Journal of Machine Learning Research 10 (2009) 475-514 Submitted 3/08; Revised 7/08; Published 2/09

Controlling the False Discovery Rate of the Association/Causality Structure Learned with the PC Algorithm

Junning Li JUNNINGL@ECE.UBC.CA
Z. Jane Wang ZJANEW@ECE.UBC.CA
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, BC, V6T 1Z4
Canada

Editors: Paolo Frasconi, Kristian Kersting, Hannu Toivonen and Koji Tsuda

Abstract

In real world applications, graphical statistical models are not only a tool for operations such as classification or prediction, but usually the network structures of the models themselves are also of great interest (e.g., in modeling brain connectivity). The false discovery rate (FDR), the expected ratio of falsely claimed connections to all those claimed, is often a reasonable error-rate criterion in these applications. However, current learning algorithms for graphical models have not been adequately adapted to the concerns of the FDR. The traditional practice of controlling the type I error rate and the type II error rate under a conventional level does not necessarily keep the FDR low, especially in the case of sparse networks. In this paper, we propose embedding an FDR-control procedure into the PC algorithm to curb the FDR of the skeleton of the learned graph. We prove that the proposed method can control the FDR under a user-specified level at the limit of large sample sizes. In the cases of moderate sample size (about several hundred), empirical experiments show that the method is still able to control the FDR under the user-specified level, and a heuristic modification of the method is able to control the FDR more accurately around the user-specified level. The proposed method is applicable to any models for which statistical tests of conditional independence are available, such as discrete models and Gaussian models.

Keywords: Bayesian networks, false discovery rate, PC algorithm, directed acyclic graph, skeleton

1. Introduction

Graphical models have attracted increasing attention in the fields of data mining and machine learning in the last decade. These models, such as Bayesian networks (also called belief networks) and Markov random fields, generally represent events or random variables as vertices (also referred to as nodes), and encode conditional-independence relationships among the events or variables as directed or undirected edges (also referred to as arcs) according to the Markov properties (see Lauritzen, 1996, Chapt. 3). Of particular interest here are Bayesian networks (see Pearl, 1988, Chapt. 3.3) that encode conditional-independence relationships according to the directed Markov property (see Lauritzen, 1996, pages 46–53) with directed acyclic graphs (DAGs) (i.e., graphs with only directed edges and with no directed cycles). The directed acyclic feature facilitates the computation of Bayesian networks because the joint probability can be factorized recursively into many local conditional probabilities.

As a fundamental and intuitive tool to analyze and visualize the association and/or causality relationships among multiple events, graphical models have become more and more explored in biomedical researches, such as discovering gene regulatory networks and modelling functional connectivity between brain regions. In these real world applications, graphical models are not only a tool for operations such as classification or prediction, but often the network structures of the models themselves are also output of great interest: a set of association and/or causality relationships discovered from experimental observations. For these applications, a desirable structure-learning method needs to account for the error rate of the graphical features of the discovered network. Thus, it is important for structure-learning algorithms to control the error rate of the association/causality relationships discovered from a limited number of observations closely below a user-specified level, in addition to finding a model that fits the data well. As edges are fundamental elements of a graph, error rates related to them are of natural concerns.

In a statistical decision process, there are basically two sources of errors: the type I errors, that is, falsely rejecting negative hypotheses when they are actually true; and the type II errors, that is, falsely accepting negative hypotheses when their alternatives, the positive hypotheses are actually true. In the context of learning graph structures, a negative hypothesis could be that an edge does not exist in the graph while the positive hypothesis could be that the edge does exist. Because of the stochastic nature of random sampling, data of a limited sample size may appear to support a positive hypothesis more than a negative hypothesis even when actually the negative hypothesis is true, or vice versa. Thus it is generally impossible to absolutely prevent the two types of errors simultaneously, but has to set a threshold on a certain type of errors, or keep a balance between the them, for instance by minimizing a certain lost function associated with the errors according to the Bayesian decision theory. For example, when diagnosing cancer, to catch the potential chance of saving a patient's life, doctors probably hope that the type II error rate, that is, the probability of falsely diagnosing a cancer patient as healthy, to be low, such as less than 5%. Meanwhile, when diagnosing a disease whose treatment is so risky that may cause the loss of eyesight, to avoid the unnecessary but great risk for healthy people, doctors probably hope that the type I error rate, that is, the probability of falsely diagnosing a healthy people as affected by the disease, to be extremely low, such as less than 0.1%. Learning network structures may face scenarios similar to the two cases above of diagnosing diseases. Given data of a limited sample size, there is not an algorithm guaranteeing a perfect recovery of the structure of the underlying graphical model, and any algorithm has to compromise on the two types of errors.

For problems involving simultaneously testing multiple hypotheses, such as verifying the existence of edges in a graph, there are several different criteria for their error-rate control (see Table 2), depending on researchers' concerns or the scenario of the study. Generally there are not mathematically or technically superior relationships among different error-rate criteria if the research scenario is not specified. One error-rate criterion may be favoured in one scenario while another criterion may be right of interest in a different scenario, just as the aforementioned examples of diagnosing diseases. In real world applications, selecting the error rate of interest is largely not an abstract question "which error rate is superior over others?", but a practical question "which error rate is the researchers' concern?" For extended discussions on why there are not general superior relationships among different error-rate criteria, please refer to Appendix C, where examples of typical research scenarios, accompanied by theoretical discussions, illustrate that each of the four error-rate criteria in Table 2 may be favoured in a certain study.

The false discovery rate (FDR) (see Benjamini and Yekutieli, 2001; Storey, 2002), defined as the expected ratio of falsely discovered positive hypotheses to all those discovered, has become an important and widely used criterion in many research fields, such as bioinformatics and neuroimaging. In many real world applications that involve multiple hypothesis testing, the FDR is more reasonable than the traditional type I error rate and type II error rate. Suppose that in a pilot study researchers are selecting candidate genes for a genetic research on schizophrenia. Due to the limited funding, only a limited number of genes can be studied thoroughly in the afterward genetic research. To use the funding efficiently, researchers would hope that 95% of the candidate genes selected in the pilot study are truly associated with the disease. In this case, the FDR is chosen as the error rate of interest and should be controlled under 5%. Simply controlling the type I error rate and the type II error rate under certain levels does not necessarily keep the FDR sufficiently low, especially in the case of sparse networks. For example, suppose a gene regulatory network involves 100 genes, where each gene interacts in average with 3 others, that is, there are 150 edges in the network. Then an algorithm with the *realized* type I error rate = 5% and the *realized* power = 90% (i.e., the *realized* type II error rate = 10%) will recover a network with $150\times 90\%=135$ correct connections and $[100\times(100-1)/2-150]\times 5\%=240$ false connections. This means that $240/(240+135)=64\%$ of the claimed connections actually do not exist in the true network. Due to the popularity of the FDR in research practices, it is highly desirable to develop structure-learning algorithms that allow the control over the FDR on network structures.

However, current structure-learning algorithms for Bayesian networks have not been adequately adapted to explicitly controlling the FDR of the claimed "discovered" networks. Score-based search methods (see Heckerman et al., 1995) look for a suitable structure by optimizing a certain criterion of goodness-of-fit, such as the Akaike information criterion (AIC), the Bayesian information criterion (BIC), or the Bayesian Dirichlet likelihood equivalent metric (BDE), with a random walk (e.g., simulated annealing) or a greedy walk (e.g., hill-climbing), in the space of DAGs or their equivalence classes.[1] It is worth noting that the restricted case of tree-structured Bayesian networks has been optimally solved, in the sense of Kullback-Leibler divergence, with Chow and Liu (1968)'s method, and that Chickering (2002) has proved that the greedy equivalence search can identify the true equivalence class in the limit of large sample sizes. Nevertheless, scores do not directly reflect the error rate of edges, and the sample sizes in real world applications are usually not large enough to guarantee the perfect asymptotic identification.

The Bayesian approach first assumes a certain prior probability distribution over the network structures, and then estimates the posterior probability distribution of the structures after data are observed. Theoretically, the posterior probability of any structure features, such as the existence of an edge, the existence of a path, or even the existence of a sub-graph, can be estimated with the Bayesian approach. This consequently allows the control of the posterior error rate of these structure features, that is, the posterior probability of the non-existence of these features. It should be pointed out that the posterior error rate is conceptually different from those error rates such as the type I error rate, the type II error rate, and the FDR, basically because they are from different statistical perspectives. The posterior error rate is defined from the perspective of Bayesian statistics. From the Bayesian perspective, network structures are assumed to be random, according to a probability distribution, and the posterior error rate is the probability of the non-existence of certain features according to the posterior probability distribution over the network structures. Given the same data,

1. An equivalence class of DAGs is a set of DAGs that encode the same set of conditional-independence relationships according to the directed Markov property.

different posterior distributions will be derived from different prior distributions. The type I error rate, the type II error rate, and the FDR are defined from the perspective of classical statistics. From the classical perspective, there is a true, yet unknown, model behind the data, and the error rates are defined by comparing with the true model. Nevertheless, a variant of the FDR, the positive false discovery rate (pFDR) proposed by Storey (2002), can be interpreted from the Bayesian perspective (Storey, 2003).

The Bayesian approach for structure learning is usually conducted with the maximum-*a posteriori*-probability (MAP) method or the posterior-expectation method. The MAP method selects the network structure with the largest posterior probability. The optimal structure is usually searched for in a score-based manner, with the posterior probability or more often approximations to the relative posterior probability (for instance the BIC score) being the score to optimize. Cooper and Herskovits (1992) developed a heuristic greedy search algorithm called K2[2] that can finish the search in a polynomial time with respect to the number of vertices, given the order of vertices. The MAP method provides us with a single network structure, the posteriorly most probable one, but does not address error rates in the Bayesian approach.

To control errors in the Bayesian approach, the network structure should be learned with the posterior-expectation method, that is, calculating the posterior probabilities of network structures, and then deriving the posterior expectation of the existence of certain structure features. Though theoretically the posterior-expectation method can control the error rate of any structure features, in practice its capacity is largely limited for computational reasons. The number of DAGs increases super-exponentially as the number of vertices increases (Robinson, 1973). For 10 vertices, there are already about 4.2×10^{18} DAGs. Though the number of equivalence classes of DAGs is much smaller than the number of DAGs, it is still forbiddingly large, empirically asymptotically decreasing to 1/13.652 of the number of DAGs, as the number of vertices increases (Steinsky, 2003). Therefore, exact inferences of posterior probabilities are only feasible for small scale problems, or under certain additional constraints. For certain prior distributions, and given the order of vertices, Friedman and Koller (2003) have derived a formula that can be used to calculate the exact posterior probability of a structure feature with the computational complexity bounded by $O(N^{D_{in}+1})$, where N is the number of vertices and D_{in} is the upper bound of the in-degree for each vertex. Considering similar prior distributions, but without the restriction on the order of vertices, Koivisto and Sood (2004) have developed a fast exact Bayesian inference algorithm based on dynamic programming that is able to compute the exact posterior probability of a sub-network with the computational complexity bounded by $O(N2^N + N^{D_{in}+1}L(m))$, where $L(m)$ is the complexity of computing a marginal conditional likelihood from m samples. In practice, this algorithm runs fairly fast when the number of vertices is less than 25. For networks with more than 30 vertices, the authors suggested setting more restrictions or combining with inexact techniques. These two breakthroughs made exact Bayesian inferences practical for certain prior distributions. However, as Friedman and Koller (2003) pointed out, the prior distributions which facilitate the exact inference are not hypothesis equivalent (see Heckerman et al., 1995), that is, different network structures that are in the same equivalence class often have different priors. The simulation performed by Eaton and Murphy (2007) confirmed that these prior distributions deviate far from the uniform distributions. This implies that the methods cannot be applied to the widely accepted uninformative prior, that is, the uniform prior distribution over DAGs. For general problems, the posterior probability of a structure feature can be approx-

2. The algorithm is named K2 because it evolved from a system named Kutato (Herskovits and Cooper, 1990).

imated with Markov chain Monte Carlo (MCMC) methods (Madigan et al., 1995). As a versatile implementation of Bayesian inferences, the MCMC method can estimate the posterior probability given any prior probability distribution. However, MCMC usually requires intensive computation and results may depend on the initial state of the randomization.

Constraint-based approaches, such as the SGS[3] algorithm (see Spirtes et al., 2001, pages 82–83), inductive causation (IC)[4] (see Pearl, 2000, pages 49–51), and the PC[5] algorithm (see Spirtes et al., 2001, pages 84–89), are rooted in the directed Markov property, the rule by which Bayesian networks encode conditional independence. These methods first test hypotheses of conditional independence among random variables, and then combine those accepted hypotheses of conditional independence to construct a partially directed acyclic graph (PDAG) according to the directed Markov property. The computational complexity of these algorithms is difficult to analyze exactly, though for the worst case, which rarely occurs in real world applications, is surely bounded by $O(N^2 2^N)$ where N is the number of vertices. In practice, the PC algorithm and the fast-causal-inference (FCI) algorithm (see Spirtes et al., 2001, pages 142–146) can achieve polynomial time if the maximum degree of a graph is fixed. It has been proved that if the true model satisfies the faithfulness constraints (see Spirtes et al., 2001, pages 13 and 81) and all the conditional-independence/dependence relationships are correctly identified, then the PC algorithm and the IC algorithm can exactly recover the true equivalence class, and so do the FCI algorithm and the IC* algorithm[6] (see Pearl, 2000, pages 51–54) for problems with latent variables. Kalisch and Bühlmann (2007) have proved that for Gaussian Bayesian networks, the PC algorithm can consistently estimate the equivalence class of an underlying sparse DAG as the sample size m approaches infinity, even if the number of vertices N grows as fast as $O(m^\lambda)$ for any $0 < \lambda < \infty$. Yet, as in practice hypotheses of conditional independence are tested with statistical inference from limited data, false decisions cannot be entirely avoided and thus the ideal recovery cannot be achieved. In current implementations of the constraint-based approaches, the error rate of testing conditional independence is usually controlled individually for each test, under a conventional level such as 5% or 1%, without correcting the effect of multiple hypothesis testing. Therefore these implementations may fail to curb the FDR, especially for sparse graphs.

In this paper, we propose embedding an FDR-control procedure into the PC algorithm to curb the error rate of the skeleton of the learned PDAGs. Instead of individually controlling the type I error rate of each hypothesis test, the FDR-control procedure considers the hypothesis tests together to correct the effect of simultaneously testing the existence of multiple edges. We prove that the proposed method, named as the PC_{fdr}-skeleton algorithm, can control the FDR under a user-specified level at the limit of large sample sizes. In the case of moderate sample sizes (about several hundred), empirical experiments show that the method is able to control the FDR under the user-specified level, and a heuristic modification of the method is able to control the FDR more accurately around the user-specified level. Schäfer and Strimmer (2005) have applied an FDR procedure to graphical Gaussian models to control the FDR of the non-zero entries of the partial correlation matrix. Different from Schäfer and Strimmer (2005)'s work, our method, built within the frame-

3. "SGS" stands for **S**pirtes, **G**lymour and **S**cheines who invented this algorithm.
4. An extension of the IC algorithm which was named as IC* (see Pearl, 2000, pages 51–54) was previously also named as IC by Pearl and Verma (1992). Here we follow Pearl (2000).
5. "PC" stands for **P**eter Spirtes and **C**lark Glymour who invented this algorithm. A modified version of the PC algorithm which was named as PC* (see Spirtes et al., 2001, pages 89–90) was previously also named as PC by Spirtes and Glymour (1991). Here we follow Spirtes et al. (2001).
6. See footnote 4.

work of the PC algorithm, is not only applicable to the special case of Gaussian models, but also generally applicable to any models for which conditional-independence tests are available, such as discrete models.

We are particularly interested in the PC algorithm because it roots in conditional-independence relationships, the backbone of Bayesian networks, and p-values of hypothesis testing represent one type of error rates. We consider the skeleton of graphs because constraint-based algorithms usually first construct an undirected graph, and then annotate it into different types of graphs while keeping the skeleton as the same as that of the undirected one.

The PC_{fdr}-skeleton algorithm is not designed to replace or claimed to be superior over the standard PC algorithm, but provide the PC algorithm with the ability to control the FDR over the skeleton of the recovered network. The PC_{fdr}-skeleton algorithm controls the FDR while the standard PC algorithm controls the type I error rate, as illustrated in Section 3.1. Since there are no general superior relationships between different error-rate criteria, as explained in Appendix C, neither be there between the PC_{fdr}-skeleton algorithm and the standard PC algorithm. In research practices, researchers first decide which error rate is of interest, and then choose appropriate algorithms to control the error rate of interest. Generally they will not select an algorithm that sounds "superior" but controls the wrong error rate. Since the purpose of the paper is to provide the PC algorithm with the control over the FDR, we assume in this paper that the FDR has been selected as the error rate of interest, and selecting the error rate of interest is out of the scope of the paper.

