasing size. As we move down the table, we can observe a general trend of improved performance for the larger networks. Moreover, this table helps give an idea of the number of parents we expect to recover with CML under the conditions of our simulations. For example, if we consider the average target set for network `munin2`, which for simplicity we will say has two parent nodes, the average recall statistic informs us that we can usually expect to recover at least one of the parents. This is under the strict definition of recovery, where undirected edges do not count.

In comparison to the global PC algorithm, the local algorithms each perform significantly better in parent recovery under the loose definition of the PRA F1 score. However, this speaks primarily to the success of the Mb recovery algorithm in identifying the correct node sets for the target neighborhoods. To reduce the uncertainty of our causal effect estimates, we need

Table 4.2: Summary of parent recovery metrics for CML averaged across all datasets, settings, and target sets used for each network in the simulation study. PRA scores are reported using strict edge counting principles (i.e., only directed edges in the estimated graph may be counted as true positives).

Network	Avg. Num. of Parents	Max. Num. of Parents	Avg. PRA Recall	Avg. PRA F1
`insurance`	1.38	5	0.36	0.28
`mildew`	1.93	8	0.67	0.71
`alarm`	2.99	9	0.54	0.59
`barley`	2.45	7	0.43	0.48
`hepar2`	4.55	11	0.24	0.30
`arth150`	1.47	7	0.34	0.32
`andes`	2.80	8	0.66	0.64
`diabetes`	3.06	8	0.63	0.67
`pigs`	3.06	8	0.65	0.67
`link`	2.56	6	0.45	0.46
`munin2`	2.16	6	0.55	0.51
`munin4`	2.70	7	0.53	0.58
`munin3`	2.73	6	0.47	0.50

to see greater specificity in defining the parent set with oriented edges into the target nodes, which we hope to achieve with the additional rules of coordinated local learning. We see this advantage in the strict definition of the score. While the SNL score completely diminishes under this score's restriction, the CML algorithm continues to significantly outperform its competitors with only mild decreases in accuracy compared to the loose score. With causal effect estimation in mind, we can clearly observe the significant benefits of a coordinated learning algorithm to estimate the parent sets for multiple target nodes.



(a) (b)

Figure 4.3: Comparisons between the global and local algorithms with respect to complexity. The distributions of (a) runtime and (b) number of CI tests used for different network sizes on a log scale.

4.2.6 Algorithm complexity

Another key contribution of local algorithms in general, and CML in particular, is the improved computational efficiency compared to global algorithms, especially as the size of the network increases. In Figure 4.3a, we find that the computing runtime of the local algorithms significantly improves in comparison to the PC algorithm, and the difference increases as the size of the network increases in magnitude. While the gains are modest for smaller networks, the median compute time drops by more than one order of magnitude for the largest networks when switching from a global to a local method. The rate of runtime increase is much slower for the local algorithms than for the global algorithm.

Additionally, since the computation of all three algorithms is dominated by CI tests, we use the total number of tests performed as a complementary metric to measure computational cost. As demonstrated in Figure 4.3b, the number of CI tests executed sharply decreases for the local algorithms when compared to the PC algorithm. For networks with more than 100 nodes, we observe a reduction of nearly two orders of magnitude in the median number of CI tests. This shows a similar pattern of improvement to the actual runtime of the complete

algorithms. We note that Figure 4.3b only includes the number of CI tests used in the skeleton recovery portions of the local algorithms, since we had difficult in extracting the exact number of tests from the Mb recovery method. However, per our discussion in Section 3.5, we can assume that the complexity of a Mb recovery algorithm is $O(|N|p)$, where N is the union of the neighborhood sets and p is the size of the network. Assuming that $|N|$ is bounded by a constant, visual inspection allows us to safely conclude that the difference in the number of CI tests between the local and global algorithms will not be substantially altered by the addition of Mb recovery CI tests, as the distribution of tests for the PC algorithm is clearly much greater than $\log p$.

4.2.7 Equivalence class accuracy

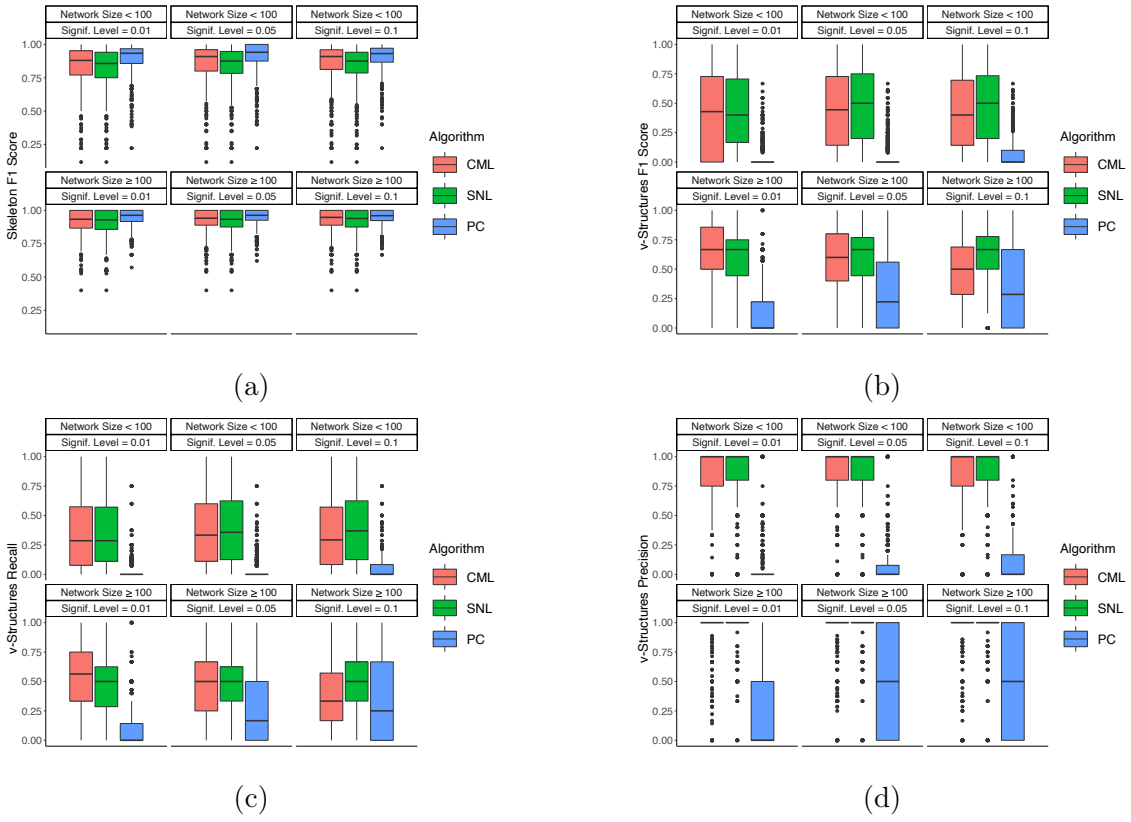


Figure 4.4: Comparison of algorithms with respect to their accuracy in recovering the skeleton and v-structures of the underlying graph. The ground truth does not include any nodes outside the target neighborhoods, even if included in the local algorithm output.

Another advantage of a local algorithm is that, compared to a global algorithm, a local method is less likely to be affected by the propagation of errors from other parts of the estimated graph. Figure 4.4 demonstrates this phenomenon in the recovery of v-structures.

Recall that the skeleton and v-structures are essential to defining an equivalence class of DAGs. Consequently, each algorithm uses a v-structure orientation step as part of its overall orientation strategy. Accurate v-structure orientation requires both the correct unshielded triple and the correct separating set, at least to the extent that it properly excludes the node

which both edges are into. The first of these requirements, an accurate skeleton, seems to be fairly evenly met by all algorithms, as seen in Figure 4.4a. In fact, the global algorithm slightly outperforms the local algorithms in skeleton recovery accuracy according to the F1 score.

However, in Figure 4.4b, we find that the local algorithms vastly outperform the global PC algorithm in v-structure orientation. This may be due to CI testing errors providing the wrong separating sets or due to conflicting v-structure orientation decisions from other nodes in the graph. Figures 4.4c and 4.4d show evidence that both false negatives and false positives are substantial components of the overall error. While CML does not always correctly identify even more than half of the v-structures in the underlying graph, as shown in Figure 4.4c, the results in Figure 4.4d confirm that the algorithm is quite precise in correctly identifying those unshielded triples which are truly v-structures. The PC algorithm, on the other hand, produces neither a good recall score nor a high precision in its estimate.

These results further bolster our motivation for using coordinated local learning. Along with the unnecessary computation requirements, global algorithms frequently lead to unnecessary errors as well. Moreover, though we are coordinating learning from different parts of the graph, which has the potential to propagate errors as well, we find minimal divergence in v-structure recovery performance from SNL to CML. Thus, CML retains the advantages of a local algorithm while having the potential for additional edge orientation, a characteristic benefit of a global algorithm.

4.2.8 Practical considerations

In the interest of practical guidance, we will take a closer look at possible sources of error for CML and attempt to draw some principles for selecting setting parameters. While each decision comes with some trade-offs, our results do provide some clear guidelines for choosing significance levels.

