Information and Media Technologies 9 (1) : 121–131 (2014)
reprinted from: Transactions of the Japanese Society for Artificial Intelligence 29 (1) : 137–147 (2014)
© Japanese Society for Artificial Intelligence

原著論文 Original Paper

A Robust Causal Discovery Algorithm against Faithfulness Violation

Takashi Isozaki Sony Computer Science Laboratories
isozaki@csl.sony.co.jp

keywords: Bayesian networks, causal discovery, conditional independence tests, structural equation models

Summary

Methods of statistical causal discovery that use conditional independence (CI) tests are attractive due to their time efficiency and applications to latent variable systems. However, they often suffer from worse inference results induced by statistical errors in CI tests than other approaches. We considered part of these errors to be due to statistically weak violations of a usually used assumption, called the causal faithfulness condition. We propose a causal discovery algorithm that can reduce the numbers of unnecessarily performed CI tests in this study and so provide accurate and fast inference without loss of theoretical correctness. We also introduce *unreliable directions*, which can reduce orientation errors caused by the locality of CI tests in the algorithm. Further, simulations are provided to demonstrate the performance of the proposed algorithm for discrete probability systems and continuous linear structural equation models.

1. Introduction

Statistical causal discovery from observational data without time-ordering information started with the research by Rebane and Pearl in the domain of artificial intelligence [Rebane 87, Pearl 00]. This attempt is obviously important in many fields such as epidemiology, bioinformatics, and econometrics besides artificial intelligence and statistics because the crucial purpose of almost all empirical sciences is to obtain laws on cause-effect relationships and it is difficult to do experiments while controlling all variables in these sciences due to their complexity. As directed acyclic graphs (DAGs) have effectively represented statistical causal relationships between interested variables, we used DAGs for this purpose in this study.

Algorithms for causal discovery are generally categorized into three approaches except for the Linear, Non-Gaussian, Acyclic causal Models (LiNGAM) approach to continuous linear non-Gaussian cases [Shimizu 06]. The first is called a constraint-based approach (e.g., [Verma 90]), which finds conditional independence (CI) relationships and uses them as constraints in identifying orientations. The second is called score-and-search, which finds the best scores such as penalized likelihoods and Bayesian scores (e.g., [Cooper 92]). The third is their hybrid (e.g., [Tsamardinos 06]). The constraint-based (CB) approach is promising because the CB algorithms are generally relatively fast and can easily be applied to both discrete and continuous variable systems (e.g., [Xie 08]) and latent variable models [Spirtes 95].

The CB approach usually identifies adjacency edges between two variables if and only if they are not conditionally independent given any subsets. Despite some algorithms that have been proposed, CB algorithms have been regarded as providing no better inference results than other approaches. The main cause is attributed to the occurrence of many errors in CI tests that also continuously generate other erroneous selections of conditioning sets and orientations subsequently in the algorithms. The large size of conditioning sets is a factor that leads to such errors in CI tests because sample sizes are likely to be insufficient in the tests. Recursive Autonomy Identification (RAI) algorithm [Yehezkel 09] was intended to reduce errors by decomposing graphs into sub-structures.

However, other statistical problems remain in the CB approach. The causal faithfulness condition [Spirtes 00] represents a perfect correspondence between conditional independence and causal graphical structures, which is usually assumed. We focused on the fact that the assumption is often statistically violated. Ramsey et al. decomposed faithfulness into adjacency and orientation faithfulness [Ramsey 06]. They only assumed the former condition and improved the PC algorithm. Nevertheless, even the former assumption was also statistically violated al-

Information and Media Technologies 9(1): 121–131 (2014)
reprinted from: Transactions of the Japanese Society for Artificial Intelligence 29(1): 137–147 (2014)
© Japanese Society for Artificial Intelligence

though the validity of faithfulness was proven from the viewpoint of probabilistic theory by Spirtes et al. [Spirtes 00] and Meek [Meek 95b]. This is because small dependencies cannot often be detected with CI tests using finite sample sizes. The edges are usually removed in the CB approach only when the hypotheses of conditional independence are not rejected. As a result, errors from missing edges often arise.

We consider that unnecessary CI tests should be detected and undone to reduce such errors. We propose a novel CB algorithm on the basis of this concept, while retaining theoretical correctness. We additionally introduce a new concept, which we call unreliable directions to reduce orientation errors caused by the locality of CI tests. A comparative study was also done in discrete probability systems with often referred to, representative, and available algorithms including state-of-the-art ones using standard data sets. We also demonstrated the effectiveness of our algorithm even in continuous variable systems using linear Gaussian structural equation models, which are often used in the social sciences, in addition to Bayesian network models.

This paper is organized as follows: the next section provides some preliminaries for probabilistic causal models and causal discovery algorithms. The new algorithm is described in detail in Section 3. Our demonstrations of it in practice are presented in Section 4. We discuss related work in Section 5 and summarize our research in Section 6.

2. Causal Networks and Causal Discovery Algorithms

This section describes some graph theoretical concepts as preliminaries and their association with statistical concepts in causal networks and a basic framework for causal discovery algorithms based on them. Also see the appendices for more basic graph-theoretical notions and supplementary descriptions.

2·1 Some Notions on Directed Graphs

Let $\mathbb{G}$ be a graph that has vertices and edges. Let $\mathbb{V}$ be a set of random variables, each corresponding to a vertex. Let $\mathbb{E}$ be a set of edges that are directed or undirected. A pair of $\mathbb{V}$ and $\mathbb{E}$ is denoted by $\langle \mathbb{V}, \mathbb{E} \rangle$. When graph $\mathbb{G} = \langle \mathbb{V}, \mathbb{E} \rangle$ has no cycled path along directed edges, $\mathbb{G}$ is called a **directed acyclic graph (DAG)**. When graph $\mathbb{G}$ has partially undirected edges, $\mathbb{G}$ is called a **partially DAG (PDAG)**. If two edges direct into Z in a relationship between three vertices, X, Y, and Z such as $X \to Z \leftarrow Y$, Z is called a **collider**. Otherwise, Z is called a **non-collider**. In addition, if X and Y are not adjacent in structure $X \to Z \leftarrow Y$, Z is called an **unshielded collider** and the structure for the three is called a **V-structure**. If X and Y are adjacent, Z is called a shielded collider.

2·2 Conditional Independencies and D-separations

Let P be a positive definite probability distribution of variables $\mathbb{V}$. If random variables X and Y are conditionally independent given a set of random variables $\boldsymbol{Z}$ in P, they are represented as $X \perp\!\!\!\perp Y \mid \boldsymbol{Z}$. If not, they are represented as $X \not\perp\!\!\!\perp Y \mid \boldsymbol{Z}$. If $\boldsymbol{Z} = \emptyset$, they are represented as $X \perp\!\!\!\perp Y$ or, explicitly $X \perp\!\!\!\perp Y \mid \emptyset$. Let us assume that a directed edge from vertex X to its child Y represents a direct effect from the variable interchangeably denoted by vertex X to child-variable Y in a pair of DAGs $\mathbb{G}$ and distribution P. The causal Markov condition (CMC) states that each variable is probabilistically independent of non-descendants conditionally given its parent sets of variables [Spirtes 00]. We assume that DAGs in this research satisfy CMC, and such DAGs are called causal graphs or causal networks. Moreover, we assume the causal faithfulness condition (CFC). CFC states that conditional independence in a DAG is necessarily involved with CMC [Spirtes 00]. In other words, conditional independencies are determined by causal structures and not by the numerical details of parameters such as their annihilations. We additionally assumed that CMC and CFC were satisfied in this work for true probability distributions without latent variables and selection biases, which is called causal sufficiency [Spirtes 00].

