A Strategy for Making Predictions Under Manipulation

Laura E. Brown [*] LAURA.E.BROWN@VANDERBILT.EDU
Department of Biomedical Informatics, Vanderbilt University, Nashville, TN 37232, USA

Ioannis Tsamardinos TSAMARD@ICS.FORTH.GR
Department of Computer Science, University of Crete and
BMI, ICS, Foundation for Research and Technology Hellas, Heraklion, Crete GR 700 13, GREECE
Department of Biomedical Informatics, Vanderbilt University, Nashville, TN 37232, USA

Editors: I. Guyon, C. Aliferis, G. Cooper, A. Elisseeff, J.-P. Pellet, P. Spirtes, and A. Statnikov

Abstract

The first Causality Challenge competition posted several causal discovery problems that require researchers to employ the full arsenal of state-of-the-art causal discovery methods, while prompting the development of new ones. Our approach used the formalism of Causal Bayesian Networks to model and induce causal relations and to make predictions about the effects of the manipulation of the variables. Using state-of-the-art, under development, or newly invented methods specifically for the purposes of the competition, we addressed the following problems in turn in order to build and evaluate a model: (a) finding the Markov Blanket of the target even under some non-faithfulness conditions (e.g., parity functions), (b) reducing the problems to a size manageable by subsequent algorithms, (c) identifying and orienting the network edges, (d) identifying causal edges (i.e., not confounded), and (e) selecting the causal Markov Blanket of the target in the manipulated distribution. The results of the competition illustrate some of the strengths and weaknesses of the state-of-the-art of causal discovery methods and point to new directions in the field. An implementation of our approach is available at http://www.dsl-lab.org for use by other researchers.

Keywords: Causal Bayesian Networks, Causal Discovery, Manipulations

1. Introduction

In order to optimally predict the effects of manipulations on a system, one needs to induce a subset of the causal relations among the parts of the system. Three key characteristics of the challenge data sets led to the choice of Causal Bayesian Networks (CBN) as the formalism to model and induce causal relations and to make predictions about the effects of the manipulation of the variables: the data contain cross-sectional measurements, the generating causal models contain no feedback loops, and the definition of causality is stochastic. A CBN is a Bayesian Network where the edges have the additional semantics that they correspond to direct causal relations. Thus, a first major assumption in our analyses is that there exists a CBN that can represent the probability distribution of the data. This in turn implies that we assume the Causal Markov Condition holds: every node X is probabilisti-

[*] Conflict of Interest Disclosure: The authors of this paper are and have been affiliated with some of the organizers of the challenge. Consequently, our submission was not eligible for any prizes but, was allowed to participate in the challenge.

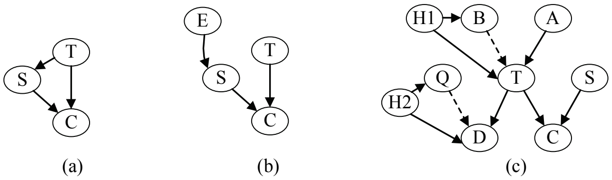


Figure 1: Causal Bayesian Networks: The unmanipulated CBN graph, $G_\emptyset$, and CBN graph $G_{\{S\}}$ where S is manipulated, are depicted in (a) and (b). In (c), a network with a hidden variables $H1$ causing both B and T, $H2$ causing both D and Q, and dashed edges (when the marginal over the observed variables, $\mathcal{O}$, is considered) is shown.

cally independent of its non-causal effects conditioned on its direct causes. An example of a graph of a CBN is shown in Figure 1(a).

1.1 Theory for Making Predictions Under Manipulation

We will denote the variable to predict with the letter T (target). Let us denote the set of variables as $\mathcal{V}$ that is partitioned into observed variables included in the data $\mathcal{O}$, and unobserved variables $\mathcal{H}$. Single variables are denoted with capital letters or with V_i where i is an index and sets of variables with bold capital letters. Let $\mathbf{M}$ denote the set of manipulated variables. For the challenge it is assumed that $\mathbf{M} \subseteq \mathcal{O}$, i.e., there are no manipulated unobserved variables. We will denote with $P_{\mathbf{M}}(\mathcal{V})$ the joint probability distribution of variables $\mathcal{V}$ when the set of manipulated variables is $\mathbf{M}$. There were three different types of tasks in the competition, each requiring a different approach, that we now explain.

1.1.1 Predictions under no manipulation

For this type of task, one could first estimate $P_\emptyset(T|\mathcal{V} \setminus \{T\})$. The estimation may be difficult and unreliable if the size of $\mathcal{V}$ is large. A Markov Blanket of T, $MB_\emptyset(T)$, for distribution $P_\emptyset$, is defined as a minimal set such that $P_\emptyset(T|\mathcal{V} \setminus \{T\}) = P_\emptyset(T|MB_\emptyset(T))$. In other words, a Markov Blanket contains the required information for optimal prediction of T, thus rendering the remaining variables superfluous and is the solution to the variable selection problem under some general conditions (Tsamardinos and Aliferis, 2003). Notice that in a CBN (by definition a minimal I-map, Pearl, 1988), a $MB_\emptyset(T)$ corresponds to the parents, children, and spouses of T in the graph (Pearl, 1988, Sec. 3.3, Corollary 6). Based on the above, our approach for this task was to identify a Markov Blanket of T, $MB_\emptyset(T)$ then learn a predictive model using only these variables.

1.1.2 Predictions under known manipulations

In this case, we assume that there is a known subset of variables $\mathbf{M} \subseteq \mathcal{O}$ that are being effectively manipulated, i.e., their values are completely determined by the external agent, that we model with variable E. As in a typical supervised learning setting, one could attempt to learn a model for $P_{\mathbf{M}}(T|\mathcal{V} \setminus \{T\})$. According to Pearl (2000) and Spirtes et al.

(2000), the joint distribution can be factorized as

$$P_{\mathbf{M}}(\mathcal{V}) = \prod_{V_i \in \mathcal{V} \setminus \mathbf{M}} P_{\emptyset}(V_i|Pa(V_i)) \cdot \prod_{V_i \in \mathbf{M}} P_{\mathbf{M}}(V_i|E)$$

where $Pa(V_i)$ are the parents (direct causes) of V_i and $P_{\mathbf{M}}(V_i|E)$ the manipulated distribution of a variable. From $P_{\mathbf{M}}(\mathcal{V})$ one could obtain $P_{\mathbf{M}}(T|\mathcal{V} \setminus \{T\})$ and solve the problem. However, this approach requires knowledge of the distributions of the manipulated variables $P_{\mathbf{M}}(V_i|E)$ that is not provided; in addition, it requires fitting the complete joint distribution of the variables that is computationally inefficient and prone to statistical errors.