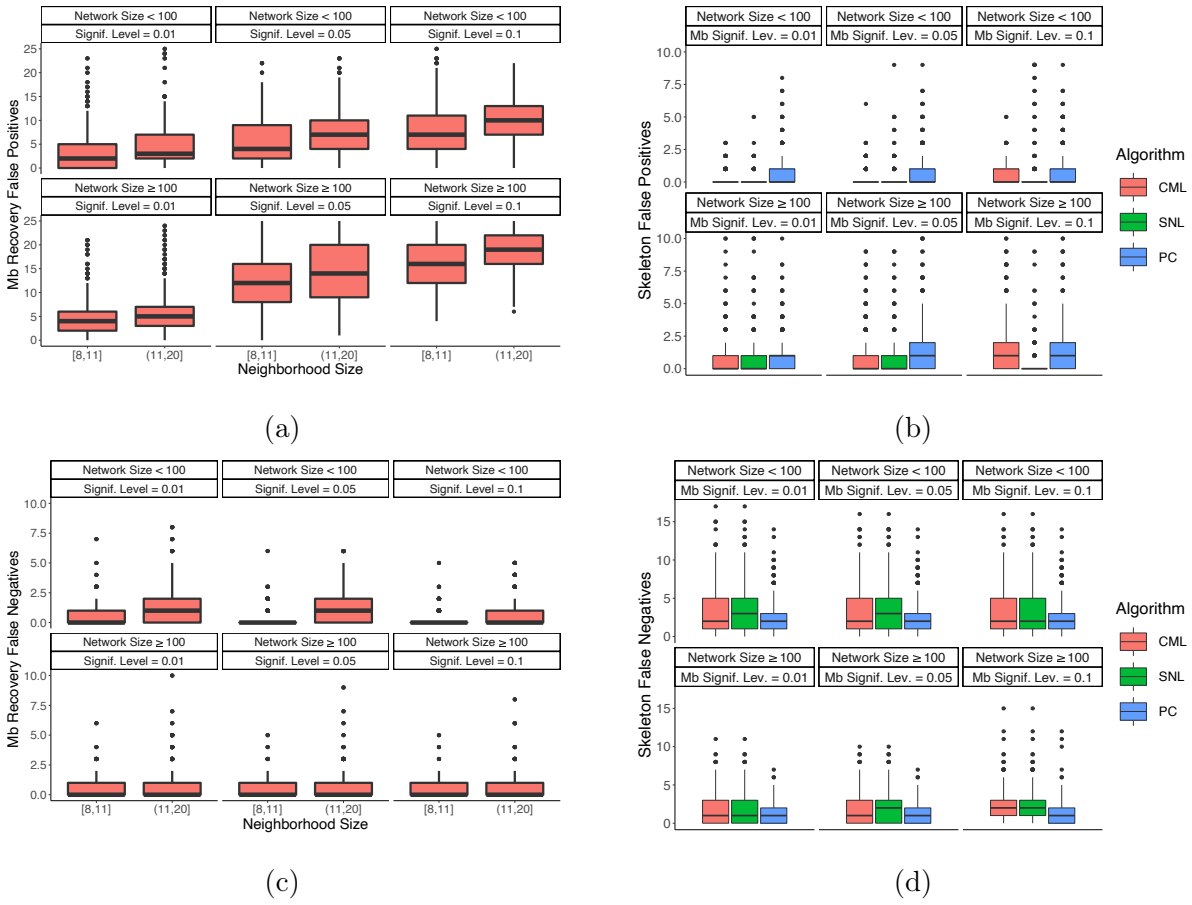


Figure 4.5: False positives in (a) Mb learning and (b) skeleton estimation. False negatives in (c) Mb learning and (d) skeleton estimation. Graphs are ordered by different combinations of network size and Mb algorithm significance level. Graphs (b) and (d) only consider simulations with $\alpha_{skel} = 0.01$. Some outliers are removed.

First, in selecting the Mb recovery algorithm significance level, we must acknowledge the asymmetry in the importance of the kinds of error which come from the algorithm output. That is, a false negative is far more consequential than a false positive with respect to selecting the Mb for the target nodes. While a false positive may be properly removed during the skeleton recovery stage of CML, there is no way to recover edges involving nodes which are not selected during Mb estimation. Thus, we can afford to be more permissive in Mb recovery, since errors from lenient selection rules may be corrected later in the algorithm. This may be

clearly seen in Figures 4.5a and 4.5b. In Figure 4.5a, for the case considering larger networks and the highest Mb learning significance level (plot in the second row, third column), we frequently observe more than 10 nodes falsely included in the Mb for larger neighborhood sets. After skeleton recovery, however, in most cases we observe only a few false positive edges for the larger Mb significance level without any major differences in performance compared to the smaller Mb significance levels. However, if Mb selection is too permissive, then the neighborhoods may become too large, which will mitigate the advantages of our local method by increasing the runtime and generating errors in defining the separating sets. If one chooses to be more permissive in Mb selection, it is prudent to balance this decision with a more restrictive threshold for skeleton estimation.

On the other hand, false negatives from Mb estimation provide a hard floor for false negatives in skeleton estimation, for the simple reason that an edge cannot be present where at least one of the nodes is not included in the graph. In Figure 4.5c, we did not encounter many false negatives across all theresholds, but wherever there are false negatives in Mb recovery, we are guaranteed at least that many false negatives in skeleton recovery, which is reflected in the higher false negative values found in Figure 4.5d.

The selection of significance thresholds depends on many factors such as the research domain, the characteristics of the particular use case, and the Mb algorithm. While each of these considerations demand considerable attention, the present discussion elucidates the balance which must be held in the relative difference between the significance thresholds. The Mb recovery significance level should be set to achieve a high recall, and with less of an emphasis on precision. The skeleton recovery significance level must aim for overall accuracy, and should be properly determined to offset the errors from Mb recovery.

4.3 Discussion

For estimating the causal structure in local neighborhoods and facilitating the estimation of causal effects of specified target nodes, the CML algorithm is a sound algorithm with demonstrable empirical benefits. Compared to existing methods, this algorithm is more efficient and scalable while maintaining a degree of accuracy comparable to or better than global methods. Though in this work we only conducted empirical analysis on Gaussian data, this method is more broadly applicable to different kinds of data and DAG models. A future direction of research is to observe the performance of our algorithm with simulated and real-world data from different distributions. Along similar lines, another potentially fruitful research direction would consider modifications to our algorithm for the use of experimental data with interventions.

4.3.1 Code

The code written to produce this simulation study is located in two online repositories for review. The first repository contains the source code for the R package which implements the CML and SNL algorithms in C++ using **Rcpp**. The package may be found at `http://github.com/stephenvsmith/CML`. The other repository contains the R script files and instructions for reproducing the simulations of our study. These scripts may be found at `http://github.com/stephenvsmith/CML-Scripts`.

4.4 Gene expression data

In this section, we apply our algorithm to the data collected in (Yao et al., 2021) which profiles approximately 1.3 million cells of the adult mouse isocortex and hippocampal formation (HPF). The isocortex is part of the cerebral cortex, covering many sensory functions and primary and secondary motor areas. HPF is a structure composed of multiple parts with

interconnected neurons forming a network related to and foundational for learning, memory, spatial navigation, and emotional regulation.

We apply our method to model local structures in a network where each variable represents the gene expression level of a particular cell. The data provides information from isolated cells processed for RNA sequencing using SMART-Seq v4. Only a subset of the available data will be used for our analysis, and the cells are conveniently catalogued over the isocortex and HPF in a manner which is strongly correlated to the spatial relationship of the cells, making one of these subsets a natural choice. As in the work of Ruiz et al. (2022), we consider the glutamatergic cells from the primary visual cortex, which contain a wide and diverse selection of cell types, and we further limit our data to cells for which injection materials are not specified. This selection reduces the number of cells from 74,973 to 7,159. Furthermore, we remove genes for which less than half of the cell expression levels were nonzero, of which many have expression measurements of exactly 0 for all cells. This reduces the number of genes from 45,768 to 10,012, which are taken across the 7,159 cells.

4.4.1 Data setup

In order to further reduce the number of genes under consideration, we calculate the coefficient of variation (CV) for each gene by measuring the ratio of the standard deviation to the mean for the expression levels across all cells. The median CV of all genes is 1.06. We then filter our dataset by removing all variables with CV less than 1.5, which leaves 1,883 genes remaining. Next, we take a random sample of 50% of cells for use in our analysis, so that our final dataset is 3,579$\times$1,883.

To identify target genes, we begin by applying the MMPC algorithm to identify the P/C set of each remaining gene. Figure 4.6 gives an idea of the size of neighborhoods we will be considering for our algorithm, and we use these results to guide our choice of targets so that we limit the size of the target neighborhoods.

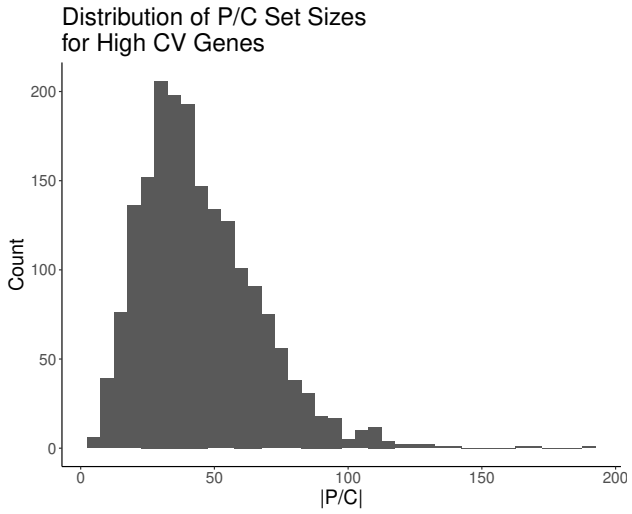


Figure 4.6: The distribution of sizes for estimated P/C sets for genes with the highest CV values. We use the MMPC algorithm and a threshold of $\alpha_{Mb} = 0.001$.

As demonstrated in Figure 4.6, there are very few genes with small P/C sets, and many of the P/C sets are very large. In selecting our targets, we choose from a pool of nodes with estimated P/C sets of cardinality less than 15. Furthermore, because we are more interested in distant or disjoint neighborhoods, we remove any nodes from the target candidate pool which contain another potential candidate within their neighborhood. Upon completing this step, we obtain our pool of qualified potential target nodes. In order to judge performance with different target set sizes, we take a random selection of two sets of targets for each target set size $|T|$, where $|T| \in \{2, 3, 4, 5\}$.

4.4.2 Parameter settings

Using a similar procedure as we did for the synthetic data, we will examine different parameter combinations to compare the performance between the algorithms, and we refer to these unique parameter combinations as settings. Setting parameters for the local algorithms still consist of the significance thresholds for the Mb and skeleton recovery algorithms, α_{Mb} and α_{skel}, and the maximum potential separating set size, ℓ_{max}. Each setting is defined by

these parameters. Since CML and SNL are in the same class of algorithms, they use the same settings $\theta_{\text{local}} = (\alpha_{skel}, \ell_{max}, \ell_{max})$, where $\theta_{\text{local}} \in \{10^{-5}, 10^{-4}\} \times \{10^{-3}, 10^{-2}\} \times \{3, 5\}$. However, because the PC algorithm does not use Mb learning for pre-processing, it uses its own parameter settings given by $\theta_{\text{global}} = (\alpha_{skel}, \ell_{max})$, where $\theta_{\text{global}} \in \{10^{-6}, 10^{-5}\} \times \{3, 5\}$. This yields 20 total combinations of algorithms and settings.