Conditional independencies in a causal network are associated with a graphical concept called **d-separation** [Pearl 88]:

[Definition 1] In a DAG that consists of a set of vertices $\mathbb{V}$, in all paths between two vertices X and Y, if a set of vertices $\boldsymbol{Z} \in \mathbb{V} \backslash \{X, Y\}$ satisfies either following condition, $\boldsymbol{Z}$ d-separates X and Y.
(i) Non-colliders included in $\boldsymbol{Z}$ are on a path between X and Y, or (ii) a path between X and Y includes colliders and those and their descendants are not included in $\boldsymbol{Z}$. □

A relationship between d-separation and conditional independence is assured by the following theorem [Verma 88]. We will thus use vertices and variables interchangeably after this.

[Theorem 1] In a causal DAG that satisfies the causal Markov condition, if two vertices X and Y are d-separated from a set of vertices $\boldsymbol{Z}$, $X \perp\!\!\!\perp Y \mid \boldsymbol{Z}$ is satisfied in the corresponding random variables. □
See Appendix B for the relationships between (conditional)

independencies and causality in causal DAGs. We assume that causal models can be represented on Bayesian networks for discrete random variables and on linear structural equation models for continuous Gaussian variables. See Appendix C for more details on these models. Both representations are based on CMC.

2·3 Causal Discovery with Constraint-based Approach

The constraint-based (CB) approach for causal discovery generally consists of two parts: adjacency and orientation identification stages. The algorithms for the approach are based on the following two propositions, which are due to the theorem of d-separations and the assumption of CFC (e.g., see [Neapolitan 04], p. 89 for the proofs).

⟨Proposition 1⟩ Two vertices in a DAG are adjacent if and only if they are not d-separated by any set of vertices. □

⟨Proposition 2⟩ If three vertices X, Y, and an unshielded collider Z in a DAG have the form of a V-structure, X and Y are d-separated by some set not including Z and not d-separated by any set including Z. □

§ 1 Identification of Undirected Graphs

Pairs of adjacent vertices can be identified according to Proposition 1, and hence an undirected causal graph is provided. We call this stage the adjacency stage. An undirected complete graph is often selected as an input to the stage (e.g., the PC algorithm [Spirtes 91]). Therefore, CB algorithms search for d-separator sets and remove the edges if they find them. The PC algorithm can be outlined in four steps as a representative and executable algorithmic instance: (i) It constructs an undirected complete graph and an empty set for any pair of X and Y as denoted by $Sepset(X,Y)$. (ii) It searches for a d-separator set, $\boldsymbol{S}$, for a currently adjacent pair, X and Y, in ascending order of the size of the set starting from zero. Candidates of d-separator sets are selected from the adjacent nodes of X or Y. (iii) If the separator set $\boldsymbol{S}$ is found in conditional independence tests such that $X \perp\!\!\!\perp Y \mid \boldsymbol{S}$, the edge is removed according to Proposition 1 and $\boldsymbol{S}$ accumulates to $Sepset(X,Y)$. (iv) Finally, an undirected graph is provided where the edges remain between the two vertices that are not d-separated by any conditional sets. $Sepset(X,Y)$ is called a separator set and used in the following orientation stage.

Conditional independence (CI) tests are generally performed to search for separator sets. The null hypothesis corresponds to conditional independence, and the opposite hypothesis to conditional dependence. G^2 [Spirtes 00, Tsamardinos 06, Xie 08], χ^2 [Spirtes 00] or conditional mutual information [Cheng 02, Yehezkel 09] is used in the CI tests for discrete variable systems. Fisher's Z tests [Anderson 03, Spirtes 00] that use correlation coefficients are usually used in the linear Gaussian structural equation models (SEMs) for continuous variable systems.

§ 2 Identification of Partially Directed Graphs

We intend to do as many orientations as possible for the next step of causal discovery on the basis of the constraints of DAGs and use of information on separator sets $Sepset$. We call this the orientation stage. Two procedures are usually used in this stage: (v) The CB algorithms identify all the V-structures on the basis of Proposition 2 by using information on separator sets $Sepset$. For instance, if X and Y are such that $X \perp\!\!\!\perp Y \mid \emptyset$ in a defined variable set $\mathbb{V}$, and for any subsets $\boldsymbol{S}_1 \subset \mathbb{V}$ and $\boldsymbol{S}_2 \subset \mathbb{V}$, $X \not\perp\!\!\!\perp Z \mid \boldsymbol{S}_1$ and $Y \not\perp\!\!\!\perp Z \mid \boldsymbol{S}_2$ are valid, we obtain V-structure $X \rightarrow Z \leftarrow Y$ according to Proposition 2. Identifying all V-structures induces orientation rules based on the constraints [Verma 92, Meek 95a]. (vi) The algorithms then perform orientation rules (see Appendix D for details). A statistically maximal distinguishable PDAG is obtained as the final output of the algorithms.

3. Proposed Algorithm

3·1 Combining Stages and Minimal Blocker Conditions

If and only if the null hypothesis of conditional independence is not rejected in the standard CB approach, the related edge is removed from the graph. The null hypotheses are often not rejected for two variables due to the finiteness of sample size even though small dependencies truly exist between them. Errors of missing edges often arise as a result. These errors can partially be attributed to weak statistical violations of CFC [Ramsey 06], which are generated by the numerical cancellations of parameters. Consequently, accuracy in causal discovery can be expected to improve if unnecessary CI tests are detected and avoided. We propose a novel CB algorithm based on this assumption, in which we combine the two stages of adjacency identification and partial orientation with some conditions. We call it the Combining Stage (CS) algorithm. The following designations have been introduced for convenience. $|\boldsymbol{S}|$ denotes the number of variables of set $\boldsymbol{S}$. When $|\boldsymbol{S}| = m$ in CI tests with conditioning set $\boldsymbol{S}$, we call the test an m-th order CI test.

First, we define blockers and minimal blockers as follows.

[Definition 2] In a DAG that has a set of vertices $\mathbb{V}$, path u between two vertices $X, Y \in \mathbb{V}$ is **blocked** by set $\boldsymbol{Z} \subseteq \mathbb{V} \backslash \{X,Y\}$ or $\emptyset$ if (i) or (ii) of Definition 1 is satisfied. When path u is blocked by $\boldsymbol{Z}$, each element of $\boldsymbol{Z}$ is a

Information and Media Technologies 9(1): 121–131 (2014)
reprinted from: Transactions of the Japanese Society for Artificial Intelligence 29(1): 137–147 (2014)
© Japanese Society for Artificial Intelligence

blocker. A minimal size set of blockers in a set of blockers for a path is called a **minimal blocker**. □

[Example 1] In a DAG that consists of four vertices X, Y, Z, and S for path $X \rightarrow Z \leftarrow S \leftarrow Y$, two sets $\emptyset$ and $\{S\}$ are both blockers for X and Y and $\emptyset$ is a minimal blocker. □

Property 1 A collider on path u between X and Y in a DAG is not a blocker on u for X and Y. □

《Proof》 The proof immediately follows from Definition 2. □

We can provide more details for minimal blockers:

Property 2 If path u between X and Y in a DAG has colliders, any vertex Z on u is not the minimal blocker on u for X or Y. □

《Proof》 If Z is a collider on u, Z is not a blocker according to Property 1. Then, if Z is not a collider on u and is on u that has another collider, Z is not the minimal blocker on u because the minimal blocker is the empty set $\emptyset$ according to Definition 2. □

The CS algorithm assumes CFC and searches for the separator sets in ascending order of sizes of the sets just as PC does. This strategy enables CS to find the minimal blockers. We here define the **Minimal Blocker Condition** as follows.

