Improving Causal Discovery By Optimal Bayesian Network Learning

Ni Y Lu,[1] **Kun Zhang,** [2] **Changhe Yuan** [3]
[1] Graduate Center, City University of New York, New York, NY
[2] Department of Philosophy, Carnegie Mellon University, Pittsburgh, PA
[3] Department of Computer Science, Queens College, Queens, NY
nlu@gradcenter.cuny.edu, kunz1@cmu.edu, changhe.yuan@qc.cuny.edu

Abstract

Many widely-used causal discovery methods such as Greedy Equivalent Search (GES), although with asymptotic correctness guarantees, have been reported to produce sub-optimal solutions on finite data, or when the causal faithfulness condition is violated. The constraint-based procedure with Boolean satisfiability (SAT) solver, and the recently proposed Sparsest Permutation (SP) algorithm have shown superb performance, but currently they do not scale well. In this work, we demonstrate that optimal score-based exhaustive search is remarkably useful for causal discovery: it requires weaker conditions to guarantee asymptotic correctness, and outperforms well-known methods including PC, GES, GSP, and NOTEARS. In order to achieve scalability, we also develop an approximation algorithm for larger systems based on the A* method, which scales up to 60+ variables and obtains better results than existing greedy algorithms such as GES, MMHC, and GSP. Our results illustrate the risk of assuming the faithfulness assumption, the advantages of exhaustive search methods, and the limitations of greedy search methods, and shed light on the computational challenges and techniques in scaling up to larger networks and handling unfaithful data.

Introduction

The goal of causal discovery is to find the underlying causal relations among different variables by analyzing observed data. Causal discovery has been an important research area in artificial intelligence, because interventions or controlled experiments are often impossible or unethical or simply too expensive. In many cases, a directed acyclic graph (DAG) is used to represent the underlying causal structure. Constraint-based methods for causal discovery(Spirtes, Glymour, and Scheines 2000), which make use of conditional independence relations among the variables, can recover a Markov equivalent class (MEC) of DAGs from the observed data.

The PC algorithm (Spirtes, Glymour, and Scheines 1993) and FCI (Spirtes, Glymour, and Scheines 2000) are traditional causal discovery algorithms, and their results are asymptotically correct under the causal Markov condition and the causal faithfulness assumption. The condition and the assumption, put together, imply that two variables are directly causally related if and only if they are not conditionally independent given any subset of the remaining variables. PC starts with a complete graph where an undirected edge exists between each pair of variables. Then it removes an edge $X - Y$ where $X \perp Y|S$ for some $S \subseteq V \setminus \{X, Y\}$, and keeps removing the edges until no such edge exists. PC assumes causal sufficiency, that is, there are no unobserved common causal variables (known as latent confounders). FCI also makes use of conditional independence relations to recover causal information and can handle latent confounders.

GES (Chickering 2002) is a greedy two-phase search algorithm in the space of MECs that optimizes a model fitting score, such as Bayesian Information Criterion (BIC) (Schwarz 1978). It first starts with a graph with no edge, and keeps adding one edge at a time if it improves score the most, until no edge can be added to further improve score. Then it checks all edges to eliminate some if removal further improves score.

Recently, the SAT-based method (SAT) has gained much attention for its superb performance in causal discovery (Hyttinen et al. 2013; Hyttinen, Eberhardt, and Järvisalo 2014). It treats causal discovery as a constraint optimization problem, encodes conditional independence and dependence as Boolean variables and formula, and tackles causal discovery with the Boolean satisfiablity solver. In fact, this constraint optimization can also be seen as a score-based method, where the score is a particular combination of all the constraints to be satisfied. This way, SAT is a special case of score-based exhaustive search. It works very well on small-scale problems (less than 8 variables).

The sparsest permutation (SP) method tries to solve the same constraint optimization problem by enumerating all permutations of variables (Raskutti and Uhler 2018). It considers each permutation as a topological causal ordering where one variable can only use as parents the variables proceeding it in the ordering, not any variables that follow it. SP finds the DAG satisfying the largest number of conditional independence (CI) relations allowed by each ordering. The DAG with the least number of edges is the optimal solution. SP is also a special case of score-based exhaustive search with the number of edges as score, given that the structure explains data well. Greedy SP (GSP) is a greedy version of SP.

NOTEARS (Zheng et al. 2018) formulates structure learn-

ing as a purely continuous optimization problem over real matrices that avoids the combinatorial constraints, achieved by a smooth characterization of acyclicity, and impressively implemented in 60 lines of Python code .

In contrast, there exist a set of methods for Bayesian network learning, aiming at finding the best Bayesian network from the given data. A* is an efficient score-based exhaustive search algorithm that finds the optimal topological ordering by following the shortest path in the order graph (Yuan, Malone, and Wu 2013), instead of the permutation space as in the SP algorithm. Each node in the order graph is a subset of variables. A* uses heuristic to prune some branches in the order graph to cut down computation.

We note that the DAGs learned by Bayesian network learning methods mentioned above do not necessarily have a causal interpretation. For instance, suppose that we are given enough data for two variables which are jointly Gaussian. Such methods will output a directed edge between them, while the direction may be arbitrary. In this case we cannot distinguish different causal structures in the same equivalence class, which share the same (conditional) independence relationships. Generally speaking, this set of methods outputs an arbitrary DAG in this MEC, which, nevertheless, give a compact representation of the joint distribution. In order for the output to have a causal interpretation, one may apply some procedures to generate the MEC from the output DAG (Meek 1995), as an additional post-processing step.