4.4.3 Cross-validation procedure

Since we use cross-validation (cv) to identify an optimal algorithm and setting, we randomly assign the 3,579 cells to 10 different sets of roughly equivalent size. Our cv procedure uses 10 folds, where each fold will be a held-out set used as the "testing data" to evaluate the model trained on the other nine folds. The results we present are generated using the held-out fold score on the "training set" models for all 10 folds, which provide the basis of comparison for the different algorithms across various settings.

For each fold of our cv procedure, we run the algorithms using data from the remaining nine folds, or the "training data," in order to learn the local structures around the target nodes, and then we calculate the model performance on the "testing data." That is, let $\mathcal{D}$ represent the entire dataset and suppose we are considering setting i. We partition the data $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_9, \mathcal{D}_{10})$. For cv fold j, we define the "training set" as $\mathcal{D}_{\text{train};j} = \mathcal{D}_{-j} = (\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{j-1}, \mathcal{D}_{j+1}, \ldots, \mathcal{D}_9, \mathcal{D}_{10})$ and the "testing set" is $\mathcal{D}_{\text{test};j} = \mathcal{D}_j$.

Using $\mathcal{D}_{\text{train};j} = (\mathbf{X}_1^{(-j)}, \mathbf{X}_2^{(-j)}, \ldots, \mathbf{X}_{p-1}^{(-j)}, \mathbf{X}_p^{(-j)})$, where p is the number of genes we are considering and $\mathbf{X}_k^{(-j)}$ contains the expression levels of gene k for all cells excluding those in the jth fold, we run CML and SNL for target nodes T and obtain the estimated local structure, denoted $G_T^{(j)}$. In addition, we run the PC algorithm once for each fold and obtain the estimated structure $G^{(j)}$ over the entire node set, from which we extract the relevant subgraphs $G_T^{(j)}$ for each node set T. For each algorithm, we then collect the estimated parent set in $G_T^{(j)}$ for each target node, $\widehat{pa}_{G_T^{(j)}}(T) = \cup_{t_m \in T} \widehat{pa}_{G_T^{(j)}}(t_m)$, which we use to estimate the

SEM. For simplicity of notation, we also write $\widehat{pa}_T^{(j)}(t_m) := \widehat{pa}_{G_T^{(j)}}(t_m)$.

In order to compare the results of our estimates, we use the cell-adjusted log-likelihood for the held-out fold, denoted $\mathcal{D}_{\text{test};j} = (\mathbf{X}_1^{(j)}, \mathbf{X}_2^{(j)}, \ldots, \mathbf{X}_{p-1}^{(j)}, \mathbf{X}_p^{(j)})$, where $\mathbf{X}_k^{(j)}$ contains the expression levels of gene k for cells in the jth fold. To compute this metric, we first estimate the SEM models for each of the target nodes $t_m \in T$ using OLS regression of $\mathbf{X}_{t_m}^{(-j)}$ on $\mathbf{X}_{\widehat{pa}_T^{(j)}(t_m)}^{(-j)}$, or the gene expression levels for all cells excluding those in fold j for the parents of target t_m in the estimated graph for target set T, which we rewrite as $\mathbf{X}_{m;T}^{(-j)}$. This regression gives us estimates $\hat{\beta}_{0m;T}^{(-j)}$, the intercept of the model, and $\hat{\boldsymbol{\beta}}_{m;T}^{(-j)}$, the model coefficients. We also obtain the standard error,

$$\hat{\sigma}_{m;T}^{(-j)} = \left(\frac{1}{n_{-j} - |\widehat{pa}_T^{(j)}(t_m)| - 1} ||\mathbf{X}_{t_m}^{(-j)} - \hat{\beta}_{0m;T}^{(-j)} \mathbf{1}_{n_{-j}} - \mathbf{X}_{m;T}^{(-j)} \hat{\boldsymbol{\beta}}_{m;T}^{(-j)}||_2^2 \right)^{1/2}, \tag{4.2}$$

where n_{-j} denotes the number of cells in $\mathcal{D}_{\text{train};j}$ and $\mathbf{1}_\eta$ is a vector of ones of length $\eta \in \mathbb{R}$. If there are no estimated parents, then we report the normalized total sum of squares. Using these results, we can compute the cell-adjusted log-likelihood on the held-out fold as

$$\ell\ell_T^{(j)} = \frac{1}{n_j} \sum_{m=1}^{|T|} \left[\frac{-n_j}{2} \log\left(2\pi(\hat{\sigma}_{m;T}^{(-j)})^2\right) - \frac{1}{2(\hat{\sigma}_{m;T}^{(-j)})^2} ||\mathbf{X}_{t_m}^{(j)} - \hat{\beta}_{0m;T}^{(-j)} \mathbf{1}_{n_j} - \mathbf{X}_{m;T}^{(j)} \hat{\boldsymbol{\beta}}_{m;T}^{(-j)}||_2^2 \right], \tag{4.3}$$

where n_j is the number of cells in $\mathcal{D}_{\text{test};j}$. We can now use the distribution of cell-adjusted log-likelihood across all cv folds to determine the optimal settings and compare the different algorithms.

However, this basis of comparison is limited in scope, since it automatically excludes nodes which are connected to the targets by an undirected edge. Consequently, in addition to using the estimated parents to evaluate the local learning methods, we also find the maximum-sized set of jointly valid parents, or the maximized parent set, from the estimated graph and recalculate the cell-adjusted log-likelihood with the new parent set. This helps broaden the perspective of comparison, since both results provide a range of possible values given the graphical output.

The primary challenge in this task is to find the set of possible parents which are jointly valid. Within the estimated equivalence class, we can only use those graphs which do not introduce any new v-structures other than those already specified by the directed edges in the output. That is, a subset of the undirected edges in the skeleton which were not oriented by the algorithm's rules can only be directed and considered jointly valid if they do not form a new v-structure. In the following discussion, we will give an outline of the steps we take to obtain the maximized parent set.

Algorithm 12 Identifying the maximized parent set

Input: graph G, target set T
Let N_v be the number of v-structures in G
Identify the set of possible parent ordered pairs $\widehat{ppa}_{\text{pair}}(T)$; set $N_{ppa} = |\widehat{ppa}_{\text{pair}}(T)|$
for $i = N_{ppa}, N_{ppa} - 1, \ldots, 0$ **do**
    for every $ppa_{max} \subseteq \widehat{ppa}_{\text{pair}}(T)$ of size i **do**
        Set $\tilde{G} \leftarrow G$
        for all $(j, t) \in ppa_{max}$ **do**
            Orient $j \rightarrow t$ in $\tilde{G}$
        end for
        if the number of v-structures in $\tilde{G}$ is N_v **then**
            Output: $\tilde{G}$
        end if
    end for
end for
Output: Original graph G

After the initial estimation of the local structure, we obtain the initial parent set of ordered pairs $\widehat{pa}_{\text{pair}}(T) := \{(i, t) \in N \times T : i \rightarrow t\}$. The maximized parent set will be a subset of the union of the parent set and the possible parents set $PPA(T) = \widehat{pa}_{\text{pair}}(T) \cup \widehat{ppa}_{\text{pair}}(T)$, where $\widehat{ppa}_{\text{pair}}(T) := \{(i, t) \in N \times T : i - t\}$. Before beginning our search, we identify all

v-structures in the estimated graph. The basic algorithm for the search begins by directing all nodes in $\widehat{ppa}_{\text{pair}}(T)$ into their respective target nodes, then checking to determine if any new v-structures are created. If not, then we return the graph with the new parent set, denoted $\tilde{G}$. If this step does create new v-structures, then we try again for subsets of $\widehat{ppa}_{\text{pair}}(T)$ with cardinality decremented by one. We repeat this procedure until we find a new graph with the same v-structures or we run out of possible parent subsets to try. We present the pseudocode for this strategy in Algorithm 12.

4.4.4 Modeling performance

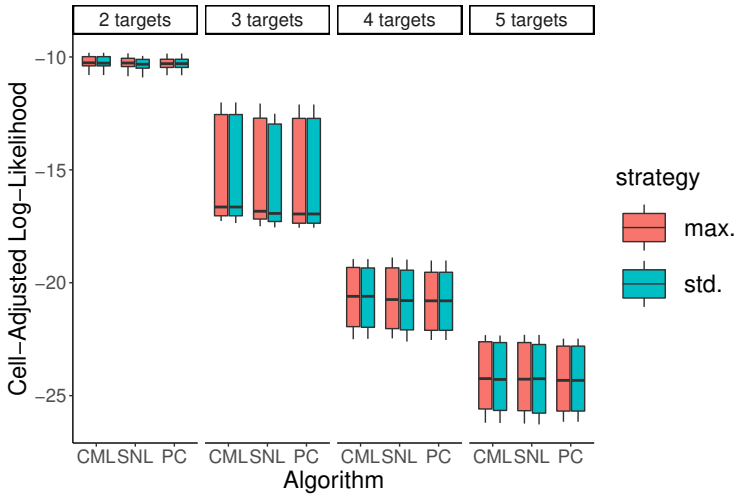


Figure 4.7: Modeling performance across different target set sizes for multiple algorithms. Side-by-side boxplots provide the results under different strategies for identifying the parent sets of the target nodes. The maximized parent set strategy (max.) identifies the largest jointly valid set of parents, and the standard strategy (std.) uses the parent set from the estimated output.

We may now consider the results produced by our modeling procedure according to cell-adjusted log-likelihood. In what follows, we compare all three kinds of algorithms across different combinations of significance levels and target set sizes. Each plot also contains

side-by-side boxplots displaying the results under two different strategies for target parent identification. The standard (std.) strategy merely uses the $\widehat{pa}_T^{(j)}(T)$ set for modeling, while the maximized parent set (max.) strategy uses the parent set obtained from Algorithm 12, which may include nodes connected to one of the targets by an undirected edge in the structure learning algorithm output.

It is difficult to discern any differences between the algorithms in Figure 4.7. Clearly, we may at least conclude that the local algorithms are competitive with the global algorithm. Moreover, we find that the variability of the score tends to increase with the number of nodes in the target set, except for the case where $|T| = 3$. However, we must go further to assess the performance of these methods.