Alternatively, we employ the concept of the Markov Blanket, to instead learn a model for $P_{\mathbf{M}}(T|MB_{\mathbf{M}}(T))$. If the causal graph is known, the $MB_{\mathbf{M}}(T)$ can be identified from it as follows. Let $G_{\emptyset}$ and $G_{\mathbf{M}}$ be the CBN graphs of the unmanipulated and manipulated distribution respectively. From Pearl (2000) and Spirtes et al. (2000), $G_{\mathbf{M}}$ results from $G_{\emptyset}$ by removing the direct causes of every variable $V_i \in \mathbf{M}$ and replacing them with an edge from an external agent performing the manipulations, E. An example is shown in Figures 1(a-b) for $\mathbf{M} = \{S\}$. Intuitively, this is justified by the fact that the manipulated variables have no other causal dependence but with the external agent. Thus, $MB_{\mathbf{M}}(T)$ is a subset of $MB_{\emptyset}(T)$ with manipulated children and their corresponding spouses removed (if a node is a spouse via multiple children, it is removed only if all of them are manipulated). Even if $MB_{\mathbf{M}}(T)$ is known, $P_{\mathbf{M}}(T|MB_{\mathbf{M}}(T))$ should be *induced from observational data following* $P_{\emptyset}$. We now present the following theorem stemming again from the more general theory of probability invariance under manipulations by Spirtes et al. (2000) (proof in Appendix):

Theorem 1 *Let $\langle G_{\emptyset}, P_{\emptyset} \rangle$ be a CBN and $\langle G_{\mathbf{M}}, P_{\mathbf{M}} \rangle$ be the resulting CBN under manipulations of variables in $\mathbf{M}$. Suppose that $T \notin \mathbf{M}$ and also that there is no manipulated child C of T in $G_{\emptyset}$ with a descendant D in $G_{\emptyset}$ that is also in $MB_{\mathbf{M}}(T)$. Then,*

$$P_{\mathbf{M}}(T|MB_{\mathbf{M}}(T)) = P_{\emptyset}(T|MB_{\mathbf{M}}(T)).$$

In other words, when the theorem holds, we can learn an optimal model for predicting T in the manipulated distribution by learning $P_{\emptyset}(T|MB_{\mathbf{M}}(T))$ from data sampled from the unmanipulated distribution. The latter of course requires knowledge of $MB_{\mathbf{M}}(T)$ which is a subset of $MB_{\emptyset}(T)$. When the theorem does not hold, then predicting T using $P_{\emptyset}(T|MB_{\mathbf{M}}(T))$ is not theoretically guaranteed to be optimal; however, the condition of the theorem is relatively strict and it is expected that it often holds in practice (of course, this claim requires further evaluation).

Notice the condition regarding the existence of a manipulated child of T and its descendant $D \in MB_{\mathbf{M}}(T)$ is important. Consider the network in Figure 1(a), where the condition does not hold when S is manipulated, and the resulting network 1(b). Then, we have:

$$P_{\emptyset}(T|MB_{\mathbf{M}}(T)) = \frac{P_{\emptyset}(T) \cdot P_{\emptyset}(S|T) \cdot P_{\emptyset}(C|S,T)}{\sum_t P_{\emptyset}(t) \cdot P_{\emptyset}(S|t) \cdot P_{\emptyset}(C|S,t)}$$

$$P_{\mathbf{M}}(T|MB_{\mathbf{M}}(T)) = \frac{P_{\mathbf{M}}(T) \cdot P_{\mathbf{M}}(do(S)) \cdot P_{\mathbf{M}}(C|S,T)}{\sum_t P_{\mathbf{M}}(t) \cdot P_{\mathbf{M}}(do(S)) \cdot P_{\mathbf{M}}(C|S,t)} = \frac{P_{\emptyset}(T) \cdot P_{\mathbf{M}}(do(S)) \cdot P_{\emptyset}(C|S,T)}{\sum_t P_{\emptyset}(t) \cdot P_{\mathbf{M}}(do(S)) \cdot P_{\emptyset}(C|S,t)},$$

where $P(do(S))$ follows Pearl's nomenclature denoting the probability of S being manipulated to obtain a specific value and if V is not manipulated then $P_{\mathbf{M}}(V|Pa(V)) = P_{\emptyset}(V|Pa(V))$ (see Pearl, 2000 for explanation and discussion). In general the top quantity takes different values from the bottom one; when the theorem does not hold, we could still fit a model from the observational data and use it in the manipulated distribution, if information about the distribution of the manipulations is provided.

From the above discussion, to identify $MB_{\mathbf{M}}(T)$ one needs to know both $MB_{\emptyset}(T)$ and the edge orientation in that graph neighborhood. So, we first attempt to learn the causal network from the training data and then derive $MB_{\mathbf{M}}(T)$ by deleting the appropriate edges. There are two potential problems with this approach, even if the network is induced perfectly. First, there may be several statistically indistinguishable networks that fit the data equally well. For example, the models $T \rightarrow X$ and $T \leftarrow X$ are indistinguishable with the $P_{\emptyset}$ distribution. We do not have a solution to this problem, which implies that some manipulated children of T may be falsely included in $MB_{\mathbf{M}}(T)$. The second problem with inducing $MB_{\mathbf{M}}(T)$ is the existence of hidden variables $\mathcal{H}$. The induced networks regard the marginal distribution over variables in $\mathcal{O}$. In Figure 1(c) an example is shown, where $\mathcal{H} = \{H1, H2\}$ and the dashed edges appear in the network capturing the marginal over $\mathcal{O}$. True causal parents and spouses (A and S) belong in $MB_{\mathbf{M}}(T)$ even when they are manipulated, but confounded parents and spouses (B and Q) should be removed when manipulated. In Section 2.5 we present newly developed methods to address this issue.

For this type of task, our general strategy was to first learn $MB_{\emptyset}(T)$, then orient the edges in that neighborhood to identify a candidate $MB_{\mathbf{M}}(T)$; subsequently, evidence about possible confounding is obtained to further remove variables if necessary (details are described in Section 2.5). Finally, a predictive model using only the variables in the estimated $MB_{\mathbf{M}}(T)$ was learned.

1.1.3 Predictions under unknown manipulations

For these tasks, the set $\mathbf{M}$ of manipulated variables is unknown. The only nodes that always belong in $MB_{\mathbf{M}}(T)$ for any $\mathbf{M} \subseteq \mathcal{O}$ are the parents of T. Thus, the safest bet for avoiding to include irrelevant or even misleading variables (depending on the sort of manipulations) in predicting T is to build a model $P_{\emptyset}(T|Pa(T))$, where $Pa(T)$ are the (non-confounded) parents (direct causes) of T.

2. General Steps of the Strategy

In order to identify the Markov Blankets to build the predictive models, several different algorithms were used in our procedure. Figure 2 summarizes the general approach followed while the subsequent sections (noted in the figure) describe the process in more detail. The first step in our strategy is to identify the $MB_{\emptyset}(T)$. If there are no manipulations in the test set distribution, an SVM model is constructed using the variables in $MB_{\emptyset}(T)$ (Section 1.1.1). If there are manipulations, a set of additional steps are taken to orient the edges in $MB_{\emptyset}(T)$ and identify non-confounded edges. Combining all this information, a set of variables is selected, either $MB_{\mathbf{M}}(T)$ or the non-confounded parents of T, depending on whether the manipulations are known or not, respectively (Section 1.1.2 and Section 1.1.3).

