JMLR: Workshop and Conference Proceedings 3: 93-105 WCCI2008 workshop on causality

Partial orientation and local structural learning of causal networks for prediction

Jianxin Yin JIANXINYIN@MATH.PKU.EDU.CN
You Zhou ZHOUYOU@PKU.EDU.CN
Changzhang Wang CHANGZHANG@PKU.EDU.CN
Ping He SUNHP@PKU.EDU.CN
Cheng Zheng ZZHENGCCHENG@PKU.EDU.CN
Zhi Geng ZGENG@MATH.PKU.EDU.CN
School of Mathematical Sciences
Peking University
Beijing 100871, China

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

Abstract

For a prediction problem of a given target feature in a large causal network under external interventions, we propose in this paper two partial orientation and local structural learning (POLSL) approaches, Local-Graph and PCD-by-PCD (where PCD denotes Parents, Children and some Descendants). The POLSL approaches are used to discover the local structure of the target and to orient edges connected to the target without discovering a global causal network. Thus they can greatly reduce computational complexity of structural learning and improve power of statistical tests. This approach is stimulated by the challenge problems proposed in IEEE World Congress on Computational Intelligence (WCCI2008) competition workshop. For the cases with and without external interventions, we select different feature sets to build prediction models. We apply the L1 penalized logistic regression model to the prediction. For the case with noise and calibrant features in microarray data, we propose a two-stage filter to correct global and local patterns of noise.

Keywords: Causal network, Local structural learning, Partial orientation.

1. Introduction

Correlations between variables are useful for prediction in the case that individuals to be predicted come from the same population as the training data. If we want to predict them after the system is manipulated by external interventions, prediction models based only on correlations may lead to awful results. For example, there is a strong correlation between a rooster's crying and sun rising. But killing the rooster cannot stop sun rising. No matter how advanced techniques and models are used based only on correlations, there may always exist some cases of external interventions which make the prediction inaccurate without causal discovery. Causal discovery is one of most important goals in various sciences, such as natural and social sciences (Pearl, 2000; Spirtes et al., 2000). In causal discovery, a key issue is to discover causes of a target feature of interest, whose main causes are generally not too many. Generally, it is difficult to discover causes and effects only from observational

data, and even harder to distinguish causes from effects. Discovering causal structures and further distinguishing causes from effects of a target are useful not only for prediction in the cases with external interventions, but also valuable for studying causal mechanisms, making decision and evaluating treatment effects.

Most of the traditional prediction approaches are based on correlations without causal discovery. For example, it is well known that for a Bayesian network a Markov blanket (MB) of a target variable is often used for prediction of the target because the target is independent of other variables conditionally on the Markov blanket. A Bayesian network is called a causal network if directed edges have causal interpretation. The causation challenge organized by Guyon et al. (Guyon et al., 2008) for IEEE WCCI2008 is to predict the effect of external interventions. When the neighbor nodes of the target in the causal network are manipulated by external interventions, we have to distinguish parent nodes (cause features) from children nodes (effect features), and then we use parent nodes (and unmanipulated children nodes if we know) to predict the target. Although there are many structural learning approaches for discovering a global network, it is well known that learning a global network is an NP-Hard problem. If we are only interested in a prediction of a target, it is inefficient and unnecessary to learn a global network.

For a prediction problem with external interventions, we propose in this paper two partial orientation and local structural learning (POLSL) approaches, Local-Graph and PCD-by-PCD (PCD means Parents, Children and some Descendants). In the POLSL approaches, we discover locally the edges connected to the target and only try to orient these edges so that we can distinguish the parents from the children of the target. We can theoretically show that the approaches can correctly obtain the edges connected to the target and their orientations. The POLSL approaches can greatly reduce computational complexity of structural learning, and their statistical test is more powerful than a global learning approach. After we select a subset of all variables according to the local structure, we use the L1 penalized logistic regression model to fit the prediction model and use the estimated conditional probability of the target variable for each individual in the test set for its classification. The L1 penalized approach is a shrinkage method which can reduce mean squared error (MSE) of prediction.

In Section 2, we describe the preprocessing and we propose a two-stage filter. In Section 3, we propose two POLSL algorithms and theoretically show their correctness. In Section 4, we use the L1 penalized logistic regression model to fit the prediction model. In Section 5, we show results of simulation and the causal challenge. Advantages of our approaches are discussed in Section 6. Details of the preprocessing are described in Appendix A, and the proofs of theorems are presented in Appendix B.