4.4.5 Parent recovery

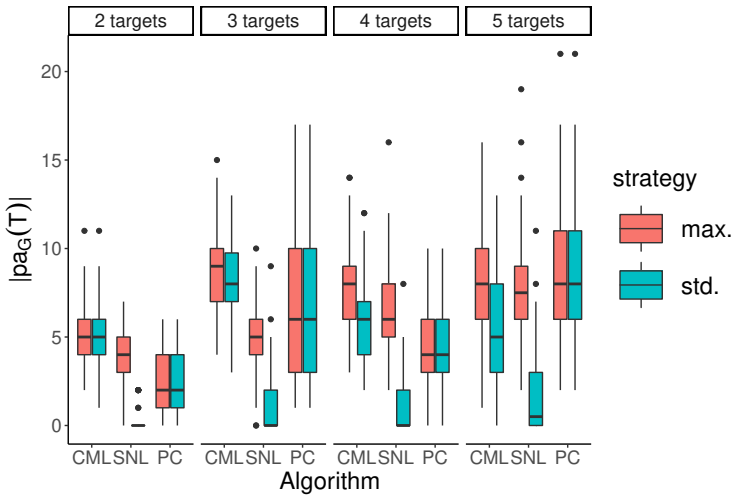


Figure 4.8: Distribution of the number of estimated parents for each algorithm by target set size and parent set identification strategy.

By analyzing the parent recovery results for each algorithm in Figure 4.8, we can gain further insight into the performance as it relates to edge orientation. For example, if we consider the SNL algorithm, we observe the detrimental effects of a lack of coordination.

Under the standard strategy for parent identification, the SNL algorithm rarely identifies any target parents, which is in stark contrast to the maximized parent set strategy, which generally competes well with the other two algorithms. Yet, this variability corresponds to greater uncertainty in our model estimation algorithms. The CML algorithm on the other hand, usually shows only minor differences in the results between the two strategies, highlighting the value of coordinated learning across target neighborhoods. The results are also competitive with those of the global PC algorithm in most cases, maintaining fairly low variability in the number of parents and generally selecting nearly as many parent nodes.

4.4.6 Runtime

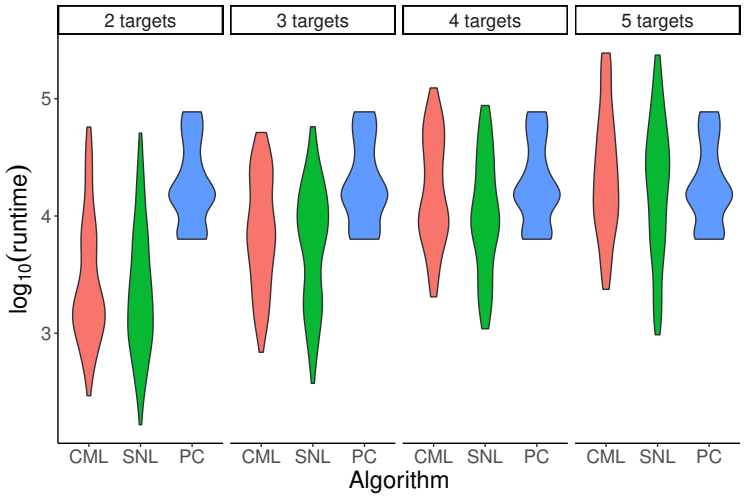


Figure 4.9: Runtime comparison for the algorithms across different target set sizes. Runtime is measured on a log scale.

Local algorithm runtime is heavily dependent on factors such as the targets chosen, the topology of the graph, and the setting threshold parameters. In Figure 4.9, we observe a high variability in the local algorithm runtime results spanning multiple orders of magnitude. In addition, we find that the runtime advantage of the local algorithm deteriorates as the number of targets increases. Extrapolating from the distribution of P/C sets presented in Figure 4.6,

only a few target nodes are required before the first- and second-order neighbor sets contain most of the node set. Therefore, due to large neighborhood sizes and the proposed best use case for our algorithm where the graph is sparse, we confine our attention to the case where two or three targets are selected. Then, we may observe significant improvement for the local algorithms in most instances, though we still observe a high variability in runtime. On this basis, we do find evidence, albeit weaker than that of the simulation study, for improvement of the local algorithms compared to the global PC.

4.4.7 Discussion

Though we cannot say that the findings from the gene expression dataset are as strong as those from the synthetic data, we still find evidence for some of the theoretical claims we make in Chapter 3 to complement the simulation results in Section 4.2. The summary results in Table 4.3 show the quality of the CML algorithm in model estimation, since CML produces the top two results in cell-normalized log-likelihood for each target set size, though it must be noted that all of the other scores are all well within one standard deviation. Furthermore, as we observed in the discussion of Figure 4.8, we find that CML improves significantly in parent orientation in comparison to SNL, further bolstering our claim of the importance of neighborhood coordination for edge orientation. Finally, apart from the larger target sets which include most of the genes in our dataset, the local algorithms still provide competitive or improved runtime compared to the global PC algorithm. This analysis provides evidence from a real-world dataset of the main advantages of coordinated learning, and sets the foundation for further exploration in applying our method to other datasets.

Algorithm	α_{Mb}	α_{skel}	ℓ_{max}	$\lvert T\rvert$	$\log_{10}$(runtime)	$\lvert NB_T\rvert$	$\lvert pa(T)\rvert$	$\lvert pa_{max}(T)\rvert$	$\ell\ell_T^{(j)}$ (sd)	$\ell\ell_T^{(j)}$ max. (sd)
CML	1e-04	1e-02	5	2	4.28	23.35	5.10	5.65	-10.24 (0.27)	-10.24 (0.27)
CML	1e-05	1e-02	5	2	3.14	17.15	4.75	4.95	-10.24 (0.28)	-10.23 (0.28)
PC	NA	1e-05	3	2	4.50	52.35	3.85	3.85	-10.26 (0.28)	-10.26 (0.28)
PC	NA	1e-06	3	2	4.18	24.80	2.30	2.30	-10.28 (0.28)	-10.28 (0.28)
SNL	1e-04	1e-02	3	2	3.88	29.85	0.40	4.05	-10.33 (0.28)	-10.28 (0.26)
SNL	1e-05	1e-03	3	2	2.96	19.70	0.10	4.10	-10.35 (0.28)	-10.25 (0.27)
CML	1e-05	1e-03	5	3	4.03	31.55	8.25	8.50	-15.78 (4.34)	-15.74 (4.36)
CML	1e-05	1e-02	3	3	3.65	38.00	8.10	9.30	-15.8 (4.31)	-15.77 (4.33)
PC	NA	1e-05	3	3	4.50	77.75	8.55	8.55	-15.95 (4.39)	-15.95 (4.39)
PC	NA	1e-05	5	3	4.71	48.40	7.35	7.35	-15.96 (4.38)	-15.96 (4.38)
SNL	1e-04	1e-03	3	3	4.06	57.90	2.05	5.70	-16 (4.19)	-15.85 (4.33)
SNL	1e-04	1e-02	3	3	4.09	57.90	1.15	3.55	-16.03 (4.19)	-15.97 (4.27)
CML	1e-05	1e-03	3	4	3.78	52.60	6.75	9.05	-20.65 (1.36)	-20.63 (1.38)
CML	1e-05	1e-02	5	4	4.25	43.15	4.90	7.05	-20.65 (1.36)	-20.64 (1.37)
PC	NA	1e-05	3	4	4.50	143.15	7.15	7.15	-20.78 (1.37)	-20.78 (1.37)
SNL	1e-04	1e-03	3	4	4.14	79.25	2.00	7.40	-20.78 (1.44)	-20.71 (1.41)
SNL	1e-05	1e-03	3	4	3.47	52.60	0.80	6.80	-20.8 (1.43)	-20.7 (1.39)
PC	NA	1e-06	3	4	4.18	78.20	3.95	3.95	-20.81 (1.38)	-20.81 (1.38)
CML	1e-05	1e-02	5	5	4.27	53.40	3.85	6.60	-24.19 (1.55)	-24.13 (1.53)
CML	1e-04	1e-03	5	5	4.94	72.70	7.85	10.25	-24.2 (1.55)	-24.18 (1.54)
SNL	1e-04	1e-03	3	5	4.57	114.80	3.95	9.20	-24.24 (1.55)	-24.22 (1.56)
PC	NA	1e-05	3	5	4.50	140.40	13.10	13.10	-24.27 (1.48)	-24.27 (1.48)
SNL	1e-05	1e-03	3	5	3.86	70.40	1.60	7.60	-24.27 (1.54)	-24.23 (1.56)
PC	NA	1e-05	5	5	4.71	81.35	9.20	9.20	-24.28 (1.49)	-24.28 (1.49)

Table 4.3: Summary statistics for different settings and target set sizes. Results are averaged across the 10 cv folds for both target sets in the category. Only the top two settings for each algorithm with respect to $\ell\ell_T^{(j)}$ are provided for each target set size. The results are given in descending order by $\ell\ell_T^{(j)}$ for each target size category. The number of estimated parents and the cell-adjusted log-likelihood using the maximized parent set strategy are denoted by $\lvert pa_{max}(T)\rvert$ and $\ell\ell_T^{(j)}$ max., respectively.

CHAPTER 5

R Package

5.1 Introduction

In the previous chapters we introduce, motivate, and demonstrate the possibilities for local structure learning with respect to a novel setting given by the multi-neighborhood problem. As graphical modeling becomes increasingly popular for researchers in a number of fields, algorithms and methods must be designed and implemented which are easy to use and fulfill specific research needs while remaining sound, consistent, and scalable. In this section, we transition from describing the design and performance of CML, our algorithm for coordinated learning across multiple target neighborhoods, and begin to discuss the implementation of our method in a new R package intended for further application and research usage.

Corresponding to the growth of interest in graphical models for machine learning and statistics, there are a substantial number of software libraries to implement the growing number of learning algorithms. Structure learning packages such as Tetrad (Ramsey et al., 2018) and visualization software such as DAGitty (Textor et al., 2017) are part of a growing ecosystem of libraries and tools designed for structure learning and causal reasoning with graphical models.

In this chapter we present **CML**, a new R (R Core Team, 2020) package developed to implement a framework for coordinated local structure learning for different kinds of datasets in the context of a multi-neighborhood problem, thus contributing a novel, complementary approach alongside existing software.

