Data Mining and Knowledge Discovery (2024) 38:4157–4183
https://doi.org/10.1007/s10618-024-01069-0

Bounding the family-wise error rate in local causal discovery using Rademacher averages

Dario Simionato[1] · **Fabio Vandin**[1]

Received: 24 October 2023 / Accepted: 10 August 2024 / Published online: 9 September 2024

Abstract
Many algorithms have been proposed to learn local graphical structures around target variables of interest from observational data, focusing on two sets of variables. The first one, called Parent–Children (PC) set, contains all the variables that are direct causes or consequences of the target while the second one, known as Markov boundary (MB), is the minimal set of variables with optimal prediction performances of the target. In this paper we introduce two novel algorithms for the PC and MB discovery tasks with rigorous guarantees on the Family-Wise Error Rate (FWER), that is, the probability of reporting any false positive in output. Our algorithms use Rademacher averages, a key concept from statistical learning theory, to properly account for the multiple-hypothesis testing problem arising in such tasks. Our evaluation on simulated data shows that our algorithms properly control for the FWER, while widely used algorithms do not provide guarantees on false discoveries even when correcting for multiple-hypothesis testing. Our experiments also show that our algorithms identify meaningful relations in real-world data.

Keywords Local causal discovery · Markov boundary · Rademacher averages · FWER

1 Introduction

One of the most fundamental and challenging problems in science is the discovery of causal relations from observational data (Pearl 2009). Bayesian networks are graphical models that are widely used to represent causal relations and have been the

Responsible editor: Matteo Riondato.

✉ Fabio Vandin
fabio.vandin@unipd.it

Dario Simionato
dario.simionato@phd.unipd.it

[1] Department of Information Engineering, University of Padua, Padua, Italy

focus of a large amount of research in data mining and machine learning. Bayesian networks represent random variables or events as vertices of graphical models, and encode conditional-independence relationships according to the (directed) Markov property among the variables or events as directed acyclic graphs (DAGs). They are a fundamental tool to represent causality relations among variables and events, and have been used to analyze data from several domains, including biology (Pe'er 2005; Sachs et al. 2005), medicine (Velikova et al. 2014), and others (Yusuf et al. 2021; Kusner and Loftus 2020).

One of the core tasks in learning Bayesian networks from observational data is the identification of local causal structures around a target variable T. In this work we focus on two related local structures. The first one is the set of parents and children (i.e., the neighbours) of T in the DAG, denoted as the parent–children set $PC(T)$. $PC(T)$ has a natural causal interpretation as the set of *direct* causes and effects of T (Spirtes et al. 2000), and the accurate identification of $PC(T)$ is a crucial step for the inference of Bayesian networks. The second structure is the Markov boundary of T, denoted as $MB(T)$. $MB(T)$ is a minimal set of variables that makes T conditionally independent of all the other variables, and comprises the elements of $PC(T)$ and the other parents of the children of T. Thus, $MB(T)$ includes all direct causes, effects, and causes of direct effects of T. Moreover, under certain assumptions, the Markov boundary is the solution of the variable selection problem (Tsamardinos and Aliferis 2003), that is, it is the minimal set of variables with optimal predictive performance for T.

In several real-world applications, such as biology (Sachs et al. 2005) and neuroscience (Bielza and Larranaga 2014), the elements in $PC(T)$ and $MB(T)$ identified from observational data provide *candidate* causal relations explored in follow-up studies and experiments, which often require significant resources (e.g., time or chemical reagents). In other areas, such as algorithmic fairness (Mhasawade and Chunara 2021; Kusner and Loftus 2020), local causal discovery can help in identifying discriminatory relationships in data. In these scenarios, it is crucial to identify *reliable* causal relations between variables, ideally avoiding any false discovery.

While the stochastic nature of random sampling implies that false discoveries cannot be avoided with absolute certainty (when at least a relation is reported), a common approach from statistics to limit false discoveries is to develop methods that rigorously bound the Family-Wise Error Rate (FWER), that is, the probability of reporting one or more false discoveries. However, current approaches for local causal discovery do not provide guarantees on false discoveries in terms of FWER, and the study of causal discovery with false positive guarantees has received scant attention in general (see Sect. 3).

1.1 Our contributions

In this paper we introduce two novel algorithms that exploit Rademacher Averages for Local structure discovery (`RAveL`) providing rigorous guarantees on the FWER: `RAveL-MB` for the MB discovery task and `RAveL-PC` for the PC identification task. To the best of our knowledge, our algorithms are the first ones to allow the

discovery of the PC set and the MB of a target variable while providing provable guarantees on false discoveries in terms of the FWER. Our algorithms crucially rely on Rademacher averages, a key concept from statistical learning theory (Bartlett and Mendelson 2002), to properly account for the multiple-hypothesis testing problem arising in local causal discovery, where a large number of statistical test for conditional independence are performed. To the best of our knowledge, this work is the first one to introduce the use of Rademacher averages in (local) causal discovery. We prove, both analytically and experimentally, that currently used approaches to discover the PC set and the MB of a target variable cannot be adapted to control the FWER simply by correcting for multiple-hypothesis testing. This is due to their additional requirement of conditional dependencies being correctly identified, which is an unreasonable assumption due to the stochastic nature of random sampling and finite sample sizes. We then introduce two test statistics to be used in independence testing with Rademacher averages, assuming that the expectation and maximum of each variable is known. Our experimental evaluation shows that our algorithms do control the FWER while allowing for the discovery of elements in the PC set and in the MB of a target variable, even when empirical estimates of the quantities of interest are used. On real data, our algorithms return a subset of variables that causally influences the target in agreement with prior knowledge.

The rest of the paper is organized as follows. Section 2 introduces the preliminary concepts used in the rest of the paper. Section 3 describes previous works related to our contribution. Section 4 describes our algorithms and their analysis, and the assumptions required by previously proposed algorithms in order to provide rigorous results in terms of the FWER. For clarity, we describe our algorithms focusing on the case of continuous variables, but our algorithms can be easily adapted to discrete and categorical variables. Section 5 describes our experimental evaluation on synthetic and real data. Finally, Sect. 6 offers some concluding remarks.

2 Preliminaries

In this section, we introduce basic notions and preliminary concepts used in the rest of the paper. More specifically, in Sect. 2.1 we formally define Bayesian networks (BNs) and the sets $PC(T)$ and $MB(T)$ for a target variable T. In Sect. 2.2 we describe the statistical testing procedure commonly used by algorithms for the identification of $PC(T)$ and $MB(T)$. In Sect. 2.3 we introduce the multiple hypotheses testing problem and the FWER. Finally, in Sect. 2.4 we introduce the concept of Rademacher averages for supremum deviation estimation.

2.1 Bayesian networks

Bayesian Networks (BNs) are convenient ways to model the influence among a set of variables $\mathbf{V}$. BNs represent interactions using a *Direct Acyclic Graph (DAG)*, and employ probability distributions to define the strength of the relations. More formally, they are defined as follows.

Definition 1 (*Bayesian network* (Neapolitan et al. 2004)) Let p be a joint probability distribution over $\mathbf{V}$. Let $G = (\mathbf{W}, \mathbf{A})$ be a DAG where the vertices $\mathbf{W}$ of G are in a one-to-one correspondence with members of $\mathbf{V}$, and such that $\forall X \in \mathbf{V}$, X is conditionally independent of all non-descendants of X, given the parents of X (i.e., the *Markov condition* holds). A *Bayesian Network (BN)* is defined as a triplet $\langle \mathbf{V}, G, p \rangle$.

A common assumption for the study of BNs is *faithfulness*, defined as follows.

Definition 2 (*Faithfulness* (Spirtes et al. 2000)) A directed acyclic graph G is *faithful* to a joint probability distribution p over variable set $\mathbf{V}$ if and only if every independence present in p is entailed by G and the Markov Condition. A distribution p is *faithful* if and only if there exists a DAG G such that G is faithful to p.

The dependencies between variables in a faithful BN can be analyzed through the study of *paths*, which are sequences of consecutive edges of any directionality (i.e. $X \to Y$ or $X \leftarrow Y$, that is, ignoring their orientation) in G. In particular, the *directional separation*, or *d-separation* (Pearl 2009), criterion can be used to study the dependence between two subsets $\mathbf{X}$ and $\mathbf{Y}$ of variables conditioning on another set $\mathbf{Z}$ of variables, such that $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{V}$ are disjoint. Informally, the criterion marks a path between any variable in $\mathbf{X}$ and any variable in $\mathbf{Y}$ as *blocked* by $\mathbf{Z}$ if the flow of dependency between the two sets is interrupted and therefore the two sets are *independent* conditioning on $\mathbf{Z}$, written $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$. Viceversa, if the two sets $\mathbf{X}$ and $\mathbf{Y}$ are conditionally dependent given $\mathbf{Z}$, denoted with $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$, the path is marked as *open*. More formally, the definition of d-separated path is the following.

Definition 3 (*d-separation* (Pearl 2009)) A path q is *d-separated*, or *blocked*, by a set of nodes $\mathbf{Z}$ if and only if:

1. q contains a *chain* $I \to M \to J$ or a *fork* $I \leftarrow M \to J$ such that $M \in \mathbf{Z}$, or
2. q contains an *inverted fork* (or *collider*) $I \to M \leftarrow J$ such that $M \notin \mathbf{Z}$ and no descendant of M is in $\mathbf{Z}$.

A set $\mathbf{Z}$ is said to d-separate $\mathbf{X}$ from $\mathbf{Y}$ if and only if $\mathbf{Z}$ blocks every path from a node in $\mathbf{X}$ to a node in $\mathbf{Y}$.

A *causal* Bayesian network is a Bayesian network with causally relevant edge semantics (Pearl 2009; Ma and Tourani 2020).

2.1.1 Local causal discovery

The task of inferring the local region of a causal BN related to a target variable T from data is called *local causal discovery*. Two sets of variables are of major importance in local causal discovery. The first set is the *parent–children set PC*(T), which contains the variables that are direct cause of T or that are its direct consequence.