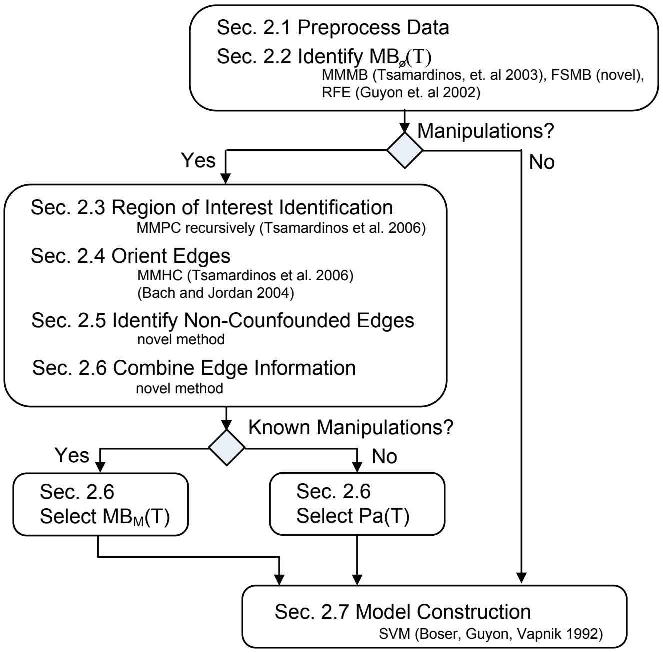

Sec. 2.1 Preprocess Data
Sec. 2.2 Identify MB∅(T)
MMMB (Tsamardinos, et. al 2003), FSMB (novel), RFE (Guyon et. al 2002)
Manipulations?
Yes
No
Sec. 2.3 Region of Interest Identification
MMPC recursively (Tsamardinos et al. 2006)
Sec. 2.4 Orient Edges
MMHC (Tsamardinos et al. 2006)
(Bach and Jordan 2004)
Sec. 2.5 Identify Non-Counfounded Edges
novel method
Sec. 2.6 Combine Edge Information
novel method
Known Manipulations?
Yes
No
Sec. 2.6 Select MBM(T)
Sec. 2.6 Select Pa(T)
Sec. 2.7 Model Construction
SVM (Boser, Guyon, Vapnik 1992)

Figure 2: Diagram illustrating the general steps of our method including the individual algorithms used.

The final set of variables is again used to construct an SVM model for predicting the cases in the manipulated test set.

Our method is publicly available online at http://www.dsl-lab.org. In order to fully automate the procedure, the released code has been modified from that used during the challenge. Wherever a difference between the competition and the released code exists, we note it in the text. The code implementing the high-level strategy is released, although some of the employed algorithms are only available as executable Matlab p-files.

2.1 Preprocessing

The data sets used in the challenge represented real world problems that required preprocessing which was tailored for each data set. For the REGED data set each variable was normalized so its mean was zero and standard deviation was one. For the SIDO data set, the variables were binary and no preprocessing was performed. For the CINA data set, variables that were not binary were treated as continuous and normalized as above; binary variables were all set to values of zero and one. For the MARTI data set, the calibrant variables were used as an indication of the position-dependent noise on the chip. For each training example, we fitted a 2D cubic spline to the values of the calibrants and then used the spline to obtain the correlated noise level at the chip location of each variable. The estimated noise was then subtracted from the value of each variable for that training sample.

2.2 Identifying $MB_{\emptyset}(T)$

Once the initial data sets have been preprocessed, the next step of our procedure was to identify the $MB_{\emptyset}(T)$. Algorithms such as HITON (Aliferis et al., 2003a) and MMMB (Tsamardinos et al., 2003a) rely on statistical tests of conditional independence. A basic assumption of these and similar methods is that if a variable is a neighbor of the target, then it will have a detectable pairwise association with the target. The general case of this assumption is that the Faithfulness Condition (Spirtes et al., 2000) holds in the causal network. However, there were no such guarantees in the problems of the competition. Thus, there could exist strong multivariate associations with the target (e.g., parity functions) whose participating variables have no detectable pairwise association with T. To address this problem we use our newly proposed algorithm called Feature Space Markov Blanket, FSMB (Brown and Tsamardinos, 2008).

2.2.1 Feature Space Markov Blanket (FSMB)

FSMB explicitly constructs a set of features, namely all the products among the variables up to a given degree d. For two variables and $d = 2$, these are V_1, V_2, V_1^2, V_2^2 and V_1V_2. It then runs HITON to find the Markov Blanket of T in this feature space. While straight-forward, this strategy does not scale up to data sets of practical sizes. A key idea in FSMB is to first learn an SVM model using a polynomial kernel that implicitly maps to this feature space consisting of all possible monomials up to a given degree d. We expect that if a feature is given a small absolute weight by the SVM, then it probably has a small association with T and there is no need to compute it and feed it to HITON. FSMB is enriched with a heuristic search to efficiently construct only the top-weighted features of the SVM model, before passing them to HITON.

This heuristic search procedure is now presented in more detail. The following standard SVM notation is used in this section; let v_k denote the predictor vector k in the data and $t_k \in \{-1, 1\}$ denote its class. Assume the use of a trained soft-margin, 1-norm SVM with full polynomial (heterogeneous) kernel $K(v_k, v_j) = \Phi(v_k) \cdot \Phi(v_j) = (v_k \cdot v_j + 1)^d$, where d is the degree of the kernel and the Lagrange multiplier vector is denoted a. The SVM model is stored as the Lagrange multipliers and support vectors, rather than explicitly constructing the feature and weight vectors of the decision function due to the large number of possible features.

In order to identify the top weighted-features without explicitly reconstructing the entire weight vector, bounds on the weights are found and updated through the search and feature construction process. Let $s_{i,j}$ be the sum of squares of the weights of all features (monomials in polynomial-kernel feature space) that involve variable i and are exactly of degree j. Then, similarly to the corresponding result for the Recursive Feature Elimination (Guyon et al., 2002) we can show that:

$$s_{i,j} = \binom{d}{j} \sum_{k=1,l=1}^{n} a_k a_l t_k t_l (H(v_k, v_l) - H(v_k^{\setminus i}, v_l^{\setminus i}))$$

where $v_k^{\setminus i}$ denotes vector v_k with the i component removed and $H(v_k, v_l) = (v_k \cdot v_l)^j$. Notice that $s_{i,j}$ is a bound on the square of the largest weight of any feature that can be constructed with variable i having degree exactly j.