5.2 Background

While there are many software developments related to causal DAG learning in other languages (Ling et al., 2022b; C. Squires, 2018), we will confine our discussion here to contributions written in R. The **bnlearn** (Scutari, 2010) and **pcalg** (Kalisch et al., 2012) packages are some of the most well-known for structure learning, since they provide a large selection of different classes of algorithms, including constraint-based and score-based global algorithms as well as local algorithms for P/C or Mb learning. In addition, the **pcalg** library provides inference procedures such as IDA and joint-IDA. The **bnlearn** package also uses the **Rgraphviz** (Hansen et al., 2020) package to provide additional plotting options for visualizing graphical models, including the output from structure learning algorithms. These packages also provide options for different kinds of data and CI tests, an indispensable feature of any structure learning package intended for practical use. For local learning, Lagani et al. (2017) developed the **MXM** package to specialize in implementing their group's feature selection algorithms, especially the statistically equivalent signature (SES) algorithm. The authors also implemented MMPC and other algorithms related to local structure learning.

In response to increased interest in biological and other applications with high-dimensional datasets, Aragam et al. (2019) introduced the **sparsebn** package to address scalability concerns and implement new methods for structure learning and parameter inference, replacing previous options which are too slow for more demanding datasets. This package does not add anything new in terms of graph visualization, since it is compatible with existing graph storage and visualization packages in three different languages. For a dataset with fewer observations, many variables, and potentially some interventional data, this package provides new methods and implementations to address challenging problems.

In addition to these, one may also consult the **deal** (Boettcher and Dethlefsen, 2003), **causaleffect** (Tikka and Karvanen, 2017), and **ParallelPC** (Le et al., 2015) packages for additional methods to use for causal inference with graphical models.

5.3 The CML package

Since we introduce an algorithm for the multi-neighborhood problem, a novel perspective for structure learning, we present **CML**, a package we developed to fill a gap in DAG learning packages. This package fills a mediating position between local and global methods, since CML is properly a local algorithm but coordinates learning from different parts of the graph in a manner similar to a global algorithm. The **CML** package is available on Github, and may be downloaded using the **devtools** library:

```
devtools::install_github("stephenvsmith/CML")
```

The CML package is built using **Rcpp** (Eddelbuettel et al., 2023a), a package which serves as an interface between R and C++ for greater computational speed. Additionally, the package uses some of the data structures and methods found in the Rcpp Armadillo library (Eddelbuettel et al., 2023b). The package also imports **bnlearn** for some internal functions, along with the **MXM** package, both of which provide implementations for Mb estimation procedures. In addition, the package carries the adjacency matrix for the `asia` network (Lauritzen and Spiegelhalter, 1988), as well as a simulated Gaussian dataset for the network with $n = 500$.

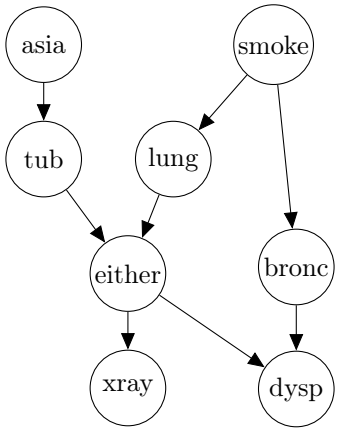


Figure 5.1: The `asia` network from **bnlearn**

To illustrate the basic usage of the package, we will demonstrate its functionality with the population version of the algorithm for the `asia` network, using the true graphical structure to identify Mbs and the d-separations in the true DAG as a CI oracle for the skeleton recovery stage. Figure 5.1 presents the `asia` DAG, and Figure 5.2 provides both the subgraph of the true graph and its estimate from the population CML algorithm for target nodes "asia" and "either." We also provide the code snippet to produce the plots in Figure 5.2. The first three lines load the library, the data matrix containing the adjacency matrix of `asia`, and the simulated Gaussian data from the network. Then, we run the algorithm using the `cml` function, using the true DAG adjacency matrix as input to ensure we use the population version. Finally, we plot the algorithm output and the subgraph of the true DAG over the target neighborhoods using the `plotOutput` function, which utilizes the functionality of **Rgraphviz**.

```
library(CML)
data("asiaDAG")
data("asiadf")
local_est <- cml(true_dag = asiaDAG,targets = c(1,6),
                 node_names = colnames(asiaDAG),verbose = FALSE)
plotOutput(local_est,asiaDAG)
```

In this example, we apply the CML algorithm to learn the local structures around nodes 1 and 6 ("asia" and "either"). Even though we use a CI oracle for the population version of the algorithm, these plots are not identical since, in principle, the edge between "asia" and "tub" may be in either direction based on the CI information available. The CML output represents the maximum amount of information which we may recover from observational data. Indeed, the CML algorithm recovers the complete skeleton and the v-structures for the subgraph over the target neighborhoods, as well as the compelled edge between "either" and "xray".

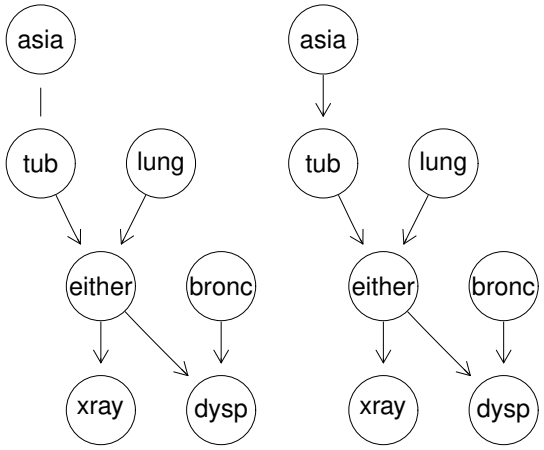


Figure 5.2: The graph on the left is the output of the CML algorithm for target nodes "asia" and "either", and the graph on the right is the subgraph of the `asia` DAG over the neighborhoods of the target nodes.

In practice, we will not have access to a CI oracle, so this exercise serves only as a proof of concept and a simple illustration of our implementation. The primary use of the package will require an input dataset, which we consider next.

5.3.1 Local structure learning from data

Calling the sample version of the `cml` method follows similar usage rules as the population version, except with additional parameters and the substitution of a dataset for the true DAG adjacency matrix.

```
cml(data = asiadf,targets = c(1,6),node_names = colnames(asiaDAG),
    lmax = 3,tol = 0.01,mb_tol = 0.05,method = "MMPC",
    test = "testIndFisher",verbose=FALSE)
```

The main arguments are `data`, `targets`, `method`, and `test`. The fuller declaration and exposition of the arguments are as follows:

- `data`: A data.frame or matrix containing observational data to be used for structure learning. The default is `NULL` in case we are considering the population version of the algorithm.

- `true_dag`: A matrix containing the adjacency matrix for the true DAG. The default is `NULL` in case we are considering the sample version of the algorithm. The purpose of this argument is mainly diagnostic and for testing purposes, not for practical application, since it allows d-separations in the graph to replace CI tests.

- `targets`: A vector containing the nodes around which we will coordinate local structure learning.

- `node_names`: A vector of strings with the names of the nodes being used in an order corresponding to that of the columns in `data` or `true_dag`.

- `lmax`: The maximum possible size of a potential separating set (ℓ_{max}). The default is 3.

- `tol`: The significance level we use during the skeleton stage of CML (α_{skel}). The default is 0.05.

- `mb_tol`: The significance level we use for the Mb recovery algorithm (α_{Mb}). The default is 0.01.

- `method`: A string with the name of the Mb recovery algorithm being used. The default choice is the MMPC algorithm.

- `test`: A string with the name of the type of CI test to be used. The default test uses Fisher's z-transformation of the partial correlation.

- `verbose`: A boolean determining whether or not to print diagnostic output. The default is `FALSE`.

The method returns a list containing the adjacency matrix of the output graph, along with various statistics including the number of CI tests, the computation times for the different stages, the frequencies of usage for the orientation rules, and the input data's mean and covariance prior to normalization and the initiation of the algorithm. The list also includes a vector of the nodes in the target neighborhoods, a list of all estimated Mbs, and the separating sets from the skeleton estimation stage.

We can consider the results of the sample version using the `plotOutput` command, which provides the output displayed in Figure 5.3.

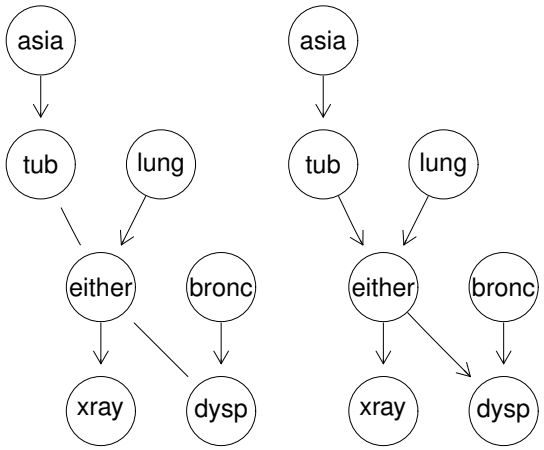


Figure 5.3: Plot output from the sample version of `cml`. The graph estimated by CML is on the left, while the subgraph over the true neighborhoods of the target nodes is on the right.

As we would expect, the sample output contains errors. Notably, both v-structures are missing from the estimated graph. However, one correctly directed edge from each of the v-structures is present in the CML output. Moreover, we also have "asia"→"tub" in the CML output, though we previously stated that the edge between "asia" and "tub" should be undirected in our discussion of Figure 5.2. These errors are related and illustrate important practical features of our implementation. According to our CI test procedure, we mistakenly identify v-structure ("asia","either","tub"). However, this conflicts with the v-structure ("tub","lung","either"), which is correctly identified. Since "tub" and "either" are

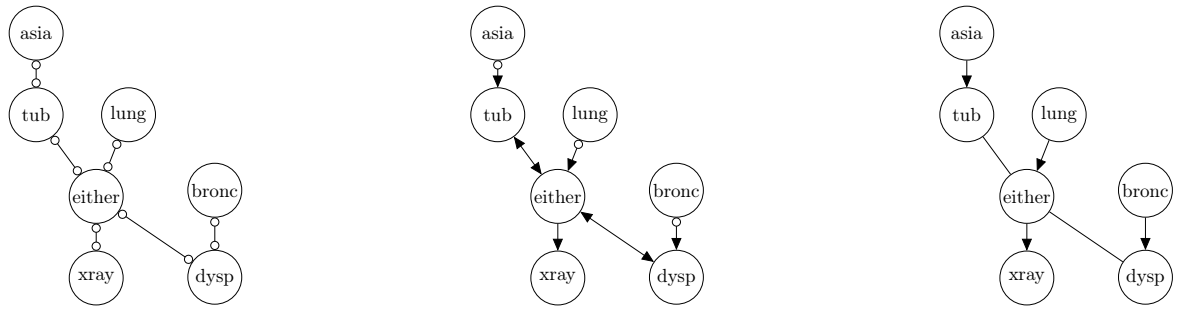


Figure 5.4: The graphical output at different times during the execution of the sample version of `cml` for the `asia` network.

nodes in the same neighborhood, we leave this edge undirected in the final output, following Remark 10. A similar line of reasoning explains the undirected edge between "either" and "dysp" in the sample output. An illustration of the stages of the sample procedure is given in Figure 5.4.