2. Preprocessing

In this section, we propose a two-stage process for filtering noise in microarray data, and we use a feature screen method to remove unnecessary features for the prediction.

2.1 A two-stage filter

For the case of observed data with noise and calibrant features (e.g., MARTI), we first centralize observations and then filter noise using a two-stage process. At the first stage,

we correct the global noise pattern. Then we treat every micro-array data separately to get a smoother output in the second stage. More details on this two-stage filter can be found in Appendix A.

After we build the model for prediction with the corrected training data, given a new micro-array with noise for predicting its target feature, we first correct it with the global regression models obtained at the first stage, then filter the noise of every feature with the local models obtained at the second stage, and finally predict its target based on the corrected features $\{\hat{r}_0^{(i)}, i = 1, \ldots, 999\}$ and a prediction model discussed in Section 4.

2.2 Feature screen and discretization

For a data set with very high dimensional space (e.g., 4932 features for SIDO), we first screen features using a sure independence screening (SIS) procedure (Fan and Lv, 2008) to reduce the dimensionality to a tractable size (e.g., 1000 features for SIDO). The SIS method is a screening method based on correlation learning which has the property that all the important variables survive after variable screening with probability tending to one.

This screen step is not necessary for other data sets, and even for a higher dimensional data set if CPU time for the following computations is not a problem.

For continuous variables, we suppose that they have a normal distribution, or we first discretize them using the supervised discretization process in the causal explorer (Aliferis et al., 2003), and suppose that the discretized variables have a multinomial distribution.

3. Partial orientation and local structural learning

After finding a Markov blanket of a target, we can obtain edges connected to a target of interest, but it is not sufficient to orient the edges connected to the target using only the variables in the Markov blanket. In this section we propose two approaches for local structural learning and partial orientation of the edges connected to the target. Let $PC(X)$ denote a set which contains all parents and children of node X, and let $PCD(X)$ denote a set which contains $PC(X)$ and may contain some descendants of X. There are a lot of algorithms which can be used to find $PCD(X)$, such as Min Max Parents and Children (MMPC) algorithm (Tsamardinos et al., 2006).

3.1 Two Algorithms: Local-Graph and PCD-by-PCD

The first approach called Local-Graph tries to find a variable set such that all v-structures connected to a target T of interest can be discovered correctly. It first finds $PCD(T)$ and then finds $PCD(X)$ for all $X \in PCD(T)$. Let $V = \{T\} \cup PCD(T) \cup [\cup_{X \in PCD(T)} PCD(X)]$. Finally it learns a directed acyclic graph (DAG) over the node set V calling an algorithm. The recursive algorithm (Xie and Geng, 2008) was used in our algorithm Local-Graph, which recursively decomposes structural learning of a large network into local learning of several small networks.

We can show below that algorithm Local-Graph can discover all v-structures connected to the target T even if the local graph returned from Local-Graph may not be a correct subgraph of the underlying DAG.

Algorithm: Local-Graph (Data D; Target T)
(a) $V = \{T\} \cup PCD(T) \cup [\cup_{X \in PCD(T)} PCD(X)]$.
(b) Construct a DAG over V with the recursive algorithm.
(c) Return the partially oriented local structure around T.

Theorem 1 *Suppose that a causal network is faithful to a probability distribution and that independence test is correctly performed by using data. Algorithm Local-Graph can correctly discover all edges and v-structures connected to a target T of interest.* ■

In the second approach called PCD-by-PCD, we extend Algorithm Local-Graph and find PCDs sequentially. In the algorithm PCD-by-PCD, we first find $PCD(T)$ of the target T and $PCD(X)$ for feature $X \in PCD(T)$, and then we sequentially find $PCD(X)$ for a feature X which is contained in the previous PCD's. During the sequential process, we find local v-structures and try to orient the edges connected to the target T as much as possible. When all of the edges connected to the target T are oriented, we stop the process and obtain all direct causes and effects of the target T. There may be some undirected edges which cannot be oriented even after we have found the PCD for every feature in the full set U of all features.