In this work, we present examples and corresponding analyses to show the potential risk of using commonly used causal discovery methods, such as PC and GES, for causal discovery, because of the rather strong assumptions they require. We further emphasize the necessity of optimal exhaustive search method for the benefit of avoiding local solutions on finite data; furthermore, adopting A* as the search method, we show that exhaustive search requires milder assumptions on data to guarantee its asymptotic correctness. Lastly, we extend A* and develop our approximation method (called Triplet A*) for better scalability. This Triplet method is rather general and can be used to scale up other exhaustive search methods as well. Experiments show that our method is better than baseline methods and can handle linear Gaussian and non-Gaussian networks.

Limitations of Existing Methods

The Causal Markov condition (CMC) and Causal Faithfulness condition (CFC) are the two major assumptions underlying many causal discovery methods (Spirtes, Glymour, and Scheines 2000).

Definition 0.1. Causal Markov Condition (CMC): Given a DAG $\mathcal{G}$ over variable set $\boldsymbol{V}$ and probability distribution $\mathbb{P}$ over $\boldsymbol{V}$, $\mathcal{G}$ and $\mathbb{P}$ satisfy the Causal Markov Condition if and only if any variable $X \in \boldsymbol{V}$ is probabilistically independent of $V \setminus \{\text{descendants}(X) \cup \text{parents}(X)\}$ given $\text{parents}(X)$.

Definition 0.2. Causal Faithfulness Condition (CFC): Given a DAG $\mathcal{G}$ over variable set $\boldsymbol{V}$ and probability distribution $\mathbb{P}$ over $\boldsymbol{V}$, $\mathcal{G}$ is faithful to $\mathbb{P}$ if and only if every conditional independence relation true in $\mathbb{P}$ is entailed by the Causal Markov Condition applied to $\mathcal{G}$.

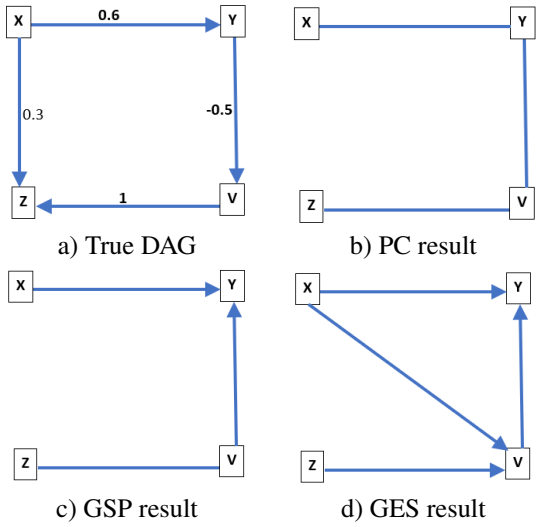

Figure 1: An unfaithful DAG where CFC fails, $X \perp Z$. Results of different algorithms are in b), c), d). The numbers on edges in a) are linear coefficients, not correlations. Data sample size 5000.

A complete DAG where each variable is connected with all other variables automatically satisfies CMC for all probability distributions, because it implies no CI relation at all, hence no useful information for causal discovery. On the other hand, CFC mandates that all the CIs in the distribution $\mathbb{P}$ should be entailed by the true DAG. The DAGs that satisfy both CMC and CFC are the optimal DAGs.

While CMC is widely accepted, CFC has been debated and found sufficient but too restrictive. It is not unusual to see violations or near-violations of CFC, as we will discuss next.

When CFC Fails

To illustrate the failure of some algorithms when CFC is violated, we use the following structural equation model:

$$\boldsymbol{X} = \boldsymbol{B}\boldsymbol{X} + \boldsymbol{e} \tag{1}$$

where $\boldsymbol{X}$ is the vector of observed random variables, $\boldsymbol{e}$ is the vector of residual errors, and $\boldsymbol{B}$ is the causal coefficient matrix. $\boldsymbol{X}$ and $\boldsymbol{e}$ are assumed to follow Gaussian distribution.

Our first example is shown in Figure 1 where structure a) is the true DAG used to generate data. The coefficients of matrix B are labeled on corresponding edges. Data contains an additional CI $X \perp Z$ despite the causal edge $X \to Z$ in the true DAG. The results by PC and GSP miss edge $X - Z$. A*, SP, SAT, and Triplet A* all produce the correct original MEC. GES returns a different MEC with the same number of edges as the true graph but a slightly worse BIC score -10801 vs the optimal -10806. The score difference is far greater than the double precision numeric error 10^{-15}, therefore, the GES result is unambiguously sub-optimal.