The skeleton plot with variant marks in Figure 5.4a shows the result of the sample CML algorithm after the Mb and skeleton estimation stages. To this point, there are no errors which are visible in the graph. However, there are errors in the definitions of the separating sets, which become apparent after initial edge orientation as displayed in Figure 5.4b. The bidirected edges appear due to the faulty separating sets, which lead us to conclude the presence of v-structures which are inaccurate. Another feature of the output at this stage is that, while some of the directed edges contain a variant mark (e.g., the edge between "asia" and "tub"), the edge between "either" and "xray" is a directed edge with an invariant tail. This is due to the fact that the former kind of directed edges come about due to v-structures, which for FCI orientation rules do not rule out the possibility of bidirected edges. The latter edge, however, is a compelled edge which does exclude the possibility of a bidirected edge. The final output in Figure 5.4c uses the neighborhood information of our algorithm to properly orient the edges by removing the variant marks and changing bidirected edges into undirected edges.

Upon completing this step, we may identify all jointly valid parent sets for the target nodes and use these as input for the `jointIda` function from the **pcalg** package to calculate the multiset of estimated causal effects of the target nodes on another node in the graph.

5.4 Conclusion

This chapter provides a brief outline of the functionality of the novel **CML** package, as well as a description of where this package fits in the larger ecosystem of structure learning software for causal graphical models. While the package currently only possesses limited options for Mb learning algorithms and different data types, we expect to extend its capabilities and incorporate some of the strategies from existing packages for wider application and cross-compatibility.

CHAPTER 6

Discussion

In this dissertation, we explore the state of local learning methods for causal DAGs and contribute a framework for coordinating learning across the local structures of multiple target nodes, jointly considered. The algorithm is both efficient and sound with respect to a novel ground truth graph which respects the topology between the neighborhoods and retains all CI information which is not underdetermined.

In Chapter 1, we provide definitions and discuss some global structure learning algorithms developed under different assumptions, especially the PC and FCI algorithms, since these contain steps which relate closely to the approach we take in addressing the multi-neighborhood problem. Though the PC algorithm is sufficient to learn the local structures for multiple target nodes, as contained in a subgraph of its output, it is computationally expensive and usually unnecessary. The FCI algorithm and its complete orientation rule set in the presence of latent variables is also important for our consideration since an optimal multi-neighborhood algorithm necessarily treats some of the nodes in the graph as latent to alleviate the complexity burden of estimating the global structure. Neither of these algorithms, however, are optimal or appropriate for local learning on multiple target nodes.

In Chapter 2, we discuss some local methods to bridge the gap between global structure learning and a preferred multi-neighborhood algorithm which does not estimate the entire graph structure. Markov blanket learning and the related field of feature selection are vast topics with a wide body of literature, and we cover only the main algorithm families as well as some newer methods. These local methods are essential for the initial pre-processing of

any multi-neighborhood algorithm, but they are insufficient for our purposes because they do not provide guidance for incorporating the overall graph topology into a local approach.

In Chapter 3, we finally introduce how to combine the relevant principles of previous work to learn the structure around multiple target nodes, especially as we may best identify a sufficient set for estimating the causal effects on other downstream nodes. With the CML algorithm, we are proposing a novel method for approaching causal structure learning across disjoint target neighborhoods without estimating the entire graph structure. This is of practical importance since global structure learning is a challenging problem laden with restrictive assumptions and high computational complexity, and practitioners interested in causal inference are usually primarily interested in a few target nodes most relevant to their study.

Though other local learning algorithms exist, to our knowledge none possess the capacity for coordinated structure learning across multiple disjoint neighborhoods where there are multiple target nodes of interest. The uniqueness of our method is also seen from the unique definition of the ground truth graph, which is a subgraph of the underlying DAG over the target neighborhoods with additional between-neighborhood ancestral edges to connect disjoint neighborhoods. We also establish soundness and consistency for our algorithm with respect to the equivalence class of the ground truth graph.

Although our algorithm builds on previous work in global and local learning, it is not a simple extension of methods such as the PC, as demonstrated by the distinction between CML and SNL, which applies the PC algorithm to each neighborhood individually. Moreover, by applying the back-door criterion, our method is more efficient for estimating the causal effects of target nodes on other nodes than applying a single Markov blanket learning algorithm to each target neighborhood individually. This is because between-neighborhood edges can help orient edges in downstream neighborhoods and thus identify accurate parent sets of the target nodes with greater specificity.

The upshot of these results is that researchers working with high-dimensional datasets

will have an efficient algorithm for causal discovery which focuses on target variables of their choosing, facilitating causal effect estimation using the back-door criterion.

In Chapter 4, we discuss the numerical results which confirm the projected strengths of our algorithm. Along with competitive or superior performance in overall accuracy and in recovering the target parents sets, we demonstrate the substantial computational savings of our algorithm compared to a global constraint-based algorithm (PC) in our simulated data results. We support our claim that coordination improves parent recovery by thorough comparisons to the SNL algorithm.

We also apply our method to a real-world gene expression dataset, which provides further evidence of at least comparable performance between CML and a global method. We find that, as long as the neighborhoods are relatively small compared to the global structure, CML provides efficiency improvements compared to the PC algorithm while identifying a similar number of nodes in the parent sets of the targets. This is also done with greater specificity of orientation than by simply applying the PC algorithm to each target neighborhood individually with the SNL algorithm.

In Chapter 5, we discuss the software package which implements the method in R, illustrating its usage and output as well as note how it may be used to facilitate joint causal effect estimation using the `jointIda` function from the **pcalg** package.

The ideas we discuss in this work point to fruitful research directions beyond the initial findings we present, which primarily establish our contention that the multi-neighborhood problem is worth greater attention and that our approach is sound and complete. Yet, this method must still be extended to receive different kinds of data as input, as well as incorporate different Mb learning algorithms and CI tests. The implementation may also be improved by storing some of the CI test results from Mb learning in order to avoid repeated computation during skeleton recovery. We can also achieve greater efficiency in our implementation by improving the separating set search such that we select subsets of the adjacency set for only one of the nodes rather than subsets of the union of the adjacency sets for both

nodes. With these changes, additional empirical analysis is required to ensure the algorithm's robustness. It will also be beneficial to include causal effect estimation comparisons in future numerical analysis to investigate our claims further. Finally, we will follow similar steps as in the work of Kalisch and Bühlmann (2007) to provide further theoretical guarantees under high-dimensional assumptions.

Bibliography

S. Acid, L. M. de Campos, and M. Fernández. Score-based methods for learning markov boundaries by searching in constrained spaces. *Data mining and knowledge discovery*, 26: 174–212, 2013.

C. F. Aliferis and I. Tsamardinos. Algorithms for large-scale local causal discovery and feature selection in the presence of limited sample or large causal neighbourhoods. 02 2003.

C. F. Aliferis, I. Tsamardinos, and A. Statnikov. Hiton: A novel markov blanket algorithm for optimal variable selection. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2003:21–5, 02 2003.

C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. 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, 2010a. URL `http://jmlr.org/papers/v11/aliferis10a.html`.

C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part ii: Analysis and extensions. *J. Mach. Learn. Res.*, 11:235–284, mar 2010b. ISSN 1532-4435.

S. A. Andersson, D. Madigan, and M. D. Perlman. A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997. ISSN 00905364. URL `http://www.jstor.org/stable/2242556`.

B. Aragam and Q. Zhou. Concave penalized estimation of sparse gaussian bayesian networks. *J. Mach. Learn. Res.*, 16(1):2273–2328, jan 2015. ISSN 1532-4435.

B. Aragam, J. Gu, and Q. Zhou. Learning large-scale bayesian networks with the sparsebn package. *Journal of Statistical Software*, 91(11):1–38, 2019. doi: 10.18637/jss.v091.i11. URL `https://www.jstatsoft.org/index.php/jss/article/view/v091i11`.

D. I. Bernstein, B. Saeed, C. Squires, and C. Uhler. Ordering-based causal structure learning in the presence of latent variables. In *International Conference on Artificial Intelligence and Statistics*, 2019. URL `https://api.semanticscholar.org/CorpusID:204800557`.

S. G. Boettcher and C. Dethlefsen. deal: A package for learning bayesian networks. *Journal of Statistical Software*, 8(20):1–40, 2003. doi: 10.18637/jss.v008.i20. URL `https://www.jstatsoft.org/index.php/jss/article/view/v008i20`.

G. Borboudakis and I. Tsamardinos. Forward-backward selection with early dropping. *J. Mach. Learn. Res.*, 20(1):276–314, jan 2019. ISSN 1532-4435.

G. Borboudakis, S. Triantafillou, and I. Tsamardinos. Tools and algorithms for causally interpreting directed edges in maximal ancestral graphs. In *Sixth European Workshop on Probabilistic Graphical Models*, 2012.

R. R. Bouckaert. Probabilistic network construction using the minimum description length principle. In M. Clarke, R. Kruse, and S. Moral, editors, *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 41–48, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg. ISBN 978-3-540-48130-0.

C. Băncioiu and R. Brad. Analyzing markov boundary discovery algorithms in ideal conditions using the d-separation criterion. *Algorithms*, 15(4), 2022. ISSN 1999-4893. doi: 10.3390/a15040105. URL `https://www.mdpi.com/1999-4893/15/4/105`.

C. Squires. *`causaldag`: creation, manipulation, and learning of causal models*, 2018. URL `https://github.com/uhlerlab/causaldag`.

E. Cai, A. McGregor, and D. Jensen. Improving the efficiency of the pc algorithm by using model-based conditional independence tests, 2022.