Let us call this bound $b_{i,j}$ and initially set it to $s_{i,j}$. We use this bound to heuristically select some features Φ_q, for an indexing q of all features, to explicitly construct and calculate the corresponding weight w_q. We expect that the features with the largest weights probably increase the corresponding $b_{i,j}$'s to which they contribute. So, we select the degree l of monomials exhibiting the largest bound $l = \mathrm{argmax}_j b_{i,j}$ and the variables V_i in that level with the largest bounds $b_{i,l}$. For example, let us assume that $l = 2$ and the variables V_1 and V_2 have the largest bounds $b_{1,2}$ and $b_{2,2}$. Then, we explicitly construct the features V_1^2, V_1V_2 and V_2^2 and calculate their corresponding weights using the formula

$$w_q = \sum_{k=1}^{n} a_k t_k \Phi_q(v_k).$$

For example, if we denote with $v_{r,z}$ the value of the r-th training example for variable z, then the weight corresponding to constructed feature V_1V_2 equals $\sum_{k=1}^{n} \sqrt{2} a_k t_k v_{k,1} v_{k,2}$. The weight w_q of each explicitly constructed feature is then subtracted from the corresponding bounds: $b_{i,j} = b_{i,j} - w_q^2$. Thus, $b_{i,j}$ always maintains the sum of the squared weights of the remaining features, not yet constructed, involving variable i of degree exactly j. A stopping criterion can determine when the bound on the remaining weights is small enough to stop the explicit calculation of the weights. Preliminary experiments showing the time-efficiency and quality of the algorithm are presented in Brown and Tsamardinos (2008).

2.2.2 Implementation of Identifying $MB_\emptyset(T)$

The MMMB algorithm (using the χ^2 test for conditional independence based on the G^2 statistic for discrete data and Fisher's z-test for continuous or mixed data) was employed to obtain a first approximation of the Markov Blanket (Tsamardinos et al., 2003a).

To estimate how good of an approximation we obtained, we employed other feature selection algorithms and constructed models using all feature sets output (see Section 2.7 for details on our procedure of building and evaluating the models). Specifically, we build models using as variable sets the output of MMMB, FSMB, RFE (Guyon et al., 2002, run using the same kernel parameters as FSMB) and all variables. If all sets exhibited similar predictive cross-validated performance (judged manually), we accepted MMMB's output as a good approximation of $MB_\emptyset(T)$. Otherwise the better performance of RFE or FSMB, indicates important variables were missed and checked the output of FSMB for additional variables participating in strong multivariate associations. If that was the case, the interaction terms and constructed features were added as part of our Markov Blanket for all subsequent steps to use[1].

At this point, we considered that we have obtained a $MB_\emptyset(T)$ that could be used for optimal prediction under no manipulation, and is a superset of the Causal Markov Blanket in any manipulated distribution (plus false positives depending on the type of manipulations).

2.3 Reducing the Size of the Problem to a Region of Interest

The previous step identifies the participants in the $MB_\emptyset(T)$. However, the methods employed do not indicate which variables are parents and which are children, i.e., the orienta-

[1] In the released code, FSMB's constructed features are always included in the Markov Blanket, if they contain variables not participating in the output of MMMB.

tion of the edges in the $G_\emptyset$. This is necessary to be able to filter out the manipulated children and their parents and obtain $MB_{\mathbf{M}}(T)$. Unfortunately, many state-of-the-art methods for orientation are unable to run on problems of the size of the tasks in the competition.

To overcome the efficiency problem, we attempted to reduce the size of the problems by identifying the variables *at most three edges away from* T in $G_\emptyset$. Therefore, rather than learn the entire global network, we focus on a smaller region engulfing the target variable. This type of learning became possible with the invention of local causal structure-learning methods such as Grow-Shrink (Margaritis and Thrun, 1999) and MMPC, where MMPC returns the parents and children of T in a network $G_\emptyset$ (Tsamardinos et al., 2003a). The idea of learning regions (subgraphs) of arbitrary size was first presented in Tsamardinos et al. (2003b). The variables in the region are identified through recursive application of a local neighborhood identification method (MMPC using the default parameter settings, Fisher's z test and χ^2 test on continuous and discrete data respectively) in a breadth-first search then, the graph is oriented as described in Section 2.4.

Restricting our attention to a region may reduce the number of edges that can be oriented. That is, it is possible for remote parts of the network to lead to orientation of edges close to or involving T. Preliminary experiments we have conducted however (publication under preparation), indicate that in many typical networks this effect is not severe and the edges in the region can be oriented as well as when using the full network. The idea of reconstructing a region of interest of limited depth around T to help orient the Markov Blanket edges has also appeared in Bai et al. (2008).

The choice of a region of depth three is explained thusly; implicitly (in search-and-score methods) or explicitly (constraint-based methods) v-structures are crucial in orientation. A v-structure occurs when the subgraph $X \rightarrow T \leftarrow Z$ exist in the true unknown graph but the edge $X - Z$ is not. To determine from data that $X - Z$ is absent we need to make sure that we have conditioned on a subset of their parents. Thus, to identify a v-structure $X \rightarrow T \leftarrow Z$ we need the parents of X and Z that are two edges away from T. The method we present in Section 2.2 requires v-structures among the parents of T, thus forcing us to induce a region of depth three.

2.4 Identifying and Orienting Edges

In this step, we run standard Bayesian Network learning algorithms on the data projected on the variables of the restricted region found in the previous step. For the case of binary data, MMHC with the default parameter settings and a χ^2 test was employed to find a high scoring network; in extensive experimentation MMHC was deemed one of the best such learning algorithms (Tsamardinos et al., 2006). For the case of continuous or mixed data, the kernel generalized variance scoring metric of Bach and Jordan (2002), with $\kappa = 0.01$ and $\sigma = 1$, was used with a greedy hill-climbing search to learn the structure. In Bach and Jordan (2002), the variable distribution is assumed Gaussian in feature space, mapped implicitly by a kernel function. This method is able to work on combinations of discrete and continuous variables and performed well compared to other algorithms and approaches targeting continuous or mixed data as shown in Fu (2005). The final structures were converted to their corresponding PDAGs with the compelled edges identified. A compelled

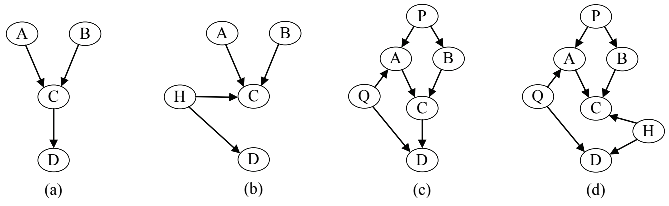


Figure 3: Four example networks to explain the Y-structure analysis.

edge $X \rightarrow T$ provides evidence (under the Faithfulness Condition) that either X causes T, or (inclusive) X and T are confounded by a hidden variable.

2.5 Dealing with Confounded Variables

To deal with hidden variables and identify confounded parents of T, or confounded spouses of T we first tried the FCI algorithm (Spirtes et al., 2000). Unfortunately, FCI could not scale up even to the reduced region found (FCI was run with version 4.3.9 of the Tetrad Project). It also failed to run even when we input several constraints to make it more efficient and specifically, to constrain the edges to the ones found by the previous step.

We then turned to the method of Mani et al. (2006) to identify a Y-structure involving a quadruple of the variables; see Figure 3(a) for such a structure. If a Y-structure faithfully captures the marginal of the four variables, *then edge $C \rightarrow D$ has to be causal*, i.e., there can be no hidden confounder of C and D, as shown in Figure 3(b). If Figure 3(b) was the case, A and D would be dependent given C and so their marginal would not faithful to Figure 3(a). There is no causal claim for the other two edges in the graph.