The second example is shown in Figure 2 where structure a) is the true DAG. There is an additional CI $X \perp Y|Z$ despite the causal edge $X \to Y$. PC and GES both miss edge

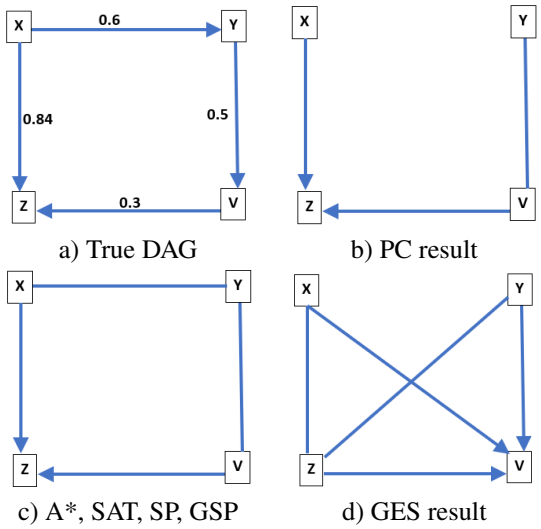

Figure 2: An unfaithful DAG where CFC fails, $X \perp Y|Z$. Results of different algorithms are in b), c), d). The numbers on edges in a) are linear coefficients, not correlations. Data sample size 5000.

$X - Y$. GES also adds spurious edges $X - V$ and $Y - Z$. A*, Triplet A*, SAT, SP, and GSP all recover the true MEC. As shown in the two examples, PC, GES and GSP do not guarantee optimal solution when CFC fails. The exhaustive methods, SAT, SP, and A*, on the other hand, find optimal MECs in both cases.

Finite Size Effect

In addition to CFC failures, greedy methods like GES can get trapped in local optima, even when the data set size is large. To see this, we generated 30 random linear Gaussian networks, 20 variables each network, and 4 neighbors per variable on average. Then we run A* and GES on n data points generated by the causal model, where n = 50, 100, 200, 500, 1k, 2k, 5k, 10k, 20k, 50k, 100k, 200k. The performance metrics, precision, recall, and F1 score are averaged over 30 networks for each sample size n.

The result is shown in Figure 3. The left three figures are plotted on log scale on x-axis so that we can see all the sample sizes better. We calculated standard deviations for all metrics, but can only show the more informative error bar plots (GES precision, PC recall, and A* F1-score) on the right hand side due to space limit (not on log scale). We run GES and A* on $\lambda = 0.5, 1, 2$, where λ is the penalization coefficient in the BIC score function Equation 2. The curves are similar in shape for these λ values, so we only show $\lambda = 1$ in the plots for better readability.

The performances of PC and A* improve as the sample size n increases, which is expected. The F1 score by GES increases to around 90% and oscillates in that region, does not approach to 1 while its recall curve approaches 1. The GES precision curve peaks around $n = 200 - 500$, and then surprisingly decreases and remains flat with standard deviation around 20% as the sample size n increases. On the other hand, the PC precision increases steadily when the sample size increases, but its recall plateaus around 94% with standard deviation around 5%. The NOTEARS does very well on precision, but its recall remains low around 67% even at large sample sizes. Starting at sample size 1000, the PC precision and F1 score exceed those of GES with various λ values. A* has very high precision and recall, both approach 100% quickly when n increases, and the standard deviation is very small, indicating that A* is clearly more reliable on randomly generated data.

To confirm that this behavior of GES is repeatable, we also generated different data sets under other parameter setting, and still observed the same behavior. We run GES with three phases (forward, backward, turning)(Kalisch et al. 2012) and two phases (forward, backward) (Chickering 2002) and the behavior is consistent. The plots show the result with three phases.

The high standard deviation in the GES precision and F1 score suggests that the GES precision is very low in a significant number of data sets. This may be partially because of violation of faithfulness on finite data. However, this does not seem to be the only reason why GES does not work well, in light of the observation that PC produces more accurate results than GES on large samples, as seen from the higher F1 score. It is possible that GES may suffer from some systematic issues in the optimization procedure on particular types of data sets.

To analyze this phenomenon of GES not converging in large sample size, we count the number of edges for each learned DAGs by GES and A* per sample size (50, 100, 200, 500...) for each of the 30 graphs. We plot the learned BIC scores for GES and A* (here we use the same score function for both methods), and the number of edges in the learned networks in Figure 4. The first plot shows that the A* and GES scores are rather close. The second plot shows that A* scores are always better than or equal to GES scores. Here, better scores means lower scores because A* minimizes the score. This verifies that GES may not be able to produce the globally optimal scores. The third one, scattered plot for number of edges, reveals that, the networks learned by GES have significantly more edges than those by A* in most cases, indicating that GES often enters some local optima and obtains some very different networks that contain more spurious edges. It is unclear why GES behaves this way. One possible explanation is that, GES does not calculate scores for all possible parent sets, so that it may often be misled by local scores and get trapped in local optima.

Scalability Issue

We have seen that exhaustive search methods give remarkable search results, especially when CFC does not hold. But Bayesian learning is NP-hard (Chickering 1996), and these exhaustive search methods do not scale well. Let N denote the number of variables. The SP time complexity grows as $\mathcal{O}(N!)$ because it enumerate all permutations. SP can handle Bayesian networks up to 10 variables. The SAT space complexity grows as $\mathcal{O}(2^{N(N-1)/2})$. SAT can handle up to 8 variables. A* is more efficient; the space complexity grows $\mathcal{O}(N2^{N-1})$ for scores and time $\mathcal{O}(2^N)$. A* can scale up to