W. Chen, M. Drton, and A. Shojaie. Causal structural learning via local graphs. *SIAM*

Journal on Mathematics of Data Science, 5(2):280–305, 2023. doi: 10.1137/20M1362796. URL https://doi.org/10.1137/20M1362796.

D. Chickering and D. Heckerman. Efficient approximations for the marginal likelihood of bayesian networks with hidden variables. *Machine Learning*, 29:181–212, 11 1997. doi: 10.1023/A:1007469629108.

D. Chickering, C. Meek, and D. Heckerman. Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*, 5, 10 2012.

D. M. Chickering. Learning equivalence classes of bayesian-network structures. *J. Mach. Learn. Res.*, 2:445–498, mar 2002. ISSN 1532-4435. doi: 10.1162/153244302760200696. URL https://doi.org/10.1162/153244302760200696.

D. M. Chickering, D. Heckerman, and C. Meek. Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*, 5:1287–1330, dec 2004. ISSN 1532-4435.

D. Colombo and M. H. Maathuis. Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.*, 15(1):3741–3782, jan 2014. ISSN 1532-4435.

D. Colombo, M. H. Maathuis, M. Kalisch, and T. S. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 40(1): 294 – 321, 2012. doi: 10.1214/11-AOS940. URL https://doi.org/10.1214/11-AOS940.

G. F. Cooper. Causal discovery from data in the presence of selection bias. In D. Fisher and H. Lenz, editors, *Pre-proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, volume R0 of *Proceedings of Machine Learning Research*, pages 140–150. PMLR, 04–07 Jan 1995. URL https://proceedings.mlr.press/r0/cooper95a.html. Reissued by PMLR on 01 May 2022.

A. P. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical*

Society. Series B (Methodological), 41(1):1–31, 1979. ISSN 00359246. URL http://www.jstor.org/stable/2984718.

L. M. de Campos. A scoring function for learning bayesian networks based on mutual information and conditional independence tests. *Journal of Machine Learning Research*, 7 (77):2149–2187, 2006. URL http://jmlr.org/papers/v7/decampos06a.html.

A. Dobra and M. West. Bayesian covariance selection. 08 2004.

D. Eddelbuettel, R. Francois, J. Allaire, K. Ushey, Q. Kou, N. Russell, I. Ucar, D. Bates, and J. Chambers. *Rcpp: Seamless R and C++ Integration*, 2023a. URL https://CRAN.R-project.org/package=Rcpp. R package version 1.0.11.

D. Eddelbuettel, R. Francois, D. Bates, B. Ni, and C. Sanderson. *RcppArmadillo: 'Rcpp' Integration for the 'Armadillo' Templated Linear Algebra Library*, 2023b. URL https://CRAN.R-project.org/package=RcppArmadillo. R package version 0.12.4.1.0.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2), apr 2004. doi: 10.1214/009053604000000067. URL https://doi.org/10.1214%2F009053604000000067.

Z. Fang, Y. Liu, Z. Geng, S. Zhu, and Y. He. A local method for identifying causal relations under markov equivalence. *Artificial Intelligence*, 305:103669, 2022. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2022.103669. URL https://www.sciencedirect.com/science/article/pii/S0004370222000091.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, December 2007. ISSN 1465-4644. doi: 10.1093/biostatistics/kxm045. URL https://doi.org/10.1093/biostatistics/kxm045. _eprint: https://academic.oup.com/biostatistics/article-pdf/9/3/432/17742149/kxm045.pdf.

N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *J. Comput. Biol.*, 7(3-4):601–620, 2000.

F. Fu and Q. Zhou. Learning sparse causal gaussian networks with experimental intervention: Regularization and coordinate descent. *Journal of the American Statistical Association*, 108:288 – 300, 2013. URL https://api.semanticscholar.org/CorpusID:12955238.

S. Fu. Efficient learning of markov blanket and markov blanket classifier. 08 2010.

S. Fu and M. Desmarais. Fast markov blanket discovery algorithm via local learning within single pass. pages 96–107, 05 2008. ISBN 978-3-540-68821-1. doi: 10.1007/978-3-540-68825-9_10.

S. Fu and M. Desmarais. Markov blanket based feature selection: A review of past decade. *Lecture Notes in Engineering and Computer Science*, 2183, 06 2010.

T. Gao and Q. Ji. Efficient markov blanket discovery and its application. *IEEE Transactions on Cybernetics*, 47(5):1169–1179, 2017. doi: 10.1109/TCYB.2016.2539338.

J. Gu and Q. Zhou. Learning big gaussian bayesian networks: Partition, estimation and fusion. *Journal of Machine Learning Research*, 21(158):1–31, 2020. URL http://jmlr.org/papers/v21/19-318.html.

J. Gu, F. Fu, and Q. Zhou. Penalized estimation of directed acyclic graphs from discrete data. *Statistics and Computing*, 29(1):161–176, Jan 2019. ISSN 1573-1375. doi: 10.1007/s11222-018-9801-y. URL https://doi.org/10.1007/s11222-018-9801-y.

X. Guo, K. Yu, F. Cao, P. Li, and H. Wang. Error-aware markov blanket learning for causal feature selection. *Information Sciences*, 589:849–877, 2022. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2021.12.118. URL https://www.sciencedirect.com/science/article/pii/S0020025521013402.

S. Gupta, D. Childers, and Z. C. Lipton. Local causal discovery for estimating causal effects. In M. van der Schaar, C. Zhang, and D. Janzing, editors, *Proceedings of the Second Conference on Causal Learning and Reasoning*, volume 213 of *Proceedings of Machine Learning Research*, pages 408–447. PMLR, 11–14 Apr 2023. URL https://proceedings.mlr.press/v213/gupta23b.html.

M. J. Ha, W. Sun, and J. Xie. PenPC: A two-step approach to estimate the skeletons of high-dimensional directed acyclic graphs. *Biometrics*, 72(1):146–155, March 2016.

K. D. Hansen, J. Gentry, L. Long, R. Gentleman, S. Falcon, F. Hahne, and D. Sarkar. *Rgraphviz: Provides plotting capabilities for R graph objects*, 2020. R package version 2.34.0.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2009. ISBN 978-0-387-84884-6. URL https://books.google.com/books?id=eBSgoAEACAAJ.

C. Heinze-Deml, M. H. Maathuis, and N. Meinshausen. Causal structure learning. *Annual Review of Statistics and Its Application*, 5(1):371–391, 2018. doi: 10.1146/annurev-statistics-031017-100630. URL https://doi.org/10.1146/annurev-statistics-031017-100630.

J. Huang and Q. Zhou. Partitioned hybrid learning of bayesian network structures. *Machine Learning*, 111(5):1695–1738, May 2022. ISSN 1573-0565. doi: 10.1007/s10994-022-06145-4. URL https://doi.org/10.1007/s10994-022-06145-4.

J. Kaddour, A. Lynch, Q. Liu, M. J. Kusner, and R. Silva. Causal machine learning: A survey and open problems, 2022. URL http://arxiv.org/abs/2206.15475.

M. Kalisch and P. Bühlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *J. Mach. Learn. Res.*, 8:613–636, may 2007. ISSN 1532-4435.

M. Kalisch, M. Mächler, D. Colombo, M. H. Maathuis, and P. Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11): 1–26, 2012. doi: 10.18637/jss.v047.i11.

D. Kaufmann, S. Parbhoo, A. Wieczorek, S. Keller, D. Adametz, and V. Roth. Bayesian markov blanket estimation. In A. Gretton and C. C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 333–341, Cadiz, Spain, 09–11 May 2016. PMLR. URL `https://proceedings.mlr.press/v51/kaufmann16.html`.

W. Khan, L. Kong, S. M. Noman, and B. Brekhna. A novel feature selection method via mining markov blanket. *Applied Intelligence*, 53(7):8232–8255, Apr 2023. ISSN 1573-7497. doi: 10.1007/s10489-022-03863-z. URL `https://doi.org/10.1007/s10489-022-03863-z`.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1): 273–324, 1997. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(97)00043-X. URL `https://www.sciencedirect.com/science/article/pii/S000437029700043X`. Relevance.

D. Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML'96, page 284–292, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc. ISBN 1558604197.

V. Lagani, G. Athineou, A. Farcomeni, M. Tsagris, and I. Tsamardinos. Feature selection with the r package mxm: Discovering statistically equivalent feature subsets. *Journal of Statistical Software*, 80(7):1–25, 2017. doi: 10.18637/jss.v080.i07. URL `https://www.jstatsoft.org/index.php/jss/article/view/v080i07`.

S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society.*

Series B (Methodological), 50(2):157–224, 1988. ISSN 00359246. URL http://www.jstor.org/stable/2345762.

T. D. Le, T. Hoang, J. Li, L. Liu, and S. Hu. Parallelpc: an r package for efficient constraint based causal exploration, 2015.

Z. Ling, B. Li, Y. Zhang, Y. Li, and H. Ling. Online markov blanket learning for high-dimensional data. *Applied Intelligence*, 53(5):5977–5997, jul 2022a. ISSN 0924-669X. doi: 10.1007/s10489-022-03841-5. URL https://doi.org/10.1007/s10489-022-03841-5.

Z. Ling, K. Yu, Y. Zhang, L. Liu, and J. Li. Causal learner: A toolbox for causal structure and markov blanket learning. *Pattern Recognition Letters*, 163:92–95, 2022b. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2022.09.021. URL https://www.sciencedirect.com/science/article/pii/S0167865522002914.

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

D. Margaritis and S. Thrun. Bayesian network induction via local neighborhoods. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/5d79099fcdf499f12b79770834c0164a-Paper.pdf.

A. Marx and J. Vreeken. Testing conditional independence on discrete data using stochastic complexity. In K. Chaudhuri and M. Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 496–505. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/marx19a.html.

C. 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.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436 – 1462, 2006. doi: 10.1214/009053606000000281. URL `https://doi.org/10.1214/009053606000000281`.

G. Michailidis and F. d'Alché Buc. Autoregressive models for gene regulatory network inference: Sparsity, stability and causality issues. *Mathematical Biosciences*, 246:326–334, 2013.

P. Nandy, M. H. Maathuis, and T. S. Richardson. Estimating the effect of joint interventions from observational data in sparse high-dimensional settings. *The Annals of Statistics*, 45(2): 647 – 674, 2017. doi: 10.1214/16-AOS1462. URL `https://doi.org/10.1214/16-AOS1462`.