The remainder of the paper is organized as follows. In Section 2, we review the PC algorithm, present the FDR-embedded PC algorithm, prove its asymptotic performance, and analyze its computational complexity. In Section 3, we evaluate the proposed method with simulated data, and demonstrate its real world applications to learning functional connectivity networks between brain regions using functional-magnetic-resonance-imaging (fMRI). Finally, we discuss the advantages and limitations of the proposed method in Section 4.

2. Controlling FDR with PC Algorithm

In this section, we first briefly introduce Bayesian networks and review the PC algorithm. Then, we expatiate on the FDR-embedded PC algorithm and its heuristic modification, prove their asymptotic performances, and analyze their computational complexity. At the end, we discuss other possible ideas of embedding FDR control into the PC algorithm.

2.1 Notations and Preliminaries

To assist the reading, notations frequently used in the paper are listed as follows:

$a,\ b,\ \cdots$	: vertices
$X_a,\ X_b,\ \cdots$	: variables respectively represented by vertices $a,\ b,\ \cdots$
$A,\ B,\ \cdots$	: vertex sets
$X_A,\ X_B,\ \cdots$	: variable sets respectively represented by vertex sets $A,\ B,\ \cdots$
V	: the vertex set of a graph
$N = \lvert V \rvert$	: the number of vertices of a graph
$a \to b$	: a directed edge or an ordered pair of vertices
$a \sim b$	: an undirected edge, or an unordered pair of vertices

E	: a set of directed edges
$E^{\sim}$	: the undirected edges derived from E, that is, $\{a \sim b \mid a \to b$ or $b \to a \in E\}$
$G = (V, E)$	: a directed graph composed of vertices in V and edges in E
$G^{\sim} = (V, E^{\sim})$	: the skeleton of a directed graph $G = (V, E)$
$\text{adj}(a, G)$	: vertices adjacent to a in graph G, that is, $\{b \mid a \to b$ or $b \to a \in E\}$
$\text{adj}(a, G^{\sim})$	: vertices adjacent to a in graph $G^{\sim}$, that is, $\{b \mid a \sim b \in E^{\sim}\}$
$a \perp b \mid C$	: vertices a and b are d-separated by vertex set C
$X_a \perp X_b \mid X_C$	: X_a and X_b are conditional independent given X_C
$p_{a \perp b \mid C}$	: the p-value of testing $X_a \perp X_b \mid X_C$

A Bayesian network encodes a set of conditional-independence relationships with a DAG $G = (V, E)$ according to the directed Markov property defined as follows.

Definition 1 ***the Directed Markov Property:*** *if A and B are d-separated by C where A, B and C are three disjoint sets of vertices, then X_A and X_B are conditionally independent given X_C, that is, $P(X_A, X_B \mid X_C) = P(X_A \mid X_C) P(X_B \mid X_C)$. (see Lauritzen, 1996, pages 46–53)*

The concept of *d-separation* (see Lauritzen, 1996, page 48) is defined as follows. A chain between two vertices a and b is a sequence $a = a_0, a_1, \ldots, a_n = b$ of distinct vertices such that $a_{i-1} \sim a_i \in E^{\sim}$ for all i=1, ... , n. Vertex b is a descendant of vertex a if and only if there is a sequence $a = a_0, a_1, \ldots, a_n = b$ of distinct vertices such that $a_{i-1} \to a_i \in E$ for all i=1, ..., n. For three disjoint subsets A, B and $C \subseteq V$, C d-separates A and B if and only if any chain π between $\forall a \in A$ and $\forall b \in B$ contains a vertex $\gamma \in \pi$ such that either

- arrows of π do not meet head-to-head at γ and $\gamma \in C$, or
- arrows of π meet head-to-head at γ and γ is neither in C nor has any descendants in C.

Moreover, a probability distribution P is *faithful* to a DAG G (see Spirtes et al., 2001, pages 13 and 81) if all and only the conditional-independence relationships derived from P are encoded by G. In general, a probability distribution may possess other independence relationships besides those encoded by a DAG.

It should be pointed out that there are often several different DAGs encoding the same set of conditional-independence relationships and they are called an *equivalence class* of DAGs. An equivalence class can be uniquely represented by a completed acyclic partially directed graph (CPDAG) (also called the essential graph in the literature) that has the same skeleton as a DAG does except that some edges are not directed (see Andersson et al., 1997).

2.2 PC Algorithm

If a probability distribution P is faithful to a DAG G, then the PC algorithm (see Spirtes et al., 2001, pages 84–89) is able to recover the equivalence class of the DAG G, given the set of the conditional-independence relationships. In general, a probability distribution may include other independence relationships besides those encoded by a DAG. The faithfulness assumption assures that the independence relationships can be perfectly encoded by a DAG. In practice, the information on conditional independence is usually unknown but extracted from data with statistical hypothesis testing. If the p-value of testing a hypothesis $X_a \perp X_b \mid X_C$ is lower than a user-specified level α

(conventionally 5% or 1%), then the hypothesis of conditional independence is rejected while the hypothesis of conditional dependence $X_a \not\perp X_b | X_C$ is accepted.

The first step of the PC algorithm is to construct an undirected graph $G^{\sim}$ whose edge directions will later be further determined with other steps, while the skeleton is kept the same as that of $G^{\sim}$. Since we restrict ourselves to the error rate of the skeleton, here we only present in Algorithm 1 the first step of the PC algorithm, as implemented in software Tetrad version 4.3.8 (see http://www.phil.cmu.edu/projects/tetrad), and refer to it as the PC-skeleton algorithm.

Algorithm 1 PC-skeleton

Input: the data X_V generated from a probability distribution faithful to a DAG G_{true}, and the significance level α for every statistical test of conditional independence
Output: the recovered skeleton $G^{\sim}$

1: Form the complete undirected graph $G^{\sim}$ on the vertex set V.
2: Let depth $d = 0$.
3: **repeat**
4: **for** each ordered pair of adjacent vertices a and b in $G^{\sim}$, that is, $a \sim b \in E^{\sim}$ **do**
5: **if** $|\text{adj}(a, G^{\sim}) \setminus \{b\}| \geq d$, **then**
6: **for** each subset $C \subseteq \text{adj}(a, G^{\sim}) \setminus \{b\}$ and $|C| = d$ **do**
7: Test hypothesis $X_a \perp X_b | X_C$ and calculate the p-value $p_{a \perp b|C}$.
8: **if** $p_{a \perp b|C} \geq \alpha$, **then**
9: Remove edge $a \sim b$ from $G^{\sim}$.
10: Update $G^{\sim}$ and $E^{\sim}$.
11: **break** the **for loop** at line 6
12: **end if**
13: **end for**
14: **end if**
15: **end for**
16: Let $d = d + 1$.
17: **until** $|\text{adj}(a, G^{\sim}) \setminus \{b\}| < d$ for every ordered pair of adjacent vertices a and b in $G^{\sim}$.

The theoretical foundation of the PC-skeleton algorithm is **Proposition 1**: if two vertices a and b are not adjacent in a DAG G, then there is a set of other vertices C that either all are neighbors of a or all are neighbors of b such that C d-separates a and b, or equivalently, $X_a \perp X_b | X_C$, according to the directed Markov property. Since two adjacent vertices are not d-separated by any set of other vertices (according to the directed Markov property), **Proposition 1** implies that a and b are not adjacent if and only if there is a d-separating C in either neighbors of a or neighbors of b. Readers should notice that the proposition does not imply that a and b are d-separated only by such a C, but just guarantees that a d-separating C can be found in either neighbors of a or neighbors b.

Proposition 1 *If vertices a and b are not adjacent in a DAG G, then there is a set of vertices C which is either a subset of* $\text{adj}(a, G) \setminus \{b\}$ *or a subset of* $\text{adj}(b, G) \setminus \{a\}$ *such that C d-separates a and b in G.* This proposition is a corollary of Lemma 5.1.1 on page 411 of book *Causation, Prediction, and Search* (Spirtes et al., 2001).

The logic of the PC-skeleton algorithm is as follows, under the assumption of perfect judgment on conditional independence. The most straightforward application of **Proposition 1** to structure learning is to exhaustively search all the possible neighbors of a and b to verify whether there is such a d-separating C to disconnect a and b. Since the possible neighbors of a and b are unknown, to guarantee the detection of such a d-separating C, all the vertices other than a and b should be searched as possible neighbors of a and b. However, such a straightforward application is very inefficient because it probably searches many unnecessary combinations of vertices by considering all the vertices other than a and b as their possible neighbors, especially when the true DAG is sparse. The PC-skeleton algorithm searches more efficiently, by keeping updating the possible neighbors of a vertex once some previously-considered possible neighbors have been found actually not connected with the vertex. Starting with a fully connected undirected graph $G^{\sim}$ (step 1), the algorithm searches for the d-separating C progressively by increasing the size of C, that is, the number of conditional variables, from zero (step 2) with the step size of one (step 16). Given the size of C, the search is performed for every vertex pair a and b (step 4). Once a C d-separating a and b is found (step 8), a and b are disconnected (step 9), and the neighbors of a and b are updated (step 10). In the algorithm, $G^{\sim}$ is continually updated, so $\text{adj}(a, G^{\sim})$ is also constantly updated as the algorithm progresses. The algorithm stops when all the subsets of the current neighbors of each vertex have been examined (step 17). The Tetrad implementation of the PC-skeleton algorithm examines an edge $a \sim b$ as two ordered pairs (a, b) and (b, a) (step 4), each time searching for the d-separating C in the neighbors of the first element of the pair (step 6). In this way, both the neighbors of a and the neighbors of b are explored.

The accuracy of the PC-skeleton algorithm depends on the discriminability of the statistical test of conditional independence. If the test can perfectly distinguish dependence from independence, then the algorithm can correctly recover the true underlying skeleton, as proved by Spirtes et al. (2001, pages 410–412). The outline of the proof is as follows. First, all the true edges will be recovered because an adjacent vertex pair $a \sim b$ is not d-separated by any vertex set C that excludes a and b. Second, if the edge between a non-adjacent vertex pair a and b has not been removed, subsets of either $\text{adj}(a, G) \setminus \{b\}$ or $\text{adj}(b, G) \setminus \{a\}$ will be searched until the C that d-separates a and b according to **Proposition 1** is found, and consequently the edge between a and b will be removed. If the judgments on conditional independence and conditional dependence are imperfect, the PC-skeleton algorithm is unstable. If an edge is mistakenly removed from the graph in the early stage of the algorithm, then other edges which are not in the true graph may be included in the graph (see Spirtes et al., 2001, page 87).

2.3 False Discovery Rate

In a statistical decision process, there are basically two sources of errors: the type I errors, that is, falsely rejecting negative hypotheses when they are actually true; and the type II errors, that is, falsely accepting negative hypotheses when their alternative, the positive hypotheses are actually true. The FDR (see Benjamini and Yekutieli, 2001) is a criterion to assess the errors when multiple hypotheses are simultaneously tested. It is the expected ratio of the number of falsely claimed positive results to that of all those claimed to be positive, as defined in Table 2. A variant of the FDR, the positive false discovery rate (pFDR), defined as in Table 2, was proposed by Storey (2002). Clearly, pFDR = FDR / $P(R_2 > 0)$, so the two measures will be similar if $P(R_2 > 0)$ is close to 1, and quite different otherwise.

Test Results	Truth		
	Negative	Positive	Total
Negative	TN (true negative)	FN (false negative)	R_1
Positive	FP (false positive)	TP (true positive)	R_2
Total	T_1	T_2	

Table 1: Results of multiple hypothesis testing, categorized according to the claimed results and the truth.

Full Name	Abbreviation	Definition
False Discovery Rate	FDR	$E(\mathrm{FP}/R_2)$ (See note *)
Positive False Discovery Rate	pFDR	$E(\mathrm{FP}/R_2 \mid R_2 > 0)$
Family-Wise Error Rate	FWER	$P(\mathrm{FP} \geq 1)$
Type I Error Rate (False Positive Rate)	α	$E(\mathrm{FP}/T_1)$
Specificity (True Negative Rate)	$1-\alpha$	$E(\mathrm{TN}/T_1)$
Type II Error Rate (False Negative Rate)	β	$E(\mathrm{FN}/T_2)$
Power (Sensitivity, True Positive Rate)	$1-\beta$	$E(\mathrm{TP}/T_2)$

Table 2: Criteria for multiple hypothesis testing. Here $E(x)$ means the expected value of x, and $P(\mathcal{A})$ means the probability of event $\mathcal{A}$. Please refer to Table 1 for related notations. * If $R_2 = 0$, FP/R_2 is defined to be 0.

The FDR is a reasonable criterion when researchers expect the "discovered" results are trustful and dependable in afterward studies. For example, in a pilot study, we are selecting candidate genes for a genetic research on Parkinson's disease. Because of the limited funding, we can only study a limited number of genes in the afterward genetic research. Thus, when selecting candidate genes in the pilot study, we hope that 95% of the selected candidate genes are truly associated with the disease. In this case, the FDR is chosen as the error rate of interest and should be controlled under 5%. Since similar situations are quite common in research practices, the FDR has been widely adopted in many research fields such as bioinformatics and neuroimaging.

In the context of learning the skeleton of a DAG, a negative hypothesis could be that a connection does not exist in the DAG, and a positive hypothesis could be that the connection exists. The FDR is the expected proportion of the falsely discovered connections to all those discovered. Learning network structures may face scenarios similar to the aforementioned pilot study, but the FDR control has not yet received adequate attention in structure learning.

Benjamini and Yekutieli (2001) have proved that, when the test statistics have positive regression dependency on each of the test statistics corresponding to the true negative hypotheses, the FDR can be controlled under a user-specified level q by Algorithm 2. In other cases of dependency, the FDR can be controlled with a simple conservative modification of the procedure by replacing H^* in Eq. (1) with $H(1+1/2,\ldots,+1/H)$. Storey (2002) has provided algorithms to control the pFDR for independent test statistics. For a review and comparison of more FDR methods, please refer to Qian and Huang (2005)'s work. It should be noted that the FDR procedures do not control the

realized FDR of a trial under q, but control the *expected value* of the error rate when the procedures are repetitively applied to randomly sampled data.

Algorithm 2 FDR-stepup

Input: a set of p-values $\{p_i | i = 1, \ldots, H\}$, and the threshold of the FDR q
Output: the set of rejected null hypotheses

1: Sort the p-values of H hypothesis tests in the ascendant order as $p_{(1)} \leq \ldots \leq p_{(H)}$.
2: Let $i = H$, and $H^* = H$ (or $H^* = H(1 + 1/2, \ldots, +1/H)$, depending on the assumption of the dependency among the test statistics).
3: **while**

$$\frac{H^*}{i} p_{(i)} > q \quad \text{and} \quad i > 0, \tag{1}$$

do
4: Let $i = i - 1$.
5: **end while**
6: Reject the null hypotheses associated with $p_{(1)}, \ldots, p_{(i)}$, and accept the null hypotheses associated with $p_{(i+1)}, \ldots, p_{(H)}$.

Besides the FDR and the pFDR, other criteria, as listed in Table 2, can also be applied to assess the uncertainty of multiple hypothesis testing. The type I error rate is the expected ratio of the type I errors to all the negative hypotheses that are actually true. The type II error rate is the expected ratio of the type II errors to all the positive hypotheses that are actually true. The family-wise error rate is the probability that at least one of the accepted positive hypotheses are actually wrong. Generally, there are not mathematically or technically superior relationships among these error-rate criteria. Please refer to Appendix C for examples of typical research scenarios where each particular error rate is favoured.

Controlling both the type I and the type II error rates under a conventional level (such as $\alpha <$ 5% or 1% and $\beta <$ 10% or 5%) does not necessarily curb the FDR at a desired level. As shown in Eq. (2), if FP/T_1 and FN/T_2 are fixed and positive, FP/R_2 approaches 1 when T_2/T_1 is small enough. This is the case of sparse networks where the number of existing connections T_2 is much smaller than the number of non-existing connections T_1.

$$\frac{\mathrm{FP}}{R_2} = \frac{\frac{\mathrm{FP}}{T_1}}{\frac{\mathrm{FP}}{T_1} + (1 - \frac{\mathrm{FN}}{T_2})\frac{T_2}{T1}}. \tag{2}$$

2.4 PC Algorithm with FDR

Steps 8–12 of the PC-skeleton algorithm control the type I error rate of each statistical test of conditional independence individually below a pre-defined level α, so the algorithm can not explicitly control the FDR. We propose embedding an FDR-control procedure into the algorithm to curb the error rate of the learned skeleton. The FDR-control procedure collectively considers the hypothesis tests related to the existence of multiple edges, correcting the effect of multiple hypothesis testing. The proposed method is described in Algorithm 3, and we name it as the $\mathrm{PC}_{\mathrm{fdr}}$-skeleton

algorithm. Similar to the PC-skeleton algorithm, $G^{\sim}$, adj$(a, G^{\sim})$ and $E^{\sim}$ are constantly updated as the algorithm progresses.