[Definition 3] Conditioning set $\boldsymbol{S}$ in a DAG, such that $|\boldsymbol{S}| > 0$ for two vertices X and Y, satisfies the Minimal Blocker Condition (MBC) if there is a path between X and Y that contains a vertex in $\boldsymbol{S}$ and no colliders. □

The following theorem ensures, on the basis of MBC, the correctness of the CS algorithm.

[Theorem 2] Suppose that the process of searching for a DAG starts with a complete graph by using an algorithm that searches conditioning sets $\boldsymbol{S}$ for all pairs of vertices X and Y such that $(X \perp\!\!\!\perp Y \,|\, \boldsymbol{S})$ in ascending order of $|\boldsymbol{S}|$ from zero, and directed or undirected edges between X and Y are removed if such sets are found. Then, if $\boldsymbol{S}$ does not satisfy the Minimal Blocker Condition, $\boldsymbol{S}$ can be removed from candidate sets of the separator sets while keeping the correctness of the algorithm given CFC. □

《Proof》 When CI tests are performed in the process of searching for a DAG for two vertices $X, Y \in \mathbb{V}$ given sets $\boldsymbol{S} \in \mathbb{V} \backslash \{X, Y\}$, if all paths containing vertex $Z \in \boldsymbol{S}$ include V-structures and all or some of them truly exist in a graph, the common minimal blocker is empty set $\emptyset$ due to Property 2. Consequently, if $(X \perp\!\!\!\perp Y \,|\, \boldsymbol{S})$ is true, $(X \perp\!\!\!\perp Y \,|\, \boldsymbol{S} \backslash Z)$ is also true and should be found in lower order CI tests that were performed before that. Then, the CI tests given sets $\boldsymbol{S}$ for X and Y can be omitted. When no paths truly exist, Z cannot be a blocker due to Definition 2, so we can omit the CI tests with conditioning set $\boldsymbol{S}$. □

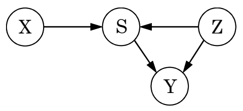


Fig. 1 Example of differences between separator sets and union of minimal blockers. Minimal blockers on two paths between X and Y are $\emptyset$ and $\{S\}$ in this DAG but separator set is $\{Z, S\}$.

We can propose an algorithm that combines adjacency and partial orientation stages with MBC, which is roughly outlined as follows. The algorithm starts from zero-th order CI tests for the complete graph. Then, V-structures are immediately identified in the results due to Proposition 2. Next, it performs first order CI tests and avoids part of them by using the information on the V-structures and Theorem 2. Some parts of the set of V-structures may be removed because the results of the CI tests may remove edges that consist of V-structures. As a result of the first order CI tests, newly identified V-structures are added to the set. These procedures are repeated until the search for separator sets finishes. This combining algorithm is expected to reduce errors by avoiding unnecessary CI tests.

It is noteworthy that separator sets are not the sum of minimal blockers, as explained in the following example.

[Example 2] Let a true DAG be the one in Figure 1, where the minimal blocker is $\emptyset$ for path $X \rightarrow S \leftarrow Z \rightarrow Y$ between X and Y. The other blocker is $\{Z\}$. In path $X \rightarrow S \rightarrow Y$, the minimal and sole blocker is $\{S\}$. However, the separator set between X and Y is $\{Z, S\}$ and not the union of the minimal blockers. The CS algorithm does not omit the CI tests for X and Y given conditioning sets $\{Z, S\}$ because the set satisfies MBC. □

The time complexity of constraint-based algorithms is typically measured by the number of CI tests. The complexity of the CS algorithm is, in the worst case, bounded by $n^2(n-1)^{k-1}/(k-1)!$ as the PC algorithm is [Spirtes 00] when no V-structures are found, where n denotes the number of vertices and k denotes the maximal degree of connectivity for any vertex. In addition, the CS algorithm tries to find V-structures with path-search algorithms, whose complexity is bounded by b^m, where b denotes a branching factor determined by edge cardinality during the search, and m denotes the depth of the path search between two vertices on CI tests, depending on path length and cardinality. The algorithm is thus not expected to be applied to large causal networks. However, we observed that several dozen variables were tractable sufficiently quickly with the CS algorithm (see Chapter 4 for practical simulations).

Information and Media Technologies 9(1): 121–131 (2014)
reprinted from: Transactions of the Japanese Society for Artificial Intelligence 29(1): 137–147 (2014)
© Japanese Society for Artificial Intelligence

3·2 *Example of Application of CS Algorithm*

We can outline the CS algorithm with MBC using the following example.

[Example 3] Let us consider a causal graph with six random variables X_1, X_2, X_3, X_4, X_5, and X_6. Let a true graph be that depicted in Figure 2 (a). The CS algorithm starts from a complete graph with the six vertices in (b) and searches for separator sets in ascending order. $(X_1 \perp\!\!\!\perp X_3)$, $(X_2 \perp\!\!\!\perp X_3)$, $(X_3 \perp\!\!\!\perp X_4)$, and $(X_3 \perp\!\!\!\perp X_5)$ are found in 0-th order CI tests, i.e., CI tests with no conditioning sets, and then corresponding vertex pairs such as $X_1 - X_3$ are removed from the complete graph. The results are shown in (c). The CS characteristically performs partial orientations in the next step, i.e., it forms V-structures. The following V-structures are obtained from independence relations and Proposition 2: $X_1 \to X_6 \leftarrow X_3$, $X_2 \to X_6 \leftarrow X_3$, $X_3 \to X_6 \leftarrow X_4$, and $X_3 \to X_6 \leftarrow X_5$, which are depicted in (d). The CS then goes to first order CI tests. The algorithm selects conditioning sets, for two objective variables X and Y, from variables that are currently adjacent to X or Y and are on the path between X and Y. Then, the following CI tests are omitted because they do not satisfy MBC: whether $(X_1 \perp\!\!\!\perp X_2 | X_6)$, $(X_1 \perp\!\!\!\perp X_4 | X_6)$, $(X_2 \perp\!\!\!\perp X_4 | X_6)$, $(X_2 \perp\!\!\!\perp X_5 | X_6)$, and $(X_4 \perp\!\!\!\perp X_5 | X_6)$. As a result of the first order tests, $(X_1 \perp\!\!\!\perp X_4 | X_2)$ and $(X_1 \perp\!\!\!\perp X_5 | X_2)$ are found, so the graph is that in (e). No new V-structures after the tests are obtained in the example. In the second order CI tests, some CI tests are also omitted by MBC such as those to determine whether $(X_1 \perp\!\!\!\perp X_2 | \{X_4, X_6\})$, $(X_1 \perp\!\!\!\perp X_2 | \{X_5, X_6\})$, and $(X_2 \perp\!\!\!\perp X_5 | \{X_1, X_4\})$. $(X_2 \perp\!\!\!\perp X_6 | \{X_1, X_4\})$, and $(X_5 \perp\!\!\!\perp X_6 | \{X_2, X_4\})$ are found in the tests and then the graph in (f) is obtained after the tests. No edges are newly directed by the orientation rules in the example, so the final graph is that depicted in (f). If we do not apply MBC, the probabilities of errors increase. For instance, if we erroneously perform a CI test to determine whether $X_1 \perp\!\!\!\perp X_2 | X_6$ or not, edge $X_1 - X_2$ can be incorrectly removed. Additionally, if this occurs, the directions of the two edges in structure $X_1 - X_6 - X_2$ cannot easily be identified. Moreover, if direction errors arise, more direction errors can be induced via the orientation rules. The CS algorithm reduces the probability of errors and performs robust causal discovery using MBC. □