Definition 4 (*Parent–children set of T* (Ma and Tourani 2020)) The *parent–children set of T*, or *PC(T)*, is the set of all parents and all children of T, i.e., the elements directly connected to T, in the DAG G.

The elements in $PC(T)$ are the only variables that cannot be d-separated from T, that is, by the Markov property, for each X in $PC(T)$: $X \not\perp\!\!\!\perp T \mid \mathbf{Z}, \forall \mathbf{Z} \subseteq \mathbf{V} \backslash \{X, T\}$. The second set is the *Markov boundary* $MB(T)$ of a target variable T, defined as follows.

Definition 5 (*Markov boundary of T* (Pearl 2009; Tsamardinos et al. 2003)) The *Markov boundary of T* or *MB(T)* is the smallest set of variables in $\mathbf{V} \backslash \{T\}$ conditioned on which all other variables are independent of T, that is $\forall Y \in \mathbf{V} \backslash MB(T), Y \neq T, T \perp\!\!\!\perp Y \mid MB(T)$.

Given its definition and the d-separation criteria, in a faithful BN $MB(T)$ is composed of all parents, children, and *spouses* (i.e., parents of children) of T (Ma and Tourani 2020), that are those variables $X \in \mathbf{V} \backslash \{T\}$ for which $\exists Y \in PC(T)$ such that $X \perp\!\!\!\perp T \mid \mathbf{Z}$ and $X \not\perp\!\!\!\perp T \mid \mathbf{Z} \cup \{Y\}$ for all $\mathbf{Z} \subseteq \mathbf{V} \backslash \{X, T\}$. $MB(T)$ is the minimal subset $\mathbf{S} \subseteq \mathbf{V}$ for which $p(T \mid \mathbf{S})$ is estimated accurately (Ma and Tourani 2020; Tsamardinos et al. 2003), therefore is the optimal solution for feature selection tasks.

2.2 Statistical testing for independence

The identification of $PC(T)$ and $MB(T)$ is based on the definitions of conditional dependence and independence between two variables X and Y. In practice, given a dataset, the conditional dependencies between variables are assessed using statistical hypothesis testing. Since a universal independence test does not exist (Shah and Peters 2020), a commonly used approach is to compute the *Pearson's linear correlation coefficient* r between two vectors $\mathbf{x}$ and $\mathbf{y}$ of k elements:

$$r_{\mathbf{x},\mathbf{y}} = \frac{\sum_{i=1}^{k} x_i y_i - k\bar{x}\bar{y}}{(k-1)s_{\mathbf{x}} s_{\mathbf{y}}} \tag{1}$$

where x_i and y_i are the i-th element $\mathbf{x}$ and $\mathbf{y}$, respectively, $\bar{x}$ and $\bar{y}$ are the sample mean of $\mathbf{x}$ and $\mathbf{y}$, respectively, and $s_{\mathbf{x}}$ and $s_{\mathbf{y}}$ are the sample standard deviations.

The vectors $\mathbf{x}$ and $\mathbf{y}$ correspond to the observations of X and Y in the data, but their definition depends on whether the test is unconditional, or conditional on a set $\mathbf{Z}$ of variables. In the first case, $\mathbf{x}$ and $\mathbf{y}$ are the vectors of observations for variables X and Y, respectively. In the second case, $\mathbf{x}$ and $\mathbf{y}$ represent the residuals of the linear regression of the observations of the variables in $\mathbf{Z}$ on the ones in X (respectively, for $\mathbf{y}$, the ones in Y). For sake of simplicity, in what follows we will use $r_{X,Y,\mathbf{Z}}$ to denote the value of $r_{\mathbf{x},\mathbf{y}}$ when $\mathbf{x}$ and $\mathbf{y}$ are obtained conditioning on the set $\mathbf{Z}$, potentially with $\mathbf{Z} = \emptyset$ (i.e., for unconditional testing), as we just described.

Under the *null hypothesis* of independence between X and Y conditional on $\mathbf{Z}$ (including the case $\mathbf{Z} = \emptyset$), the expected value of $r_{X,Y,\mathbf{Z}}$ is 0, and the statistic

$t = \frac{r_{X,Y,\mathbf{Z}}}{\sqrt{(1-r^2_{X,Y,\mathbf{Z}})/(k-2)}}$ follows a *Student's t* distribution with and $k-2$ degrees of freedom. The dependence between X and Y is then usually assessed by computing (with *Student's t* distribution) the *p-value* for the test statistic t, that is the probability that the statistic is greater or equal than t under the null hypothesis of independence. In practice, algorithms for local causal discovery (e.g., Tsamardinos et al. 2003; Pearl et al. 2007) consider X and Y as independent (unconditionally or conditional on $\mathbf{Z}$) if the p-value is greater than a threshold δ (common values for δ are 0.01 or 0.05), while X and Y are considered as dependent otherwise.

2.3 Multiple hypotheses testing

As described above, in testing for the independence of two variables X and Y, they are considered dependent if the p-value of the corresponding test is below a threshold δ. It is easy to see that such procedure guarantees that if X and Y are independent, then the probability of a *false discovery*, that is *falsely rejecting* their independence, is at most δ. The situation is drastically different when a large number N of hypotheses are tested, as in the case of local causal discovery. In this case, if the same threshold δ is used for every test, the expected number of false discoveries can be as large as δN. Therefore, it is necessary to correct for multiple hypothesis testing (MHT), with the goal of providing guarantees on false discoveries. A commonly used guarantee is provided by the *Family-Wise Error Rate (FWER)*, which is the probability of having at least one false discovery among all the tests. A common approach to control the FWER is the so called *Bonferroni correction* (Bonferroni 1936), which performs each test with a corrected threshold $\delta_{test} = \delta/N$ (a simple union bound shows that the resulting FWER is at most δ).

2.4 Supremum deviation and Rademacher averages

While Bonferroni correction does control the FWER, it conservatively assumes the worst-case scenario (of independence) between *all* null hypotheses. This often leads to a high number of *false negatives* (i.e. false null hypotheses that are not rejected). We now describe Rademacher averages (Bartlett and Mendelson 2002; Koltchinskii and Panchenko 2000), which allow to compute *data-dependent* confidence intervals for *all hypotheses simultaneously*, leading to improved tests for MHT scenarios (Pellegrina et al. 2022). Rademacher averages are a concept from statistical learning theory commonly used to measure the complexity of a family of functions and that, in general, also provide a way to probabilistically bound the deviation of the empirical means of the functions in the family from their expected values.

Let $\mathcal{F}$ be a family of functions from a domain $\mathcal{D}$ to $[a, b] \subset \mathbb{R}$ and let $\mathcal{S}$ be a sample of m i.i.d. observations from an unknown data generative distribution $\mathcal{W}$ over $\mathcal{D}$. We define the *empirical sample mean* $\hat{\mathbb{E}}_{\mathcal{S}}[f]$ of a function $f \in \mathcal{F}$ and its *expectation* $\mathbb{E}[f]$ as

$$\hat{\mathbb{E}}_{\mathcal{S}}[f] \doteq \frac{1}{m} \sum_{s_i \in \mathcal{S}} f(s_i) \text{ and } \mathbb{E}[f] \doteq \mathbb{E}_{\mathcal{W}} \left[\frac{1}{m} \sum_{s_i \in \mathcal{S}} f(s_i) \right]. \tag{2}$$

Note that $\mathbb{E}[f] = \mathbb{E}_{\mathcal{W}}[f]$, that is, the expected value of the empirical mean corresponds to the expectation according to distribution $\mathcal{W}$. A measure of the maximum deviation of the empirical mean from the (unknown) expectation for every function $f \in \mathcal{F}$ is given by the *supremum deviation* (SD) $D(\mathcal{F}, \mathcal{S})$ that is defined as

$$D(\mathcal{F}, \mathcal{S}) = \sup_{f \in \mathcal{F}} |\hat{\mathbb{E}}_{\mathcal{S}}[f] - \mathbb{E}[f]|. \tag{3}$$

Computing $D(\mathcal{F}, \mathcal{S})$ exactly is not possible given the unknown nature of $\mathcal{W}$, therefore bounds are commonly used. An important quantity to estimate tight bounds on the SD is the *Empirical Rademacher Average* (ERA) $\hat{R}(\mathcal{F}, \mathcal{S})$ of $\mathcal{F}$ on $\mathcal{S}$, defined as

$$\hat{R}(\mathcal{F}, \mathcal{S}) \doteq \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i f(s_i) \right] \tag{4}$$

where $\boldsymbol{\sigma}$ is a vector of m i.i.d. Rademacher random variables, i.e. for which each element σ_i equals 1 or -1 with equal probability. ERA is an alternative of *VC dimension* for computing the expressiveness of a set $\mathcal{S}$ over class function $\mathcal{F}$, whose main advantage is that it provides tight *data-dependent* bounds while the *VC dimension* provides *distribution-free* bounds that are usually fairly conservative (Mitzenmacher and Upfal 2017, chap. 14).

Computing the exact value of $\hat{R}(\mathcal{F}, \mathcal{S})$ is often infeasible since the expectation is taken over 2^m elements. A common approach is then to estimate $\hat{R}(\mathcal{F}, \mathcal{S})$ using a Monte-Carlo approach with n samples of $\boldsymbol{\sigma}$. The n-samples Monte-Carlo Empirical Rademacher Average (n-MCERA) $\hat{R}^n_m(\mathcal{F}, \mathcal{S}, \boldsymbol{\sigma})$ is defined as

$$\hat{R}^n_m(\mathcal{F}, \mathcal{S}, \boldsymbol{\sigma}) \doteq \frac{1}{n} \sum_{j=1}^{n} \sup_{f \in \mathcal{F}} \frac{1}{m} \sum_{s_i \in S} \sigma_{j,i} f(s_i) \tag{5}$$

with $\boldsymbol{\sigma}$ being a $m \times n$ matrix of i.i.d. Rademacher random variables. n-MCERA is useful to derive probabilistic upper bounds to the SD, as the following.