The PC$_{\text{fdr}}$-skeleton and the PC-skeleton algorithms share the same search strategy, but differ on the judgment of conditional independence. The same as the PC-skeleton algorithm, the PC$_{\text{fdr}}$-skeleton algorithm increases d, the number of conditional variables, from zero (step 3) with the step size of one (step 25), and also keeps updating the neighbors of vertices (steps 14 and 15) when some previously-considered possible neighbors have been considered not connected (step 13). The PC$_{\text{fdr}}$-skeleton algorithm differs from the PC-skeleton algorithm on the inference of d-separation, with its steps 11–20 replacing steps 8–12 of the PC-skeleton algorithm. In the PC-skeleton algorithm, two vertices are regarded as d-separated once the conditional-independence test between them yields a p-value larger than the pre-defined significant level α. In this way, the type I error of each statistical test is controlled separately, without consideration of the effect of multiple hypothesis testing. The PC$_{\text{fdr}}$-skeleton algorithm records in $p^{max}_{a\sim b}$ the up-to-date maximum p-value associated with an edge $a \sim b$ (steps 9 and 10), and progressively removes those edges whose non-existence is accepted by the FDR procedure (step 12), with $P^{max} = \{p^{max}_{a\sim b}\}_{a\neq b}$ and the pre-defined FDR level q being the input. The FDR procedure, **Algorithm 2**, is invoked at step 12, either immediately after every element of P^{max} has been assigned a valid p-value for the first time, or later once any element of P^{max} is updated.

The $p^{max}_{a\sim b}$ is the upper bound of the p-value of testing the hypothesis that a and b are d-separated by at least one of the vertex sets C searched in step 7. According to the directed Markov property, a and b are not adjacent if and only if there is a set of vertices $C \subseteq V \setminus \{a,b\}$ d-separating a and b. As the algorithm progresses, the d-separations between a and b by vertex sets $C_1, \ldots, C_K \subseteq V \setminus \{a,b\}$ are tested respectively, and consequently a sequence of p-values $p_1, \ldots, p_K$ are calculated. If we use $p^{max}_{a\sim b} = \max_{i=1}^{K} p_i$ as the statistic to test the negative hypothesis that there is, though unknown, a C_j among $C_1, \ldots, C_K$ d-separating a and b, then due to

$$P(p^{max}_{a\sim b} \leq p) = P(p_i \leq p \text{ for all } i = 1, \ldots, K) \leq P(p_j \leq p) = p, \tag{3}$$

$p^{max}_{a\sim b}$ is the upper bound of the p-value of testing the negative hypothesis. Eq. (3) also clearly shows that the PC-skeleton algorithm controls the type I error rate of the negative hypothesis, since its step 8 is equivalent to "if $p^{max}_{a\sim b} < \alpha$, then ... " if $p^{max}_{a\sim b}$ is recorded in the PC-skeleton algorithm.

The statistical tests performed at step 8 of the PC$_{\text{fdr}}$-skeleton algorithm generally are not independent with each other, since the variables involved in two hypotheses of conditional independence may overlap. For example, conditional-independence relationships $a \perp b_1 | C$ and $a \perp b_2 | C$ both involve a and C. It is very difficult to prove whether elements of P^{max} have positive regression dependency or not, so rigorously the conservative modification of Algorithm 2, should be applied at step 12. However, since $p^{max}_{a\sim b}$ is probably a loose upper bound of the p-value of testing $a \nsim b$, in practice we simply apply the FDR procedure that is correct for positive regression dependency.

It should be noted that different from step 9 of the PC-skeleton algorithm, step 14 of the PC$_{\text{fdr}}$-skeleton algorithm may remove edges other than just $a \sim b$, because the decisions on other edges can be affected by the updating of $p^{max}_{a\sim b}$.

A heuristic modification of the PC$_{\text{fdr}}$-skeleton algorithm is to remove $p^{max}_{a\sim b}$ from P^{max} once edge $a \sim b$ has been deleted from $G^{\sim}$ at step 14. We name this modified version as the PC$_{\text{fdr}*}$-skeleton algorithm. In the PC$_{\text{fdr}}$-skeleton algorithm, $p^{max}_{a\sim b}$ is still recorded in P^{max} and input to the FDR procedure after the edge $a \sim b$ has been removed. This guarantees that the algorithm can asymptotically keep the FDR under the user-specified level q (see Section 2.5). The motivation of

Algorithm 3 PC_{fdr}-skeleton

Input: the data X_V generated from a probability distribution faithful to a DAG G_{true}, and the FDR level q for the discovered skeleton

Output: the recovered skeleton $G^{\sim}$

1: Form the complete undirected graph $G^{\sim}$ on the vertex set V
2: Initialize the maximum p-values associated with edges as $P^{max} = \{p^{max}_{a\sim b} = -1\}_{a\neq b}$.
3: Let depth $d = 0$.
4: **repeat**
5: **for** each ordered pair of adjacent vertices a and b in $G^{\sim}$, that is, $a \sim b \in E^{\sim}$ **do**
6: **if** $|\text{adj}(a, G^{\sim}) \setminus \{b\}| \geq d$, **then**
7: **for** each subset $C \subseteq \text{adj}(a, G^{\sim}) \setminus \{b\}$ and $|C| = d$ **do**
8: Test hypothesis $X_a \perp X_b | X_C$ and calculate the p-value $p_{a\perp b|C}$.
9: **if** $p_{a\perp b|C} > p^{max}_{a\sim b}$, **then**
10: Let $p^{max}_{a\sim b} = p_{a\perp b|C}$.
11: **if** every element of P^{max} has been assigned a valid p-value by step 10, **then**
12: Run the FDR procedure, **Algorithm 2**, with P^{max} and q as the input.
13: **if** the non-existence of certain edges are accepted, **then**
14: Remove these edges from $G^{\sim}$.
15: Update $G^{\sim}$ and $E^{\sim}$.
16: **if** $a \sim b$ is removed, **then**
17: **break** the **for loop** at line 7.
18: **end if**
19: **end if**
20: **end if**
21: **end if**
22: **end for**
23: **end if**
24: **end for**
25: Let $d = d + 1$.
26: **until** $|\text{adj}(a, G^{\sim}) \setminus \{b\}| < d$ for every ordered pair of adjacent vertices a and b in $G^{\sim}$.

* A heuristic modification at step 15 of the algorithm is to remove from P^{max} the $p^{max}_{a\sim b}$s whose associated edges have been deleted from $G^{\sim}$ at step 14, that is, to update P^{max} as $P^{max} = \{p^{max}_{a\sim b}\}_{a\sim b\in E^{\sim}}$ right after updating $E^{\sim}$ at step 15. This heuristic modification is named as the **$\text{PC}_{\text{fdr}*}$-skeleton algorithm**.

the heuristic modification is that if an edge has been eliminated, then it should not be considered in the FDR procedure any longer. Though we cannot theoretically prove the asymptotic performance of the heuristic modification in the sense of controlling the FDR, it is shown to control the FDR closely around the user-specified level in our empirical experiments and gain more detection power than that of the PC_{fdr}-skeleton algorithm (see Section 3).

2.5 Asymptotic Performance

Here we prove that the PC_{fdr}-skeleton algorithm is able to control the FDR under a user-specified level q ($q > 0$) at the limit of large sample sizes if the following assumptions are satisfied:

(A1) The probability distribution P is faithful to a DAG G_{true}.

(A2) The number of vertices is fixed.

(A3) Given a fixed significant level of testing conditional-independence relationships, the power of detecting conditional-dependence relationships with statistical tests approaches 1 at the limit of large sample sizes. (For the definition of power in hypothesis testing, please refer to Table 2.)

Assumption (A1) is generally assumed when graphical models are applied, although it restricts the probability distribution P to a certain class. Assumption (A2) is usually implicitly stated, but here we explicitly emphasize it because it simplifies the proof. Assumption (A3) may seem demanding, but actually it can be easily satisfied by standard statistical tests, such as the likelihood-ratio test introduced by Neyman and Pearson (1928), if the data are identically and independently sampled. Two statistical tests that satisfy Assumption (A3) are listed in Appendix B.

The detection power and the FDR of the PC_{fdr}-skeleton algorithm and its heuristic modification at the limit of large sample sizes are elucidated in Theorems 1 and 2. The detailed proofs are provided in Appendix A.

Theorem 1 *Assuming (A1), (A2) and (A3), both the PC_{fdr}-skeleton algorithm and its heuristic modification, the $\text{PC}_{\text{fdr}*}$-skeleton algorithm, are able to recover all the true connections with probability one as the sample size approaches infinity:*

$$\lim_{m\to\infty} P(E^{\sim}_{true} \subseteq E^{\sim}) = 1,$$

where $E^{\sim}_{true}$ denotes the set of the undirected edges derived from the true DAG G_{true}, $E^{\sim}$ denotes the set of the undirected edges recovered with the algorithms, and m denotes the sample size.

Theorem 2 *Assuming (A1), (A2) and (A3), the FDR of the undirected edges recovered with the PC_{fdr}-skeleton algorithm approaches a value not larger than the user-specified level q as the sample size m approaches infinity:*

$$\limsup_{m\to\infty} \text{FDR}(E^{\sim}, E^{\sim}_{true}) \leq q,$$

where $\text{FDR}(E^{\sim}, E^{\sim}_{true})$ *is defined as*

$$\begin{cases} \text{FDR}(E^{\sim}, E^{\sim}_{true}) &= E\left[\frac{|E^{\sim}\setminus E^{\sim}_{true}|}{|E^{\sim}|}\right], \\ \text{Define } \frac{|E^{\sim}\setminus E^{\sim}_{true}|}{|E^{\sim}|} &= 0, \quad \text{if } |E^{\sim}| = 0. \end{cases}$$

2.6 Computational Complexity

The PC_{fdr}-skeleton algorithm spends most of its computation on performing statistical tests of conditional independence at step 8 and controlling the FDR at step 12. Since steps 13 to 19 of the PC_{fdr}-skeleton algorithm play a role similar to steps 8 to 12 of the PC-skeleton algorithm do, and

all the other parts of both algorithms employ the same search strategy, the computation spent by the PC_{fdr}-skeleton algorithm on statistical tests has the same complexity as that by the PC-skeleton algorithm. The only extra computational cost of the PC_{fdr}-skeleton algorithm is at step 12 for controlling the FDR.

The computational complexity of the search strategy employed by the PC algorithm has been studied by Kalisch and Bühlmann (2007) and Spirtes et al. (see 2001, pages 85–87). Here to make the paper self-contained, we briefly summarize the results as follows. It is difficult to analyze the complexity exactly, but if the algorithm stops at the depth $d = d_{max}$, then the number of conditional-independence tests required is bounded by

$$T = 2C_N^2 \sum_{d=0}^{d_{max}} C_{N-2}^d,$$

where N is the number of vertices, C_N^2 is the number of combinations of choosing 2 un-ordered and distinct elements from N elements, and similarly C_{N-2}^2 is the number of combinations of choosing from $N-2$ elements. In the worst case that $d_{max} = N-2$, the complexity is bounded by $2C_N^2 2^{N-2}$. The bound usually is very loose, because it assumes that no edge has been removed until $d = d_{max}$. In real world applications, the algorithm is very fast for sparse networks.

The computational complexity of the FDR procedure, Algorithm 2, generally is $O(\, H\log(H) + H) = O(H\log(H))$ where $H = C_N^2$ is the number of input p-values. The sorting at step 1 costs $H\log(H)$ and the "while" loop from step 3 to step 5 repeats H times at most. However, if the sorted P^{max} is recorded during the computation, each time when an element of P^{max} is updated at step 10 of the PC_{fdr}-skeleton algorithm, the complexity of keeping the updated P^{max} sorted is only $O(H)$. With this optimization, the complexity of the FDR-control procedure is $O(H\log(H))$ at its first operation, and is $O(H)$ later. The FDR procedure is invoked only when $p_{a\perp b|C} > p_{a\sim b}^{max}$. In the worst case that $p_{a\perp b|C}$ is always larger than $p_{a\sim b}^{max}$, the complexity of the computation spent on the FDR control in total is bounded by $O(C_N^2\log(C_N^2) + TC_N^2) = O(N^2\log(N) + TN^2)$ where T is the number of performed conditional-independence tests. This is a very loose bound because it is rare that $p_{a\perp b|C}$ is always larger than $p_{a\sim b}^{max}$.

The computational complexity of the heuristic modification, the $\text{PC}_{\text{fdr}*}$-skeleton algorithm, is the same as that of the PC_{fdr}-skeleton algorithm, since they share the same search strategy and both employ the FDR procedure. In the $\text{PC}_{\text{fdr}*}$-skeleton algorithm, the size of P^{max} keeps decreasing as the algorithm progresses, so each operation of the FDR procedure is more efficient. However, since the $\text{PC}_{\text{fdr}*}$-skeleton algorithm adjusts the effect of multiple hypothesis testing less conservatively, it may remove less edges than the PC_{fdr}-skeleton algorithm does, and invokes more conditional-independence tests. Nevertheless, their complexity is bounded by the same limit in the worst case.

It is unfair to directly compare the computational time of the PC_{fdr}-skeleton algorithm against that of the PC-skeleton algorithm, because if the q of the former is set at the same value as the α of the latter, the former will remove more edges and perform much less statistical tests, due to its more stringent control over the type I error rate. A reasonable way is to compare the time spent on the FDR control at step 12 against that on conditional-independence tests at step 8 in each run of the PC_{fdr}-skeleton algorithm. If P^{max} is kept sorted during the learning process as aforementioned, then each time (except the first time) the FDR procedure just needs linear computation time (referring to the size of P^{max}) with simple operations such as division and comparing two numerical values. Thus, we suspect that the FDR procedure will not contribute much to the total computation time of the

structure learning. In our simulation study in Section 3.1, the extra computation added by the FDR control was only a tiny portion, less than 0.5%, to that spent on testing conditional independence, performed with the Cochran-Mantel-Haenszel (CMH) test (see Agresti, 2002, pages 231–232), as shown in Tables 3 and 4.

2.7 Miscellaneous Discussions Discussions

An intuitive and attracting idea of adapting the PC-skeleton algorithm to the FDR control is to "smartly" determine such an appropriate threshold of the type I error rate α that will let the errors be controlled at the pre-defined FDR level q. Given a particular problem, it is very likely that the FDR of the graphs learned by the PC-skeleton algorithm is an monotonically increasing function of the pre-defined threshold α of the type I error rate. If this hypothesis is true, then there is a one-to-one mapping between α and q for the particular problem. Though we cannot prove this hypothesis rigorously, the following argument may be enlightening. Instead of directly focusing on FDR = $E(\mathrm{FP}/R_2)$ (see Table 2), the expected ratio of the number of false positives (FP) to the number of accepted positive hypotheses (R_2), we first focus on $E(\mathrm{FP})/E(R_2)$, the ratio of the expected number of false positives to the expected number of accepted positive hypotheses, since the latter is easier to link with the type I error rate according to Eq. (2), as shown in Eq. (4),

$$\frac{E(\mathrm{FP})}{E(R_2)} = \frac{E\left(\frac{\mathrm{FP}}{T_1}\right)}{E\left(\frac{\mathrm{FP}}{T_1} + (1-\frac{\mathrm{FN}}{T_2})\frac{T_2}{T1}\right)} = \frac{E\left(\frac{\mathrm{FP}}{T_1}\right)}{E\left(\frac{\mathrm{FP}}{T_1}\right) + \left(1 - E\left(\frac{\mathrm{FN}}{T_2}\right)\right)\frac{T_2}{T_1}} = \frac{\alpha}{\alpha + (1-\beta)\frac{T_2}{T_1}}, \tag{4}$$

where α and β are the type I error rate and the type II error rate respectively. A sufficient condition for $E(\mathrm{FP})/E(R_2)$ being a monotonically increasing function of the type I error rate includes (I) $(1-\beta)/\alpha > \partial(1-\beta)/\partial\alpha$, (II) $T_1 > 0$ and (III) $T_2 > 0$, where $\partial(1-\beta)/\partial\alpha$ is the derivative of $(1-\beta)$ over α. If $(1-\beta)$, regarded as a function of α, is a concave curve from (0, 0) to (1,1), then condition (I) is satisfied. Recall that $(1-\beta)$ versus α actually is the receiver operating characteristic (ROC) curve, and that with an appropriate statistic the ROC curve of a hypothesis test is usually a concave curve from (0, 0) to (1,1), we speculate that condition (I) is not difficult to satisfy. With the other two mild conditions (II) $T_1 > 0$ and (III) $T_2 > 0$, we could expect that $E(\mathrm{FP})/E(R_2)$ is a monotonically increasing function of α. $E(\mathrm{FP})/E(R_2)$ is the ratio of the expected values of two random variables, while $E(\mathrm{FP}/R_2)$ is the expected value of the ratio of two random variables. Generally, there is not a monotonic relationship between $E(\mathrm{FP})/E(R_2)$ and $E(\mathrm{FP}/R_2)$. Nevertheless, if the average number of false positives, $E(\mathrm{FP})$, increases proportionally faster than that of the accepted positives, $E(R_2)$, we speculate that under certain conditions, the FDR = $E(\mathrm{FP}/R_2)$ also increases accordingly. Thus the FDR may be a monotonically increasing function of the threshold α of the type I error rate for the PC-skeleton algorithm.