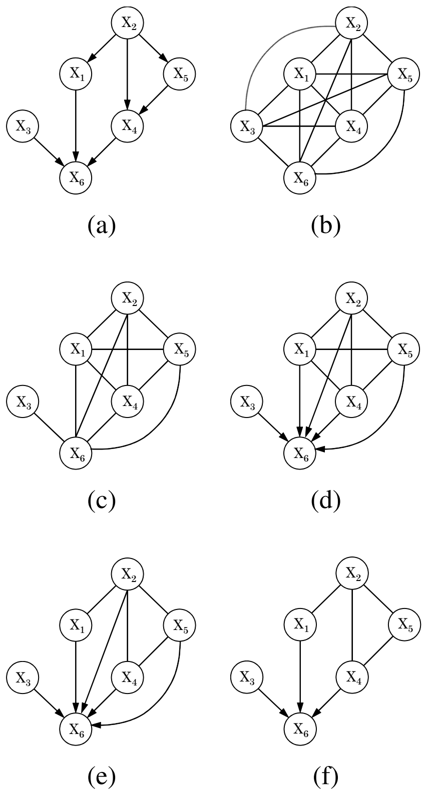


Fig. 2 Example applied with CS algorithm: (a): true graph, (b): initial complete graph, (c): undirected graph after 0th-order CI tests, (d): PDAG after 0th-order CI tests, (e): PDAG after 1st-order CI tests, and (f): PDAG after 2nd-order CI tests and final graph is obtained.

3·3 *k-Minimal Blocker Condition*

The minimal blocker condition (MBC) can have a parameter, and MBC with the determined parameter is called the **k-Minimal Blocker Condition (k-MBC)**. The algorithm, in k-th order CI tests, uses V-structures recognized by up to $(k-1)$-th order CI tests. The V-structures recognized by higher order statistics in the practical use of the algorithm are less reliable, and MBCs with such V-structures are also less reliable. We can then determine a maximal order k of V-structures used in MBCs in advance. If we set such value k, we put off constructing new V-structures recognized by higher order CI tests than k until the end of the adjacency identification stage. This parameter is expected to make CS more robust for practical use. Ramsey et al. decomposed CFC to adjacency faithfulness and orientation faithfulness and only assumed the former [Ramsey 06] (also see Chapter 5). It can be said that k-MBC assumes lower order orientation faithfulness and may render CS robust to some extent against weak violations of higher order adjacency faithfulness. We do not discuss how maximal order k is determined in this paper.

3·4 *Dependency Measures and Bi-directional Resolutions*

The following two subsections describe procedures to make the algorithm more robust against statistical errors in CI tests, which we consider necessary in almost all CB approaches. A p-value can be obtained from the CI tests performed on the CS algorithm, and the null hypothesis

Information and Media Technologies 9(1): 121–131 (2014)
reprinted from: Transactions of the Japanese Society for Artificial Intelligence 29(1): 137–147 (2014)

is rejected if the p-value is less than a significance level. We regard the variable sets in the CI tests as conditionally independent if they cannot be rejected in the same way previous researchers have done. The p-value is used as follows in the CS algorithm as a metric of the degree of dependence between nodes. The smaller the p-value, the higher the (conditional) dependence between related nodes. This use of p-values resolves the problem as follows.

As inconsistent structures often emerge in the CB approach due to the locality of CI tests and statistical errors, we need to take countermeasures against them. For example, if $X \perp\!\!\!\perp Y \mid \boldsymbol{S}_1$, $X \not\perp\!\!\!\perp Z$, and $Y \not\perp\!\!\!\perp Z$ given any sets, and $Z \notin \boldsymbol{S}_1$ are obtained, the relationships generate $X \rightarrow Z \leftarrow Y$, and if $Z \perp\!\!\!\perp W \mid \boldsymbol{S}_2$, $Y \not\perp\!\!\!\perp Z$, and $Y \not\perp\!\!\!\perp W$ given any sets, and $Y \notin \boldsymbol{S}_2$ are obtained, they generate $Z \rightarrow Y \leftarrow W$. That means $Y \leftrightarrow Z$, i.e., it is bi-directionally inconsistent. We resolve the inconsistency in the CS algorithm using p-values: we select the set with the highest p-values from the inconsistency sets of V-structures, remove the others from V-structure sets, and disorient edges in the loser groups. This procedure is outlined in Figure 3 and described in Algorithm 1, where we have denoted *Vstr* as the V-structure sets.

3·5 Unreliable Directions

The CB approach also has another weakness that is due to the locality of CI tests and statistical errors, which is the same as the case in the previous subsection. That is, this approach may produce another contradictory CI test result. As this may lead to other incorrect orientations caused by the orientation rules, induced errors should be avoided. To discuss this further, we need to define *unreliable directions* as follows. If two V-structures ($X \rightarrow Z \leftarrow W$ and $Y \rightarrow Z \leftarrow W$) are obtained in a DAG $\mathbb{G}$, and a V-structure ($X \rightarrow Z \leftarrow Y$), which is induced from the former patterns, cannot be allowed, both statistically from CI tests, the two directions ($X \rightarrow Z$ and $Y \rightarrow Z$) are called **unreliable directions**. The CS algorithm resolves the inconsistency by disorienting these unreliable edges if found. The disorientation also reduces induced errors of direction in the successive full orientation stage. There is an example of these in Figure 4. Let us assume that the following four CI relationships hold: ($X \perp\!\!\!\perp W \mid \emptyset$), ($Y \perp\!\!\!\perp W \mid \emptyset$), ($X \not\perp\!\!\!\perp Y \mid \emptyset$), and ($X \perp\!\!\!\perp Y \mid Z$). $X \rightarrow Z \leftarrow W$ and $Y \rightarrow Z \leftarrow W$ are then obtained from the zero-th order conditional independence. In addition, $X \rightarrow Z \leftarrow Y$ is induced as seen in (a) but it contradicts $X \perp\!\!\!\perp Y \mid Z$. This contradictory result can happen due to the locality of CI tests. We conservatively regard directions $X \rightarrow Z$ and $Y \rightarrow Z$ as being unreliable and so disorient both. A PDAG in (b) is obtained as a result.

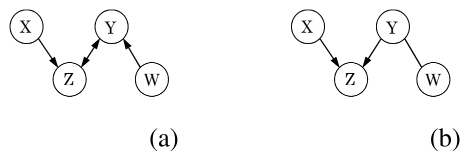


Fig. 3 Example of inconsistent errors in V-structures: When $X \perp\!\!\!\perp Y \mid \boldsymbol{S}_1$ and $Z \notin \boldsymbol{S}_1$ are valid with p-value 0.90, and $Z \perp\!\!\!\perp W \mid \boldsymbol{S}_2$ and $Y \notin \boldsymbol{S}_2$ are valid with p-value 0.30, bi-directional edges are generated as seen in (a). CS algorithm selects V-structure with highest p-value and a PDAG is obtained as in (b).