Theorem 1 (Th. 3.1 of Pellegrina et al. (2022)) *Let $\delta \in (0, 1)$. For ease of notation let*

$$\tilde{R} = \hat{R}^n_m(\mathcal{F}, \mathcal{S}, \boldsymbol{\sigma}) + 2z \sqrt{\frac{\ln \frac{4}{\delta}}{2nm}} \tag{6}$$

With a probability of at least $1 - \delta$ over the choice of $\mathcal{S}$ and $\boldsymbol{\sigma}$, it holds

$$D(\mathcal{F},\mathcal{S}) \leq 2\tilde{R} + \frac{\sqrt{c(4m\tilde{R} + c\ln\frac{4}{\delta})\ln\frac{4}{\delta}}}{m} + \frac{c\ln\frac{4}{\delta}}{m} + c\sqrt{\frac{\ln\frac{4}{\delta}}{2m}} \tag{7}$$

where $z = \max\{a, b\}$ *and* $c = b - a$.

Theorem 1 allows us to obtain confidence intervals around the empirical mean containing the expectation with probability at least $1 - \delta$ for all functions in $\mathcal{F}$ simultaneously.

3 Related work

Given a target variable T, the task of finding $MB(T)$ is strictly related to the discovery of $PC(T)$. A common approach for MB discovery consists of creating a candidate set of elements in $MB(T)$ by running a PC discovery algorithm twice (first on T, and then on all the elements reported as member of $PC(T)$) to find the elements at distance at most 2 from T, and then to eliminate false positives, which are those elements that are not parents, children, or spouses of T. Various algorithms follow this general scheme (Tsamardinos et al. 2003; Aliferis et al. 2003; Pearl et al. 2007; Aliferis et al. 2010), each one with a different variant that aims at minimizing the number of independence tests *actually* performed and their degrees of freedom to reduce the amount of data required. However, as described in Sect. 4.3, this does not decrease the number of statistical tests to be considered for MHT correction, since *a priori* all tests could *potentially* be performed. Among such algorithms, Pearl et al. (2007) proposed *PCMB* and proved its correctness under the assumption of all statistical tests being correct, that is, not returning *any* false positive or false negative. *PCMB* discovers $MB(T)$ by exploiting an auxiliary function *GetPC*, which returns $PC(T)$, and that in turn uses another function *GetPCD* returning a set containing all parents, all children, and some descendants of T. *PCMB* at first discovers the variables in $PC(T)$ by calling *GetPC*(T) and then builds a set $\mathbf{S}$ of candidate spouses by repeating the PC discovery on each element just retrieved. Lastly, a filtering operation is performed to remove false positives from $\mathbf{S}$ and to select only the actual spouses of T that are then returned in output together with $PC(T)$. (See Algorithm 1, Algorithm 2, and Algorithm 3 of Sect. 4.1 for the pseudocode of *GetPCD*, *GetPC*, and *PCMB*.)

A different approach has been proposed for *IAMB* (Tsamardinos et al. 2003) that incrementally grows a candidate set of elements in $MB(T)$ without searching for $PC(T)$, and then performs a false positive removal phase. *IAMB* starts with defining an empty set $\mathbf{E}$ of candidate elements in $MB(T)$ and, in the growing phase, it adds one variable $Y \in \mathbf{V}$ to $\mathbf{E}$ if it is dependent to the T conditioning on $\mathbf{E}$, otherwise it removes Y from the analysis. At the end of the growing phase, the set $\mathbf{E}$ will be a Markov blanket of T (i.e. a set of elements condition upon which T is independent of the rest of the variables) but it might not be $MB(T)$, that is also defined as the *minimal* Markov blanket of T. In order to return $MB(T)$, *IAMB* therefore performs a clean-up procedure that removes the elements $Y \in \mathbf{E}$ from $\mathbf{E}$ if they are independent

of T conditioning on $\mathbf{E}\backslash\{Y\}$. (See Algorithm 4 of Sect. 4.1 for the pseudocode of *IAMB*.)

Both *PCMB* and *IAMB* do not report false positives only under the assumption of not having any false positive and any false negative. Such assumptions are unrealistic in real-world scenarios due to noise in the data, finite sample sizes, and probabilistic guarantees of statistical tests, especially in multiple hypotheses scenarios. Our algorithms `RAveL-PC` and `RAveL-MB` do not require such assumptions to identify $PC(T)$ and $MB(T)$ with guarantees on the FWER.

To the best of our knowledge, the study of local causal discovery with guarantees on false discoveries has received scant attention. Tsamardinos et al. (2008) introduced the problem of MHT in the context of local causal discovery, and proposed to use the Benjamini-Hochberg correction (Benjamini and Hochberg 1995) to estimate the False Discovery Rate (FDR) of elements retrieved by $PC(T)$ discovery algorithms. However, such work does not provide an algorithm with guarantees for $MB(T)$. To the best of our knowledge, no method has focused on local causal discovery while bounding the FWER, which is extremely important in domains where false positives are critical or where follow-up studies require significant resources (e.g., biology and medicine).

Additional works focused on the more general task of BN inference. In Armen and Tsamardinos (2014), the authors extended the analysis of Tsamardinos et al. (2008) from the local discovery task to the BN inference while (Li and Wang 2009; Liu et al. 2012; Strobl et al. 2019) re-implemented the PC algorithm for BN structure discovery using the Benjamini and Yekutieli (2001) correction for the FDR, the former focusing on the skeleton retrieving and the latter deriving bounds on edge orientation as well. Our work instead focuses on *local* causal discovery tasks.

Rademacher averages have been successfully used to speed-up data mining tasks (e.g., pattern mining Riondato and Upfal 2015; Riondato and Upfal 2018; Pellegrina et al. 2022; Santoro et al. 2020; Pellegrina and Vandin 2023). To the best of our knowledge, ours is the first work to introduce their use in (local) causal discovery.

4 Algorithms for local causal discoveries with FWER guarantees

In this section we describe algorithms to obtain $PC(T)$ and $MB(T)$ with guarantees on the FWER. First, we discuss in Sect. 4.1 the requirements for previously proposed algorithms *PCMB* and *IAMB* to obtain guarantees on the FWER. In particular, we show that they require unrealistic assumptions that are not met in practice, as confirmed by our experimental evaluation (see Sect. 5). We then present in Sect. 4.2 our algorithms `RAveL-PC` and `RAveL-MB` for the computation of $PC(T)$ and $MB(T)$ with guarantees on the FWER. Finally, in Sect. 4.3 we describe how our algorithms perform effective independence testing by combining a novel test statistic with Rademacher averages.

4.1 Analysis and limitations of *PCMB* and *IAMB*

The algorithms presented in Sect. 3 are correct under the assumption that the independence tests result in no false positive *and* no false negative (Pearl et al. 2007; Tsamardinos et al. 2003). In this section we determine milder sufficient conditions that allow *GetPC* (Pearl et al. 2007) to control the FWER for the PC discovery task, and *PCMB* (Pearl et al. 2007) and *IAMB* (Tsamardinos et al. 2003) to control the FWER for the MB discovery task. In all cases, a first requirement is that the independence tests performed by the algorithms must account for MHT in order to bound the FWER. However, we also show that an additional requirement on the ability to identify dependent variables (i.e., on the *power* of the tests) is needed. In particular, we refer to the situation where *all tests* on dependent variables correctly reject the null hypothesis of independence as the *infinite power assumption*. In some cases, we consider the infinite power assumption only for independence tests between pairs of variables that are directly connected in the underlying DAG. We refer to such situation as the *local infinite power assumption*.

4.1.1 PCMB

Both *GetPC* and *PCMB* make use of a subroutine called *GetPCD* (Pearl et al. 2007) whose aim is to return a set containing the elements in $PC(T)$ and eventually some elements of the set $Descendants(T)$, that is, the set of descendants of T by applying a sequence of independence tests. In this section we will study under which conditions each method does not output any false positive, and how each subroutine result may affect the output of other algorithms.

We first start by studying under which conditions *GetPCD* (Pearl et al. 2007) returns a false positive in output.

Theorem 2 (Study of false positives in *GetPCD*) *An element $X \notin PC(T) \cup Descendants(T)$ is returned from GetPCD only if not all the parents of T are detected or the null hypotheses of some independence tests is wrongly rejected.*

Proof Let us recall that an element $X \in \mathbf{V}$ returned by $GetPCD(T, \mathbf{V})$ is a false negative if and only if $X \notin PC(T) \cup Descendants(T)$, that is X is either not connected to T in G or X is connected to T but it is not its parent, children or descendant (e.g. X is parent of a parent of T).

It is easy see that an element is returned by *GetPCD* only if it is not removed at lines 9 and 19 of Algorithm 1, which means that the null hypothesis of tests at lines 8 and 18 gets always rejected.[1] The independence test rejects the null hypothesis of independence of T from X conditioning on $\mathbf{Z} = sep[X]$ only if the two variables are

[1] The conditions in the "if" clause do not evaluate to `true` since an element may be added and then subsequently removed leading to the end of the repeat cycle because *PCD* did not change, but there still are elements in *canPCD* i.e. unremoved elements.

truly conditionally dependent (which means that conditioning on $\mathbf{Z} = sep[X]$ there is an open path between X and T), or if the null hypothesis gets *wrongly* rejected.

Let us now study the two topological cases of X being disconnected to T and of X being connected to T.

Disconnected case. Let X be disconnected from T. Since there are no paths from X to T (therefore no open paths from X to T), X cannot be conditional dependent from T conditioning on any set $\mathbf{Z}$. Therefore X may be returned by *GetPCD* only if independence tests at lines 8 and 18 always wrongly reject the null hypothesis.

Connected case. Let $X \notin PC(T) \cup Descendants(T)$ be connected to T. X is returned in output only if in any iteration of the cycle the null hypothesis on tests at lines 8 and 18 is wrongly rejected or if $T \not\perp\!\!\!\perp X \mid Sep[X]$, meaning there is an open path conditioning on $Sep[X]$.