However, even though the FDR of the PC-skeleton algorithm may decrease as the pre-defined significant level α decreases, the FDR of the PC-skeleton algorithm still cannot be controlled at the user-specified level for general problems by "smartly" choosing an α beforehand, but somehow has to be controlled in a slightly different way, such as the $\mathrm{PC}_{\mathrm{fdr}}$-skeleton algorithm does. First, the value of such an α for the FDR control depends on the true graph, but unfortunately the graph is unknown in problems of structure learning. According to Eq. (2), the realized FDR is a function of the realized type I and type II error rates, as well as T_2/T_1, which in the context of structure learning is the ratio of the number of true connections to the number of non-existing connections. Since

T_2/T_1 is unknown, such an α cannot be determined completely in advance without any information about the true graph, but has to be estimated practically from the observed data. Secondly, the FDR method we employ is such a method that estimates the α from the data to control the FDR of multiple hypothesis testing. The output of the FDR algorithm is the rejection of those null hypotheses associated with *p*-values $p_{(1)}, \ldots, p_{(i)}$ (see Algorithm 2). Given $p_{(1)} \leq \ldots \leq p_{(H)}$, the output is equivalent to the rejection of all those hypotheses whose *p*-values are smaller than or equal to $p_{(i)}$. In other words, it is equivalent to setting $\alpha = p_{(i)}$ in the particular multiple hypothesis testing. Thirdly, the PC_{fdr}-skeleton algorithm is a valid solution to combining the FDR method with the PC-skeleton algorithm. Because the estimation of the α depends on *p*-values, and *p*-values are calculated one by one as the PC-skeleton algorithm progresses with hypothesis tests, the α cannot be estimated separately before the PC-skeleton algorithm starts running, but the estimation has to be embedded within the algorithm, like in the PC_{fdr}-skeleton algorithm.

Another idea on the FDR control in structure learning is a two-stage algorithm. The first stage is to draft a graph that correctly includes all the existing edges and their orientations but may also include non-existing edges as well. The second stage is to select the real parents for each vertex, with the FDR controlled, from the set of potential parents determined in the first stage. The advantage of this algorithm is that the selection of real parent vertices in the second stage is completely decoupled from the determination of edge orientations, because all the parents of each vertex have been correctly connected with the particular vertex in the first stage. However, a few concerns about the algorithm should be noticed before researchers start developing this two-stage algorithm. First, to avoid missing many existing edges in the first stage, a considerable number of non-existing edges may have to be included. To guarantee a perfect protection of the existing edges given any randomly sampled data, the first stage must output a graph whose skeleton is a fully connected graph. The reason for this is that the type I error rate and the type II error rate contradict each other and the latter reaches zero generally when the former approaches one (see Appendix C). Rather than protecting existing edges perfectly, the first stage should trade off between the type I and the type II errors, in favour of keeping the type II error rate low. Second, selecting parent vertices from a set of candidate vertices in the second stage, in certain sense, can be regarded as learning the structure of a sub-graph locally, in which error-rate control remains as a crucial problem. Thus erro-rate control is still involved in both of the two stages. Though this two-stage idea may not essentially reduce the problem of the FDR control to an easier task, it may break the big task of simultaneously learning all edges to many local structure-learning tasks.

3. Empirical Evaluation

The PC_{fdr}-skeleton algorithm and its heuristic modification are evaluated with simulated data sets, in comparison with the PC-skeleton algorithm, in the sense of the FDR, the type I error rate and the power. The PC_{fdr}-skeleton and the PC-skeleton algorithms are also applied to two real functional-magnetic-resonance-imaging (fMRI) data sets, to check whether the two algorithms correctly curb the error rates that they are supposed to control in real world applications.

3.1 Simulation Study

The simulated data sets are generated from eight different DAGs, shown in Figure 1, with the number of vertices N = 15, 20, 25 or 30, and the average degree of vertices D = 2 or 3. The DAGs are generated as follows:

(1) Sample $\frac{N\times D}{2}$ undirected edges from $\{a \sim b | a,b \in V \text{ and } a \neq b\}$ with equal probabilities and without replacement to compose an undirected graph $G^{\sim}_{true}$.

(2) Generate a random order $\succ$ of vertices with permutation.

(3) Orientate the edges of $G^{\sim}$ according to the order $\succ$. If a is before b in the order $\succ$, then orientate the edge $a \sim b$ as $a \rightarrow b$. Denote the orientated graph as a DAG G_{true}.

For each DAG, we associate its vertices with (conditional) binary probability distributions as follows, to extend it to a Bayesian network.

(1) Specify the strength of (conditional) dependence as a parameter $\delta > 0$.

(2) Randomly assign each vertex $a \in V$ with a dependence strength $\delta_a = 0.5\delta$ or -0.5δ, with equal possibilities.

(3) Associate each vertex $a \in V$ with a logistic regression model

$$\Delta = \sum_{b \in \text{pa}[a]} X_b \delta_b,$$
$$P(X_a = 1 | X_{\text{pa}[a]}) = \frac{\exp(\Delta)}{1+\exp(\Delta)},$$
$$P(X_a = -1 | X_{\text{pa}[a]}) = \frac{1}{1+\exp(\Delta)},$$

where $\text{pa}[a]$ denotes the parent vertices of a.

The parameter δ reflects the strength of dependence because if the values of all the other parent variables are fixed, the difference between the conditional probabilities of a variable X_a = 1 given a parent variable X_b = 1 and -1 is

$$\left|\text{logit}[P(X_a = 1 | X_b = 1, X_{\text{pa}[a]\setminus\{b\}})] - \text{logit}[P(X_a = 1 | X_b = -1, X_{\text{pa}[a]\setminus\{b\}})]\right| = |2\delta_b| = \delta,$$

where the logit function is defined as $\text{logit}(x) = \log(\frac{x}{1-x})$.

Since the accuracy of the PC-skeleton algorithm and its FDR versions is related to the discriminability of the statistical tests, we generated data with different values of δ (δ= 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0) to evaluate the algorithms' performances with different power of detecting conditional dependence. The larger the absolute value of δ is, the easier the dependence can be detected with statistical tests. Because statistical tests are abstract queries yielding p-values about conditional independence for the structure-learning algorithms, the accuracy of the algorithms is not determined by the particular procedure of a statistical test, or a particular family of conditional probability distributions but by the discriminability of the statistical tests. Given a fixed sample size, the stronger the conditional-dependence relationships are, the higher discriminability the statistical tests have. By varying the dependence strength δ of the *binary* conditional probability distributions, we have varied the discriminability of the statistical tests, as if by varying the dependence strength of *other* probability distribution families. To let the readers intuitively understand the dependence strength of these δ values, we list as follows examples of probability pairs whose logit contrasts are equal to these δ values:

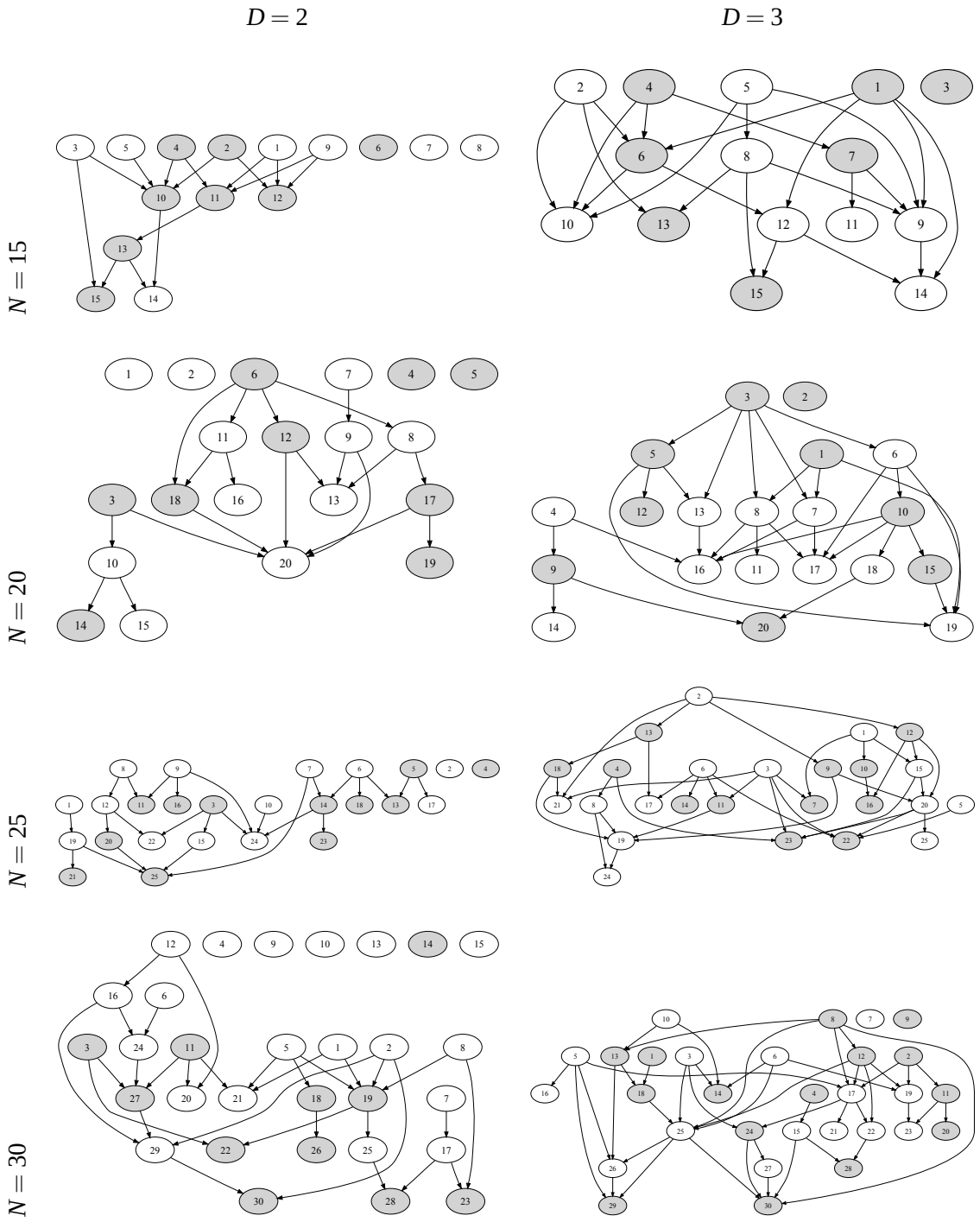

Figure 1: DAGs used in the simulation study. N denotes the number of vertices and D denotes the average degree of the vertices. Unshaded vertices are associated with positive dependence strength 0.5δ, and shaded ones are associated with negative dependence strength -0.5δ.

0.5 = logit(0.5622) - logit(0.4378), 0.6 = logit(0.5744) - logit(0.4256),
0.7 = logit(0.5866) - logit(0.4134), 0.8 = logit(0.5987) - logit(0.4013),
0.9 = logit(0.6106) - logit(0.3894), 1.0 = logit(0.6225) - logit(0.3775).

.

In total, we performed the simulation with 48 Bayesian networks generated with all the combinations of the following parameters:

$$\begin{aligned} N &= 15, 20, 25, 30; \\ D &= 2, 3; \\ \delta &= 0.5, 0.6, 0.7, 0.8, 0.9, 1.0. \end{aligned}$$

From each Bayesian network, we repetitively generated 50 data sets each of 500 samples to estimate the statistical performances of the algorithms. A non-model-based test, the Cochran-Mantel-Haenszel (CMH) test (see Agresti, 2002, pages 231–232), was employed to the test conditional independence among random variables. Both the significant level α of the PC-skeleton algorithm and the FDR level q of the PC_{fdr}-skeleton algorithm and its heuristic modification were set at 5%.

Figures 2, 3 and 4 respectively show the empirical FDR, power and type I error rate of the algorithms, estimated from the 50 data sets repetitively generated from each Bayesian network, with error bars indicating the 95% confidence intervals of these estimations. The PC_{fdr}-skeleton algorithm controls the FDR under the user-specified level 5% for all the 48 Bayesian networks, and the $\text{PC}_{\text{fdr}*}$-skeleton algorithm steadily controls the FDR closely around 5%, while the PC-skeleton algorithm yields the FDR ranging from about 5% to about 35%, and above 15% in many cases, especially for those sparser DAGs with the average degree of vertices $D = 2$. The PC_{fdr}-skeleton algorithm is conservative, with the FDR notably lower than the user-specified level, while its heuristic modification controls the FDR more accurately around the user-specified level, although the correctness of the heuristic modification has not been theoretically proved. As the discriminability of the statistical tests increases, the power of all the algorithms approaches 1. When their FDR level q is set at the same value as the α of the PC-skeleton algorithm, the PC_{fdr}-skeleton algorithm and its heuristic modification control the type I error rate more stringently than the PC-skeleton algorithm does, so their power generally is lower than that of the PC-skeleton algorithm. Figure 4 also clearly shows, as Eq. 3 implies, that it is the type I error rate, rather than the FDR, that the PC-skeleton algorithm controls under 5%.

Figure 5 shows the average computational time spent during each run of the PC_{fdr}-skeleton algorithm and its heuristic modification on the statistical tests of (conditional) independence at step 8 and the FDR control at step 12. The computational time was estimated on the platform of an Intel Xeon 1.86GHz CPU and 4G RAM, and with the code implemented in Matlab R14. Tables 3 and 4 show the average ratios of the computational time spent on the FDR control to that spent on the statistical tests. The average ratios are not more than 2.57‰ for all the 48 Bayesian networks. The relatively small standard deviations, as shown in brackets in the tables, indicate that these estimated ratios are trustful. Because the PC_{fdr}-skeleton algorithm and its heuristic modification employ the same search strategy as the PC-skeleton algorithm does, this result evidences that the extra computation cost to achieve the control over the FDR is trivial in comparison with the computation already spent by the PC-skeleton algorithm on statistical tests.