Algorithm 1 Resolve Inconsistency

```
1: Input Vstr
2: Select inconsistency V-structure sets in Vstr
3: Calculate p-values on independence of all sets
4: Select highest p-values set
5: Remove other sets from Vstr and disorient their edges
6: return Vstr
```

The pseudocode for the whole version of the CS algorithm is listed in Algorithm 2, where ADJ_X denotes the set of adjacent nodes of node X. We can use k-MBCs to reduce additive errors caused by statistical power shortages in practical use.

4. Numerical Evaluation

We carried out numerical simulations to evaluate the performance of the CS algorithm in practice, which was implemented in C++ language. We also compared the CS algorithm with other often referred to algorithms in terms of the accuracy of DAG-structure recovery in both discrete and continuous random variable systems, using known DAGs that have usually been used in recent previous research.

4·1 Experiments in Discrete Variables

There are three main approaches to inferring a DAG structure for discrete probability systems, which is represented by Bayesian networks (BNs: see Appendix C): CB, score-and-search (SS), and their hybrids. We used the following algorithms, which are publicly available and have often been used in recent research on each approach (CB/SS/hybrids): PC [Spirtes 91] (CB), Three Phase Dependency Analysis (TPDA) [Cheng 02] (CB), Sparse Candidate (SC) [Friedman 99] (SS), and Max-Min Hill-Climbing (MMHC) [Tsamardinos 06] (hybrids). SC and MMHC recently demonstrated high levels of performances in structure recovery tasks over a broad range of training sample sizes [Tsamardinos 06, Xie 08]. MMHC has been re-

Information and Media Technologies 9 (1) : 121–131 (2014)
reprinted from: Transactions of the Japanese Society for Artificial Intelligence 29 (1) : 137–147 (2014)

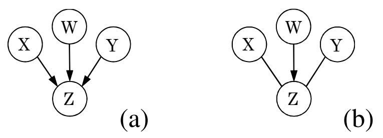


Fig. 4 Example of unreliable directions: if CI tests derive $(X \perp\!\!\!\perp W \mid \emptyset)$, $(Y \perp\!\!\!\perp W \mid \emptyset)$, $(X \not\perp\!\!\!\perp Y \mid \emptyset)$, and $(X \perp\!\!\!\perp Y \mid Z)$, DAG in (a) is obtained but has inconsistent error. We regard $X \to Z$ and $Y \to Z$ as unreliable directions that disorient both edges in (b).

Algorithm 2 Combining Stage Algorithm

```
 1: form a complete undirected graph $\mathbb{G}$ and empty sets for any pair of
    $X, Y \in \mathbb{V}$ as $Sepset(X,Y)$.
 2: set $n = 0$
 3: repeat
 4:   for each $X \in \mathbb{V}$ do
 5:     for each $Y \in \mathrm{ADJ}_X$ do
 6:       for each subset $\boldsymbol{S} \subseteq \{\mathrm{ADJ}_X \setminus Y\}$ such that $|\boldsymbol{S}| = n$ do
 7:         set $m = 0$
 8:         for each $Z \in \boldsymbol{S}$ do
 9:           if a path between $X$ and $Y$ containing $Z$ does not
              contain a V-structure then
10:             set $m = 1$
11:             break
12:           end if
13:         end for
14:         if $m = 0$ then
15:           continue
16:         else if $X \perp\!\!\!\perp Y \mid \boldsymbol{S}$ then
17:           remove the edge $X - Y$ from $\mathbb{G}$
18:           add $\boldsymbol{S}$ to $Sepset(X,Y)$
19:         end if
20:       end for
21:     end for
22:   end for
23:   for each unshielded triple $\langle X, Z, Y \rangle$ in $\mathbb{G}$ do
24:     orient it as $X \to Z \leftarrow Y$ iff $Z$ is not in $Sepset(X,Y)$
25:     add them to $Vstr(X,Z,Y)$
26:   end for
27:   set $n = n + 1$
28: until $|\mathrm{ADJ}_X| \leq n$ for all $X \in \mathbb{V}$
29: Resolve Inconsistency (Vstr)
30: find all unreliable directions and disorient those edges
31: orient edges using the orientation rules
32: return a partially DAG $\mathbb{G}$
```

garded as an especially state-of-the-art algorithm in recent research [Xie 08, Yehezkel 09]. We were aided by the Causal Explorer algorithm library [Aliferis 03] working on the MatLab 7.8 platform to run all algorithms other than CS. The significance level of CI tests using the G^2 likelihood in PC and MMHC was 0.05. The threshold of CI tests using mutual information used in TPDA was 0.01. The score metric used in SC and MMHC was the Bayesian Dirichlet equivalence uniform (BDeu) [Heckerman 95] and its equivalent sample size was 10. The maximum sizes of candidates for SC, parent-child sets for MMHC, and conditioning sets for PC and CS corresponded to 10, 10, and 5. All these settings for the metric and parameters were suggested by their authors.

We also used G^2 tests on CS to identify conditional independence with a significance level of 0.01, which was determined from preliminary trials. The results from its approximation accuracy were wholly fixed on the accuracy of causal discovery in CS. We used CS with 0-MBCs specifically with MBCs using V-structures derived from only marginal independence relationships (i.e., $(X \perp\!\!\!\perp Y \mid \emptyset)$), due to the reason mentioned in Section 3·3. Additionally, the CS algorithm selected nodes in ascending order of numbers of currently adjacent nodes in line 4 of Algorithm 2 and used a heuristic of PC [Spirtes 00] for speeding up.*1

We used the following five Bayesian networks (BNs), which neither had very many variables nor very many parameters because we wanted to evaluate algorithms by excluding influences caused by the difference in statistics that were used: Alarm with 37 nodes, 46 edges, and 509 parameters in conditional probability tables (nps, for short), Carpo with 60 nodes, 74 edges, and 342 nps, Hailfinder with 56 nodes, 66 edges, and 2656 nps, Insurance with 27 nodes, 52 edges, and 1008 nps, and Water with 32 nodes, 66 edges, and 10083 nps. Extra information can be obtained through the Bayesian Network Repository,*2 where the structures and probability distributions of these networks are also available. We sampled training cases from the distributions of the known networks. From each BN, 10 datasets were randomly sampled in each size of 1000, 2000, 5000, and 10000. The statistical quantities used here are the average over 10 runs of an algorithm on each sample size from the identical distribution of DAGs.