By assuming of not having wrong rejections of the null hypotheses, $\mathbf{Z} = Parents(T)$ d-separates X and T by definition of parents since X is not a descendant of T. This implies that if some parent of T is undetected, then it may not be possible to d-separate X from T. □

Algorithm 1 *GetPCD*$(T, \mathbf{V})$ (Pearl et al. 2007)

Input: target variable T, set $\mathbf{V}$ of variables
Output: $\{X \in \mathbf{V} \mid X \in PC(T) \vee X \in descendants(T)\}$

1 $PCD \leftarrow \emptyset$;
2 $CanPCD \leftarrow \mathbf{V} \setminus \{T\}$;
3 **repeat**
4 /* *Remove false positives from CanPCD* */ ;
5 **foreach** $X \in CanPCD$ **do**
6 $Sep[X] \leftarrow \arg\min_{\mathbf{Z} \subseteq PCD} dep(T, X \mid \mathbf{Z})$;
7 **foreach** $X \in CanPCD$ **do**
8 **if** $T \perp\!\!\!\perp X \mid Sep[X]$ **then**
9 $CanPCD \leftarrow CanPCD \setminus \{X\}$;
10 /* *Add the best candidate to PCD* */ ;
11 $Y \leftarrow \arg\max_{X \in CanPCD} dep(T, X \mid Sep[X])$;
12 $PCD \leftarrow PCD \cup \{Y\}$;
13 $CanPCD \leftarrow CanPCD \setminus \{Y\}$;
14 /* *Remove false positives from PCD* */;
15 **foreach** $X \in PCD$ **do**
16 $Sep[X] \leftarrow \arg\min_{\mathbf{Z} \subseteq PCD \setminus \{X\}} dep(T, X \mid \mathbf{Z})$;
17 **foreach** $X \in PCD$ **do**
18 **if** $T \perp\!\!\!\perp X \mid Sep[X]$ **then**
19 $PCD \leftarrow PCD \setminus \{X\}$;
20 **until** *PCD does not change*;
21 **return** *PCD;*

We can then determine under which conditions *GetPCD* is able to control the FWER.

Theorem 3 *GetPCD*$(T, \mathbf{V})$ *outputs a set of elements in PC*(T) *or a descendant of T with FWER lower than* δ *if the FWER of the set of all the independence tests performed by GetPCD is below* δ *and the local infinite power assumption holds.*

Proof By analyzing *GetPCD* structure as in Theorem 2, an element is returned only if both independence tests at lines 8 and 18 of Algorithm 1 reject the null hypothesis. Therefore the algorithm outputs a false positive if under infinite power assumption for elements directly connected at least one independence test returns a false positive. Let us define the events $E =$ "*GetPCD*$(T, \mathbf{V})$ *outputs a false positive*" and $E_i =$ "*the i-th independence test returns a false positive*". We then have

$$FWER = P(E) \leq P\left(\cup_i E_i\right) \leq \delta$$

by definition of FWER. □

We now provide sufficient conditions for bounding the FWER of the elements returned by *GetPC* (Pearl et al. 2007).

Theorem 4 *GetPC*$(T, \mathbf{V})$ *outputs a set of elements in PC*(T) *with FWER* $\leq \delta$ *if the independence tests performed by GetPC have FWER* $\leq \delta$ *and the local infinite power assumption holds.*

Proof *GetPC* outputs a false positive only if at least one call to *GetPCD* at lines 2–3 of Algorithm 2 outputs a false positive and, under the infinite power assumption while testing the independence of elements directly connected, this happens only if at least one independence test outputs a false positive. Let us define the events $E =$ "*GetPC*$(T, \mathbf{V})$ *outputs a false positive*" and $E_i =$"*the i-th independence test returns a false positive*". We then have

$$FWER = P(E) \leq P\left(\cup_i E_i\right) \leq \delta$$

by definition of FWER. □

Algorithm 2 *GetPC*$(T, \mathbf{V})$ (Pearl et al. 2007)

Input: target variable T, set $\mathbf{V}$ of variables
Output: $PC(T)$
1 $PC \leftarrow \emptyset$;
2 **foreach** $X \in GetPCD(T, \mathbf{V})$ **do**
3 **if** $T \in GetPCD(X, \mathbf{V})$ **then**
4 $PC \leftarrow PC \cup \{X\}$
5 **return** PC;

The following proves that similar requirements are needed for *PCMB* (Pearl et al. 2007) to have guarantees on the FWER.

Theorem 5 *PCMB*$(T, \mathbf{V})$ *outputs a set of elements in MB*(T) *with FWER* $\leq \delta$ *if the independence tests performed by PCMB have FWER* $\leq \delta$ *and the infinite power assumption holds.*

Proof *PCMB* outputs a false positive only if there is a false positive in any independence test performed by *GetPC* calls at lines 2 and 6 of Algorithm 3, or if tests at lines 8 and 9 return a false negative or a false positive, respectively. Given the infinite power assumption and Theorem 4, *PCMB* outputs a false positive only if at least one independence test outputs a false positive and by defining the events $E =$ "*PCMB*$(T, \mathbf{V})$ *outputs a false positive*" and $E_i =$"*the i-th independence test returns a false positive*" we have

$$FWER = P(E) \leq P\left(\cup_i E_i\right) \leq \delta$$

by definition of FWER. □

Algorithm 3 *PCMB*$(T, \mathbf{V})$ (Pearl et al. 2007)

Input: target variable T, set $\mathbf{V}$ of variables
Output: $MB(T)$
1 /* *Add true positives to MB* */ ;
2 $PC \leftarrow GetPC(T, \mathbf{V})$;
3 $MB \leftarrow PC$;
4 /* *Add more true positives to MB* */ ;
5 **foreach** $Y \in PC$ **do**
6 **foreach** $X \in GetPC(Y, \mathbf{V})$ **do**
7 **if** $X \notin PC$ **then**
8 find $\mathbf{Z}$ such that $T \perp\!\!\!\perp X \mid \mathbf{Z}$ and $T, X \notin \mathbf{Z}$;
9 **if** $T \not\perp\!\!\!\perp X \mid \mathbf{Z} \cup Y$ **then**
10 $MB \leftarrow MB \cup \{X\}$;
11 **return** MB;

4.1.2 IAMB

The following result proves analogous requirements of Sect. 4.1.1 for *IAMB*.

Theorem 6 *IAMB*$(T, \mathbf{V})$ *outputs a set of elements in MB*(T) *with FWER* $\leq \delta$ *if the independence tests performed by IAMB have FWER* $\leq \delta$ *and the infinite power assumption holds.*

Proof *IAMB* outputs a false positive only if an element $X \notin MB(T)$ gets added to MB at lines 5–6, and it does not get removed from MB at lines 10–11 of Algorithm 4. Under the infinite power assumption, all elements in $PC(T)$ get added at lines 5–6 by definition of PC, therefore X gets returned by *IAMB* only if independence tests at lines 10–11 output a false positive. Then, by defining the events $E =$ "*GetPC*$(T, \mathbf{V})$ *outputs a false positive*" and $E_i =$"*the i-th independence test returns a false positive*", we have

$$FWER = P(E) \leq P\left(\cup_i E_i\right) \leq \delta$$

by definition of FWER. □

Algorithm 4 *IAMB*$(T, \mathbf{V})$ (Tsamardinos et al. 2003)

Input: target variable T, set $\mathbf{V}$ of variables
Output: $MB(T)$
1 /* *Add true positives to MB* */ ;
2 $MB \leftarrow \emptyset$;
3 **repeat**
4 $\quad Y \leftarrow \arg\max_{X \in \mathbf{V} \setminus MB \setminus \{T\}} dep(T, X, MB)$;
5 $\quad$ **if** $T \not\perp\!\!\!\perp Y \mid MB$ **then**
6 $\quad\quad MB \leftarrow MB \cup \{Y\}$;
7 **until** *MB does not change*;
8 /* *Remove false positives from MB* */ ;
9 **foreach** $X \in MB$ **do**
10 $\quad$ **if** $T \perp\!\!\!\perp X \mid MB \setminus \{X\}$ **then**
11 $\quad\quad MB \leftarrow MB \setminus \{X\}$;
12 **return** MB;

4.1.3 Relaxation of the infinite power assumption

Note that the results above require the (local) infinite power assumption to hold in order to have guarantees on the FWER of the output of previously proposed algorithms. In fact, if the (local) infinite power assumption does not hold, such algorithms may output false positives even when *all* independence tests do not return a single false positive. We now present three such examples by considering the subgraph of Fig. 1 in Sect. 5 between variables $\mathbf{V} = \{C_1, A_2, B_2, C_2\}$ with edges $\mathbf{E} = \{C_1 \to A_2, C_1 \to B_2, A_2 \to C_2, B_2 \to C_2\}$. Moreover, our experimental evaluation in Sect. 5 shows that these situations do happen in practice.

Scenario 1: The infinite power assumption holds only for directly connected elements. Let us study the subgraph previously described under only local infinite power assumption. Let us suppose to run $PCMB(C_1, \mathbf{V})$ and that the call at line 2 correctly returned $GetPC(C_1, \mathbf{V}) = \{A_2, B_2\}$. Let us further suppose that $GetPC(A_2, \mathbf{V}) = \{C_1, C_2\}$ and that a false negative arises when testing the

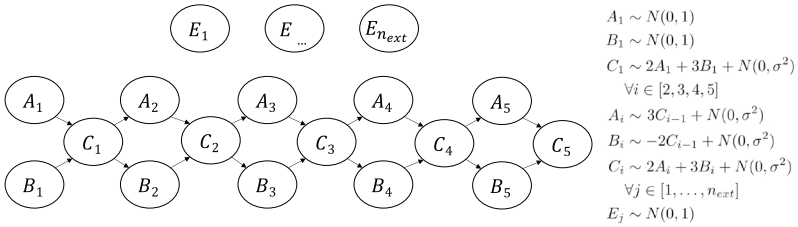

Fig. 1 Bayesian Network used for synthetic data generation, parametrized by two values σ^2 and n_{ext}. After drawing all the observations $\mathbf{x}$ for a particular variable X, $\mathbf{x}$ is normalized such that $mean(\mathbf{x}) = 0$ and $var(\mathbf{x}) = 1$, then the values for the descendants of X are sampled

unconditional dependence between C_1 and C_2, leading to the choice of $\mathbf{Z} = \emptyset$ on line 8. If the conditional independence test at line 9 correctly assesses the conditional dependence of C_1 and C_2 conditioning on A_2, then C_2 is wrongly considered a spouse of C_1.