These undirected edges may have different directions in DAGs of the Markov equivalent class. Theoretically we can show that the PCD-by-PCD algorithm is correct, that is, it can correctly find, at each step, edges and local v-structures of the global DAG. Let $A \| B$ denote an operation adding the list B to the tail of the list A. For example, $[1,3,5] \| [2,4] = [1,3,5,2,4]$ which is an ordinal sequence.

Algorithm: PCD-by-PCD (Data D; Target T)
1. Initialization:
Set $canV = PCD(T)$. ($canV$ is an ordinal waiting list whose PCD will be found)
Set $V = \{T\}$. (V is a set of variables whose PCD has been obtained)
2. Repeat
(a) Take X from the head of the list $canV$.
(b) Get $PCD_X = PCD(X)$.
(c) $V = V \cup \{X\}$.
(d) For each $Z \in (V \cap PCD_X)$, create an undirected edge (X, Z) if $Z \in PCD_X$ and $X \in PCD_Z$.
(e) Within V, discover possible v-structures only for the triple of X and other two variables in V if an intermediate node is not in the separator set of two nonadjacent nodes.
(f) If we find new v-structures, orient other edges between nodes in V if each opposite of them creates either a directed cycle or a new v-structure (Meek, 1995).
(g) $canV = canV \| (PCD_X \setminus V)$. (Add new variables to the tail of the waiting list)
Until (1) all edges connecting T are oriented, or (2) $canV = \emptyset$, or (3) $V = U$.
3. Return The partially oriented local structure around T.

Theorem 2 *Suppose that a causal network is faithful to a probability distribution and that independence test is correctly performed by using data. Then algorithm PCD-by-PCD correctly obtains edges connected to the target T, and further it returns the same orientations of these edges as a partially directed graph for the Markov equivalence class of the underlying global causal network.* ■

Algorithm PCD-by-PCD sequentially finds $PCD(X)$ of node X that is nearest to the target T among all nodes whose PCDs have not been found at the present step, and it finds $PCD(X)$ at most once for each node X. Thus its computational complexity depends on the algorithm for finding $PCD(X)$. If the number of nodes in the full set U is too large to find all PCDs, then we can stop the algorithm by limiting the maximum size of the set V. The likelihood ratio test statistic G^2 is used in our algorithms for testing conditional independencies.

3.2 Comparison between algorithms

There are several other approaches which can be used for local structural learning. One is the MB-based approach in which we first find the MB of the target and then learn the local structure over the MB and the target. Another is the Markov Blanket Fan Search (MBFS) algorithm proposed by Ramsey (2006). Below we use examples to make comparisons of the Local-Graph, PCD-by-PCD, MB-based and MBFS algorithms.

Example 1. We use the underlying causal network in Figure 1 (a) to compare the MB-based and Local-Graph algorithms. The local structures obtained from the MB-based and Local-Graph algorithms are shown in Figure 1 (b) and (c) respectively. The dashed line between nodes 1 and 7 in Figure 1 (b) denotes the edge which may be false. It can be seen that the MB-based algorithm cannot orient the v-structure $7 \to T \leftarrow 1$.

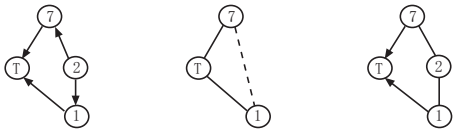

Figure 1: Comparison between the MB-based and Local-Graph algorithms.

Example 2. The underlying causal network in Figure 2 (a) is used to compare the Local-Graph and MBFS algorithms. The local structures in Figure 2 (b) and (c) are obtained from the Local-Graph and MBFS algorithms respectively. The dashed lines in Figure 2 (b) and (c) denote the edges which may be false. For example, the dashed lines $(2, 4)$ and $(3, 4)$ in Figure 2 (b) are determined a true edge and a false edge at the later step respectively, see Figure 2 (c). Although the v-structure $7 \to T \leftarrow 1$ is obtained from the Local-Graph algorithm, it cannot orient the undirected edge $T - 2$. The MBFS algorithm can correctly orient the edge as $T \leftarrow 2$.