The PC, TPDA, and CS algorithms generate the maximal distinguishable PDAG as mentioned in Section 2·3·2, which is also called a complete PDAG [Chickering 95], of the recovered DAGs, while the SC and MMHC algorithms provide a DAG. Then, we transform the DAG provided by the SC and MMHC into a complete PDAG using Chickering's algorithm [Chickering 95], as Tsamardinos et al. and Yehezkel and Lerner did [Tsamardinos 06, Yehezkel 09]. Tsamardinos et al. suggested evaluating the quality of recovered PDAGs using the structural Hamming distance (SHD) metric, while they pointed out that Kullback-Leibler (KL) divergence and the BDeu score were not always suitable for the purpose [Tsamardinos 06]. Following Tsamardinos et al. and Yehezkel and Lerner, we used five types of inference errors to evaluate recovered networks: extra edges (EEs), missing edges (MEs), extra direction (ED), missing direction (MD), and reversed direction (RD) errors [Tsamardinos 06, Yehezkel 09]. EE (ME)

*1 For given variable X, test with variables Y in ascending order of dependence on X is conditional on subsets of variables that are in descending order of dependence on X.
*2 http://www.cs.huji.ac.il/site/labs/compbio/Repository/

errors represented extra (missing) edges that appeared in the recovered (true) graph but not in the true (recovered) graph. ED (MD) errors were due to edge directions that appeared in the recovered (true) graph but not in the true (recovered) graph. We also used the total directional errors (DE), DE = ED + MD + RD. SHD summed all five structural errors.

The simulation results are listed in Table 1. The numerical values are 10 run-averaged ones relative to that of MMHC following the approach by Tsamardinos et al. and Yehezkel and Lerner [Tsamardinos 06, Yehezkel 09], and the bold numbers denote the best scores in each network and sample size as the results of two-sided t-tests with a significance level of 0.05 (then, multiple winners can exist unless significant differences are obtained). Values significantly smaller than 1.0 mean superiority over MMHC. As has been found in recent research, the hybrid (MMHC) method outperformed the PC and TPDA algorithms, which are CB methods. The CS algorithm, which is also a CB approach, is obviously superior to SC, PC, and TPDA for all networks and all sample sizes. In addition, the best 19 results in the 20 patterns of the experiment were provided by CS, while the best 13 results provided by MMHC meant that the CS algorithm was rarely outdone by MMHC and was often superior to MMHC in the experiment. As far as we know, few constraint-based algorithms have exhibited such effectiveness compared with MMHC by using *averaged* SHD for a *broad range* of sample sizes.[*3]

Table 2 summarizes the number of CI tests performed by the CS and MMHC algorithms for a sample size of 10000 because the numbers monotonically increased for both algorithms and all networks. CS performed relatively fewer CI tests *on average* and had *more stable numbers* than MMHC did. Table 3 lists the normalized averages of running time on both algorithms for the same sample size, where averaged times were normalized by the shortest times in each algorithm to absorb the differences in programming languages in the algorithms. These timing results also demonstrate CS ran *stably*. The table also lists the raw data on running time in seconds as a reference without data input time in CS. The data were obtained under the following running conditions: OS: 64 bit Windows 7, CPU: Intel Core i7 870, 2.93 GHz, and RAM: 12.0 GB. The results indicated CS ran sufficiently fast.

*3 Xie and Geng achieved better results for SHD in discrete variable systems [Xie 08], and Yehezkel and Lerner carried out a DAG recovery experiment with only 500 sample sizes [Yehezkel 09].

4·2 *Experiments for Linear Gaussian SEMs*

We also undertook the following simulations for the Gaussian case using a linear structural equation model (see Appendix C). As the CB approach could only be applied to models with continuous variables, we used the PC and TPDA implemented in the Causal Explorer algorithm library package. Fisher's Z tests were used for CI tests (see Section 2·3·1). The significance levels were 0.05 for PC and 0.01 for TPDA and CS, which are the default settings for PC and TPDA. Samples were generated from the Alarm network, whose coefficients were randomly generated from the uniform distribution from (-1.5,-0.5) ∪ (0.5,1.5). We applied the CS algorithm to construct a DAG for each of the three models with different coefficient sets, and then we compared them using SHD with the DAG obtained from the PC and TPDA algorithms. The averaged SHDs for each such model were calculated for 10 runs in each data set of 1000, 2000, 5000, and 10000 cases. The averaged SHDs were obtained in the same way as the discrete cases, which were normalized by those of PC. The results are listed in Table 4, where the three different coefficient SEMs are denoted as Alarms 1, 2, and 3. CS obviously outperformed PC and TPDA because the SHDs of CS ranged from 0.3 to 0.5 against those of PC for each coefficient model and sample size, while the SHDs of TPDA were similar to those of PC.

4·3 *Large Network-Tractability of CS*

The CS algorithm does not seem suitable for causal discovery in systems with large numbers of variables due to its use of path-search algorithms as mentioned in Section 3·1. We investigated tractability in variable sizes with Alarm5, Munin, Hailfinder5, and Pigs datasets, which are all available from Tsamardinos et al.'s Web site: http://www.dsl-lab.org/supplements/mmhc_paper/mmhc_index.html. The networks respectively had 185, 189, 280 and 441 variables and 265, 282, 458 and 592 edges. The processing time for Alarm5 ranged, in five runs, from 8.9 sec. to 10.1 sec. (ave.: 9.6 sec.) and for Hailfinder5 from 114 sec. to 189 sec. (ave.: 151 sec.) with the depth-first path search algorithm for detecting V-structures for the 5000 samples in each network with the same computational conditions as in Section 4·1. However, the time for Munin ranged from 1 h. 59 min. to 2 h. 43 min. (ave.: 2 h. 17 min.) despite smaller numbers of variables and edges than those for Hailfinder5. The practical time complexity of CS thus did not simply depend on the numbers of variables and edges for large networks even though sparsity in causal structures was needed. The processes in the Pigs network remained unfinished within 24 h. for two of five

Table 1 Averaged normalized results for structural Hamming distance, which have been normalized by dividing by corresponding value for MMHC algorithm for particular network and sample size (N). Normalized value smaller than one means fewer structural errors in algorithm than MMHC. Bold numbers denote best scores as results of two-sided t-tests with significance level of 0.05.

Network	Algorithm	N=1000	N=2000	N=5000	N=10000
Alarm	CS	**1.2**	**1.2**	**0.7**	**0.5**
	PC	2.5	2.9	3.5	2.5
	TPDA	3.0	3.1	3.8	2.7
	SC	2.5	2.3	2.8	2.3
	MMHC	**1.0**	**1.0**	1.0	1.0
Carpo	CS	**0.9**	**0.9**	**0.8**	**0.8**
	PC	1.2	1.2	1.4	1.6
	TPDA	1.3	1.3	1.5	1.8
	SC	1.6	1.8	2.2	3.0
	MMHC	1.0	1.0	1.0	1.0
Hailfinder	CS	**1.2**	**0.9**	**0.9**	**1.0**
	PC	3.6	3.3	3.7	3.9
	TPDA	4.2	3.3	3.3	3.0
	SC	1.5	1.4	1.8	1.7
	MMHC	**1.0**	**1.0**	**1.0**	**1.0**
Insurance	CS	**1.0**	**1.1**	**0.9**	**1.0**
	PC	1.6	1.5	1.6	1.8
	TPDA	2.1	1.7	1.7	1.9
	SC	1.3	1.5	1.6	1.8
	MMHC	**1.0**	**1.0**	**1.0**	**1.0**
Water	CS	**1.0**	**1.0**	1.2	**0.9**
	PC	3.5	3.7	4.3	4.2
	TPDA	1.2	1.2	1.3	1.3
	SC	1.3	1.5	1.8	1.8
	MMHC	**1.0**	**1.0**	**1.0**	1.0

Table 2 Averaged number of conditional independence tests for 10000 sample sizes in CS and MMHC algorithms.