Scenario 2: No infinite power assumption. Consider as an example the calculus of $GetPC(C_2, \mathbf{V})$ in the subgraph previously described. Let us suppose that a false negative occurs when testing the unconditional independencies between C_2 and A_2 and between C_1 and A_2. Let us further suppose $\mathbf{Z} = \{A_2, B_2\}$ to be the only set for which the null hypothesis of independence between C_1 and C_2 is not rejected. Then $GetPCD(C_2, \mathbf{V})$ will contain C_1 (because the independence conditioning on $\mathbf{Z} = \{A_2, B_2\}$ is never tested), and similarly $GetPCD(C_1, \mathbf{V})$ will contain C_2 leading C_1 to be returned by $GetPC(C_2, \mathbf{V})$.

Scenario 3: No infinite power assumption and *GetPC* does not return false positives. Let us finally consider a situation in which the infinite power assumption does not hold and *GetPC* does not return any false positive, as this may be the case of a modification of the algorithms proposed by Pearl et al. (2007) using Bonferroni correction. Let us suppose $GetPC(C_1, \mathbf{V}) = \{A_2\}$, and $GetPC(C_2, \mathbf{V}) = \{A_2\}$. Let us suppose line 8 to return $\mathbf{Z} = \emptyset$, and the conditional independence test at line 9 to correctly assess the conditional dependence of C_1 and C_2 conditioning on A_2. Under these assumptions, C_2 is wrongly considered a spouse of C_1. Note that this scenario differs from the first because the local infinite power assumption does not hold, leading to a partial discovery of the variables in $PC(C_1)$ whose elements are not enough to d-separate C_1 and C_2.

4.2 Algorithms `RAveL-PC` and `RAveL-MB`

As shown in Sect. 4.1, controlling the FWER of every independence test is not sufficient for bounding the FWER of the variables returned by current state-of-the-art algorithms for PC and MB discovery. In addition, infinite statistical power is a strong assumption which is impossible to test and ensure in real-world scenarios. Motivated by these observations, we developed `RAveL-PC` and `RAveL-MB`, two algorithms for the discovery of elements in PC and MB, respectively, that control the FWER of their outputs without making any assumption on statistical power.

RAveL-MB follows the same overall approach used by previously proposed algorithms (e.g., *PCMB*, see Sect. 3): it first identifies elements in $PC(T)$ and adds them to $MB(T)$, and then tests the spouse condition on elements at distance 2 from T, that are variables $Y \in PC(X)$ with $X \in PC(T)$ and $Y \notin PC(T)$. The pseudocode of RAveL-MB is shown in Algorithm 5. RAveL-MB initializes MB to the output of the function RAveL-PC(T,$\mathbf{V}$,δ) (line 1), which returns a subset of $PC(T)$. For each element $X \in MB$ (line 2), RAveL-MB computes RAveL-PC(X,$\mathbf{V}$,δ) and, for every returned element Y that is not already in MB (line 3), an independence test of T on Y conditioning on $\mathbf{V}\backslash\{Y,T\}$ using function test_indep(T,Y,$\mathbf{V}\backslash\{Y,T\}$,$\delta$) is performed to test whether Y is a spouse of T with respect to X (line 4). If such test determines the conditional dependence between T and Y, then Y is added to MB (line 5). Finally, after analyzing all variables originally in MB, RAveL-MB outputs the set of elements in the Markov Boundary (line 6).

Note that the spouse condition is tested by conditioning only on the set $\mathbf{V}\backslash\{Y,T\}$. This is sufficient, since it is a set conditioned on which T and Y are d-connected if and only if Y is directly connected or is a spouse of T. In fact, if Y does not belong to any of these elements, then Y is connected to T through paths that contain chains or forks whose middle element is in $\mathbf{V}\backslash\{Y,T\}$. That is, Y is connected to T only through d-blocked paths.

Algorithm 5 RAveL-MB(T,$\mathbf{V}$,δ)

Input: target variable T, set $\mathbf{V}$ of variables, threshold $\delta \in (0,1]$
Output: A subset of $MB(T)$ with FWER lower than δ.

1 $MB \leftarrow$ RAveL-PC(T,$\mathbf{V}$,δ) ;
2 **foreach** $X \in MB$ **do**
3 **foreach** $Y \in$ *RAveL-PC*(X,$\mathbf{V}$,δ) ***and*** $Y \notin MB$ **do**
4 **if** ***not*** *test_indep*(T,Y,$\mathbf{V} \setminus \{Y,T\}$,$\delta$) **then**
5 $MB \leftarrow MB \cup \{Y\}$;
6 **return** MB;

RAveL-MB uses algorithm RAveL-PC(X,$\mathbf{V}$,δ) (shown in Algorithm 6) for the discovery of variables of a set $\mathbf{V}$ that are in $PC(X)$. The parameter δ controls the overall FWER of the procedure. RAveL-PC(X,$\mathbf{V}$,δ) identifies $PC(X)$ by using the definition of parent–children set, that is, $Y \in PC(X)$ gets returned if only if all independence tests between X and Y reject the null hypothesis.

Algorithm 6 RAveL-PC(T,$\mathbf{V}$,δ)

Input: target variable T, set $\mathbf{V}$ of variables, threshold $\delta \in (0, 1]$
Output: A subset of $PC(T)$ with FWER lower than δ.

```
PC ← V \ {T};
foreach X ∈ V \ {T} do
    foreach Z ⊆ V \ {X, T} do
        if test_indep(T, X, Z, δ) then
            PC ← PC \ {X};
return PC;
```

Both algorithms RAveL-MB and RAveL-PC employ a function, denoted as test_indep(X,Y,$\mathbf{Z}$,δ), that performs the independence test between $X, Y \in \mathbf{V}$ conditioning on $\mathbf{Z} \subseteq \mathbf{V}$ while controlling the FWER of *all testable hypotheses* with threshold δ, and returns true only if the null hypothesis gets rejected. Practical details on our implementation of test_indep(X,Y,$\mathbf{Z}$,δ) are provided in Sect. 4.3.

The following results prove that RAveL-PC and RAveL-MB control the FWER of PC and MB, respectively.

Theorem 7 RAveL-PC(T,$\mathbf{V}$,δ) *outputs a set of elements in PC*(T) *with FWER* $\leq \delta$.

Proof Note that the number of false positives of RAveL-PC(T,$\mathbf{V}$,δ) is greater than 0 if and only if there is at least one variable X of $\mathbf{V}\backslash\{T\}$ that is not in $PC(T)$ and is in the set PC reported by RAveL-PC(T,$\mathbf{V}$,δ). A variable X is returned in PC if and only if all independence tests between T and X (conditioning on the various sets $\mathbf{Z} \subseteq \mathbf{V}\backslash\{X, T\}$) reject the null hypothesis. Therefore RAveL-PC(T,$\mathbf{V}$,δ) reports a false positive only if at least one independence test returns a false positive, which happens with probability at most δ by definition of test_indep(T,X,$\mathbf{Z}$,δ). □

Theorem 8 RAveL-MB(T,$\mathbf{V}$,δ) *outputs a set of elements in MB*(T) *with FWER* $\leq \delta$.

Proof The set of RAveL-MB(T,$\mathbf{V}$,δ) output elements is the union of the set O_1 of variables returned by RAveL-PC(T,$\mathbf{V}$,δ), and the set O_2 of candidate spouses Y for which test_indep(T,Y,$\mathbf{V}\backslash\{Y, T\}$,$\delta$) rejects the null hypothesis. Then, a necessary condition to return a false positive is that at least one between sets O_1 and O_2 contains a false positive. The last event happens if and only if all calls to test_indep(T,X,$\mathbf{Z}$) returns at least a false positive, which happens with probability at most δ. □

The choice of $\mathbf{V}\backslash\{Y, T\}$ as conditioning set for testing the spouse condition is a consequence of RAveL-PC returning, with probability at least $1 - \delta$, a subset of $PC(T)$, and of any superset of $PC(T)$ allowing the discovery of spouses by RAveL-MB. We note that prior knowledge may be incorporated in the algorithm, if available, by conditioning on smaller set of variables, therefore increasing the precision of independence tests.

4.3 Rademacher averages for independence testing

Note that our algorithms `RAveL-PC` and `RAveL-MB` both rely on the availability of function `test_indep`$(X,Y,\mathbf{Z},\delta)$, which assesses the independence between $X, Y \in \mathbf{V}$ conditioning on $\mathbf{Z} \subseteq \mathbf{V}$ and returns `true` only if the null hypothesis gets rejected, while controlling the FWER of *all testable hypotheses* below a threshold δ.

The naïve implementation of `test_indep`$(X,Y,\mathbf{Z},\delta)$ would be to perform a standard statistical test (see Sect. 2.2) and use Bonferroni correction (see Sect. 2.3) to correct for multiple hypothesis testing. In particular, this requires to use a modified threshold δ/N for every hypothesis, where N is the maximum number of hypotheses that could be tested. Therefore, N is the maximum number of conditional independencies[2] between the variables in $\mathbf{V}$, that is $N = \mathbf{V}(\mathbf{V}-1)2^{\mathbf{V}-3}$. Note that the value of N grows exponentially with $\mathbf{V}$, leading to a Bonferroni correction which is very conservative and, therefore, to a high number of false negatives (independence tests between dependent variables for which the null hypothesis does not get rejected).