Example 3. For the underlying causal network in Figure 3 (a), the MBFS and PCD-by-PCD algorithms output the local structures in Figure 3 (b) and (c) respectively. The MBFS algorithm cannot orient the undirected edge $T - 2$, while the PCD-by-PCD algorithm can

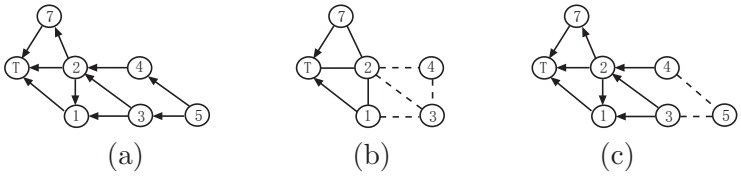

Figure 2: Comparison between the Local-Graph and MBFS algorithms.

do that correctly. The dashed lines in Figure 3 have a similar meaning to those in Example 2.

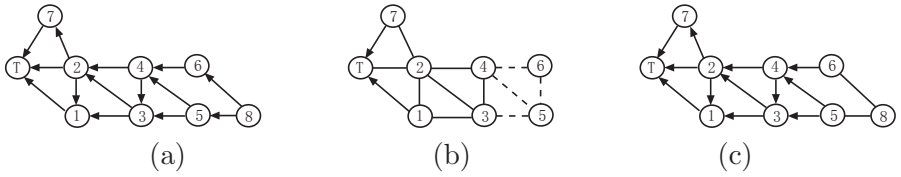

Figure 3: Comparison between the MBFS and PCD-by-PCD algorithms.

From the above examples, we can see that these algorithms discover local structures over different neighbor areas. The MB-based algorithm tries the smallest neighbor area, the Local-graph one the second smallest, the MBFS one the third, and the PCD-by-PCD one extends the neighbor area continuously until all edges connected to the target are oriented or the neighbor area has been extended to all variables. Thus the MB-based, Local-Graph, MBFS and PCD-by-PCD algorithms become in turn to be more complete in terms of orientations.

3.3 Computational complexity of the algorithms

From the previous subsection, it can be seen that all these algorithms need to find PCDs. Thus the number $\#PCD$ of times of finding PCDs can be used as the computational complexity of an algorithm. Let K denote the maximum size of PCDs for all nodes. For the MB-based algorithm, we need to find $\#PCD = O(K)$ PCDs to obtain the MB of the target and then we find a local structure over the MB. For the Local-Graph algorithm, we also need to find $\#PCD = O(K)$ PCDs to obtain the set V and then we find a local structure over V. For the MBFS algorithm, we find $\#PCD = O(K^2)$ PCDs. For the PCD-by-PCD algorithm, the number $\#PCD$ depends on the underly network, which may be smaller than that of the MB-based algorithm, such as the underlying network in Figure 1 (a), or which may larger than that of the MBFS algorithm, such as the network in Figure 3 (a). When there is an undirected path connected to the target T with a length L in the process, the number $\#PCD$ is $O(K^L)$. To stop the PCD-by-PCD algorithm early for the presence of a long undirected path, we can add a stop condition (4): the size of V is larger than a given constant C. Notice that the MB-based and Local-Graph algorithms need additional computation for finding a local structure over the MB and the set V.

Computational complexity of structural learning is exponential with respect to the size of a node set, although the sizes of the MB and V are generally small.

4. Prediction

We first select features based on the causal discovery results discussed in the previous section and return a single set of selected features without rank. For the cases with and without external interventions, we select different feature sets to build prediction models. For the data set without manipulation (numbered 0), all the features in the Markov blanket (MB) of a target T are used to predict the target. For the data set with a known manipulated feature set (numbered 1), we drop the manipulated variables in the children set and drop the spouses of T whose children common with T have been all dropped, and we use all parent variables and unmanipulated children and the parents of unmanipulated children in the MB of T. For the data set with an unknown manipulated variable set (numbered 2), only the parent features of the target are used. When the feature sets that are used for prediction are sensitive to significance levels and other parameters, we may use a union of these sets and then predict the target with a shrinkage method to remove the redundant features. This approach of feature selection is defensibly heuristic since it may drop useful variables in some cases.