We found this idea interesting but did not apply the algorithm as given by Mani et al. (2006) because the conditions to identify such a structure are restrictive (e.g., A and B need to be unconditionally independent). Instead, we extended the general idea to identify causal edges in more general settings, where the pairs A and B, or A and D may be conditionally independent instead of unconditionally, such as in Figure 3(c) (this is mentioned as future work in Mani et al. 2006). We proved (proof omitted for scope) and implemented a test based on the following proposition:

Proposition 2 *Let $\mathcal{V} = \mathcal{O} \cup \mathcal{H}$ be a set of variables, $\mathcal{O} \cap \mathcal{H} = \emptyset$; $P(\mathcal{V})$ is faithful to a CBN $\langle G, P \rangle$ and $I(X;Y|\mathbf{Z})$ denotes independence of X and Y given the conditioning set $\mathbf{Z}$ and $\neg I(X;Y|\mathbf{Z})$ denotes dependence. For the distinct variables $A, B, C, D \in \mathcal{O}$ when the following conditions hold:*

1. $\forall \mathbf{S} \subseteq \mathcal{O}, \neg I(A;C|\mathbf{S})$ *4.* $\exists \mathbf{Z}_1 \subseteq \mathcal{O}, I(A;B|\mathbf{Z}_1)$
2. $\forall \mathbf{S} \subseteq \mathcal{O}, \neg I(B;C|\mathbf{S})$ *5.* $\neg I(A;B|\mathbf{Z}_1 \cup \{C\})$
3. $\forall \mathbf{S} \subseteq \mathcal{O}, \neg I(D;C|\mathbf{S})$ *6.* $\exists \mathbf{Z}_2 \subseteq \mathcal{O}, I(A;D|\mathbf{Z}_2)$ *and* $C \in \mathbf{Z}_2$

then, there is a causal path $C \rightarrow \ldots \rightarrow D$ in G, where the intermediate variables belong in $\mathcal{H}$ (are hidden).

We call this set of conditions collectively the *Y-test for the variables A, B, C, and D*. In our implementation, we apply the Y-test for every quadruple of distinct variables A, B, C, D in the region of interest around T[2]. If all conditions (1) - (6) are satisfied then we considered the edge $C \rightarrow D$ as causal and without possible confounding. We applied the Y-test only once per quadruple of variables and reused cached results for improved efficiency as follows: If an edge $A - C$ (ignoring the direction) exists in the region of interest then $\forall \mathbf{S} \subseteq \mathcal{O}, \neg I(A; C|\mathbf{S})$, or MMPC would have discovered a *d*-separating set for A and C. Thus, condition (1) of the proposition holds. Similarly, if the edges $B - C$ and $C - D$ exist in the region of interest, the quadruple passes the first three conditions. If the edges $A - B$ and $A - D$ are not in the region of interest, it implies that MMPC has discovered subsets $\mathbf{Z}_1$ and $\mathbf{Z}_2$ that *d*-separate the two pairs of variables respectively: condition (4) and the first part of (6) also hold. Condition (5) is checked with an additional test of independence, using the specific $\mathbf{Z}_1$ found by MMPC when removing the edge $A - B$. Finally, it is checked whether $C \in \mathbf{Z}_2$, the subset found by MMPC when removing the edge $A - D$.

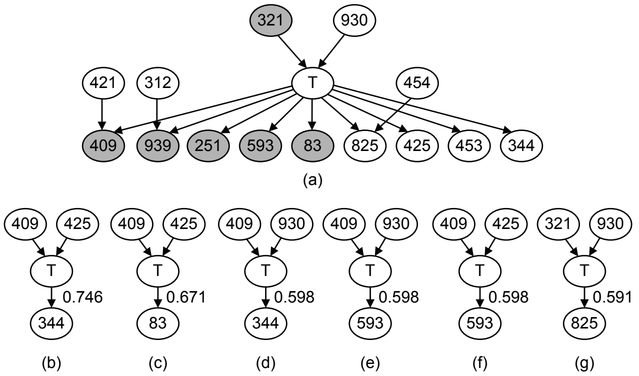
Multiple applications of the Y-test for different quadruple of variables may provide conflicting information for an edge $C \rightarrow D$. We devised two weighting schemes to rank the strength of evidence a single Y-test provides. First, a value was calculated as the minimum *p*-value returned by the independence tests of conditions (4) and (6). Let this value be referred to as the *p-score* of the Y-test. This value represents the closest the independence conditions (4) and (6) were to failing to pass the threshold for accepting dependence. Second, a ratio of the BDeu score of the Y-structure (including the nodes in the conditioning sets) to the BDeu score of an empty DAG was assessed. In preliminary tests on known networks, the BDeu score metric was not consistently informative; therefore, the p-score was used in further analysis.

2.6 Combining Information to Identify $MB_{\mathbf{M}}(T)$

We used the PDAG at the end of Section 2.4 to obtain the orientation of some edges and the method of Section 2.5 to obtain both orientation and causal evidence for some edges, i.e,. that they are non-confounded. The information from these two sources may be incomplete (some edges are not oriented or could appear due to possible confounding phenomena) and conflicting. This information was combined manually and subjectively during the competition; however, for testing purposes during the post-challenge analysis and to be able to release a fully automated algorithm, we have replaced the manual step with an automated method. The latter attempts to follow as close as possible our thought process during the challenge.

We present the method following an example using the REGED1 data set. Figure 4 illustrates and summarizes the different information sources. Figure 4(a) shows the Markov Blanket variables extracted from the PDAG of Section 2.4. The shaded nodes indicate the manipulated variables in REGED1. In addition, all possible Y-structures involving edges of the Markov Blanket were identified and scored. Figures 4(b)-(g) show the top six Y-structures centered on the target node ranked by the maximum p-score. Finally, the table in 4(h) lists for each variable the number of times it is determined to be a child of T and the maximum p-score among those instances. There were no Y-structures (A, B, X, T) that

[2] In our actual implementation the symmetrical test for B, $\exists \mathbf{Z}_3 \subseteq \mathcal{O}, I(B; D|\mathbf{Z}_3)$ and $C \in \mathbf{Z}_3$ is also checked, although theoretically not necessary.



	Num.	Max
Node	Child	P-score
930	3	0.148
321	4	0.185
409	3	0.185
939	7	0.185
251	8	0.185
825	7	0.591
593	11	0.598
425	6	0.656
453	12	0.656
83	13	0.671
344	11	0.764

(h)

Figure 4: Information available to determine $MB_{\mathbf{M}}(T)$ for REGED: (a) the DAG involving the MB variables determined by the search-and-score procedure (variables manipulated in REGED1 are shaded), (b)-(g) the top valid Y-tests ranked by p-score, and (h) a table of the variables from (a) considered to be either parents or children along with the number of valid Y-tests where the node appears as a child of T and the top p-score when this occurs.

passed the Y-test with an edge $X \rightarrow T$ where $X \in MB_{\emptyset}(T)$; therefore, the Y-tests alone did not give any strong evidence for a variable to be a parent of the target.