The high number of tests is not a feature of our algorithms only, but it is, in essence, shared by other widely used algorithms such as IAMB and PCMB (see Sect. 3). In fact, for both algorithms, the potential number of independence tests they perform can be as high as $N = \mathbf{V}(\mathbf{V}-1)2^{\mathbf{V}-3}$, even if a smaller number of tests may be considered in practice, depending on the output of the tests in previous steps, and a proper MHT correction depends on the maximum number of tests that could be performed.

Our solution to make our algorithms `RAveL-PC` and `RAveL-MB` practical is to implement `test_indep`$(X,Y,\mathbf{Z},\delta)$ exploiting Rademacher averages to obtain data-dependent bounds and confidence intervals. The key idea is to estimate confidence intervals around the empirical test statistics so that they contain the true values *simultaneously* with probability $1-\delta$. In this way, testing for independence corresponds to check whether a confidence interval contains the expected value of the test statistic under the null hypothesis of independence.

To implement the idea described above, we express Eq. 1 as an additive function on the samples as follows. Let us assume that the observations $\mathbf{x}$ of each variable X follow a probability distribution $\mathcal{X}$ taking values in $[-1,1]$ and with mean 0. (Alternatively, we assume the knowledge of the mean of $\mathcal{X}$, i.e. $\mu_{\mathcal{X}}$, and its maximum absolute value $\max_{\mathcal{X}}$,[3] and that all variables have been centered around 0 (i.e. by subtracting $\mu_{\mathcal{X}}$) and then normalized by dividing for $\max_{\mathcal{X}} - \mu_{\mathcal{X}}$.) The assumption on the mean being 0 is not necessary, as one could obtain analogous bounds also for other values of the mean. However, considering the mean to be 0, potentially after rescaling, leads to tight bounds on the SD (as the values for z in Theorem 1 are

[2] N counts, in fact, the total number of possible conditional independencies between any couple of variables by considering the symmetry property of independence tests, that is testing the (conditional) independence of X from Y is equivalent to testing the one of Y from X.

[3] Such knowledge may either come from knowledge about the generative process, or from previous estimates for such feature. In the latter case, we assume that those estimates are reliable representations of $\mu_{\mathcal{X}}$ and $\max_{\mathcal{X}}$.

lower for functions centered around 0). The assumption on the boundedness of each function is instead required by the analytical tools we use.

Let $s_1, s_2, \dots, s_k$ be the samples in the dataset $\mathcal{S} = \{s_1, s_2, \dots, s_k\}$, where each s_i is a collection of observations $s_i = \{v_1^i, v_2^i, \dots\}$ of variables in $\mathbf{V}$, where v_j^i is the observation of the j-th variable $V_j \in \mathbf{V}$ in sample s_i. Given two variables $X, Y \in \mathbf{V}$, and a set of variables $\mathbf{Z} \subset \mathbf{V}$, we define the following function $\tilde{r}_{X,Y,\mathbf{Z}}(s_i)$ on a sample s_i as

$$\tilde{r}_{X,Y,\mathbf{Z}}(s_i) = k\frac{x_i y_i}{k-1}, \tag{8}$$

where the conditioning set $\mathbf{Z}$ does not explicitly appear in the term $k\frac{x_i y_i}{k-1}$ but it is used in the definition of the values in $\mathbf{x}$ and $\mathbf{y}$ as in Sect. 2.2.

We then define the following modified version $\tilde{r}$ of Pearson's r coefficient, which we refer to as the *modified r statistic* (or *ModR*):

$$\tilde{r}_{X,Y,\mathbf{Z}} = \frac{1}{k}\sum_{i=1}^{k} \tilde{r}_{X,Y,\mathbf{Z}}(s_i). \tag{9}$$

By considering the family $\mathcal{F}$ of functions defined by $\tilde{r}_{X,Y,\mathbf{Z}}$ for each pair X, Y of variables and each set $\mathbf{Z} \subseteq \mathbf{V}\backslash\{X, Y\}$, we have that the n-MCERA (Eq. 5) is

$$\hat{R}_k^n(\mathcal{F}, \mathcal{S}, \boldsymbol{\sigma}) \doteq \frac{1}{n}\sum_{j=1}^{n} \sup_{\tilde{r}_{X,Y,\mathbf{Z}} \in \mathcal{F}} \frac{1}{k}\sum_{i=1}^{k} \sigma_{j,i}\tilde{r}_{X,Y,\mathbf{Z}}(s_i). \tag{10}$$

After the n-MCERA has been computed as above, we compute a bound $\mathcal{B}$ to the supremum deviation $D(\mathcal{F}, \mathcal{S})$ according to Theorem 1, which allows us to obtain confidence intervals around the empirical $\tilde{r}_{X,Y,\mathbf{Z}}$ as

$$CI_{X,Y,\mathbf{Z}} = \left[\tilde{r}_{X,Y,\mathbf{Z}} - \mathcal{B}, \tilde{r}_{X,Y,\mathbf{Z}} + \mathcal{B}\right] \tag{11}$$

with the guarantee that, *simultaneously* for all $\tilde{r}_{X,Y,\mathbf{Z}} \in \mathcal{F}$, $CI_{X,Y,\mathbf{Z}}$ contains the expected value of $\tilde{r}_{X,Y,\mathbf{Z}}$ with probability at least $1 - \delta$. Then, for a pair X, Y of variables and a set $\mathbf{Z} \subseteq \mathbf{V}\backslash\{X, Y\}$, we reject the null hypothesis of independence between X, Y conditioning on $\mathbf{Z}$ (i.e., `test_indep`(X,Y,$\mathbf{Z}$,δ) returns `true`) if $CI_{X,Y,\mathbf{Z}}$ does not contain the value 0. In practice, we replace the unknown quantities $\mu_{\mathcal{X}}$ and $\max_{\mathcal{X}}$ with their empirical estimates, that is, we replace $\mu_{\mathcal{X}}$ with the empirical sample mean $\bar{\mathbf{x}}$ and $\max_{\mathcal{X}}$ with $\max_{\mathbf{x}}$.

We finally propose another test statistic on a sample s_i, which we refer to as the *r-centered statistic* (or $\tilde{r}^c$), defined as

$$\tilde{r}^c_{X,Y,\mathbf{Z}}(s_i) = \frac{x_i y_i}{(max\{x_i, y_i\})^2} \tag{12}$$

where $\mathbf{x}$ and $\mathbf{y}$ are defined as previously (see Sect. 2.2) and are assumed to be centered around 0. The same independence testing procedure described for $\tilde{r}_{X,Y,\mathbf{Z}}$ applies for the empirical average of $\tilde{r}^c_{X,Y,\mathbf{Z}} = \frac{1}{k}\sum_{i=1}^{k} \tilde{r}^c_{X,Y,\mathbf{Z}}(s_i)$, since its expectation is zero under independence assumption and data centered around zero as follows.

Theorem 9 *Let $\mathcal{W}$ be the joint distribution of the variables X, Y, and $\mathbf{Z}$. If X and Y are independent, then $\mathbb{E}_{\mathcal{W}}[\tilde{r}^c_{X,Y,\mathbf{Z}}] = 0$.*

Proof We have that

$$\mathbb{E}_{\mathcal{W}}\left[\tilde{r}^c_{X,Y,\mathbf{Z}}\right] = \mathbb{E}_{\mathcal{W}}\left[\frac{1}{k}\sum_{i=1}^{k}\frac{x_i y_i}{(max\{x_i, y_i\})^2}\right]$$

which is proportional to $\mathbb{E}_{\mathcal{W}}[\hat{\mathbb{E}}_{\mathcal{S}}[XY]]$ (see Sect. 2.4 for definitions of $\mathbb{E}_{\mathcal{W}}$ and $\hat{\mathbb{E}}_{\mathcal{S}}$). Under the independence assumption, we have $\mathbb{E}_{\mathcal{W}}[\hat{\mathbb{E}}_{\mathcal{S}}[XY]] = \mathbb{E}_{\mathcal{W}}[\hat{\mathbb{E}}_{\mathcal{S}}[X]] \times \mathbb{E}_{\mathcal{W}}[\hat{\mathbb{E}}_{\mathcal{S}}[Y]]$, and the result follows since $\mathbb{E}_{\mathcal{W}}[\hat{\mathbb{E}}_{\mathcal{S}}[X]] = \mathbb{E}_{\mathcal{X}}[\hat{\mathbb{E}}_{\mathcal{S}}[X]] = 0$. □

5 Experimental evaluation

This section describes the experimental evaluation performed to empirically assess our algorithms. In Sect. 5.1 we compare `RAveL-PC` and `RAveL-MB` performances with other state-of-the-art methods on synthetic data. Section 5.2 present the analysis on two real world datasets (see the Appendix for details). We implemented[4] `RAveL-PC`, `RAveL-MB`, and the other algorithms considered in this section in Python 3. On each run we assumed no prior knowledge of the data distributions values for each variable X therefore we empirically normalized and centralized our observations using the empirical mean and maximum.[5] While the formal guarantees for our methods hold only assuming the knowledge of $\mu_{\mathcal{X}}$ and $\max_{\mathcal{X}}$, the experimental results in this section show that our methods still control the FWER below the desired threshold.

5.1 Synthetic data

We used synthetic data to evaluate `RAveL-PC` and `RAveL-MB` against state-of-the-art algorithms for the task of PC and MB discovery, respectively. In this scenario, each variable is a linear combination of its parents values plus Gaussian noise. The related structural model (shown in Fig. 1) is composed of 15 connected variables and n_{ext} external variables, and it is specified by two parameters: σ^2 which controls the amount of noise in the estimations, and n_{ext} which sets the number of external variables.

In these experiments we set the rejection threshold $\delta = 0.05$, which is a common value in literature, and we run each algorithm on increasing size datasets. We repeated

[4] Code available at https://github.com/VandinLab/RAveL.

[5] We empirically normalized each dataset $\mathcal{S}$ by dividing it by the highest absolute maximum value found among all datasets of the same size as $\mathcal{S}$. This method has been utilized to ensure a reliable estimation of the maximum value for each dataset size and reduce sampling variability.