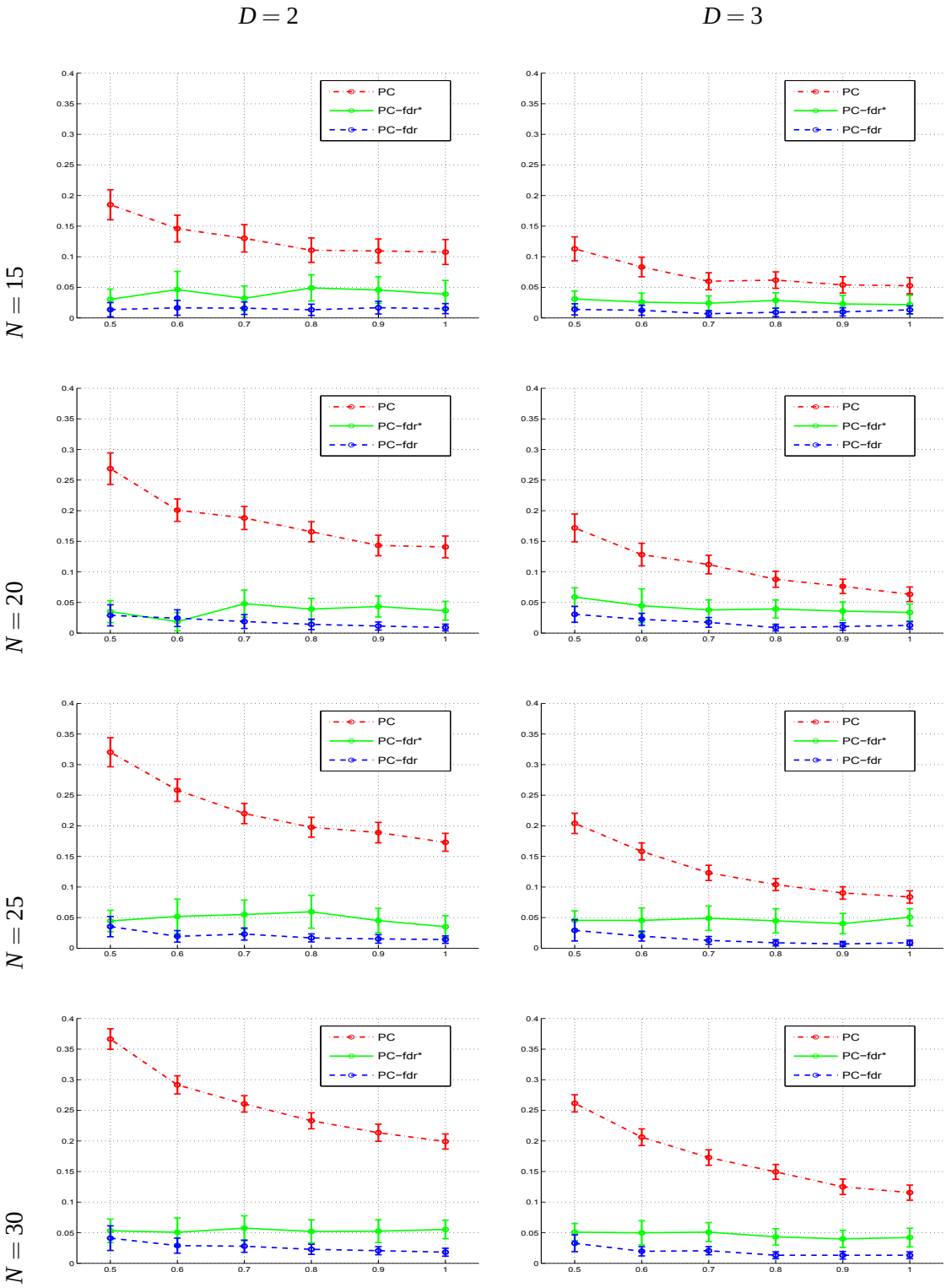

Figure 2: The FDR (with 95% confidence intervals) of the PC-skeleton algorithm, the PC_{fdr}-skeleton algorithm and the PC_{fdr*}-skeleton algorithm on the DAGs in Figure 1, as the dependence parameter δ shown on the x-axes increases from 0.5 to 1.0.

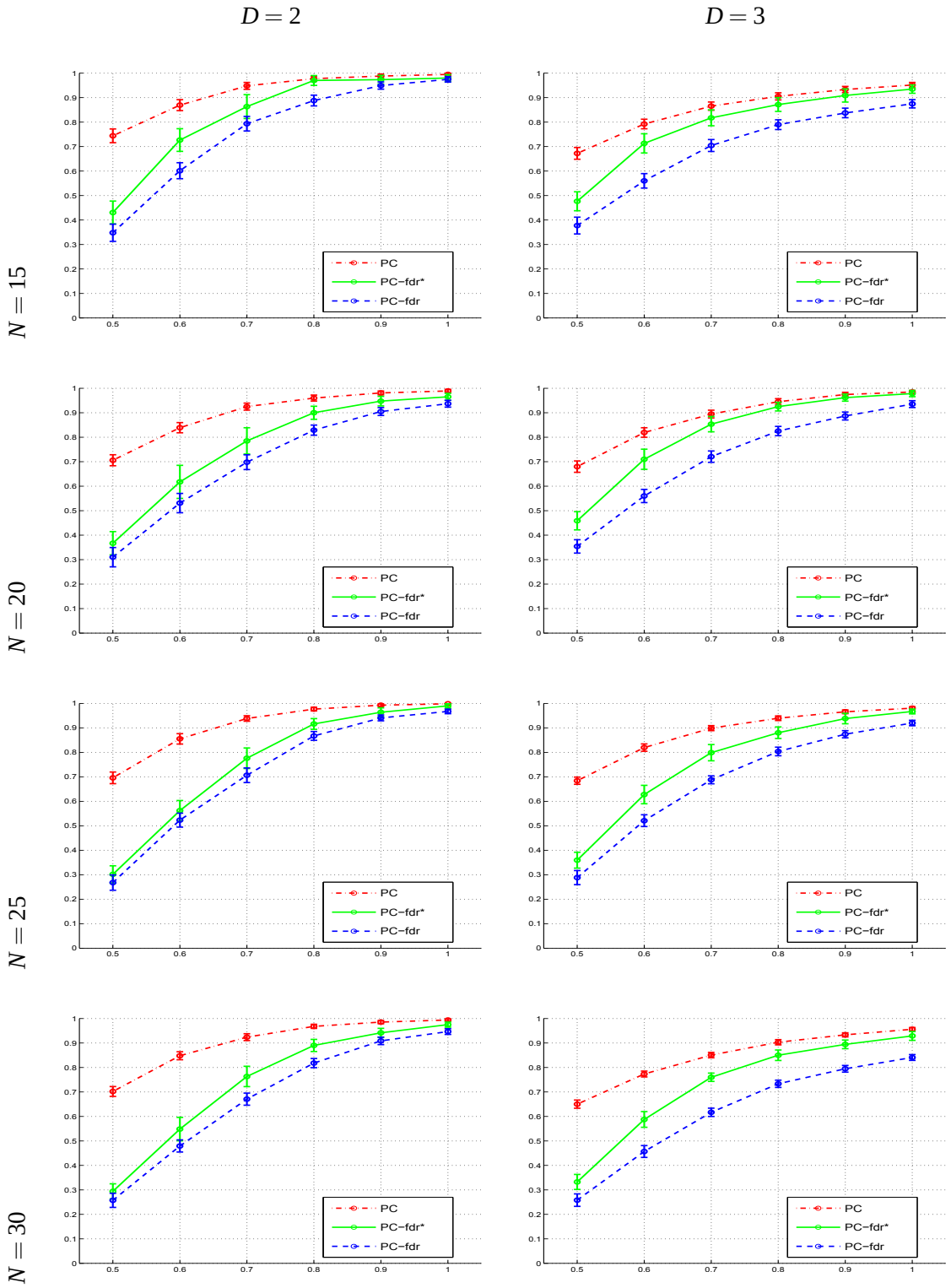

Figure 3: The power (with 95% confidence intervals) of the PC-skeleton algorithm, the PC_{fdr}-skeleton algorithm and the PC_{fdr*}-skeleton algorithm on the DAGs in Figure 1, as the dependence parameter δ shown on the x-axes increases from 0.5 to 1.0.

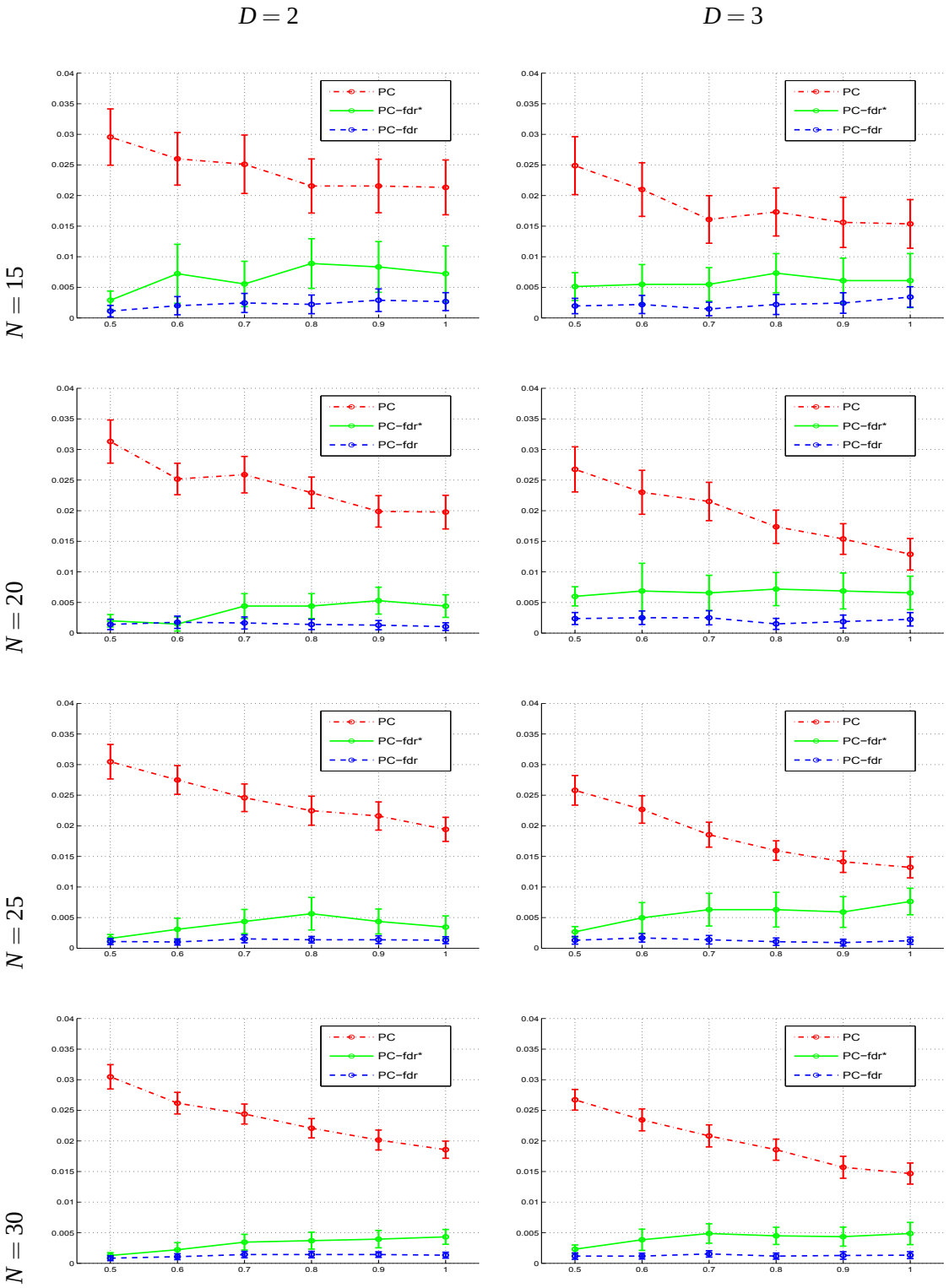

Figure 4: The type I error rates (with 95% confidence intervals) of the PC-skeleton algorithm, the PC_{fdr}-skeleton algorithm and the PC_{fdr*}-skeleton algorithm on the DAGs in Figure 1, as the dependence parameter δ shown on the x-axes increases from 0.5 to 1.0.

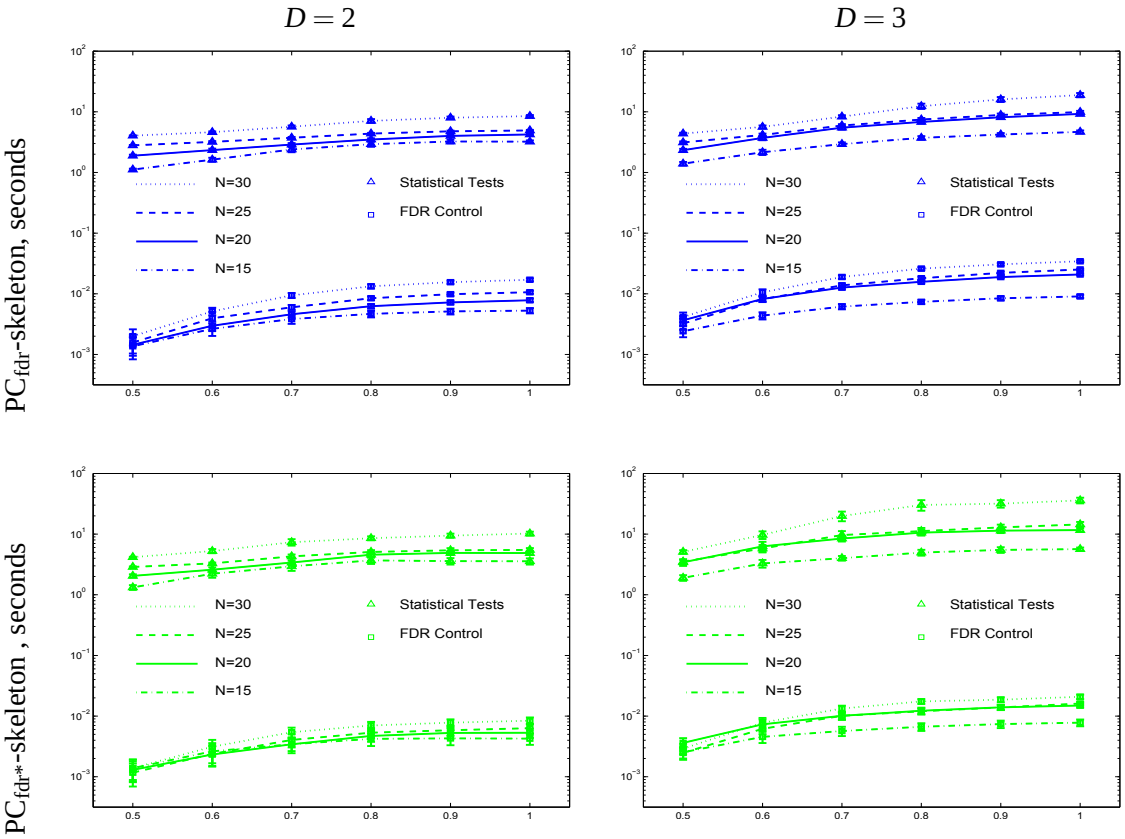

Figure 5: The average computational time (in seconds, with 95% confidence intervals) spent on the FDR control and statistical tests during each run of the PC_{fdr}-skeleton algorithm and its heuristic modification.

	δ	N=15	N=20	N=25	N=30
$D = 2$	0.5	1.11e-03 (3.64e-04)	7.19e-04 (2.32e-04)	5.44e-04 (1.79e-04)	4.81e-04 (1.37e-04)
	0.6	1.48e-03 (2.03e-04)	1.24e-03 (2.15e-04)	1.21e-03 (3.32e-04)	1.09e-03 (1.44e-04)
	0.7	1.58e-03 (1.68e-04)	1.61e-03 (2.01e-04)	1.59e-03 (1.31e-04)	1.64e-03 (1.09e-04)
	0.8	1.63e-03 (1.64e-04)	1.81e-03 (1.61e-04)	1.93e-03 (1.20e-04)	1.89e-03 (1.04e-04)
	0.9	1.59e-03 (1.50e-04)	1.83e-03 (1.50e-04)	2.06e-03 (1.19e-04)	1.95e-03 (9.63e-05)
	1.0	1.64e-03 (1.51e-04)	1.88e-03 (1.59e-04)	2.15e-03 (1.12e-04)	2.01e-03 (9.01e-05)
$D = 3$	0.5	1.69e-03 (3.70e-04)	1.50e-03 (2.55e-04)	9.80e-04 (1.90e-04)	9.10e-04 (1.52e-04)
	0.6	2.06e-03 (2.82e-04)	2.22e-03 (1.45e-04)	1.93e-03 (1.71e-04)	1.82e-03 (1.47e-04)
	0.7	2.11e-03 (1.84e-04)	2.36e-03 (1.24e-04)	2.31e-03 (1.19e-04)	2.29e-03 (1.12e-04)
	0.8	2.02e-03 (1.68e-04)	2.35e-03 (1.20e-04)	2.45e-03 (1.28e-04)	2.20e-03 (1.15e-04)
	0.9	2.04e-03 (1.50e-04)	2.34e-03 (9.05e-05)	2.53e-03 (1.15e-04)	2.03e-03 (1.23e-04)
	1.0	1.99e-03 (1.41e-04)	2.28e-03 (9.18e-05)	2.57e-03 (9.66e-05)	1.92e-03 (1.25e-04)

Table 3: The average ratios (with their standard deviations in brackets) of the computational time spent on the FDR control to that spent on the statistical tests during each run of the PC_{fdr}-skeleton algorithm.