We now describe how to identify the parents of T. We consider as possible parents all variables returned by FSMB as neighbors of T. First, we identify the variables with strong evidence of being parents of T. These are the ones that appear as parents in the PDAG of the edge orientation phase of Section 2.4. We sort them by the number of times they appear as non-confounded parents of T in Y-tests. In our example, these are variables with indexes $\{930, 321\}$ (Figure 4(a)). Then, we filter out the variables with strong indication that they are indeed children of T; these are variables X for which the edge $T \rightarrow X$ gets a high p-score in some Y-test, i.e., they have maximum p-score above a threshold (arbitrarily set to 0.5). In our example, these are variables $\{825, 593, 425, 453, 83, 344\}$ (Figure 4(h)). The remaining variables $\{409, 939, 251\}$ are those without strong evidence that they are either parents or children. These are sorted in decreasing order of the ratio of valid Y-tests as a parent to that as a child; ties are broken with preference to variables appearing less often as children of T in Y-tests. The final list to consider thus is $\{930, 321, 409, 939, 251\}$. During the competition, several subsets of this list were tried and a final decision was made among those submissions that ranked in the top 25% of all competitors. The automated procedure simply uses a threshold on the number of times the variables appear as children of T to remove the tail of the list.

If the complete $MB(T)$ is sought and not just the parents of T, we also need to identify the children and spouses of T. As children we consider the remaining non-manipulated variables adjacent to the target; in our example, these are variables with indexes $\{825, 425, 453, 344\}$. The spouses of the selected children are found from the PDAGs orientation: $\{454\}$ (alternatively, we could have used the same procedure for the identification of the parents of T as above, to identify the parents of the children of T).

Table 1: Results on Challenge Data Sets: The Fnum, Fscore, Dscore, Tscore and Ranking is given for each version of the data sets; the results in (a) represent the final challenge submission and (b) show the results if the $MB_{\emptyset}(T)$ is used for every variable list regardless of considering manipulations. In (a), the number of entries, the overall ranking and average Tscore are given for each data problem. The cells are shaded in the colored quartile information: green - best 25%, yellow - best 50%, orange - worst 50%, and red worst 25%.

	Final Challenge Submission							Unmanipulated MB used for all Data Sets				
	Fnum	Fscore	Dscore	Tscore	Ranking			Fnum	Fscore	Dscore	Tscore	Ranking
CINA0	101	0.8496	0.9717	0.9721	9	Num. Entries / Total	7 / 277	101	0.8496	0.9717	0.9721	9
CINA1	5	0.4716	0.9316	0.5113	23	Average Tscore	0.6015	101	0.5795	0.9717	0.8581	4
CINA2	5	0.4716	0.9316	0.3210	25	Overall Ranking	23 / 25	101	0.5795	0.9717	0.6917	8
MARTI0	24	0.5869	0.9952	0.9681	8	Num. Entries / Total	2 / 233	24	0.5869	0.9948	0.9824	7
MARTI1	17	0.5643	0.9951	0.7837	9	Average Tscore	0.8083	24	0.5985	0.9948	0.8477	9
MARTI2	3	0.4985	0.6973	0.6730	10	Overall Ranking	9 / 19	24	0.7429	0.9948	0.6971	9
REGED0	15	0.8571	1.0000	0.9998	2	Num. Entries / Total	5 / 355	15	0.8571	1.0000	0.9998	2
REGED1	9	0.7851	1.0000	0.9673	4	Average Tscore	0.9423	15	0.7825	1.0000	0.9280	14
REGED2	3	1.0000	0.9728	0.8600	1	Overall Ranking	1 / 30	15	1.0000	1.0000	0.7231	9
SIDO0	13	0.5115	0.9356	0.9230	12	Num. Entries / Total	2 / 242	13	0.5015	0.9365	0.9237	12
SIDO1	4	0.5003	0.8587	0.6073	12	Average Tscore	0.6909	13	0.5012	0.9365	0.6626	11
SIDO2	4	0.5003	0.8587	0.5426	14	Overall Ranking	12 / 28	13	0.5012	0.9365	0.5713	11

(a) (b)

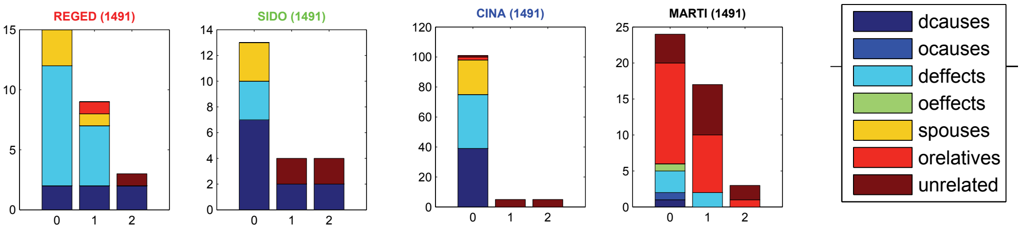
In our effort to automate the above procedure after the challenge, we noticed that the procedure was not stable. Specifically, the lists of variables output and the corresponding models produced, varied significantly under different ordering of the variables in the data set. To alleviate the problem we augmented the procedure with a model-averaging-type step where we run the orientation procedure several times with different parameters (namely, we vary the equivalent sample size in the Bayesian Score and the kernel parameters for the scoring metric of Bach and Jordan, 2002). Only the variables that appear consistently across parameter combinations remain in consideration. The procedure has been validated in the post-challenge tests set by the organizers and was found stable and robust under permutations of the variables and subsampling of the data.

2.7 Building Predictive Models

Once the variable list was determined for each data set, a final classification SVM model was trained on only the variable list members (Boser et al., 1992). An n-fold cross-validation design was used to select the optimal parameters: type of kernel (polynomial or Gaussian), kernel parameters (degree of kernel $\in \{1, 2, 3, 4\}$ or sigma $\in \{10^{-4}, 10^{-3}, \ldots, 10^{0}\}$), and C value $\in \{10^{-4}, 10^{-3}, \ldots, 10^{1}\}$. The value of n ranged from 5 to 10 based on the sample size available in the training sample. Once the best parameters were selected, a final SVM model was trained and used to predict the values for the test data sets.

3. Results

The classification performance (AUC reported as Tscore in the challenge results) is ultimately how the challenge submissions were rated. Table 1(a) presents the Fnum, Fscore, Dscore, Tscore, and ranking of our final submission for each data set version. The number of entries before the final submission, the average Tscore (across the versions of a data set), and the overall ranking (generated from the average Tscore) are also shown in the table.



(a) (b) (c) (d)

Figure 5: The selected features' relationship to the target variable, where dcauses = direct causes, deffects = direct effects, ocauses = other causes (indirect), oeffects = other effects (indirect), spouses = parent of direct effect, orelatives = other relatives, and unrelated = completely irrelevant.