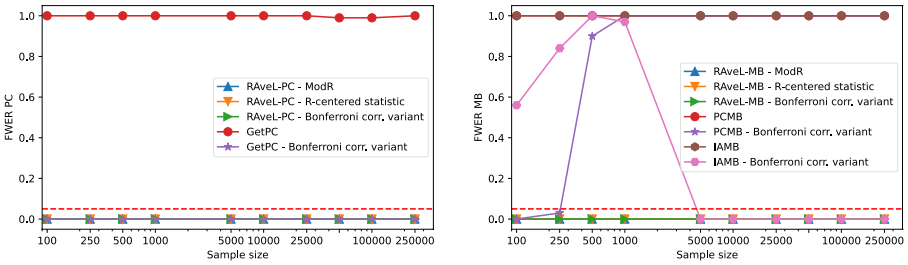

Fig. 2 Empirical FWER of various PC discovery (a) and MB discovery (b) algorithms on synthetic data for different sample sizes. FWER is the fraction of 100 trials in which at least one false positive is reported. The dashed line represents the bound $\delta = 0.05$ to the FWER used in the experiments

each trial 100 times and used $n = 1000$ for the n-MCERA. For each dataset, we considered all variables as target variable T in turn and run the algorithms for each choice of T. (Note that the number N of potential hypotheses tested is still the same as defined in Sect. 4.3.). Lastly, we limited our algorithms to consider only conditioning sets $\mathbf{Z}$ of at most 2 variables (except for the independence test at line 6 of `RAveL-MB`) for avoiding the analysis of all the exponential number N of hypotheses. We chose such value since each variable X is d-separated by each $Y \notin PC(X)$ by conditioning on a $\mathbf{Z}$ of size at most 2, and by running the algorithms on synthetic data allowing higher maximum sizes, we observed no differences in results w.r.t. the ones we are presenting.

In the first experiment, we compared different local causal discovery algorithms on the BN obtained setting $\sigma^2 = 1$ and $n_{ext} = 15$. For the PC discovery task, we compared two versions of *GetPC* (Pearl et al. 2007), the original one (without any correction for MHT) and one adaptation that uses Bonferroni correction, with three versions of `RAveL-PC`: one that uses the modified r statistic (or *ModR*) defined in Eq. 9, another that exploits $\tilde{r}^c$, and a variant of `RAveL-PC` that uses Bonferroni correction instead of Rademacher averages for MHT. Figure 2a shows the estimated FWER of each method (that is, the fraction of trials in which at least a false positive is reported). The results confirm our analysis in Sect. 4.2, and we observe that, for the specific BN we consider, the adaptation of *GetPC* that uses Bonferroni correction has FWER below the threshold, even if this is not guaranteed from our theoretical analysis.

For the MB discovery task, we compared two versions of *PCMB* (Pearl et al. 2007) and of *IAMB* (Tsamardinos et al. 2003), the original ones (without any correction for MHT) and two adaptations that use Bonferroni correction, with three versions of `RAveL-MB`: one that uses the modified r statistic defined in Eq. 9, another that exploits $\tilde{r}^c$, and a variant of `RAveL-MB` that uses Bonferroni correction instead of Rademacher averages for MHT. Figure 2b shows the FWER of each method. The results confirms `RAveL-MB` (with both statistics) and its variant to be the only algorithms with guarantees on the FWER at any sample size, that is without infinite power assumption. Moreover, note that *PCMB* reports false positives with high probability even if its PC discovery method *GetPC* does not. This is due to elements at distance 2 from T that are correctly identified as candidate spouses, but for which the spouse condition used by *PCMB* results in a false positive due to false negatives in $PC(T)$, as described in Sect. 4.1.3 (scenario 3).

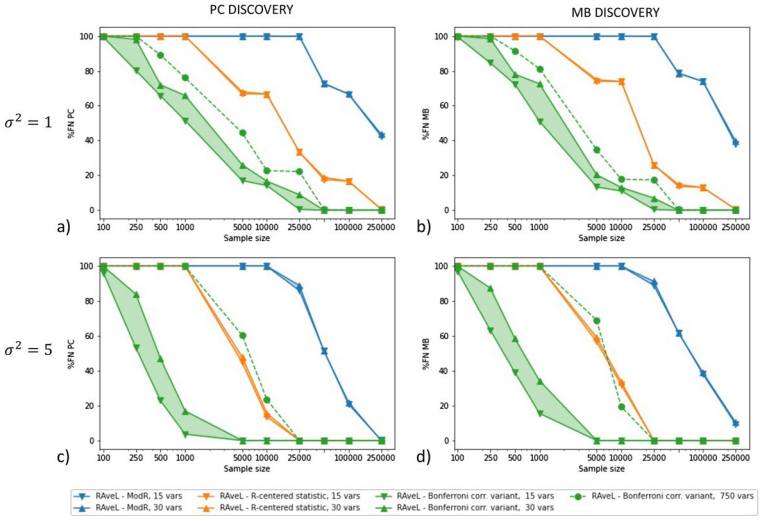

Fig. 3 Empirical FN% of `RAveL-PC` (**a,c**) and `RAveL-MB` (**b,d**) on synthetic data for different sample sizes in two data generative scenarios. We sampled data from Fig. 1 in two scenarios with different noise level: $\sigma^2 = 1$ for (**a,b**), and $\sigma^2 = 5$ for (**c,d**). FN% is the mean percentage of false negatives out of 100 trials. In each experiment we compared the approach that uses the Pearson's R test with Bonferroni correction, and two implementations that exploits Rademacher averages, one using the modified r statistic *ModR* defined in Eq. 9, and another with $\tilde{r}^c$. Solid lines represent experiments on datasets with $n_{ext} = 0$ and $n_{ext} = 15$, and performance gaps between the two are highlighted. Dashed lines show simulated results on datasets with $n_{ext} = 750$

We then assessed the fraction of false negatives for our algorithms, which are the only ones with guarantees on the FWER, on datasets with sample sizes up to 250,000 elements by repeating each trial 100 times. Figure 3 summarizes (with solid lines) these results on a scenario with $\sigma^2 = 1$ (in Fig. 3a, b) and another with $\sigma^2 = 5$ (in Fig. 3c, d). For each setting, we run the algorithms by considering a different number of variables ($n_{ext} = 0$ and $n_{ext} = 15$), and we highlighted the difference in performances between the two cases. The results show how the approaches based on Rademacher averages do not suffer from the addition of external variables (i.e. their FN% are equivalent), as opposed to the versions of `RAveL-PC` and `RAveL-MB` that exploit the Bonferroni correction, whose performances degrade by increasing the number of variables under analysis. Both behaviors are expected as the Bonferroni correction becomes stricter since the number N of hypotheses to test increases (see Sect. 2.3), while the bound to the supremum deviation remains stable as the complexity of the function class $\mathcal{F}$ does not increase.[6] Motivated by these observations, we simulated the performances of `RAveL-PC` and `RAveL-MB` variants that exploit Bonferroni correction in a high-dimensional scenario with 750 total variables, and we reported them as well in Fig. 3 (dashed lines).

Figure 3a, b shows differences between the approach that exploits Rademacher averages with the modified r statistic defined in Eq. 9 and the one that exploits $\tilde{r}^c$, with the FN% of the first one decreasing for datasets with more than 10,000 samples

[6] The most complex statistics in $\mathcal{F}$ are in fact the ones for which there is indepencence between **x** and **y**, that are the ones with the highest variance.

and the latter one just at 5000 samples. Such difference is due to the normalization procedure applied to the data for using the former test statistic (see Sect. 4.3). Such procedure allows us to bound the test statistic (and therefore to use the Rademacher averages) but it also lowers the test statistic value as the sample size increases (since it will increase the chances of observing more extreme values) degrading the statistical power and requiring more accurate estimates of the bound $\mathcal{B}$ to the supremum $D(\mathcal{F}, S)$. Despite lowering the test statistic and degrading the statistical power, however, such procedure does not lead to any false positive in output, as our algorithms are correct without requiring any infinite power assumption. $\tilde{r}^c$ instead is not affected by such issue and shows higher statistical power, highlighting the importance of the choice of the test statistic. From Fig. 3a, b we also observe that the use of Bonferroni correction leads to a high statistical power, even with a high number of variables, in the $\sigma^2 = 1$ scenario. Such trend does not hold when $\sigma^2 = 5$ and the dimensionality is high (Fig. 3c, d), for which `RAveL-PC` and `RAveL-MB` that exploit $\tilde{r}^c$ have more statistical power than algorithmic variants with Bonferroni correction.

5.2 Real datasets

We tested our algorithms on the Boston housing dataset (Harrison and Rubinfeld 1978), which contains data about house prices in Boston suburbs, considering the median price of homes in each suburb as target T. Since the number of variables for such dataset is small, we used the Bonferroni variant of our algorithms `RAveL-PC` and `RAveL-MB`, with $\delta = 0.01$. Given the small number of observations (506 samples), we limited our analysis to conditioning sets $\mathbf{Z}$ of size at most 2 for maintaining a high statistical power in the independence testing. Both algorithms reported in output two variables, one related to the number of rooms per house, and the other to the median income of the suburb residents, that clearly influence the median price of the houses in the neighborhood. The first variable is a common indicator of the price of a house, while the second confirms the intuition that between two identical houses, the one built in a wealthier neighborhood has a higher price.

We finally tested our algorithms on the Framingham dataset (see Appendix 2), that provides information about the development of coronary heart disease (CHD) in 10 years for 3656 citizens of the city of Framingham, with 16 features describing health status and lifestyle. Given the relatively small number of samples, we limited our analysis to conditioning sets $\mathbf{Z}$ of size at most 2 for maintaining an high statistical power in the independence testing. We preprocessed the dataset by removing samples with missing data and binary features that were highly unbalanced, for which therefore we would not have had enough statistical power to test our assumptions.[7] We tested `RAveL-PC` and `RAveL-MB` variants using Bonferroni correction with $\delta = 0.05$ and got in output, for both discovery tasks, three variables: *Age*, *Systolic Blood Pressure*, and *Glucose*. Such results are supported by the World Health Organization guidelines.[8] Overall, our results on real data provide empirical evidence that our algorithms identify meaningful causal relations while avoiding false positives.