	δ	N=15	N=20	N=25	N=30
$D = 2$	0.5	8.88e-04 (2.57e-04)	5.93e-04 (2.25e-04)	3.94e-04 (1.60e-04)	3.24e-04 (1.15e-04)
	0.6	1.12e-03 (2.99e-04)	8.82e-04 (3.19e-04)	7.08e-04 (2.58e-04)	5.86e-04 (1.81e-04)
	0.7	1.17e-03 (2.76e-04)	1.04e-03 (2.92e-04)	9.02e-04 (1.51e-04)	7.45e-04 (1.37e-04)
	0.8	1.16e-03 (2.66e-04)	1.04e-03 (1.83e-04)	1.03e-03 (1.38e-04)	8.23e-04 (1.24e-04)
	0.9	1.22e-03 (2.73e-04)	1.08e-03 (1.91e-04)	1.06e-03 (8.76e-05)	8.37e-04 (1.35e-04)
	1.0	1.21e-03 (2.68e-04)	1.11e-03 (2.09e-04)	1.12e-03 (1.04e-04)	8.31e-04 (1.05e-04)
$D = 3$	0.5	1.33e-03 (3.42e-04)	1.01e-03 (1.86e-04)	6.74e-04 (1.54e-04)	5.49e-04 (1.08e-04)
	0.6	1.44e-03 (3.46e-04)	1.19e-03 (1.63e-04)	1.08e-03 (2.08e-04)	7.97e-04 (8.83e-05)
	0.7	1.43e-03 (2.41e-04)	1.20e-03 (1.10e-04)	1.09e-03 (1.51e-04)	7.19e-04 (7.80e-05)
	0.8	1.36e-03 (1.38e-04)	1.17e-03 (9.36e-05)	1.10e-03 (1.20e-04)	6.48e-04 (9.32e-05)
	0.9	1.35e-03 (1.34e-04)	1.24e-03 (1.03e-04)	1.10e-03 (8.31e-05)	6.19e-04 (6.57e-05)
	1.0	1.39e-03 (1.72e-04)	1.29e-03 (1.01e-04)	1.14e-03 (9.77e-05)	5.98e-04 (4.86e-05)

Table 4: The average ratios (with their standard deviations in brackets) of the computational time spent on the FDR control to that spent on the statistical tests during each run of the PC_{fdr*}-skeleton algorithm.

3.2 Applications to Real fMRI Data

We applied the PC_{fdr}-skeleton and the PC-skeleton algorithms to real-world research tasks, studying the connectivity network between brain regions using functional magnetic resonance imaging (fMRI). The purpose of the applications is to check whether the two algorithms correctly curb the error rates in real world applications. The purpose of the applications is not, and also should not be, to answer the question "which algorithm, the PC_{fdr}-skeleton or the PC-skeleton, is superior?", for the following reasons. Basically, the two algorithms control different error rates between which there is not a superior relationship (see Appendix C). Secondly, the error rate of interest for a specific application is selected largely not by mathematical superiority, but by researchers' interest and the scenario of research (see Appendix C). Thirdly, the simulation study has clearly revealed the properties of and the differences (not superiority) between the two algorithms. Lastly, the approximating graphical models behind the real fMRI data are unknown, so the comparison on the real fMRI data is rough, rather than rigorous.

The two algorithms were applied to two real fMRI data sets, one including 11 discrete variables and 1300 observations, and the other including 25 continuous variables and 1098 observations. The first data set, denoted by "the bulb-squeezing data set", was collected from 10 healthy subjects each of whom was asked to squeeze a rubber bulb with their left hand at three different speeds or at a constant force, as cued by visual instruction. The data involve eleven variables: the speed of squeezing and the activities of the ten brain regions listed in Table 5. The speed of squeezing is coded as a discrete variable with four possible values: the high speed, the medium speed, the low speed, and the constant force. The activities of the brain regions are coded as discrete variables with three possible values: high activation, medium activation and low activation. The data of each subject include 130 time points. The data of the ten subjects are pooled together, so in total there are 1300 time points. For details of the data set, please refer to Li et al. (2008).

The second data set, denoted by "the sentence-picture data set", was collected from a single subject performing a cognitive task. In each trial of the task, the subject was shown in sequence an affirmative sentence and a simple picture, and then answered whether the sentence correctly

Full Name	Abbreviation
Left/Right anterior cingulate cortex	L_ACC, R_ACC
Left/Right lateral cerebellar hemispheres	L_CER, R_CER
Left/Right primary motor cortex	L_M1, R_M1
Left/Right pre-frontal cortex	L_PFC, R_PFC
Left/Right supplementary motor cortex	L_SMA, R_SMA

Table 5: Brain regions involved in the bulb-squeezing data set. The prefixes "L" or "R" in the abbreviations stand for "Left" or "Right", respectively.

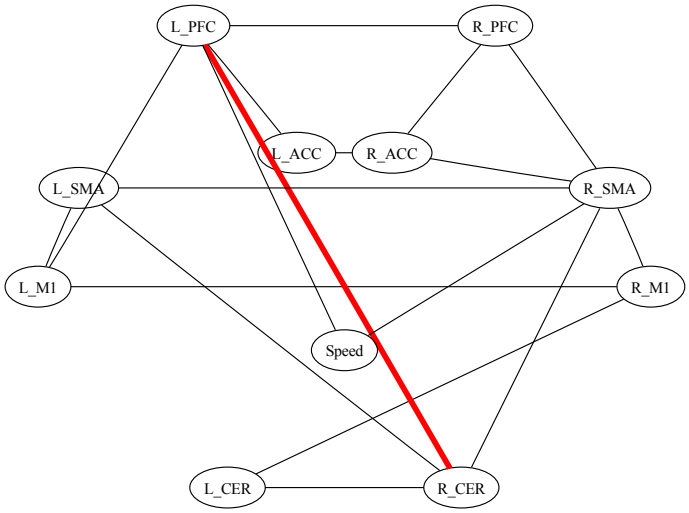

Figure 6: The networks learned from the bulb-squeezing data set, by the PC_{fdr}-skeleton and the PC-skeleton algorithms. For ease of comparison, the networks learned by the two algorithms are overlaid. Thin solid black edges are those connections detected by both the two algorithms; thick solid red edges are those connections detected only by the PC-skeleton algorithm. For the full names of the brain regions, please refer to Table 5.

described the picture. In half of the trials, the picture was presented first, followed by the sentence. In the remaining trials, the sentence was presented first, followed by the picture. The data involve the activities of 25 brain regions, as listed in Table 6, encoded as continuous variables, at 1098 time points. For details of the data set, please refer to Keller et al. (2001) and Mitchell et al. (2004).

The PC_{fdr}-skeleton and the PC-skeleton algorithms were applied to both the bulb-squeezing and the sentence-picture data sets. Both the FDR level q of the PC_{fdr}-skeleton algorithm and the type-I-error-rate level α of the PC-skeleton algorithm were set at 5%. For the bulb-squeezing data set, all of whose variables are discrete, conditional independence was tested with Pearson's Chi-square test; for the sentence-picture data set, all of whose variables are continuous, conditional independence was tested with the t-test for partial correlation coefficients (Fisher, 1924).

The networks learned from the bulb-squeezing data set and the networks learned from the sentence-picture data set are shown in Figures 6 and 7 respectively. For ease of comparison, the

Full Name	Abbreviation
Calcarine fissure	CALC
Left/Right dorsolateral prefrontal cortex	L_DLPFC, R_DLPFC
Left/Right frontal eye field	L_FEF, R_FEF
Left inferior frontal gyrus	L_IFG
Left/Right inferior parietal lobe	L_IPL, R_IPL
Left/Right intraparietal sulcus	L_IPS, R_IPS
Left/Right inferior temporal lobule	L_IT, R_IT
Left/Right opercularis	L_OPER, R_OPER
Left/Right posterior precentral sulcus	L_PPREC, R_PPREC
Left/Right supramarginal gyrus	L_SGA, R_SGA
Supplementary motor cortex	SMA
Left/Right superior parietal lobule	L_SPL, R_SPL
Left/Right temporal lobe	L_T, R_T
Left/Right triangularis	L_TRIA, R_TRIA

Table 6: Brain regions involved in the sentence-picture data set. The prefixes "L" or "R" in the abbreviations stand for "Left" or "Right", respectively.

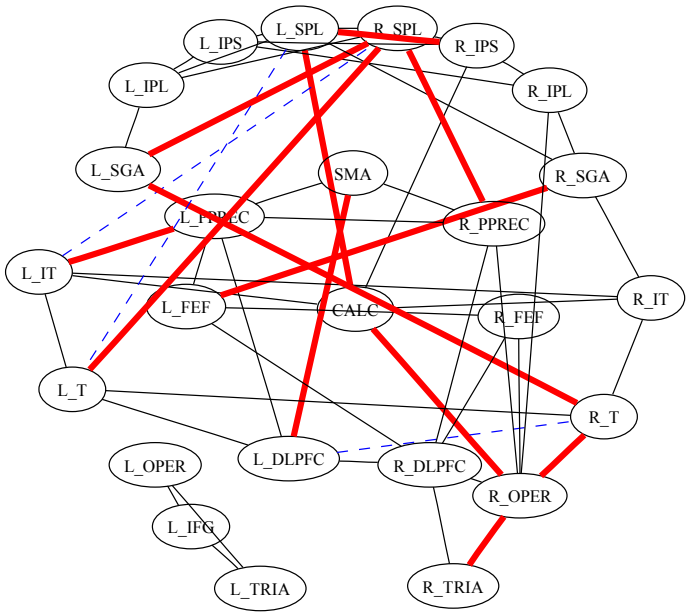

Figure 7: The networks learned from the sentence-picture data set, by the PC_{fdr}-skeleton and the PC-skeleton algorithms. For ease of comparison, the networks learned by the two algorithms are overlaid. Thin solid black edges are those connections detected by both the two algorithms; thin dashed blue edges are those connections detected only by the PC_{fdr}-skeleton algorithm; thick solid red edges are those connections detected only by the PC-skeleton algorithm. For the full names of the brain regions, please refer to Table 6.

Bulb-Squeezing

	Assumed Truth		*Realized* Detection			
	Exist	Non-Exist	Correct	False	FDR	Type I Error Rate
PC_{fdr}	17	$\frac{11*(11-1)}{2} - 17 = 38$	17	0	**0.00%**	0.00%
PC			17	1	5.56%	**2.63%**

Sentence-Picture

	Assumed Truth		*Realized* Detection			
	Exist	Non-Exist	Correct	False	FDR	Type I Error Rate
PC_{fdr}	39	$\frac{25*(25-1)}{2} - 39 = 261$	39	3	**7.14%**	1.14%
PC			39	12	23.5%	**4.60%**

Table 7: The *realized* error rates of the PC_{fdr}-skeleton and the PC algorithms on the bulb-squeezing and sentence-picture data sets, under the TI assumption that all and only those connections detected by both of the two algorithms truly exist.

networks learned by the two algorithms are overlaid. Thin solid black edges are those connections detected by both the two algorithms; thin dashed blue edges are those detected only by the PC_{fdr}-skeleton algorithm; thick solid red edges are those detected only by the PC-skeleton algorithm. In Figure 6, there are 17 thin solid black edges, 0 thin dashed blue edge and 1 thick solid red edge; in Figure 7, there are 39 thin solid black edges, 3 thin dashed blue edges and 12 thick solid red edges.

The results intuitively, though not rigorously, support our expectation of the performances of the two algorithms in real world applications. First, since the data sets are relatively large, with the sample sizes more than 1000, it is expected that both algorithms will recover many of the existing connections, and consequently the networks recovered by the two algorithms may share many common connections. This is consistent with the fact that in Figures 6 and 7 there are many thin solid black edges, that is, the connections recovered by both algorithms.

Second, since the PC_{fdr}-skeleton algorithm is designed to control the FDR while the PC-skeleton algorithm to control the type I error rate, it is expected that the two algorithms will control the corresponding error rate under or around the pre-defined level, which is 5% in this study. To verify whether the error rates were controlled as expected, we need to know which connections really exist and which do not. Unfortunately, this is very difficult for real data sets, because unlike the simulated data, the true models behind the real data are unknown, and in the literature, researchers usually tend to report evidences supporting the existence of connections rather than supporting the non-existence. However, since the sample sizes of the two data sets are relatively large, more than 1000, we can speculate that both of the two algorithms have recovered most of the existing connections. Extrapolating this speculation a bit, we intuitively assume that those connections detected by both of the two algorithms truly exist while all the others do not. In other words, we assume that all and only the thin black edges in the figures truly exist. We refer to this assumption as the "True Intersection" (TI) assumption. The statistics about Figures 6 and 7, under the TI assumption, are listed in Table 7. The *realized* FDR of the PC_{fdr}-skeleton algorithm on the bulb-squeezing and sentence-picture data sets are 0.00% and 7.14%, respectively; the *realized* type I error rate of the PC-skeleton algorithm on the bulb-squeezing and sentence-picture data sets are 2.63% and 4.60%, respectively. Considering that the *realized* error rate, as a statistic extracted from just a trial, may

slightly deviate from its expected value, these results, derived under the TI assumption, support that the two algorithms controlled the corresponding error rate under the pre-defined level 5%.

Third, according to Eq. (2), the sparser and the larger the true network is, the higher the FDR of the PC-skeleton algorithm will be. For the bulb-squeezing data set, there are 11 vertices, and under the TI assumption, 17 existing connections and 38 non-existing connections. In this case, the *realized* FDR of the PC-skeleton algorithm is only 5.56% (Table 7). For the sentence-picture data set, there are 25 vertices, and under the TI assumption, 39 existing connections and 261 non-existing connections. In this case, the *realized* FDR of the PC-skeleton algorithm rises to 23.5% (Table 7). This notable increase of the *realized* FDR is consistent with the prediction based on Eq. (2).

It should be noted that the preceding arguments are rough rather than rigorous, since they are based on the TI assumption rather than the true models behind the data. However, because the true models behind the real data are unknown, the TI assumption is a practical and intuitive approach to assess the performance of the two algorithms in the two real world applications.

4. Conclusions and Discussions

We have proposed a modification of the PC algorithm, the PC_{fdr}-skeleton algorithm, to curb the false discovery rate (FDR) of the skeleton of the learned Bayesian networks. The FDR-control procedure embedded into the PC algorithm collectively considers the hypothesis tests related to the existence of multiple edges, correcting the effect of multiple hypothesis testing. Under mild assumptions, it is proved that the PC_{fdr}-skeleton algorithm can control the FDR under a user-specified level q ($q > 0$) at the limit of large sample sizes (see Theorem 2). In the cases of moderate sample size (about several hundred), empirical experiments have shown that the method is still able to control the FDR under the user-specified level. The $\text{PC}_{\text{fdr}*}$-skeleton algorithm, a heuristic modification of the proposed method, has shown better performance in the simulation study, steadily controlling the FDR closely around the user-specified level and gaining more detection power, although its asymptotic performance has not been theoretically proved. Both the PC_{fdr}-skeleton algorithm and its heuristic modification can asymptotically recover all the edges of the true DAG (see Theorem 1). The idea of controlling the FDR can be extended to other constraint-based methods, such as the inductive causation (IC) algorithm (see Pearl, 2000, pages 49–51) and the fast-causal-inference (FCI) algorithm (see Spirtes et al., 2001, pages 142–146).

The simulation study has also shown that the extra computation spent on achieving the FDR control is almost negligible when compared with that already spent by the PC algorithm on statistical tests of conditional independence. The computational complexity of the new algorithm is closely comparable with that of the PC algorithm.

As a modification based on the PC algorithm, the proposed method is modular, consisting of the PC search strategy, statistical tests of conditional independence and an FDR-control procedure. Different statistical tests and FDR-control procedures can be "plugged in", depending on the data type and the statistical model. Thus, the method is applicable to any models for which statistical tests of conditional independence are available, such as discrete models and Gaussian models.

It should be noted that the PC_{fdr}-skeleton algorithm is not proposed to replace the PC-skeleton algorithm. Instead, it provides an approach to controlling the FDR, a certain error-rate criterion for testing the existence of multiple edges. When multiple edges are involved in structure learning, there are different applicable error-rate criteria, such as those listed in Table 2. The selection of these criteria depends on researchers' interest and the scenarios of studies, which is beyond the scope of

this paper. When the FDR is applied, the PC_{fdr}-skeleton algorithm is preferable; when the type I error rate is applied, the PC-skeleton algorithm is preferable. The technical difference between the two algorithms is that the PC_{fdr}-skeleton algorithm adaptively adjusts the type I error rate according to the sparseness of the network to achieve the FDR control, while the PC-skeleton algorithm fixes the type I error rate.

Currently the FDR control is applied only to the skeleton of the graph, but not to the directions of the edges yet. The final output of the PC algorithm is a partially directed acyclic graph that uniquely represents an equivalence class of DAGs, so a possible improvement for the PC_{fdr}-skeleton algorithm is to extend the FDR control to the directions of the recovered edges. Because both type I and type II errors may lead to wrong directions in the later steps of the PC algorithm, minimizing direction errors may lead to a related, yet different, error-control task.

The asymptotic performance of the PC_{fdr}-skeleton algorithm has only been proved under the assumption that the number of vertices is fixed. Its behavior when both the number of vertices and the sample size approach infinity has not been studied yet. Kalisch and Bühlmann (2007) proved that for Gaussian Bayesian networks, the PC algorithm consistently recovers the equivalence class of the underlying sparse DAG, as the sample size m approaches infinity, even if the number of vertices N grows as quickly as $O(m^\lambda)$ for any $0 < \lambda < \infty$. Their idea is to adaptively decrease the type I error rate α of the PC-skeleton algorithm as both the number of vertices and the sample size increase. It is desirable to study whether similar behavior can be achieved with the PC_{fdr}-skeleton algorithm if the FDR level q is adjusted appropriately as the sample size increases.