3.1 What Went Well

The specific implementation of our strategy performs well on the REGED data set achieving the top overall ranking. The strategy also exhibits decent performance on the unmanipulated data sets, version "0". This indicates that our implementation is approximating $MB_{\emptyset}(T)$ well. This is corroborated by the organizers' post-challenge analysis, shown in Figure 5. In 3 of the 4 data sets (REGED, SIDO, and CINA) the method is performing well at identifying members of $MB_{\emptyset}(T)$. In fact, in those three data sets only ~2 false positives are added to the Markov Blanket (the number of false negatives is undisclosed). Notice that our algorithms were able to accurately identify CINA's $MB_{\emptyset}(T)$ numbering close to 100 variables. On MARTI it seems that $MB_{\emptyset}(T)$ was not accurately found, however we believe this is due to our inability to handle the noise correctly. Evidence to this is provided by the following experiment: the post-challenge analysis included other teams' preprocessed data for MARTI; re-running our method on the preprocessed data provided by Dr. Guyon we see a marked improvement in our performance (in particular on the MARTI0 data, where our method has proven to do well in all other cases). Specifically, the Tscore on MARTI0 improves from 0.9681 to a score of 0.9910 resulting in an improved ranking on that data set from eighth to fifth and corroborating that we approximate well the $MB_{\emptyset}(T)$ (the actual false positives and false negatives have not been released for post-challenge submissions).

3.2 What Went Wrong

While our methods performed well at identifying the unmanipulated Markov Blanket, the identification of the manipulated Markov Blanket was very poor on all but the REGED data set. This indicates that our methods for orienting the edges of $MB_{\emptyset}(T)$ performed poorly. We now provide some possible explanations.

Unfortunately, we spent most our time on the REGED data sets and the development of new methods, leaving little time for the rest of the data sets. Most importantly, we set out to solve a more difficult problem than what the organizers had set, namely inducing causality in the presence of hidden variables and violations of faithfulness. These are two important issues in real data sets, but did not occur in the challenge: FSMB identified between 0-4 features per data set that were added for consideration; these features were often considered spouses, or other relatives when selecting $MB_{\mathbf{M}}(T)$ and did not make much difference in performance. Also, there were actually no hidden variables in the challenge data sets. More specifically, all the variables participating in the models from which data were simulated, were also included in the released data sets. Because of the way data were simulated, the problematic confounding effect we described never occurred. We spent a significant amount of time on this problem is because the FAQ of the competition specifically declared that there may be missing variables (a problem for many real-world analyses).

Also, our submissions were overly conservative in regards to including false positive variables, i.e, variables not in $MB_{\mathbf{M}}(T)$. However, it turns out that for this challenge, false negatives degrade performance significantly more than false positives (also see discussion in the organizers' post-challenge analysis online Appendix B, Challenge Website 2008). This is exemplified by the following post-challenge experiment: we submitted a new set of entries where the variable list for each data set version was the $MB_{\emptyset}(T)$, a superset of $MB_{\mathbf{M}}(T)$. The results for these submissions are shown in Table 1(b) and can be contrasted with the challenge results in 1(a). On REGED, the performance is degraded since we were already ranking 1st on this task. On CINA, the challenge submission choice of $MB_{\mathbf{M}}(T)$ was both incorrect and very conservative, especially in light of the large size of the Markov Blanket and number of possible parents. The use of $MB_{\emptyset}(T)$ improved the performance and these results rank as high as fourth for CINA1. For MARTI and SIDO, the new submission returns a similar or slightly better ranking to that of the challenge submission. This analysis, while only over the limited data sets of this challenge, suggests that without an edge orientation procedure to supply correct information to differentiate the parents and children, letting $MB_{\emptyset}(T)$ be the default manipulated Markov Blanket is a reasonable approach. In addition, we believe that a model averaging approach would also greatly improve the robustness of identifying the $MB_{\mathbf{M}}(T)$ and make it more resilient to edge-orientation errors.

Regarding the CINA data sets, we note that they consisted of a mixture of discrete and continuous variables. Many of the algorithms employed by our strategy heavily rely on tests of independence. Our implementations of these tests however, have been developed targeting only all discrete or all continuous variables and were not designed for mixed types of variables. Regarding the SIDO data sets, we were informed after the completion of the challenge that it contained variables created by the binarization of other variables. For example a variable V taking values $v_1, \ldots, v_k$ is converted to the binary variables $B_1, \ldots, B_k$ taking values $B_i = I(V = v_i)$, where I is the indicator function. The newly created variables B_i are all inter-dependent, since knowing $B_i = 1$ implies that $B_j = 0$, for $i \neq j$. Graphically, the new set of variables $\{B_i\}$ would consist of a *clique in the PDAG of a network*. If V is a parent of T in the original network, then all B_i's are connected to T and among each other. This reduces the identifiable Y-structures by our procedure and confuses all traditional search-and-score Bayesian Network learning algorithms. The problem stemming from binarization of variables points to an interesting future research direction.

Finally, due to the time pressure, several parts of our strategy were not fully optimized. We did not optimize the model construction procedure and just used standard SVMs with cross-validation. Most importantly, we did not have the time to fully test and optimize the novel algorithms and procedures for these tasks.

4. Lessons Learned and Conclusions

The most important outcome of our participation to the challenge is the experience gained and realization of several theoretical and practical issues as well as ideas that emerged for future directions in the field. We now distill some of these in the following.

Knowledge of the causal structure is theoretically necessary for making optimal predictions under manipulations. This is exemplified, in our opinion, in this challenge by the difference between the top non-causal submissions and the theoretical optimum performance; see the organizers' post-challenge analysis online (Challenge Website, 2008, Figures 3-6). Regarding the state-of-the-art in causal discovery, we believe there exists efficient, scalable, and publicly available code to learn the Markov Blanket. In fact, several other top participants also used our package Causal Explorer (Aliferis et al., 2003b) implementing such algorithms. These methods perform well on a range of high-dimensional data sets involving discrete, continuous, and mixed data. However, we also note that there is a shortage of reliable and efficient, publicly-available code or software packages that are meant to identify hidden variables or non-confounded variables. Of those available (e.g., Tetrad's FCI implementation), they are unable to scale to the size of the challenge problems (even when reduced to a region of depth 3). In addition, we observe that the state-of-the-art methods employed to learn the orientation did not perform well. Consequently, we were unable to reliably identify the manipulated Markov Blanket.

Regarding important implementation issues, we note that reducing the size of the problem to a region of depth 3 greatly improved the efficiency of the later applied methods; this reduction allowed the orientation procedures to complete in minutes rather than hours or days if the full variable set was considered. Several algorithms heavily depend on statistical tests that ought to be tailored for the problem at hand. Binarized variables pose a problem to causal-discovery methods at the moment.

In summary, we presented a general strategy for predicting a quantity under manipulations of a system. It relies on identifying $MB_{\mathbf{M}}(T)$ and fitting a model for $P_{\emptyset}(T|MB_{\mathbf{M}}(T))$ from the observational data. The steps of the strategy are shown in Figure 2. They are implemented by existing algorithms and augmented with novel procedures for detecting certain kinds of violations of faithfulness and for detecting non-confounded causal edges. Overall, this challenge provided us with an opportunity to develop, apply, and compare methods for causal discovery on realistic, challenging problems and initiating new avenues of research.

References

C.F. Aliferis, I. Tsamardinos, and A. Statnikov. HITON, A Novel Markov Blanket Algorithm for Optimal Variable Selection. In *Proceedings of the American Medical Informatics Association Conference(AMIA)*, pages 21–25, 2003a.