Algorithm	Alarm	Carpo	Hailfinder	Insurance	Water
CS	3.8K	5.5K	6.2K	4.9K	1.3K
MMHC	2.9K	19.5K	61.1K	5.9K	1.3K

datasets, and the time ranged from 7 h. to 20 h. 24 min. for the others. CS hence seemed to be able to handle 100 to 200 variables without having to make improvements to the V-structure-finding algorithm.

5. Related Work

Ramsey et al. decomposed the causal faithfulness condition into two conditions, which were called adjacency faithfulness and orientation faithfulness [Ramsey 06]. Adjacency faithfulness means that if two variables are adjacent, there are then not any separator sets for them. Orientation faithfulness means that two variables are conditionally independent given correct separator sets. Ramsey et al. assumed adjacency faithfulness and checked for violations of orientation faithfulness to robustly orient edges. However, the CS algorithm is proposed here to avoid errors in CI tests due to both kinds of violations of faithfulness.

Spirtes et al. discussed *undirected paths* required to contain separator sets for CI tests [Spirtes 00]. Our proposed constraint (i.e., MBCs) is thereby an extension of the condition on undirected paths to that on partially directed paths.

Two CB algorithms were recently proposed, which are both based on decomposing a graph to sub-graphs. The first is the Recursively Autonomy Identification (RAI) algorithm advocated by Yehezkel and Lerner [Yehezkel 09] who intended to avoid CI tests with large conditioning sets by decomposing a graph to sub-graphs. The method of decomposition hence seemed especially effective for large networks. The second recursive method was proposed by Xie and Geng [Xie 08]. Their algorithm depended rather on the assumption of correctness of CI tests because an edge between X and Y is removed if X and Y are conditionally independent given the set of *all other variables*. Therefore, their method seemed effective for parametric continuous variable systems because the number of parameters was generally much smaller than that of discrete variables. It may be effective to combine such decomposing algorithms with CS to obtain more accurate causal discovery algorithms.

Table 3 Averaged normalized running time for 10000 sample sizes in CS and MMHC algorithms. These results indicate CS ran stably. Raw data in CS have also been provided for reference in terms of seconds (denoted as *s*), where second values denote standard deviations.

Algorithm	Alarm	Carpo	Hailfinder	Insurance	Water
CS	3.0	5.0	8.0	3.6	1.0
	(2.0±0.1s)	(3.3±0.3s)	(5.3±0.8s)	(2.4±0.1s)	(0.7±0.1s)
MMHC	1.8	11.7	123.1	4.2	1.0

Table 4 Averaged normalized results of structural Hamming distance, which are normalized by dividing by corresponding value of PC algorithm for particular model for Alarms 1, 2, and 3 and sample size (N). Normalized value smaller than one means fewer structural errors in algorithm than PC. Bold numbers denote best scores as results of two-sided t-tests with significance level of 0.05.

Network	Algorithm	N=1000	N=2000	N=5000	N=10000
	CS	**0.5**	**0.4**	**0.3**	**0.3**
Alarm 1	PC	1.0	1.0	1.0	1.0
	TPDA	1.0	1.0	1.0	1.0
	CS	**0.5**	**0.4**	**0.4**	**0.4**
Alarm 2	PC	1.0	1.0	1.0	1.0
	TPDA	1.0	0.9	0.9	0.9
	CS	**0.5**	**0.4**	**0.4**	**0.4**
Alarm 3	PC	1.0	1.0	1.0	1.0
	TPDA	1.0	1.0	1.0	1.0

6. Conclusion

The constraint-based (CB) approach for causal models is dependent on the causal faithfulness condition (CFC), which states perfect correspondence of causal graphical representations with the causal Markov condition to conditional independence (CI) of probability distributions. However, the CB approach often suffers from many errors in CI tests. We regard that this is rather due to statistically weak violations of CFC. Thus, we focused on the problem and proposed an algorithm, which we call the Combining Stage (CS) algorithm. We defined the Minimal Blocker Conditions (MBCs) and the CS algorithm to avoid unnecessary CI tests that could be the source of errors by using MBCs. The MBCs also ensured the accuracy of CS under CFC. We also introduced *unreliable directions* to reduce direction errors due to the locality of CI tests.

In a comparative study using five repository datasets with 1000–10000 samples for discrete variables, the algorithm worked better in practice than the current prominent algorithms (PC, TPDA, and SC) and MMHC, which is a state-of-the-art algorithm, from the viewpoint of the numbers of best DAG-recovery results. In addition, as the number of CI tests was stably reduced and was smaller than that with MMHC on average, CS ran fast stably for datasets with several dozen variables. CS produced obviously better recovery results for linear Gaussian structural equation models (SEMs) with continuous variables than the PC and TPDA algorithms that were applied. These results demonstrate that avoiding CI tests and unreliable directions are useful for the CB approach.

Acknowledgments

The author would like to thank Manabu Kuroki for the helpful discussions and his suggestions, Ryosuke Ohata for his assistance with the experiments, and Mimpei Morishita for his comments on an earlier version of this paper. The author is also grateful to the referees for their valuable comments to improve this manuscript and Mario Tokoro and Hiroaki Kitano of Sony Computer Science Laboratories, Inc. for their support.

◇ References ◇

[Aliferis 03] Aliferis, C. F., Tsamardinos, I., Statnikov, A., and Brown, L. E.: Causal Explorer: A Causal Probabilistic Network Learning Toolkit for Biomedical Discovery, in *Proc. of International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences*, pp. 371–376 (2003)

[Anderson 03] Anderson, T. W.: *An Introduction to Multivariate Statistical Analysis*, John Wiley & Sons, Inc., Hoboken, NJ, second edition (2003)

[Cheng 02] Cheng, J., Greiner, R., Kelly, J., Bell, D., and Liu, W.: Learning Bayesian Networks from Data: An Information-Theory Based Approach, *Artificial Intelligence*, Vol. 137, No. 1-2, pp. 43–90 (2002)

[Chickering 95] Chickering, D. M.: A Transformational Characterization of Equivalent Bayesian Network Structures, in *Proc. of Conference on Uncertainty in Artificial Intelligence*, pp. 87–98 (1995)

[Cooper 92] Cooper, G. and Herskovits, E.: A Bayesian Method for the Induction of Probabilistic Networks from Data, *Machine Learning*, Vol. 9, No. 4, pp. 309–347 (1992)

[Friedman 99] Friedman, N., Nachman, I., and Peér, D.: Learning Bayesian Network Structure from Massive Datasets: The "Sparse Candidate" Algorithm, in *Proc. of Conference on Uncertainty in Artificial Intelligence*, pp. 206–215 (1999)

[Heckerman 95] Heckerman, D., Geiger, D., and Chickering, D.: Learning Bayesian Networks: The Combination of Knowledge and Statistical Data, *Machine Learning*, Vol. 20, No. 3, pp. 197–243 (1995)

[Meek 95a] Meek, C.: Causal Inference and Causal Explanation with Background Knowledge, in *Proc. of Conference on Uncertainty in*

Artificial Intelligence, pp. 403–410 (1995)

[Meek 95b] Meek, C.: Strong Completeness and Faithfulness in Bayesian networks, in *Proc. of Conference on Uncertainty in Artificial Intelligence*, pp. 411–418 (1995)

[Neapolitan 04] Neapolitan, R. E.: *Learning Bayesian Networks*, Prentice Hall, Upper Saddle River, NJ (2004)

[Pearl 88] Pearl, J.: *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Mateo, CA (1988)

[Pearl 00] Pearl, J.: *Causality, models, reasoning, and inference*, Cambridge University Press, New York, NY (2000)

[Ramsey 06] Ramsey, J., Spirtes, P., and Zhang, J.: Adjacency-Faithfulness and Conservative Causal Inference, in *Proc. of Conference on Uncertainty in Artificial Intelligence*, pp. 401–408 (2006)

[Rebane 87] Rebane, G. and Pearl, J.: The Recovery of Causal Poly-Trees from Statistical Data, in *Proc. of Workshop on Uncertainty in Artificial Intelligence*, pp. 222–228 (1987)

[Reichenbach 56] Reichenbach, H.: *The Direction of Time*, Dover Publications, Mineola, NY (1956), Republication of the work published by University of California Press, Berkely.