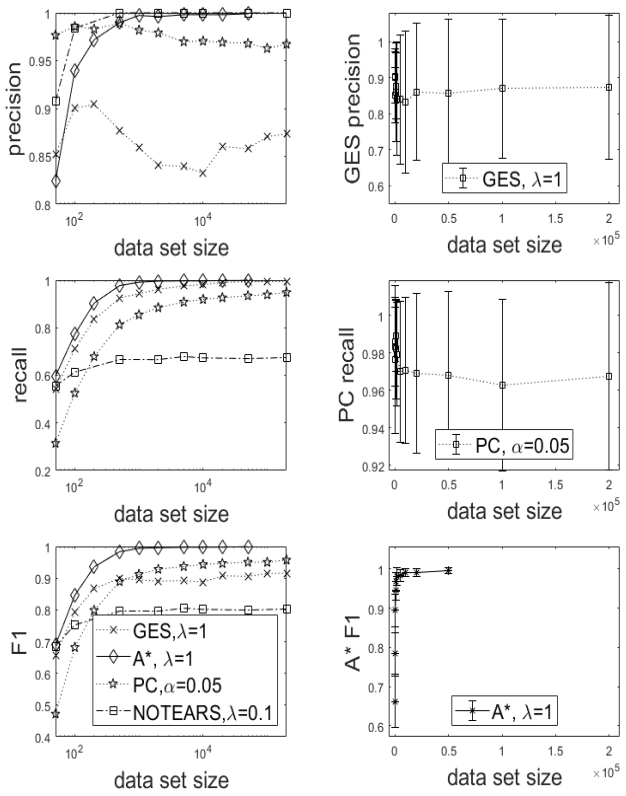

Figure 3: Finite Size Effect, average over 30 random DAGs, 20 variables each DAG, 4 neighbors per variable on average. $\lambda = 1$ in BIC for A*. GES was run with $\lambda = 0.5, 1, 2$, only $\lambda = 1$ shown.

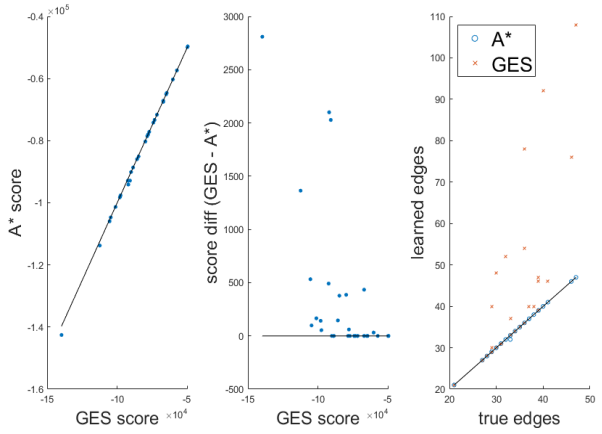

Figure 4: Score and edge comparison, the same 30 random DAGs as in Figure 3, 20 variables each DAG, 4 neighbors per variable on average. $\lambda = 1$ in BIC for A* and GES . Sample size $n = 10,000$.

30 variables for any networks or more variables with sparser networks. So it is very desirable to scale up these methods to handle larger systems.

Weaker Condition for Exhaustive Search

As shown in Figures 1 and 2, exhaustive search methods such as A*, SAT, and SP, outperform PC, GES and GSP, when CFC fails. These exhaustive search algorithms guarantee optimal solution under the Frugality condition (Forster et al. 2018), which is strictly weaker than CFC.

Definition 0.3. Frugality Condition: Given a probability distribution $\mathbb{P}$ on $\boldsymbol{V}$, the true DAG over $\boldsymbol{V}$ is in the set of maximally frugal DAGs for $\boldsymbol{V}$.

A DAG $\mathcal{G}$ is more frugal than DAG $\mathcal{G}'$ if $\mathcal{G}$ has fewer edges than $\mathcal{G}'$. The maximally frugal DAGs have the least number of edges while satisfying CMC. The Frugality condition acknowledges that there might be unfaithful conflicting CIs in the data, and it chooses the MEC that allows maximum number of CIs while ignoring the rest of CIs.

Here we will show that BIC score-based exhaustive search guarantees optimal solution asymptotically under a similar condition to Frugality. The BIC score (Schwarz 1978) is defined as

$$\text{BIC} = -\log L + \lambda k \log n \tag{2}$$

where L is the maximized value of the likelihood function of the model based on the observed data, k is the number of parameters in the model, and n is the number of observed data points. The second term in BIC penalizes the number of parameters k, avoiding over-fitting by favoring simpler models. The parameter $\lambda = \frac{1}{2}$ in the original definition of BIC, but larger values are often used to adjust penalty. In our DAG case, each directed edge corresponds to at least one parameter. So k increases strictly monotonically with the number of edges.

When $n \to \infty$, BIC will select the most parsimonious model if the true model is in the class of models under consideration (Nishii 1984; Shibata 1981). BIC Score has the following asymptotic consistency.

1. As $n \to \infty$, the true structure G* (or any I-equivalent structure) minimizes the score.
2. Asymptotically, spurious edges will not contribute to likelihood and will be penalized.
3. Required edges will be added due to linear growth of likelihood term compared to logarithmic growth of model complexity.

BIC score is calculated per variable per parent set. That is, for each variable $X \in \boldsymbol{V}$, BIC($X|\boldsymbol{S}$) is calculated for each $S \subset \boldsymbol{V}$. Then the BIC scores of all variables are summed up in a DAG. The goal is to find the DAG that minimizes the total BIC score.

In order to discuss the condition under which BIC score-based exhaustive search guarantees optimal solution asymptotically, we introduce the following definition.