A Matlab® package of the PC_{fdr}-skeleton algorithm and its heuristic modification is downloadable at www.junningli.org/software.

Acknowledgments

The authors thank Dr. Martin J. McKeown for sharing the functional magnetic resonance imaging (fMRI) data (Li et al., 2008) and helpful discussions.

Appendix A. Proof of Theorems

To assist the reading, we list below notations frequently used in the proof:

$G^{\sim}_{true}$: the skeleton of the true underlying Bayesian network.

$\mathcal{A}_{a\sim b}$: the event that edge $a \sim b$ is in the graph recovered by the PC_{fdr}-skeleton algorithm.

$\mathcal{A}_{E^{\sim}_{true}}$: $\mathcal{A}_{E^{\sim}_{true}} = \bigcap_{a\sim b \in E^{\sim}_{true}} \mathcal{A}_{a\sim b}$, the joint event that all the edges in $G^{\sim}_{true}$, the skeleton of the true DAG, are recovered by the PC_{fdr}-skeleton algorithm.

$E^{\not\sim}_{true}$: the set of the undirected edges that are not in $G^{\sim}_{true}$.

$p_{a\sim b}$: the value of $p^{max}_{a\sim b}$ when the PC_{fdr}-skeleton algorithm stops.

$C^{*}_{a\sim b}$: a certain vertex set that d-separates a and b in G_{true} and that is also a subset of either $\mathrm{adj}(a, G^{\sim}_{true}) \setminus \{b\}$ or $\mathrm{adj}(b, G^{\sim}_{true}) \setminus \{a\}$, according to Proposition 1. $C^{*}_{a\sim b}$ is defined only for vertex pairs that are not adjacent in the true DAG G_{true}.

$p^*_{a\sim b}$: the p-value of testing $X_a \perp X_b | X_{C^*_{a\sim b}}$. The conditional-independence relationship may not be really tested during the process of the PC_{fdr}-skeleton algorithm, but $p^*_{a\sim b}$ can still denote the value as if the conditional-independence relationship was tested.

H^*: the value in Eq. (1) that is either H or $H(1+1/2,\ldots,+1/H)$, depending on the assumption of the dependency of the p-values.

Lemma 1 *If as m approaches infinity, the probabilities of K events $\mathcal{A}_i(m),\cdots,\mathcal{A}_K(m)$ approach 1 at speed*

$$P(\mathcal{A}_i(m)) = 1 - o(\beta(m))$$

where $\lim_{m\to\infty}\beta(m) = 0$ and K is a finite integer, then the probability of the joint of all these events approaches 1 at speed

$$P\left(\bigcap_{i=1}^{K}\mathcal{A}_i(m)\right) \geq 1 - Ko(\beta(m))$$

as m approaches infinity.

Proof
$\because \bigcap_{i=1}^{K}\mathcal{A}_i(m) = \overline{\bigcup_{i=1}^{K}\overline{\mathcal{A}}_i(m)}.$
$\therefore P\left(\bigcap_{i=1}^{K}\mathcal{A}_i(m)\right) = 1 - P\left(\bigcup_{i=1}^{K}\overline{\mathcal{A}}_i(m)\right) \geq 1 - \sum_{i=1}^{K}P(\overline{\mathcal{A}}_i(m))$
$= 1 - \sum_{i=1}^{K}[1 - P(\mathcal{A}_i(m))] = 1 - \sum_{i=1}^{K}o(\beta(m)) = 1 - Ko(\beta(m)).$ ■

Corollary 1 *If $\mathcal{A}_i(m),\cdots,\mathcal{A}_K(m)$ are a finite number of events whose probabilities each approach 1 as m approaches infinity:*

$$\lim_{m\to\infty} P(\mathcal{A}_i(m)) = 1,$$

then the probability of the joint of all these events approaches 1 as m approaches infinity:

$$\lim_{m\to\infty} P\left(\bigcap_{i=1}^{K}\mathcal{A}_i(m)\right) = 1.$$

Lemma 2 *If there are F ($F \geq 1$) false hypotheses among H tested hypotheses, and the p-values of the all the false hypotheses are smaller than or equal to $\frac{F}{H^*}q$, where H^* is either H or $H(1+1/2,\ldots,+1/H)$, depending on the assumption of the dependency of the p-values, then all the F false hypotheses will be rejected by the FDR procedure, Algorithm 2.*

Proof
Let p_i ($i = 1,\cdots,H$) denote the p-value of the ith hypothesis, p_f denote the maximum of the p-values of the F false hypotheses, and r_f denote the rank of p_f in the ascending order of $\{p_i\}_{i=1,\cdots,H}$.
$\because p_f$ is the maximum of the p-values of the F false hypotheses.
$\therefore r_f = |\{p_i | p_i \leq p_f\}| \geq F.$
$\therefore \frac{H^*}{r_f}p_f \leq \frac{H^*}{F}p_f.$

$\because p_f \leq \frac{F}{H^*} q$.
$\therefore \frac{H^*}{r_f} p_f \leq \frac{H^*}{F} p_f \leq q$.
$\therefore$ Hypotheses with p-values not greater than p_f will be rejected.
$\because$ The p-values of the F false hypotheses are not greater than p_f.
$\therefore$ All the F false hypotheses will be rejected by the FDR procedure, Algorithm 2. ■

Proof of **Theorem 1**
If there is not any edge in the true DAG G_{true}, then the proof is trivially $E^{\sim}_{true} = \emptyset \subseteq E^{\sim}$. In the following part of the proof, we assume $E^{\sim}_{true} \neq \emptyset$. For the PC$_{\text{fdr}}$-skeleton algorithm and its heuristic modification, whenever the FDR procedure, Algorithm 2, is invoked, $p^{max}_{a\sim b}$ is always less than $\max_{C \in V\setminus\{a,b\}} \{p_{a\perp b|C}\}$, and the number of p-values input to the FDR algorithm is always not more than C_N^2. Thus, according to Lemma 2, if

$$\max_{a\sim b \in E^{\sim}_{true}} \left\{ \max_{C \in V\setminus\{a,b\}} \{p_{a\perp b|C}\} \right\} \leq \frac{|E^{\sim}_{true}|}{C_N^2 \sum_{i=1}^{C_N^2} \frac{1}{i}} q, \tag{5}$$

then all the true connections will be recovered by the PC$_{\text{fdr}}$-skeleton algorithm and its heuristic modification. Let $\mathcal{A}'_{a\perp b|C}$ denote the event

$$p_{a\perp b|C} \leq \frac{|E^{\sim}_{true}|}{C_N^2 \sum_{i=1}^{C_N^2} \frac{1}{i}} q,$$

$\mathcal{A}'_{E^{\sim}_{true}}$ denote the event of Eq. (5), and $\mathcal{A}_{E^{\sim}_{true}}$ denote the event that all the true connections are recovered by the PC$_{\text{fdr}}$-skeleton algorithm and its heuristic modification.
$\because \mathcal{A}'_{E^{\sim}_{true}}$ is a sufficient condition for $\mathcal{A}_{E^{\sim}_{true}}$, according to Lemma 2.
$\therefore \mathcal{A}_{E^{\sim}_{true}} \supseteq \mathcal{A}'_{E^{\sim}_{true}}$.
$\therefore P(\mathcal{A}_{E^{\sim}_{true}}) \geq P(\mathcal{A}'_{E^{\sim}_{true}})$.
$\because \mathcal{A}'_{E^{\sim}_{true}}$ is the joint of a limited number of events as

$$\mathcal{A}'_{E^{\sim}_{true}} = \bigcap_{a\sim b \in E^{\sim}_{true}} \bigcap_{C \subseteq V\setminus\{a,b\}} \mathcal{A}'_{a\perp b|C},$$

and $\lim_{m\to\infty} P(\mathcal{A}'_{a\perp b|C}) = 1$ according to Assumption (A3).
$\therefore$ According to Corollary 1, $\lim_{m\to\infty} P(\mathcal{A}'_{E^{\sim}_{true}}) = 1$.
$\therefore 1 \geq \lim_{m\to\infty} P(\mathcal{A}_{E^{\sim}_{true}}) \geq \lim_{m\to\infty} P(\mathcal{A}'_{E^{\sim}_{true}}) = 1$.
$\therefore \lim_{m\to\infty} P(\mathcal{A}_{E^{\sim}_{true}}) = 1$. ■

Lemma 3 *Given any FDR level $q > 0$, if the p-value vector $P = [p_1, \cdots, p_H]$ input to Algorithm 2 is replaced with $P' = [p'_1, \cdots, p'_H]$, such that (1) for the those hypotheses that are rejected when P is the input, p'_i is equal to or less than p_i, and (2) for all the other hypotheses, p'_i can be any value between 0 and 1, then the set of rejected hypotheses when P' is the input is a superset of those rejected when P is the input.*

Proof
Let R and R' denote the sets of the rejected hypotheses when P and P' are respectively input to the FDR procedure.

If $R = \emptyset$, then the proof is trivially $R' \supseteq \emptyset = R$.

If $R \neq \emptyset$, let us define $\alpha = \max_{i \in R} p_i$ and $\alpha' = \max_{i \in R} p'_i$. Let $r = |R|$ denote the rank of α in the ascending order of P and r' denote the rank of α' in the ascending order of P'.
$\because p'_i \leq p_i$ for all $i \in R$.
$\therefore \alpha' \leq \alpha$.
$\because \alpha' = \max_{i \in R} p'_i$.
$\therefore r' \geq |R| = r$.
$\because \frac{H^*}{r}\alpha \leq q$.
$\therefore \frac{H^*}{r'}\alpha' \leq \frac{H^*}{r}\alpha \leq q$.
$\therefore$ When P' is the input, hypotheses with p'_i smaller than or equal to α' will be rejected.
$\because p'_i \leq \alpha', \forall i \in R$.
$\therefore R \subseteq R'$, equivalently $R' \supseteq R$. ■

Corollary 2 *Given any FDR level $q > 0$, if the p-value vector $P = [p_1, \cdots, p_H]$ input to Algorithm 2 is replaced with $P' = [p'_1, \cdots, p'_H]$ such that $p'_i \leq p_i$ for all $i = 1, \cdots, H$, then the set of rejected hypotheses when P' is the input is a superset of those rejected when P is the input.*

Proof of **Theorem 2**
Let $E^{\sim}_{stop}$ and $E^{\nsim}_{stop}$ denote the undirected edges respectively recovered and removed by the PC_{fdr}-skeleton algorithm when the algorithm stops. Let sequence $P_1^{max}, \cdots, P_K^{max}$ denote the values of P^{max} when the FDR procedure is invoked at step 12 as the algorithm progresses, in the order of the update process of P^{max}, and let $E_k^{\nsim}$ denote the set of removable edges indicated by the FDR procedure, with P_k^{max} as the input. $E_k^{\nsim}$ may include edges that have already been removed.
$\because$ The PC_{fdr}-skeleton algorithm accumulatively removes edges in $E_k^{\nsim}$.
$\therefore E^{\nsim}_{stop} = \bigcup_{k=1}^{K} E_k^{\nsim}$.
$\because P^{max}$ is updated increasingly at step 10 of the algorithm.
$\therefore$ According to **Corollary 2**, $E_1^{\nsim} \subseteq \cdots \subseteq E_K^{\nsim}$.
$\therefore E^{\nsim}_{stop} = \bigcup_{k=1}^{K} E_k^{\nsim} = E_K^{\nsim}$.

Let $P = \{p_{a \sim b}\}$ denote the value of P^{max} when the PC_{fdr}-skeleton algorithm stops.
$\because$ The FDR procedure is invoked whenever P^{max} is updated.
$\therefore$ The value of P^{max} does not change after the FDR procedure is invoked for the last time.
$\therefore P = P_K^{max}$.
$\therefore E^{\sim}_{stop}$ is the same as the edges recovered by directly applying the FDR procedure to P.

The theorem is proved through comparing the result of the PC_{fdr}-skeleton algorithm with that of applying the FDR procedure to a virtual p-value set constructed from P. The virtual p-value set P^* is defined as follows.

For a vertex pair $a \sim b$ that is not adjacent in the true DAG G_{true}, let $C^*_{a \sim b}$ denote a certain vertex set that d-separates a and b in G_{true} and that is also a subset of either $\text{adj}(a, G^{\sim}_{true}) \setminus \{b\}$ or

$\text{adj}(b, G^{\sim}_{true}) \setminus \{a\}$. Let us define $P^* = \{p^*_{a\sim b}\}$ as:

$$p^*_{a\sim b} = \begin{cases} p_{a\perp b|C^*_{a\sim b}} & : \quad a \sim b \in E^{\nsim}_{true}, \\ p_{a\sim b} & : \quad a \sim b \in E^{\sim}_{true}. \end{cases}$$

Though $p_{a\perp b|C^*_{a\sim b}}$ may not be actually calculated during the process of the algorithm, $p_{a\perp b|C^*_{a\sim b}}$ still can denote the value as if it was calculated. Let us design a virtual algorithm, called *Algorithm**, that recovers edges by just applying the FDR procedure to P^*, and let $E^{\sim *}$ denote the edges recovered by this virtual algorithm. This algorithm is virtual and impracticable because the calculation of P^* depends on the unknown $E^{\sim}_{true}$, but this algorithm exists because $E^{\sim}_{true}$ exists. For any vertex pair a and b that is not adjacent in G_{true}:
$\because$ X_a and X_b are conditional independent given $X_{C^*_{a\sim b}}$.
$\therefore$ $p_{a\perp b|C^*_{a\sim b}}$ follows the uniform distribution on [0, 1].
$\therefore$ The FDR of *Algorithm** is under q.

When all the true edges are recovered by the PC$_{\text{fdr}}$-skeleton algorithm, that is, $E^{\sim}_{true} \subseteq E^{\sim}_{stop}$, the conditional independence between X_a and X_b given $X_{C^*_{a\sim b}}$ is tested for all the falsely recovered edges $a \sim b \in E^{\nsim}_{true} \bigcap E^{\sim}_{stop}$, because for these edges, subsets of $\text{adj}(a, G_{true}) \setminus \{b\}$ and subsets of $\text{adj}(a, G_{true}) \setminus \{b\}$ have been exhaustively searched and $C^*_{a\sim b}$ is one of them. Therefore, $p_{a\sim b} \geq p^*_{a\sim b}$ for all $a \sim b \in E^{\sim}_{stop}$ when event $\mathcal{A}_{E^{\sim}_{true}}$ happens. Consequently, according to Lemma 3, if event $\mathcal{A}_{E^{\sim}_{true}}$ happens, $E^{\sim}_{stop} \subseteq E^{\sim *}$.

Let $q(E^{\sim})$ denote the realized FDR of reporting $E^{\sim}$ as the recovered skeleton of the true DAG:

$$q(E^{\sim}) = \begin{cases} \frac{|E^{\sim} \bigcap E^{\nsim}_{true}|}{|E^{\sim}|} & : \quad E^{\sim} \neq \emptyset, \\ 0 & : \quad E^{\sim} = \emptyset. \end{cases}$$

The FDRs of the PC$_{\text{fdr}}$-skeleton algorithm and *Algorithm** are $E[q(E^{\sim}_{stop})]$ and $E[q(E^{\sim *})]$ respectively. Here $E[x]$ means the expected value of x.

$\because$ $E[q(E^{\sim}_{stop})] = E[q(E^{\sim}_{stop})|\mathcal{A}_{E^{\sim}_{true}}]P(\mathcal{A}_{E^{\sim}_{true}}) + E[q(E^{\sim}_{stop})|\overline{\mathcal{A}}_{E^{\sim}_{true}}]P(\overline{\mathcal{A}}_{E^{\sim}_{true}})$
$\leq Q + P(\overline{\mathcal{A}}_{E^{\sim}_{true}})$, where $Q = E[q(E^{\sim}_{stop})|\mathcal{A}_{E^{\sim}_{true}}]P(\mathcal{A}_{E^{\sim}_{true}})$.
$\therefore$ $\limsup_{m\to\infty} E[q(E^{\sim}_{stop})] \leq \limsup_{m\to\infty} Q + \limsup_{m\to\infty} P(\overline{\mathcal{A}}_{E^{\sim}_{true}})$.
$\because$ $\lim_{m\to\infty} P(\mathcal{A}_{E^{\sim}_{true}}) = 1$, according to **Theorem 1**.
$\therefore$ $\limsup_{m\to\infty} P(\overline{\mathcal{A}}_{E^{\sim}_{true}}) = \lim_{m\to\infty} P(\overline{\mathcal{A}}_{E^{\sim}_{true}}) = 0$.
$\therefore$ $\limsup_{m\to\infty} E[q(E^{\sim}_{stop})] \leq \limsup_{m\to\infty} Q$.
$\because$ $Q \leq E[q(E^{\sim}_{stop})]$.
$\therefore$ $\limsup_{m\to\infty} Q \leq \limsup_{m\to\infty} E[q(E^{\sim}_{stop})]$.
$\therefore$ $\limsup_{m\to\infty} E[q(E^{\sim}_{stop})] = \limsup_{m\to\infty} Q = \limsup_{m\to\infty} E[q(E^{\sim}_{stop})|\mathcal{A}_{E^{\sim}_{true}}]P(\mathcal{A}_{E^{\sim}_{true}})$.