[Shimizu 06] Shimizu, S., Hoyer, P. O., Hyvärinen, A., and Kerminen, A.: A Linear Non-Gaussian Acyclic Model for Causal Discovery, *Journal of Machine Learning Research*, Vol. 7, pp. 2003–2030 (2006)

[Spirtes 91] Spirtes, P. and Glymour, C.: An Algorithm for Fast Recovery of Sparse Causal Graphs, *Social Science Computer Review*, Vol. 9, No. 1, pp. 62–72 (1991)

[Spirtes 95] Spirtes, P., Meek, C., and Richardson, T.: Causal Inference in the Presence of Latent Variables and Selection Bias, in *Proc. of Conference on Uncertainty in Artificial Intelligence*, pp. 499–506 (1995)

[Spirtes 00] Spirtes, P., Glymour, C., and Scheines, R.: *Causation, Prediction and Search*, MIT Press, Cambridge, MA, second edition (2000)

[Tsamardinos 06] Tsamardinos, I., Brown, L. E., and Aliferis, C. F.: The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm, *Machine Learning*, Vol. 65, No. 1, pp. 31–78 (2006)

[Verma 88] Verma, T. and Pearl, J.: Causal Networks: Semantics and Expressiveness, in *Proc. of Workshop on Uncertainty in Artificial Intelligence*, pp. 352–359 (1988)

[Verma 90] Verma, T. and Pearl, J.: Equivalence and Synthesis of Causal Models, in *Proc. of Conference on Uncertainty in Artificial Intelligence*, pp. 220–227 (1990)

[Verma 92] Verma, T. and Pearl, J.: An Algorithm for Deciding If a Set of Observed Independencies Has a Causal Explanation, in *Proc. of Conference on Uncertainty in Artificial Intelligence*, pp. 323–330 (1992)

[Xie 08] Xie, X. and Geng, Z.: A Recursive Method for Structural Learning of Directed Acyclic Graphs, *Journal of Machine Learning Research*, Vol. 9, pp. 459–483 (2008)

[Yehezkel 09] Yehezkel, R. and Lerner, B.: Bayesian Network Structure Learning by Recursive Autonomy Identification, *Journal of Machine Learning Research*, Vol. 10, pp. 1527–1570 (2009)

〔担当委員：Koichi HIRATA〕

Received June 26, 2013.

◇ Appendix ◇

A. Graph Theoretical Notions

We have described some graph theoretical concepts used in this paper as preliminaries in this appendix.

Let $\mathbb{G}$ be a graph that has vertices and edges. Let $\mathbb{V}$ be a set of random variables, each corresponding to a vertex. Let $\mathbb{E}$ be a set of edges that are directed or undirected. When a pair of two vertices X and Y, denoted by $\langle X,Y\rangle$, satisfies $\langle X,Y\rangle \in \mathbb{E}$ and $\langle Y,X\rangle \notin \mathbb{E}$, a directed edge is drawn from X to Y and X is called Y's parent and Y is called X's child. When two vertices X and Y satisfy $\langle X,Y\rangle \in \mathbb{E}$ and $\langle Y,X\rangle \in \mathbb{E}$, an undirected edge is drawn between X and Y. When either a directed or an undirected edge exists between two vertices X and Y, we say X and Y are **adjacent**. When every pair of vertices is adjacent in a graph $\mathbb{G}$, $\mathbb{G}$ is called a complete graph. When a sequence of distinct $n+1$ vertices $A_0, A_1, \ldots, A_n$ is $\langle A_{i-1}, A_i\rangle \in \mathbb{E}$ or $\langle A_i, A_{i-1}\rangle \in \mathbb{E}$ for all $i = 1,\ldots,n$, it is called a **path** of length n. When a path from X to Y exists that consists of directed edges, X is called an ancestor of Y and Y is called a descendant of X. When Z is not a descendant of X, Z is called a non-descendant of X.

B. Relationships between Conditional Independencies and Causality

Reichenbach found relationships between some (conditional) independencies and causalities from the viewpoint of macroscopic physics [Reichenbach 56], and both CMC and causal structure inference with the CB approach are based on this knowledge. According to Reichenbach, if the following marginal or conditional independencies are held by three random variables X, Y, and Z, such independencies are related to the following cause-effect relations: (i) $X \perp\!\!\!\perp Y \mid Z$ means $X \to Z \to Y$, $Y \to Z \to X$, or $X \leftarrow Z \to Y$, and (ii) $X \perp\!\!\!\perp Y$ and $X \not\perp\!\!\!\perp Y \mid Z$ mean $X \to Z \leftarrow Y$. As the relations of (i) cannot be probabilistically or statistically distinguished from one another, which is called Markov equivalence [Spirtes 00], they are represented by undirected edges in PDAG. On the other hand, the relation of (ii) can be distinguished, and hence it is used for identifying orientations in the CB causal discovery algorithms.

C. Probabilistic Graphical Models as Causal Networks

A joint probability distribution on Bayesian networks has the following decomposed form of conditional probabilities for n discrete variables:

$$P(X_1,\ldots,X_n) = \prod_{i=1}^{n} P(X_i \mid Pa(X_i)),$$

where $Pa(X_i)$ denotes a set of parents' variables of a variable X_i.

Causal models for continuous variables are generally represented by functional causal models [Pearl 00]. We have assumed the following linear Gaussian structural equation models (SEMs) in this study:

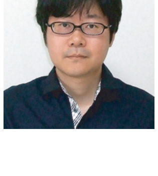
$$X_i = \sum_{X_k \in Pa(X_i)} \gamma_{ik} X_k + u_k, \qquad i = 1,\ldots n,$$

where $Pa(X_i)$ denotes a set of parents' variables of a variable X_i, γ is a set of coefficients, and each u_k is an independent Gaussian noisy term.

D. Orientation Rules

There are three orientation rules [Verma 92, Meek 95a] where there is no prior knowledge: (1) If X and Z are not adjacent and $X \to Y - Z$, then $X \to Y \to Z$. (2) If X and Y are adjacent and there exist other paths from X to Y, then $X \to Y$. (3) If X and Z are not adjacent and $X \to Y \leftarrow Z$, $X - W - Y$, and $Z - W$, then $W \to Z$. These three rules are induced by the constraints of being a DAG and inhibit the generation of new V-structures.

Author's Profile



Isozaki, Takashi (Member)

received his B.S. from Tokyo Institute of Technology and his M.S. from Tohoku University, in physics in 1995 and 1997. He received his Ph.D. in engineering from the University of Electro-Communications in 2010. He has been an associate researcher at Sony Computer Science Laboratories, Inc. since 2010 after working at Fuji Xerox Co., Ltd. His current research interests include statistical data analysis and causal discovery.