Definition 0.4. Optimal BIC Condition (OBC): Given a probability distribution $\mathbb{P}$ on $\boldsymbol{V}$, the true DAG $\mathcal{G}^*$ over $\boldsymbol{V}$ satisfies the Optimal BIC Condition if for every DAG

$\mathcal{G}$ that is Markov relative to $\mathbb{P}$ and $\mathcal{G} \notin \text{MEC}(\mathcal{G}^*)$ then $\text{BIC}(\mathcal{G}) > \text{BIC}(\mathcal{G}^*)$.

We will follow similar arguments in (Raskutti and Uhler 2018) to show that BIC score-based exhaustive search algorithms only require OBC.

Theorem 0.1. BIC score-based exhaustive search algorithms find the MEC of the true DAG if and only if the true DAG satisfies the Optimal BIC Condition.

Proof. If the true DAG $\mathcal{G}^*$ satisfies the Optimal BIC Condition and the DAG returned by exhaustive search $\mathcal{G}_{ES} \notin \text{MEC}(\mathcal{G}^*)$ then $\text{BIC}(\mathcal{G}_{ES}) > \text{BIC}(\mathcal{G}^*)$, contradicting minimizing BIC score by exhaustive search. Hence, we must have $\mathcal{G}_{ES} \in \text{MEC}(\mathcal{G}^*)$.

Suppose the true DAG does not satisfy the Optimal BIC Condition, there must exist $\mathcal{G}$ that is Markov relative to $\mathbb{P}$, $\mathcal{G} \notin \text{MEC}(\mathcal{G}^*)$ and $\text{BIC}(\mathcal{G}) \leq \text{BIC}(\mathcal{G}^*)$. If $\text{BIC}(\mathcal{G}) = \text{BIC}(\mathcal{G}^*)$, then exhaustive search will return an arbitrary one or both of them. If $\text{BIC}(\mathcal{G}) < \text{BIC}(\mathcal{G}^*)$ then exhaustive search will return $\mathcal{G}$. In both cases, $\mathcal{G} \notin \text{MEC}(\mathcal{G}^*)$.

□

Remark. With the BIC asymptotic consistency on spurious edges and required edges, at large sample sizes, the learned graph $\mathcal{G}_{ES}$ with the optimal BIC has all the required edges and no spurious edges, therefore, $\mathcal{G}_{ES}$ shares the same skeleton as $\mathcal{G}^*$, $E(\mathcal{G}_{ES}) = E(\mathcal{G}^*)$. Since both $\mathcal{G}_{ES}$ and $\mathcal{G}^*$ have the likelihood, it is very tempting to claim that the two DAGs must share the same v-structures, hence belong to the same Markov equivalent class, and we can drop the requirement of Optimal BIC Condition in Theorem 0.1. How can two DAGs with the same skeleton but different v-structures (hence different set of CIs) have the same likelihood ? Theoretically we cannot rule out the case, or need more in-depth proof to show that the probability of multiple Markov in-equivalent DAGs at optimal BIC score approaches zero asymptotically. That is why we require $\text{MEC}(\mathcal{G}^*)$ is the only Markov equivalent class that gives the optimal BIC score. This is not an overly restrictive requirement by any means. Similar to the Frugality, OBC is strictly weaker than CFC (See Figures 1 and 2).

In the unthinkable case where there are multiple Markov in-equivalent DAGs with the same skeleton at the optimal BIC score, they should all be considered true DAGs because they all have the same likelihood. Exhaustive search algorithms cannot distinguish these Markov in-equivalent DAGs based on BIC score.

In Figure 1, the true DAG 1a) entails $X \perp V|Y$ and $Y \perp Z|\{X, V\}$, while the GES DAG 1d) entails $X \perp Z$ and $Y \perp Z|\{X, V\}$. They both satisfy CMC because the data contains all three CIs. They are both maximally frugal with 4 edges. But they do not have the same BIC score. So the BIC score-based exhaustive search is able to select the true MEC.

Challenges in Scaling up Exhaustive Search

There have been enormous amount of effort to scale up the exact exhaustive search algorithms to handle larger networks by divide-and-conquer (Kojima et al. 2010; Friedman, Nachman, and Peér 1999). However, no attempt has been successful or practical. Friedman, Nachman, and Peér (1999) proposed a divide-and-conquer approach to a **digraph** (directed graph, not a regular un-directed skeleton) with a small subset of variables that partitions the digraph into two clusters, called sepset. Their scheme also requires that all the potential parents of the variables in a sepset must be on the same side of the partition. It is quite restricted to require a skeleton to be fully directed as input to Bayesian learning problem. In most cases, data are collected without knowing the full causal relations among them, hence, no directions. Even if the input is a directed graph, a partition has to be carefully chosen so that, for each boundary variable, all its parents are in one cluster. But such an ideal partition might not exist.

The intrinsic difficulty of the divide-and-conquer approach is that, there are always boundary variables when dividing a strongly connected graph, and there is simply no way to put all potential parent variables on one side of a partition for each boundary variable. Therefore, when applying exhaustive search in one cluster, there is no guarantee that the optimal parents can be found for boundary variables. As a consequence, not all edges in the search result for a partial cluster can be trusted.