[7] Dataset and preprocessing information on the Appendix.

[8] More information available on the official site https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds) [Accessed: March 2023].

6 Conclusions

In this paper we presented two algorithms, `RAveL-PC` and `RAveL-MB`, for the task of local causal discovery. In contrast to state-of-the-art approaches, our algorithms provide guarantees on false discoveries in terms of bounding the FWER. Our algorithms use Rademacher averages to to properly account for multiple hypothesis testing, and our experimental evaluation shows that our algorithms properly control for false discoveries. Our algorithms can be extended to other (e.g., non-linear) test statistics and to other tests. In particular, Rademacher averages provide appealing time-effective alternatives for independence testing with test statistics whose distributions are unknown, since in such scenarios a typical solution is to rely on permutation testing, which require to analyze a large number of permuted datasets in order to achieve high statistical power. Interesting research directions include the application of our framework to recently proposed independence tests (Bellot and van der Schaar 2019), improving the efficiency of our algorithms, and exploiting them for structure discovery. An additional direction for future research is to relax our framework assumptions. In particular, the assumption on knowledge of the mean of each variable may be relaxed by considering empirical centralization (Cousins and Riondato 2020) (i.e., by subtracting the *observed* mean in the data).

Appendix 1: Variables in Boston housing dataset

Variables description follows from the paper describing the dataset (Harrison and Rubinfeld 1978).

Variable name	Explanation
CRIM	Per capita crime rate by town
ZN	Proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	Proportion of non-retail business acres per town.
CHAS	Charles River dummy variable (1 if tract bounds river; 0 otherwise)
NOX	Nitric oxides concentration (parts per 10 million)
RM	Average number of rooms per dwelling
AGE	Proportion of owner-occupied units built prior to 1940
DIS	Weighted distances to five Boston employment centres
RAD	Index of accessibility to radial highways
TAX	Full-value property-tax rate per $10,000
PTRATIO	Pupil-teacher ratio by town
B	$1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town
LSTAT	% lower status of the population
MEDV	Median value of owner-occupied homes in $1000's

Appendix 2: Framingham dataset

Dataset and variable description are taken from https://www.kaggle.com/datasets/dileep070/heart-disease-prediction-using-logistic-regression. Variables "CurrentSmoker", "PrevalentStroke", "PrevalentHyp", and "Diabetes" were removed in the data preprocessing phase.

Variable name	Explanation
Age	Age of the patient (Continuous - Although the recorded ages have been truncated to whole numbers, the concept of age is continuous)
Current Smoker	Whether or not the patient is a current smoker (Nominal)
Cigs Per Day	The number of cigarettes that the person smoked on average in one day. (can be considered continuous as one can have any number of cigarettes, even half a cigarette.)
BP Meds	Whether or not the patient was on blood pressure medication (Nominal)
Prevalent Stroke	Whether or not the patient had previously had a stroke (Nominal)
Prevalent Hyp	Whether or not the patient was hypertensive (Nominal)
Diabetes	Whether or not the patient had diabetes (Nominal)
Tot Chol	Total cholesterol level (Continuous)
Sys BP	Systolic blood pressure (Continuous)
Dia BP	Siastolic blood pressure (Continuous)
BMI	Body Mass Index (Continuous)
Heart Rate	Heart rate (Continuous - In medical research, variables such as heart rate though in fact discrete, yet are considered continuous because of large number of possible values.)
Glucose	Glucose level (Continuous)
10 year risk of coronary heart disease CHD	Binary: "1", means "Yes", "0" means "No"

Acknowledgements This work is supported, in part, by the Italian Ministry of University and Research (MUR), under PRIN Project No. 2022TS4Y3N - EXPAND (scalable algorithms for EXPloratory Analyses of heterogeneous and dynamic Networked Data) and the initiative "Departments of Excellence" (Law 232/2016).

Author contributions D.S. and F.V. wrote the main manuscript text.

Funding Open access funding provided by Università degli Studi di Padova within the CRUI-CARE Agreement.

Declarations

Competing interests The authors declare no competing interests.

References

Aliferis CF, Statnikov A, Tsamardinos I, Mani S, Koutsoukos XD (2010) Local causal and Markov blanket induction for causal discovery and feature selection for classification part i: algorithms and empirical evaluation. JMLR 11(1):171–234

Aliferis CF, Tsamardinos I, Statnikov A (2003) Hiton: a novel Markov blanket algorithm for optimal variable selection. In: Proceedings of AMIA, pp 21–25

Armen AP, Tsamardinos I (2014) Estimation and control of the false discovery rate of Bayesian network skeleton identification. Tech. rep., TR-441. U. of Crete, pp 1–79

Bartlett PL, Mendelson S (2002) Rademacher and Gaussian complexities: risk bounds and structural results. JMLR 3:463–482

Bellot A, van der Schaar M (2019) Conditional independence testing using generative adversarial networks. In: Advances in neural information processing systems, 32, pp 1-11

Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc Ser B 57(1):289–300

Benjamini Y, Yekutieli D (2001) The control of the false discovery rate in multiple testing under dependency. Ann Stat 29(4):1165–1188

Bielza C, Larranaga P (2014) Bayesian networks in neuroscience: a survey. Front Comput Neurosci 8(131):1–23

Bonferroni C (1936) Teoria statistica delle classi e calcolo delle probabilita. Istituto Superiore di Scienze Economiche e Commericiali di Firenze 8:3–62

Cousins C, Riondato M (2020) Sharp uniform convergence bounds through empirical centralization. In: Advances in Neural Information Processing Systems 33, pp 15123-15132

Harrison D Jr, Rubinfeld DL (1978) Hedonic housing prices and the demand for clean air. J Environ Econ Manag 5(1):81–102

Koltchinskii V, Panchenko D (2000) Rademacher processes and bounding the risk of function learning. In: High dimensional probability II, Birkhäuser Boston, pp 443–457

Kusner MJ, Loftus JR (2020) The long road to fairer algorithms. Nature 578(7793):34–36

Li J, Wang ZJ (2009) Controlling the false discovery rate of the association/causality structure learned with the pc algorithm. J Mach Learn Res 10:475–514

Liu A, Li J, Wang ZJ, McKeown MJ (2012) A computationally efficient, exploratory approach to brain connectivity incorporating false discovery rate control, a priori knowledge, and group inference. Comput Math Methods Med 2012:1–14

Ma S, Tourani R (2020) Predictive and causal implications of using Shapley value for model interpretation. KDD Workshop on Causal Discovery, PMLR 2020, pp 23–28

Mhasawade V, Chunara R (2021) Causal multi-level fairness. In: Proceedings of the AAAI/ACM conference on AI, ethics, and society, pp 784–794

Mitzenmacher M, Upfal E (2017) Probability and computing, 2nd edn. Cambridge University Press, Cambridge

Neapolitan RE et al (2004) Learning Bayesian networks. Pearson Prentice Hall, Boston

Pearl J (2009) Causality, 2nd edn. Cambridge University Press, Cambridge

Pe'er D (2005) Bayesian network analysis of signaling networks: a primer. Science's STKE 2005(281):1–12

Pellegrina L, Cousins C, Vandin F, Riondato M (2022) Mcrapper: Monte-Carlo Rademacher averages for poset families and approximate pattern mining. ACM Trans Knowl Discov Data 16(6):1–29

Pellegrina L, Vandin F (2023) Silvan: estimating betweenness centralities with progressive sampling and non-uniform Rademacher bounds. ACM Trans Knowl Discov Data 18(3):1–55

Pena JM, Nilsson R, Björkegren J, Tegnér J (2007) Towards scalable and data efficient learning of Markov boundaries. Int J Approx Reason 45(2):211–232

Riondato M, Upfal E (2015) Mining frequent itemsets through progressive sampling with rademacher averages. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, p 1005–1014

Riondato M, Upfal E (2018) Abra: approximating betweenness centrality in static and dynamic graphs with Rademacher averages. ACM Trans Knowl Discov Data 12(5):1–38

Sachs K, Perez O, Pe'er D, Lauffenburger DA, Nolan GP (2005) Causal protein-signaling networks derived from multiparameter single-cell data. Science 308(5721):523–529

Santoro D, Tonon A, Vandin F (2020) Mining sequential patterns with vc-dimension and Rademacher complexity. Algorithms 13(5), 123:1-34

Shah RD, Peters J (2020) The hardness of conditional independence testing and the generalised covariance measure. Ann Stat 48(3):1514–1538

Spirtes P, Glymour CN, Scheines R, Heckerman D (2000) Causation, prediction, and search. MIT Press, Cambridge

Strobl EV, Spirtes PL, Visweswaran S (2019) Estimating and controlling the false discovery rate of the pc algorithm using edge-specific p-values. ACM Intell Syst Technol 10(5):1–37

Tsamardinos I, Aliferis CF (2003) Towards principled feature selection: relevancy, filters and wrappers. In: Proceeding of the 9th international workshop on artificial intelligence and statistics, PMLR, p 300–307

Tsamardinos I, Aliferis CF, Statnikov A (2003) 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, p 673–678

Tsamardinos I, Aliferis CF, Statnikov AR, Statnikov E (2003) Algorithms for large scale Markov blanket discovery. In: Proceedings of the 16th international FLAIRS conference, p 376–381

Tsamardinos I, Brown LE (2008) Bounding the false discovery rate in local Bayesian network learning. In: Proceedings of the 23rd AAAI conference on artificial intelligence, p 1100–1105

Velikova M, van Scheltinga JT, Lucas PJ, Spaanderman M (2014) Exploiting causal functional relationships in Bayesian network modelling for personalised healthcare. Int J Approx Reason 55(1):59–73

Yusuf F, Cheng S, Ganapati S, Narasimhan G (2021) Causal inference methods and their challenges: the case of 311 data. In: Proceedings of the 22nd annual international conference on digital government research, p 49–59