I. Ng, S. Zhu, Z. Chen, and Z. Fang. A graph autoencoder approach to causal structure learning, 2019.

T. Niinimäki and P. Parviainen. Local structure discovery in bayesian networks. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, UAI'12, page 634–643, Arlington, Virginia, USA, 2012. AUAI Press. ISBN 9780974903989.

J. M. Ogarrio, P. Spirtes, and J. Ramsey. A hybrid causal search algorithm for latent variable models. In A. Antonucci, G. Corani, and C. P. Campos, editors, *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, volume 52 of *Proceedings of Machine Learning Research*, pages 368–379, Lugano, Switzerland, 06–09 Sep 2016. PMLR. URL `https://proceedings.mlr.press/v52/ogarrio16.html`.

Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44 vol.1, 1993. doi: 10.1109/ACSSC.1993.342465.

J. Pearl. *Causality*. Causality: Models, Reasoning, and Inference. Cambridge University Press, 2009. ISBN 978-0-521-89560-6.

J. Pellet and A. Elisseeff. Using markov blankets for causal structure learning. *J. Mach. Learn. Res.*, 9:1295–1342, jun 2008. ISSN 1532-4435.

J. M. Peña, J. Björkegren, and J. Tegnér. Scalable, efficient and correct learning of markov boundaries under the faithfulness assumption. In L. Godo, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 136–147, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31888-0.

J. M. Peña, R. Nilsson, J. Björkegren, and J. Tegnér. Towards scalable and data efficient learning of markov boundaries. *International Journal of Approximate Reasoning*, 45(2):211–232, 2007. ISSN 0888-613X. doi: https://doi.org/10.1016/j.ijar.2006.06.008. URL `https://www.sciencedirect.com/science/article/pii/S0888613X06000600`. Eighth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005).

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020. URL `https://www.R-project.org/`.

J. Ramsey, P. Spirtes, and J. Zhang. Adjacency-faithfulness and conservative causal inference. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'06, page 401–408, Arlington, Virginia, USA, 2006. AUAI Press. ISBN 0974903922.

J. D. Ramsey, K. Zhang, M. Glymour, R. S. Romero, B. Huang, I. Ebert-Uphoff, S. Samarasinghe, E. A. Barnes, and C. Glymour. Tetrad—a toolbox for causal discovery. In *8th international workshop on climate informatics*, 2018.

T. Richardson and P. Spirtes. Ancestral graph Markov models. *The Annals of Statistics*, 30(4):962 – 1030, 2002. doi: 10.1214/aos/1031689015. URL `https://doi.org/10.1214/aos/1031689015`.

G. Ruiz, O. H. M. Padilla, and Q. Zhou. Sequentially learning the topological ordering of directed acyclic graphs with likelihood ratio scores. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=4pCjIGIjrt.

J. Runge. Conditional independence testing based on a nearest-neighbor estimator of conditional mutual information. In A. Storkey and F. Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 938–947. PMLR, 09–11 Apr 2018. URL https://proceedings.mlr.press/v84/runge18a.html.

A. Sboner and C. F. Aliferis. Modeling clinical judgment and implicit guideline compliance in the diagnosis of melanomas using machine learning. In *AMIA 2005, American Medical Informatics Association Annual Symposium, Washington, DC, USA, October 22-26, 2005*. AMIA, 2005. URL https://knowledge.amia.org/amia-55142-a2005a-1.613296/t-001-1.616182/f-001-1.616183/a-133-1.616290/a-134-1.616287.

F. Schlüter. A survey on independence-based markov networks learning. *Artificial Intelligence Review*, 42(4):1069–1093, jun 2012. doi: 10.1007/s10462-012-9346-y. URL https://doi.org/10.1007%2Fs10462-012-9346-y.

M. Scutari. Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software*, 35(3):1–22, 2010. URL http://www.jstatsoft.org/v35/i03/.

M. Scutari. Dirichlet bayesian network scores and the maximum relative entropy principle. *Behaviormetrika*, 45(2):337–362, Oct 2018. ISSN 1349-6964. doi: 10.1007/s41237-018-0048-x. URL https://doi.org/10.1007/s41237-018-0048-x.

B. Sierra and P. Larrañaga. Predicting survival in malignant skin melanoma using bayesian networks automatically induced by genetic algorithms. an empirical comparison between different approaches. *Artificial Intelligence in Medicine*, 14(1):215–230, 1998. ISSN 0933-3657. doi: https://doi.org/10.1016/S0933-3657(98)00024-4. URL https://www.

sciencedirect.com/science/article/pii/S0933365798000244. Selected Papers from AIME '97.

M. Singh, M. Provan, and P. Langley. Induction of selective bayesian network classifiers. *Machine Learning*, 2, 1996.

S. Smith and Q. Zhou. Coordinated multi-neighborhood learning on a directed acyclic graph. Manuscript in preparation for submission.

A. Sondhi and A. Shojaie. The reduced pc-algorithm: Improved causal structure learning in large random networks. *Journal of Machine Learning Research*, 20(164):1–31, 2019. URL http://jmlr.org/papers/v20/17-601.html.

P. Spirtes. An anytime algorithm for causal inference. In T. S. Richardson and T. S. Jaakkola, editors, *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, volume R3 of *Proceedings of Machine Learning Research*, pages 278–285. PMLR, 04–07 Jan 2001. URL https://proceedings.mlr.press/r3/spirtes01a.html. Reissued by PMLR on 31 March 2021.

P. Spirtes, C. Meek, and T. Richardson. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, page 499–506, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. The MIT press, 2nd edition, 2000.

A. Srivastava, S. P. Chockalingam, and S. Aluru. A parallel framework for constraint-based bayesian network learning via markov blanket discovery. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15, 2020. doi: 10.1109/SC41405.2020.00011.

A. Statnikov, I. Tsamardinos, Y. Dosbayev, and C. F. Aliferis. Gems: A system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. *International Journal of Medical Informatics*, 74(7):491–503, 2005. ISSN 1386-5056. doi: https://doi.org/10.1016/j.ijmedinf.2005.05.002. URL https://www.sciencedirect.com/science/article/pii/S1386505605000523. MedInfo 2004.

J. Textor, B. van der Zander, M. S. Gilthorpe, M. Liśkiewicz, and G. Ellison. Robust causal inference using directed acyclic graphs: the R package 'dagitty'. *International Journal of Epidemiology*, 45(6):1887–1894, 01 2017. ISSN 0300-5771. doi: 10.1093/ije/dyw341. URL https://doi.org/10.1093/ije/dyw341.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246. URL http://www.jstor.org/stable/2346178.

S. Tikka and J. Karvanen. Identifying causal effects with the r package causaleffect. *Journal of Statistical Software*, 76(12):1–30, 2017. doi: 10.18637/jss.v076.i12. URL https://www.jstatsoft.org/index.php/jss/article/view/v076i12.

I. Tsamardinos and C. F. Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. In C. M. Bishop and B. J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, volume R4 of *Proceedings of Machine Learning Research*, pages 300–307. PMLR, 03–06 Jan 2003. URL https://proceedings.mlr.press/r4/tsamardinos03a.html. Reissued by PMLR on 01 April 2021.

I. Tsamardinos, C. F. Aliferis, and A. 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. doi: 10.1145/956750.956838. URL https://doi.org/10.1145/956750.956838.

I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Algorithms for large scale markov blanket discovery. In I. Russell and S. M. Haller, editors, *FLAIRS Conference*, pages 376–381. AAAI Press, 2003b. ISBN 1-57735-177-0. URL http://dblp.uni-trier.de/db/conf/flairs/flairs2003.html#TsamardinosAS03.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. *Machine Learning*, 65(1):31–78, 2006.

C. Uhler, G. Raskutti, P. Bühlmann, and B. Yu. Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics*, 41(2):436–463, 2013. ISSN 00905364, 21688966. URL http://www.jstor.org/stable/23566569.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. *Probabilistic and Causal Inference*, 1990. URL https://api.semanticscholar.org/CorpusID:27807863.

M. J. Vowels, N. C. Camgoz, and R. Bowden. D'ya like dags? a survey on structure learning and causal discovery. *ACM Comput. Surv.*, 55(4), nov 2022. ISSN 0360-0300. doi: 10.1145/3527154. URL https://doi.org/10.1145/3527154.

Z. Yao, C. T. J. van Velthoven, T. N. Nguyen, J. Goldy, A. E. Sedeno-Cortes, F. Baftizadeh, D. Bertagnolli, T. Casper, M. Chiang, K. Crichton, S. Ding, O. Fong, E. Garren, A. Glandon, N. W. Gouwens, J. Gray, L. T. Graybuck, M. J. Hawrylycz, D. Hirschstein, M. Kroll, K. Lathia, C. Lee, B. Levi, D. McMillen, S. Mok, T. Pham, Q. Ren, C. Rimorin, N. Shapovalova, J. Sulc, S. M. Sunkin, M. Tieu, A. Torkelson, H. Tung, K. Ward, N. Dee, K. A. Smith, B. Tasic, and H. Zeng. A taxonomy of transcriptomic cell types across the isocortex and hippocampal formation. *Cell*, 184(12):3222–3241.e26, 2021. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2021.04.021. URL https://www.sciencedirect.com/science/article/pii/S0092867421005018.

S. Yaramakala and D. Margaritis. Speculative markov blanket discovery for optimal feature

selection. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 4 pp.–, 2005. doi: 10.1109/ICDM.2005.134.

K. Yu, L. Liu, and J. Li. Learning markov blankets from multiple interventional data sets. *IEEE Transactions on Neural Networks and Learning Systems*, 31(6):2005–2019, 2020. doi: 10.1109/TNNLS.2019.2927636.

J. Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16):1873–1896, 2008a. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2008.08.001. URL `https://www.sciencedirect.com/science/article/pii/S0004370208001008`.

J. Zhang. Causal reasoning with ancestral graphs. *Journal of Machine Learning Research*, 9, 07 2008b. doi: 10.1145/1390681.1442780.

K. Zhang, J. Peters, D. Janzing, and B. Schölkopf. Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, page 804–813, Arlington, Virginia, USA, 2011. AUAI Press. ISBN 9780974903972.

Y. Zhang, Z. Zhang, K. Liu, and G. Qian. An improved iamb algorithm for markov blanket discovery. *JCP*, 5:1755–1761, 11 2010. doi: 10.4304/jcp.5.11.1755-1761.