Algorithm: Scaling up A*

To overcome the difficulty in scaling up exhaustive search, we propose an approximation method based on exhaustive search method A*. By CMC, we assume that in a cluster containing variables X, Y, Z, and all their parents and children, the exhaustive search methods can find the optimal edges among X, Y, Z, equivalent to those in the globally optimal DAG. We can run exhaustive search algorithms on smaller clusters locally to discover the causal networks around each variable, combine the results, and resolve conflict if any. The combined result, however, is not exact, because of potential conflicts among the DAG's of all clusters.

The most natural choice of cluster for scaling up is the Markov Blanket (MB). The MB for a variable i includes its parents $\text{Pa}(i)$, children and spouses (other parents of children). We can obtain a MB through some methods such as the PC-algorithm (Spirtes, Glymour, and Scheines 1993), Max-Min Markov Blanket algorithm (MMMB)(Tsamardinos, Aliferis, and Statnikov 2003). We can run A* on each Markov blanket and combine the results. However, this single MB method has several drawbacks. The first drawback is that it may produce spurious V-structures when missing spouses. If $X \to Y$ in the true DAG, but other parents of Y are missing in the MB, then the edge $X - Y$ can take any direction, because $\text{Score}(X|Y) = \text{Score}(Y|X)$. The second drawback is the big size of MB when a variable has many children, because all spouses are included. This happens in denser networks.

Triplet Method

To avoid the drawbacks of the single Markov blanket method, we propose a new approach by combining neighbors around each triple $X - Y - Z$. Here is a brief description of the algorithm.

First, obtain initial guess of parents and children for each variable. Second, for each variable X and each pair of neighbors (Y, Z) of X, form a cluster consisting of X, Y, Z and their direct neighbors, and run exhaustive search on each cluster. Third, combine the results from all clusters.

Because only X, Y, Z have all their parents present in the triplet cluster, we can only trust the edges among these three variables. If there is a V-structure, record the directed edges. Otherwise, record the undirected edges. If an edge is already undirected, and later it joins a V-structure, then it becomes directed consistent with the V-structure. If an edge is already directed by some V-structure, then it will not become undirected any more. This is not a perfect approach to resolve conflicts among clusters, as some edge orientation might depend on order of traversal. But it is simple and effective, and works pretty well in our experiments. More sophisticated approaches can also be devised with more book keeping.

While examining the triple $X - Y - Z$, exhaustive search might choose the optimal network by adding edge $X - Z$, directed or undirected, if $X \perp Z|Y$ is an unfaithful CI from the data and edge $X - Z$ improves score. This is how it can handle some missing edges caused by a common type of unfaithfulness. This addresses the first drawback of single MB approach.

The triplet clusters are usually smaller than the biggest MB in a dense network. For example, when each variable has 5 neighbors, for triple $X - Y - Z$, the triplet cluster size is 4 (neighbors of X) + 3 (neighbors of Y) + 4 (neighbors of Z) + 3 (X, Y, Z) = 14. The largest single MB (including the variable itself) for Y, assuming 5 children, can contain 5*4 (spouses) + 5 (children) + 1 (Y itself) = 26 variables. It will take much longer (over $2^{26-14} = 4096$ times) to run exhaustive search on a cluster of 26 variables than that of 14 variables. This partially addresses the second drawback of single MB approach.

Suppose a variable has m neighbors at most, then we need run A* routine at most $m(m-1)/2$ times for each variable, and the total number of A* runs is at most $Nm(m-1)/2$ for a system of N variables. The total running time upper bound is $\mathcal{O}(Nm^2 2^{3m})$. As long as m is reasonably small, for example, $m \leq 7$, which is much larger than the tree width used in many large networks (usually $m = 2$), the running time is proportional to N.

Simulations and Experiments

For comparison with other methods on handling unfaithful networks, we run Triplet A* on the unfaithful network in Figure 1 and Figure 2. Triplet A* finds the correct MEC.

To compare the performances of GES, SAT, GSP, MMHC (Tsamardinos, Brown, and Aliferis 2006), A*, and Triplet A*, we use the R package by (Hyttinen, Eberhardt, and Järvisalo 2014) to generate random DAGs. First, we start with seven variables, due to the restriction of SAT. We use the 'ges' and 'pc' functions in the R package pcalg, the 'mmhc.skel' function in R package MXM, and the 'gsp' function in the Python package causaldag (Solus et al. 2020).

Figure 5 shows the performance plots vs edge density. F1, the harmonic mean of precision and recall, is a better measure than precision and recall alone. The SAT method performs very well, but A* and Triplet A* can match SAT on the balanced F1 plot. GES also performs as well as SAT on sparse networks, but under-performs on denser networks.