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

F.R. Bach and M.I. Jordan. Learning Graphical Models with Mercer Kernels. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS-02)*, pages 1009–1016, 2002.

X. Bai, R. Padman, J. Ramsey, and P. Spirtes. Tabu Search-Enhanced Graphical Models for Classification in High Dimensions. *INFORMS JOURNAL ON COMPUTING*, 20(3):423–437, Oct. 2008.

B. Boser, I. Guyon, and V. Vapnik. An Training Algorithm for Optimal Margin Classifiers. In *Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

L.E. Brown and I. Tsamardinos. Markov Blanket-Based Variable Selection in Feature Space. Technical Report TR-08-XX, Vanderbilt Univeristy, 2008.

Challenge Website. Causation and prediction challenge.
http://clopinet.com/isabelle/Projects/WCCI2008/Analysis.html, 2008.

L.D. Fu. A Comparison of State-of-the-Art Algorithms for Learning Bayesian Network Structure from Continuous Data. Master's thesis, Vanderbilt University, 2005.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 46(1-3):389–422, 2002.

S. Mani, P. Spirtes, and G.F. Cooper. A Theoretical Study of Y structures for Causal Discovery. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 314–323, 2006.

D. Margaritis and S. Thrun. Bayesian network induction via local neighborhoods. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS-99)*, 1999.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmannn Publishers, San Mateo, CA, 1988.

J. Pearl. *Causality, Models, Reasoning, and Inference.* Cambridge University Press, Cambridge, U.K., 2000.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* MIT Press, Cambridge, MA, 2nd edition, 2000.

Tetrad Project. http://www.phil.cmu.edu/projects/tetrad/.

I. Tsamardinos and C.F. Aliferis. Towards Principled Feature Selection: Relevancy, Filters and Wrappers. In *Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

I. Tsamardinos, C.F. Aliferis, and A. Statnikov. Time and Sample Efficeint Discovery of Markov Blankets and Direct Causal Relations. In *Proceedings of Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 673–678, 2003a.

I. Tsamardinos, C.F. Aliferis, A. Statnikov, and L.E. Brown. Scaling-Up Bayesian Network Learning to Thousands of Variables Using Local Learning Techniques. Technical Report TR-03-02, Vanderbilt University, March 2003b.

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

Appendix A.

Theorem 1 *Let $\langle G_{\emptyset}, P_{\emptyset}\rangle$ be a CBN and $\langle G_{\mathbf{M}}, P_{\mathbf{M}}\rangle$ be the resulting CBN under manipulations of variables in $\mathbf{M}$. Suppose that $T \notin \mathbf{M}$ and also that there is no manipulated child C of T in $G_{\emptyset}$ with a descendant D in $G_{\emptyset}$ that is also in $MB_{\mathbf{M}}(T)$. Then,*

$$P_{\mathbf{M}}(T|MB_{\mathbf{M}}(T)) = P_{\emptyset}(T|MB_{\mathbf{M}}(T)).$$

Proof

We base the proof of the theorem on the more general theory of probability invariance under manipulations found in Spirtes et al. (2000). Let G be the original graph $G_{\emptyset}$ with the additional exogenous variable E representing the manipulating agent and edges from E to any manipulated variable in $\mathbf{M}$. All graph operations that follow in the proof are on G (in the terminology of Spirtes et al. (2000) G is the combined graph G_{comb}). Then $P_{\emptyset}(\mathbf{Y}|\mathbf{Z}) = P_{\mathbf{M}}(\mathbf{Y}|\mathbf{Z})$, if $Dsep(E, \mathbf{Y}|\mathbf{Z})$, where $\mathbf{Y}$, $\mathbf{Z}$ are two disjoint sets and $Dsep(E, \mathbf{Y}|\mathbf{Z})$ denotes the d-separation of E from $\mathbf{Y}$ given $\mathbf{Z}$ in G. Thus, we just need to show that $Dsep(T; E|MB_{\mathbf{M}}(T))$ under the conditions $\mathcal{C}$:

> There is no pair of variables C, D such that:
>
> 1. $E \rightarrow C \leftarrow T$
> 2. $C \rightsquigarrow D$
> 3. $D \in MB_{\mathbf{M}}(T)$

where $C \rightsquigarrow D$ denotes a directed path from C to D. Let us assume that the d-separation does not hold when conditions $\mathcal{C}$ do, and reach a contradiction. Recall that there are no incoming edges to E since it is an exogenous variable and no edge from E to T.

Since the d-separation does not hold, there must be an open path from E to T that is not blocked by $MB_{\mathbf{M}}(T)$. Take a path of the form $E \rightarrow \cdots P \rightarrow T$. $P \in MB_{\mathbf{M}}(T)$ under any manipulation and so we condition on it and it blocks the path. Thus, since there is an open path, it must be of the form $E \rightarrow \cdots C \leftarrow T$. For the path to be open, for each collider on it, we must be conditioning on either the collider or a descendant of the collider. Let us now consider the last collider on the path, which can be (1) C itself, or (2) some other node G.

Case (1): The open path is of the form $E \rightarrow \cdots C \leftarrow T$ and C is the last collider on it. We also distinguish two subcases, either (1a) the path is of the form $E \rightarrow C \leftarrow T$, or (1b) of the form $E \rightarrow \cdots S \rightarrow C \leftarrow T$. If (1a) is true, since C is a collider on the open path of case (1) we must be conditioning on either itself or a descendant of it $D \in MB_{\mathbf{M}}(T)$. Since, in (1a) C is manipulated, $C \notin MB_{\mathbf{M}}(T)$ and we cannot be conditioning on C itself. Thus, there is a $D \in MB_{\mathbf{M}}(T)$, descendant of C and conditions $\mathcal{C}$ all hold reaching a contradiction.

If (1b)is true, then S cannot belong in $MB_{\mathbf{M}}(T)$ or it would block the path by conditioning on it. Thus, $S \notin MB_{\mathbf{M}}(T)$ and the only way for this to be possible is if C is manipulated and so $E \rightarrow C \leftarrow T$ holds. Similarly to case (1a) we then conclude that conditions $\mathcal{C}$ should hold, reaching a contradiction.

Case (2): The open path is of the form $E \rightarrow \cdots G \leftarrow \cdots \leftarrow C \leftarrow T$ and G is the last collider on the path. If $C \in MB_{\mathbf{M}}(T)$ then we condition on it and it blocks the path. Thus,

$C \notin MB_{\mathbf{M}}(T)$ which means C is manipulated and so $E \rightarrow C \leftarrow T$ holds. For the path to be open, given that G is a collider we must be conditioning on a node $D \in MB_{\mathbf{M}}(T)$ that is either G itself or a descendant of it. In either case, D must be a descendant of C too since there is a directed path $G \leftarrow \cdots \leftarrow C$ (notice this path cannot be of the form $G \leftarrow Q \rightarrow C$ or C and not G would be the last collider on the path $E \rightarrow \cdots G \leftarrow \cdots \leftarrow C \leftarrow T$). Thus, case (2) implies conditions $\mathcal{C}$ hold, again contrary to what we assumed. ■