Similarly, $\limsup_{m\to\infty} E[q(E^{\sim *})] = \limsup_{m\to\infty} E[q(E^{\sim *})|\mathcal{A}_{E^{\sim}_{true}}]P(\mathcal{A}_{E^{\sim}_{true}})$.

$\because$ Given event $\mathcal{A}_{E^{\sim}_{true}}$, $E^{\sim}_{true} \subseteq E^{\sim}_{stop} \subseteq E^{\sim *}$.

$\therefore$ Given event $\mathcal{A}_{E^{\sim}_{true}}$,

$$q(E^{\sim}_{stop}) = \frac{|E^{\sim}_{stop}| - |E^{\sim}_{true}|}{|E^{\sim}_{stop}|} = 1 - \frac{|E^{\sim}_{true}|}{|E^{\sim}_{stop}|} \leq 1 - \frac{|E^{\sim}_{true}|}{|E^{\sim *}|} = \frac{|E^{\sim *}| - |E^{\sim}_{true}|}{|E^{\sim *}|} = q(E^{\sim *}).$$

$\therefore \limsup_{m\to\infty} E[q(E^{\sim}_{stop})|\mathcal{A}_{E^{\sim}_{true}}]P(\mathcal{A}_{E^{\sim}_{true}}) \leq \limsup_{m\to\infty} E[q(E^{\sim *})|\mathcal{A}_{E^{\sim}_{true}}]P(\mathcal{A}_{E^{\sim}_{true}})$.
$\therefore \limsup_{m\to\infty} E[q(E^{\sim}_{stop})] \leq \limsup_{m\to\infty} E[q(E^{\sim *})]$.
$\because Algorithm^{*}$ controls the FDR under q.
$\therefore E[q(E^{\sim *})] \leq q$.
$\therefore \limsup_{m\to\infty} E[q(E^{\sim *})] \leq q$.
$\therefore \limsup_{m\to\infty} E[q(E^{\sim}_{stop})] \leq q$. ■

Appendix B. Statistical Tests with Asymptotic Power Equal to One

Assumption (A3) on the asymptotic power of detecting conditional dependence appears demanding, but actually the detection power of several standard statistical tests approaches one as the number of identically and independently sampled observations approaches infinity. Listed as follows are two statistical tests satisfying Assumption (A3) for Gaussian models or discrete models.

B.1 Fisher's z Transformation on Sample Partial-correlation-coefficients for Gaussian Models

In multivariate Gaussian models, X_a and X_b are conditional independent given X_C if and only if the partial-correlation-coefficient of X_a and X_b given X_C is zero (see Lauritzen, 1996, pages 129–130). The partial-correlation-coefficient ρ is defined as:

$$\begin{aligned}
\rho &= \frac{\text{Cov}[Y_a, Y_b]}{\sqrt{\text{Var}[Y_a]\text{Var}[Y_b]}}, \\
Y_a &= X_a - < W_a, X_C >, \\
Y_b &= X_b - < W_b, X_C >, \\
W_a &= \arg\min_w E[(X_a - < w, X_C >)^2], \\
W_b &= \arg\min_w E[(X_b - < w, X_C >)^2].
\end{aligned}$$

The sample partial-correlation-coefficient $\hat{\rho}$ can be calculated from m i.i.d. samples $[x_{ai}, x_{bi}, x_{Ci}]$ $(i = 1, \cdots, m)$ as:

$$\begin{aligned}
\hat{\rho} &= \frac{\frac{1}{m}\sum_{i=1}^{m}[(\hat{y}_{ai}-\bar{y}_a)(\hat{y}_{bi}-\bar{y}_b)]}{\sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_{ai}-\bar{y}_a)^2 \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_{bi}-\bar{y}_b)^2}}, \\
\bar{y}_a &= \frac{1}{m}\sum_{i=1}^{m}\hat{y}_{ai}, \\
\bar{y}_b &= \frac{1}{m}\sum_{i=1}^{m}\hat{y}_{bi}, \\
\hat{y}_{ai} &= x_{ai} - < \hat{W}_a, x_{Ci} >, \\
\hat{y}_{bi} &= x_{bi} - < \hat{W}_b, x_{Ci} >, \\
\hat{W}_a &= \arg\min_{w} \sum_{i=1}^{m}(x_{ai} - < w, x_{Ci} >)^2, \\
\hat{W}_b &= \arg\min_{w} \sum_{i=1}^{m}(x_{bi} - < w, x_{Ci} >)^2.
\end{aligned}$$

The asymptotic distribution of $z(\hat{\rho})$, where $z(x)$, the Fisher's z transformation (see Fisher, 1915), is defined as

$$z(x) = \frac{1}{2}\log\frac{1+x}{1-x},$$

is the normal distribution with mean $z(\rho)$ and variance $1/(m-|C|-3)$ (see Anderson, 1984, pages 120–134). When the type I error rate is kept lower than α, the power of detecting $\rho \neq 0$ with Fisher's z transformation is the probability that $\sqrt{m-|C|-3}\, z(\hat{\rho})$ falls in the range $(-\infty, \Phi^{-1}(\alpha/2)]$ or $[\Phi^{-1}(1-\alpha/2), +\infty)$, where Φ is the cumulative distribution function of the standard normal distribution and Φ^{-1} is its inverse function. Without loss of generality, we assume the true partial-correlation-coefficient ρ is greater than zero, then the asymptotic power is

$$\begin{aligned}
&\lim_{m\to\infty} \text{Power} \geq \lim_{m\to\infty} P\left(\sqrt{m-|C|-3}\, z(\hat{\rho}) \geq \Phi^{-1}(1-\alpha/2)\right) \\
&= \lim_{m\to\infty}\left(1 - \Phi[\Phi^{-1}(1-\alpha/2) - \sqrt{m-|C|-3}\, z(\rho)]\right) = (1-\Phi[-\infty]) = 1.
\end{aligned}$$

B.2 The Likelihood-ratio Test Generally Applicable to Nested Models

The likelihood ratio is the ratio of the maximum likelihood of a restricted model to that of a saturated model (see Neyman and Pearson, 1928). Let $f(x, \theta)$ denote the probability density function of a random vector x parametrized with $\theta = [\theta^1, \cdots, \theta^k]$. The null hypothesis restricts θ to a set Ω specified with r $(r \leq k)$ constraints

$$\xi_1(\theta) = \xi_2(\theta) = \cdots = \xi_r(\theta) = 0.$$

Given i.i.d. observations $x_1, \cdots, x_m$, let $L(\theta)$ denote the likelihood function

$$L(\theta) = \prod_{i=1}^{m} f(x_i, \theta).$$

The likelihood ratio Λ given the observations is defined as

$$\Lambda = \frac{\sup L(\theta)}{\sup_{\theta\in\Omega} L(\theta)}.$$

Wald (1943) has proved that under certain assumptions on $f(x,\theta)$ and $\xi_1(\theta),\cdots,\xi_r(\theta)$, the limit distribution of the statistic $-2log\Lambda$ is the χ^2_r distribution with r degrees of freedom if the null hypothesis true. If the null hypothesis is not true, the distribution of $-2log\Lambda$ approaches the non-central $\chi^2_r(\lambda)$ distribution with r degrees of freedom and the non-central parameter

$$\begin{aligned}\lambda &= mD(\theta) \geq 0,\\ D(\theta) &= \sum_{i=1}^{k}\sum_{j=1}^{k} \frac{\xi_i(\theta)\xi_j(\theta)}{\sum_{p=1}^{k}\sum_{q=1}^{k} \frac{\frac{\partial\xi_i}{\partial\theta^p}\frac{\partial\xi_j}{\partial\theta^p}}{E\left[-\frac{\partial^2 f(x,\theta)}{\partial\theta^p\partial\theta^q}\right]}}.\end{aligned}$$

If $D(\theta)>0$, then $\lim_{m\to\infty}\lambda=\infty$. Let t ($t<\infty$) denote the threshold of rejecting the null hypothesis with type I error rate under α ($\alpha>0$). The asymptotic power of detecting a θ that is not in Ω and whose $D(\theta)$ is greater than 0 is $\lim_{\lambda\to\infty} P(\chi^2_r(\lambda)>t)$. The mean and the variance of the $\chi^2_r(\lambda)$ distribution is $u=r+\lambda$ and $\sigma^2=2(r+2\lambda)$, respectively. When λ is large enough,

$$P\left(\chi^2_r(\lambda)>t\right) \geq P\left(t<\chi^2_r(\lambda)<u+(u-t)\right) = 1-P\left(|\chi^2_r(\lambda)-u|\geq u-t\right).$$

According to Chebyshev's inequality,

$$P\left(|\chi^2_r(\lambda)-u|\geq u-t\right) \leq \frac{\sigma^2}{(u-t)^2} = \frac{2(r+2\lambda)}{(r+\lambda-t)^2}.$$

$\therefore$ When λ is large enough, $P(\chi^2_r(\lambda)>t)\geq 1-\frac{2(r+2\lambda)}{(r+\lambda-t)^2}$.
$\because \lim_{m\to\infty}\lambda=\infty$ and both r and t are fixed.
$\therefore \lim_{m\to\infty}\frac{2(r+2\lambda)}{(r+\lambda-t)^2}=0$.
$\therefore \lim_{m\to\infty} P(\chi^2_r(\lambda)>t)=1$.

Appendix C. Error Rates of Interest

Statistical decision processes usually involve choices between negative hypotheses and their alternatives, positive hypotheses. In the decision, there are basically two sources of errors: the type I errors, that is, falsely rejecting negative hypotheses when they are actually true; and the type II errors, that is, falsely accepting negative hypotheses when their alternatives, the positive hypotheses are actually true. In the context of learning graph structures, a negative hypothesis could be that an edge does not exist in the graph, while the positive hypothesis could be that the edge does exist. It is generally impossible to absolutely prevent the two types of errors simultaneously, because observations of a limited sample size may appear to support a positive hypothesis more than a negative hypothesis even when actually the negative hypothesis is true, or vice versa, due to the stochastic nature of random sampling. Moreover, the two types of errors generally contradict each other. Given a fixed sample size and a certain statistic extracted from the data, decreasing the type I errors will increase the type II errors, and vice versa. To guarantee the absolute prevention of the type I errors in any situations, one must accept all negative hypotheses, which will generally lead the type II error rate to be one, and vice versa. The contradiction between the two types of errors is clearly revealed by the monotone increase of receiver operating characteristic (ROC) curves. Thus

the errors must be controlled by setting a threshold on a certain type of errors, or trading off between them, for instance, by minimizing a certain lost function associated with the errors according to the Bayesian decision theory.

Rooted in the two types of errors, there are several different error-rate criteria (as listed in Table 2) for problems involving simultaneously testing multiple hypotheses, such as verifying the existence of edges in a graph. The type I error rate is the expected ratio of the type I errors to all the negative hypotheses that are actually true; the type II error rate is the expected ratio of the type II errors to all the positive hypotheses that are actually true; the false discovery rate (FDR) (see Benjamini and Yekutieli, 2001; Storey, 2002), is the expected ratio of falsely accepted positive hypotheses to all those accepted positive hypotheses; the family-wise error rate is the probability that at least one of the accepted positive hypotheses is actually wrong.

Generally, there are no mathematically or technically superior relationships among these error-rate criteria. Each of these error rates may be favoured in certain research scenarios. For example:

- We are diagnosing a dangerous disease whose treatment is so risky that may cause the loss of eyesight. Due to the great risk of the treatment, we hope that less than 0.1% of healthy people will be falsely diagnosed as patients of the disease. In this case, the type I error rate should be controlled under 0.1%.

- We are diagnosing cancer patients. Because failure in detecting the disease will miss the potential chance to save the patient's life, we hope that 95% of the cancer patients will be correctly detected. In this case, the type II error rate should be controlled under 5%.

- In a pilot study, we are selecting candidate genes for a genetic research on Parkinson's disease. Because of the limited funding, we can only study a limited number of genes in the afterward genetic research, so when selecting candidate genes in the pilot study, we hope that 95% of the selected candidate genes are truly associated with the disease. In this case, the FDR will be chosen as the error rate of interest and should be controlled under 5%.

- We are selecting electronic components to make a device. Any error in any component will cause the device to run out of order. To guarantee the device functions well with a probability higher than 99%, the family-wise error rate should be controlled under 1%.

In these examples, the particular error-rate criteria are selected by reasons beyond mathematical or technical superiority, but by the researchers' interest, to minimize a certain lost function associated with the errors according to the Bayesian decision theory. Learning network structures in real world applications may face scenarios similar to the above examples.

The excellent discrimination between negative hypotheses and positive hypotheses cannot be achieved by "smartly" setting a threshold on a "superior" error-rate criterion. Setting a threshold on a certain type of error rate is just choosing a cut-point on the ROC curve. If the ROC curve is not sharp enough, any cut-point on the curve away from the ends (0,0) and (1,1) still leads to considerable errors. To discriminate more accurately between a negative hypothesis and a positive hypothesis, one must design a better statistic or increase the sample size to achieve a sharper ROC curve.

References

A. Agresti. *Categorical Data Analysis (2nd edition)*. John Wiley & Sons, Inc., 2002.

T. W. Anderson. *An Introduction to Multivariate Statistical Analysis (2nd edition)*. John Wiley & Sons, Inc., 1984.

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.

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

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

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

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

D. Eaton and K. Murphy. Bayesian structure learning using dynamic programming and MCMC. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, 2007.

R. A. Fisher. Frequency distribution of the values of the correlation coefficients in samples from an indefinitely large population. *Biometrika*, 10(4):507–521, 1915.

R. A. Fisher. The distribution of the partial correlation coefficient. *Metron*, 3:329–332, 1924.

N. Friedman and D. Koller. Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1):95–125, 2003.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

E. H. Herskovits and G. F. Cooper. Kutato: An entropy-driven system for the construction of probabilistic expert systems from databases. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 54–62, 1990.

M. Kalisch and P. Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, 8:613–636, 2007.

T. A. Keller, M. A. Just, and V. A. Stenger. Reading span and the time-course of cortical activation in sentence-picture verification. In *Annual Convention of the Psychonomic Society*, 2001.

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

S. L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford University Press, Oxford, New York, 1996.

J. Li, Z. J. Wang, S. J. Palmer, and M. J. McKeown. Dynamic Bayesian network modelling of fMRI: A comparison of group analysis methods. *NeuroImage*, 41:398–407, 2008.

D. Madigan, J. York, and D. Allard. Bayesian graphical models for discrete data. *International Statistical Review*, 63(2):215–232, 1995.

T. M. Mitchell, R. Hutchinson, R. S. Niculescu, F. Pereira, X. Wang, M. Just, and S. Newman. Learning to decode cognitive states from brain images. *Machine Learning*, 57(1):145–175, 2004.

J. Neyman and E. S. Pearson. On the use and interpretation of certain test criteria for purposes of statistical inference: Part I. *Biometrika*, 20A:175–240, 1928.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

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

J. Pearl and T. S. Verma. A statistical semantics for causation. *Statistics and Computing*, 2(2): 91–95, 1992.

H. Qian and S. Huang. Comparison of false discovery rate methods in identifying genes with differential expression. *Genomics*, 86(4):495–503, 2005.

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

J. Schäfer and K. Strimmer. An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764, 2005.

P. Spirtes and C. Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9:62–72, 1991.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. The MIT Press, 2001.

B. Steinsky. Enumeration of labelled chain graphs and labelled essential directed acyclic graphs. *Discrete Mathematics*, 270(1-3):267–278, 2003.

J. D. Storey. A direct approach to false discovery rates. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 64(3):479–498, 2002.

J. D. Storey. The positive false discovery rate: A Bayesian interpretation and the q-value. *The Annals of Statistics*, 31(6):2013–2035, 2003.

A. Wald. Tests of statistical hypotheses concerning several parameters when the number of observations is large. *Transactions of the American Mathematical Society*, 54(3):426–482, 1943.