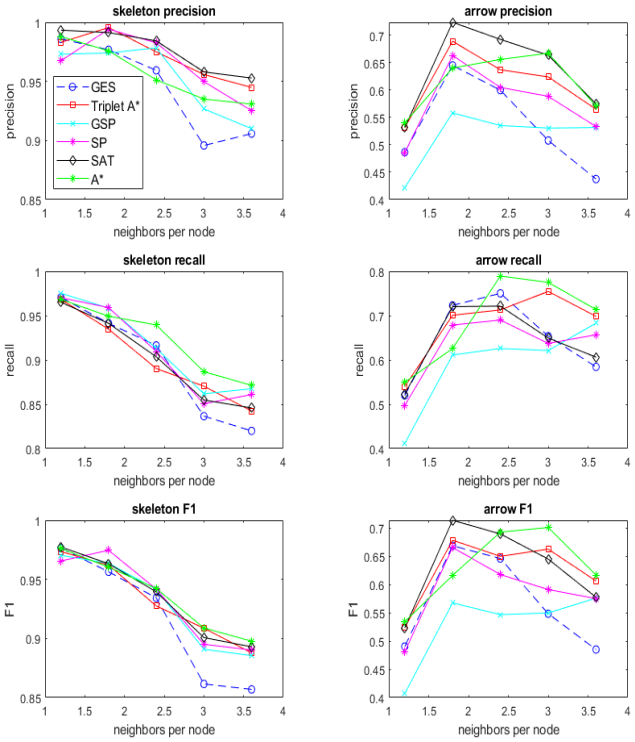

Figure 5: SAT, GES, GSP, SP, A*, and Triplet A* performance on 7-variable networks. Each plot is averaged over 50 random DAGs, 500 data points each DAG

After demonstrating its good performance on small networks, we also apply GES, GSP, PC, LiNGAM (Shimizu et al. 2006), and Triplet A* on larger networks with 60 variables, with various edge densities, shown in Figure 6. Triplet A* performs as well as GES on sparse networks, and slightly outperforms GES on denser networks. We then use the squared Gaussian distribution for the noise distribution in the BIC score, with the search results given in Figure 7. One can see that the BIC score is rather robust to Gaussian and non-Gaussian noises, as there is no clear performance difference between Figure 6 with Gaussian noise and Figure 7 with squared Gaussian noise.

The GES precision also decreases with edge density, but the Triplet A* precision is relatively stable. The Triplet A* orientation recall is lower than GES, partially because edges are oriented only by V-structures.

Since we use the MMMB skeleton before running A*, the quality of the skeleton is important. The simulation result shows that when the skeleton does not miss any edge, the Triplet A* precision and recall are much higher than GES. When there is some unfaithful edge missed in the MMMB skeleton, the Triplet performance can be affected, but it can recover some missing edges.

Note that each method has some parameter to tune. For

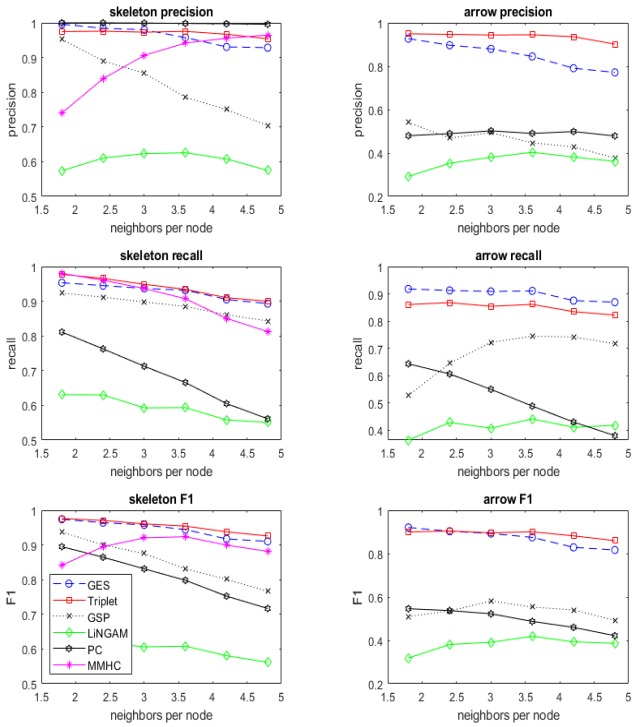

Figure 6: 60-variable networks, Gaussian noise. Each graph is averaged over 50 random DAGs, 500 data points each DAG.

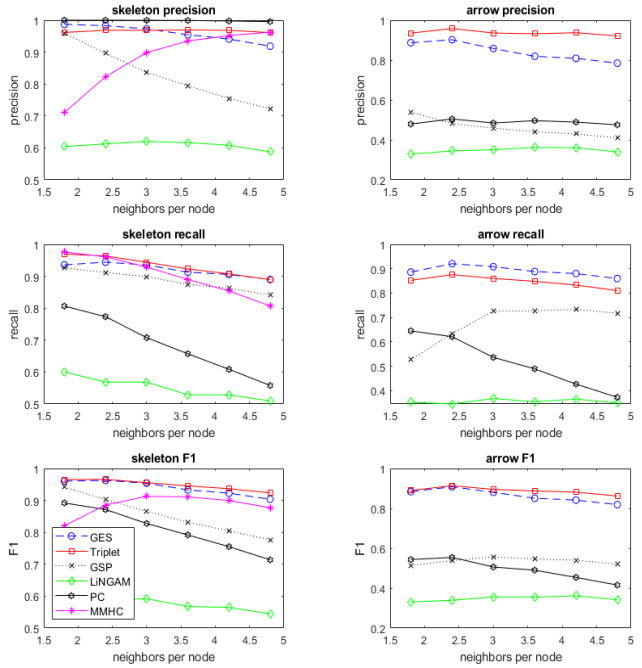

Figure 7: 60-variable networks. Gaussian noise squared, but sign unchanged. Each graph is averaged over 50 random DAGs, 500 data points each DAG.

GES and A*, the λ in BIC score affects the performance at small sample sizes. For large sample sizes, we find that λ has no clear impact on the learned structures. In our experiments, we use $\lambda = 0.5, 1, 2$. For PC, SAT, and SP, the α in CI test is important for performance, and it varies dramatically for different methods. We find $\alpha = 0.0001$ optimal for GSP on networks of 60 variables at sample size 500. For SAT and SP, we use $\alpha \in [0.2, 0.4]$ to get optimal performance. For PC, we try $\alpha = 0.01, 0.05, 0.1$, and $\alpha = 0.05$ gives the highest F1 score. It seems that different methods require different optimal α for CI test.

Conclusion

For small networks, exhaustive methods such as A*, SAT, and SP outperform PC, GES, GSP, and NOTEARS. A* scales better than SAT and SP. Greedy methods such as GES and GSP suffer from local optima. For larger networks, Triplet A* performs at least as well as GES, and outperforms GES in denser networks.

These findings are worth emphasizing. First, in practice it might be risky to assume faithfulness. Second, score-based methods, especially when equipped with an optimal search procedure, demonstrate certain advantages, compared to constraint-based methods, as they are less susceptible to data unfaithfulness than conditional independence test. Third, greedy methods may frequently give sub-optimal results on finite data. Fourth, the proposed triplet method works well in practice. For future work, we will investigate the precision-saturation phenomenon of GES, discussed in Section 2, and accordingly develop its improved version.

Acknowledgments

We want to thank Brandon Malone for his great help in using his open-source A* library, and thank Antti Hyttinen for helpful instructions and updates on how to run his R package that implements the SAT method. We thank Chandler Squires for instructions on running the SP/GSP Python package. We thank Wayne Lam, Joseph Ramsey, Peter Spirtes for helpful discussions regarding uniqueness of optimal solutions. KZ would like to acknowledge the support by the United States Air Force under Contract No. FA8650-17-C-7715 and a generous gift from Apple. We also want to credit the Google MLPACK project for its efficient C++ library on machine learning, especially various regression functions.

References

Chickering, D. 1996. Learning Bayesian Networks is NP-Complete. In Fisher, D.; and Lenz, H.-J., eds., *Learning from Data. Lecture Notes in Statistics*, volume 112, 121 – 130. New York, NY: Springer.

Chickering, D. M. 2002. Optimal Structural Identification with Greedy Search. *Journal of Machine Learning Research* 3: 507–554.

Forster, M.; Raskutti, G.; Stern, R.; and Weinberger, N. 2018. The frugal inference of causal relations. *The British Journal for the Philosophy of Science* 69(3): 821–848.

Friedman, N.; Nachman, I.; and Peér, D. 1999. Learning Bayesian Network Structure from Massive Datasets: The Sparse Candidate Algorithm. *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence* 55(1): 206–215.

Hyttinen, A.; Eberhardt, F.; and Järvisalo, M. 2014. Constraint-based causal discovery: Conflict resolution with answer set programming. *Uncertainty in Artificial Intelligence* 340–349.

Hyttinen, A.; Hoyer, P. O.; Eberhardt, F.; and Järvisalo, M. 2013. Discovering Cyclic Causal Models with Latent Variables: A General SAT-Based Procedure. *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence, UAI* .

Kalisch, M.; Mächler, M.; Colombo, D.; Maathuis, M. H.; and Bühlmann, P. 2012. Causal Inference Using Graphical Models with the R Package pcalg. *Journal of Statistical Software* 47(11): 1–26.

Kojima, K.; Perrier, E.; Imoto, S.; and Miyano, S. 2010. Optimal Search on Clustered Structural Constraint for Learning Bayesian Network Structure. *Journal of Machine Learning Research* 11: 285–310.

Meek, C. 1995. Strong completeness and faithfulness in BAyesian networks. *Proceedings of the Proceedings of the Eleventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)* 411–419.

Nishii, R. 1984. Asymptotic properties of criteria for selection of variables in multiple regression. *Annals of Statistics* 12: 758–765.

Raskutti, G.; and Uhler, C. 2018. Learning directed acyclic graph models based on sparsest permutations. *arXiv:1307.0366* .

Schwarz, G. E. 1978. Estimating the dimension of a model. *Annals of Statistics* 6(2): 461–464.

Shibata, R. 1981. An optimal selection of regression variables. *Biometrika* 68(1): 45–54.

Shimizu, S.; Hoyer, P. O.; Hyvärinen, A.; and Kerminen, A. J. 2006. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research* 7: 2003–2030.

Solus, L.; Wang, Y.; Matejovicova, L.; and Uhler, C. 2020. Consistency guarantees for permutation-based causal inference algorithms. *arXiv:1702.03530* .

Spirtes, P.; Glymour, C.; and Scheines, R. 1993. *Causation, Prediction, and Search*. Springer-Verlag.

Spirtes, P.; Glymour, C.; and Scheines, R. 2000. *Causation, Prediction, and Search*. MIT Press.

Tsamardinos, I.; Aliferis, C. F.; and Statnikov, A. 2003. Time and sample efficient discovery of Markov blankets and direct causal relations. *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining* 673–678.

Tsamardinos, I.; Brown, L.; and Aliferis, C. 2006. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning* 65(1): 31–78.

Yuan, C.; Malone, B.; and Wu, X. 2013. Learning optimal Bayesian networks: A shortest path perspective. *Journal of Artificial Intelligence Research* 48: 23–65.

Zheng, X.; Aragam, B.; Ravikumar, P.; and Xing, E. P. 2018. DAGs with NO TEARS: Continuous optimization for structure learning. *NeurIPS* .