Next we apply the L1 penalized logistic regression model with the single set of selected features to the target prediction. We use the estimated probability of the target feature for each individual in the test set for its classification. Let X denote a feature vector, and let Y denote a binary target feature of interest with mean $\mu = E(Y)$. Consider a generalized linear model (GLM) with the logit link function

$$\log \frac{\mu}{1-\mu} = \beta^\top x.$$

The objective function is defined as $-\log$ (likelihood function) with a penalization on the L_1-norm of coefficients, $\|\beta\|_1$,

$$f(\beta, \lambda) = -l(\beta) + \lambda \|\beta\|_1, \tag{1}$$

where λ is a constant. Then a L_1-regularization path algorithm (Park and Hastie, 2007) is used to minimize the objective function $f(\cdot)$ with respect to β and to find the full solution path for (1). On the solution path we select a λ value with 5-fold cross validation (CV) in the training data set which minimizes the prediction error.

5. Numerical Studies

In this section, we first evaluate POLSL algorithms via simulations and then we interpret our results of the causal challenge.

5.1 Evaluation via Simulation

We consider the toy-example: LUCAS (LUng CAncer Simple set) network as shown in Figure 1(a) in Guyon et al. (2008). We repeatedly do 100 simulations and give average

values for each case of different sample size n and significance level α. For each simulation, we draw a training data from the distribution with parameters given on the website: http://www.causality.inf.ethz.ch/data/LUCAS.html. The manipulated features for LUCSA1 and LUCAS2 are shown respectively in Figure 1 (b) and (c) in Guyon et al. (2008), and the manipulated features for test data are drawn randomly which are independent of their parents.

In Table 1, we show the simulation results of discovering the parent set (PA), the MB set and the children (CH) set of the target 'Lung Cancer' with Min Max Hill Climbing (MMHC), MB-based, Local-Graph and PCD-by-PCD algorithms. Feature scores (Fscores) increase with sample size n increasing and are not significantly different. The MMHC algorithm takes CPU time the most, the MB-based algorithm the second, Local-Graph the third and PCD-by-PCD algorithm the least.

In Table 2, we show test scores (Tscores) for different logistic regressions. The middle columns are for linear models, the last two columns are for logistic models with the second order interaction terms. Tscores increase with sample size n increasing and are not significantly different for test data sets labeled 0 and 1. But for test data labeled 2, Tscores based on causal knowledge (here we use the true causal structure) are higher than those without causal knowledge. The methods without/with shrinkage are not significantly different for linear models, but Tscores are quite different for models with interactions. It may because a linear model has a few of parameters, but a model with interactions has a larger number of parameters.

5.2 Results of Causal Challenge

The problems and results of feature selection and prediction for four data sets in the causal challenge are introduced by Guyon et al. (2008). We apply both of our two algorithms Local Graph and PCD-by-PCD on each of the four task data sets. The parameters used are the default value of the MMPC algorithm in Causal Explorer toolkit (Aliferis et al., 2003). Our results are shown in Figure 4. We just select a single unsorted feature subset (ulist) without ranking features (slist), and we submitted a single set of predictions based on the ulist for each test data set. We focused on causal discovery and we tried to minimize the number of features (ulist) selected for prediction. Using the POLSL approaches, we discover a small number of important features which can dominate main causal relationships with a target of interest. As shown in Figure 4, we selected 15 features from 999 features for REGED, 11 from 999 for MARTI, 16 from 4932 for SIDO and 24 from 132 for CINA. There are only two direct causes of the target in the underlying causal graphs of REGED and MARTI, and both of them are contained in our feature sets, see the histograms of REGED and MARTI in Figure 4. Also for CINA and SIDO, a large proportion of our features are direct causes; especially for SIDO, 13 direct causes are contained in our set of 15 features. The Tscore and the rank (rk) of our results in Figure 4 may be improved by chance by using a slist of ranked features because the best Tscore over all feature set sizes is retained under the rules of the challenge. This can be seen in Figure 2 (a) in Guyon et al. (2008) that relative Tscore of our results are not the best comparing with the Tscore which is the best over nested feature subsets. However, under the rule of pairwise comparison using the same number of features, our Fscore and Tscore are at the Pareto front, as shown in Figure

n	α	Set / Time	MMHC	MB-based	Local-Graph	PCD-by-PCD
100	.05	PA	$.657 \pm .169$	$.723 \pm .145$	$.728 \pm .137$	$.702 \pm .160$
		MB	$.764 \pm .103$	$.764 \pm .077$	$.781 \pm .087$	$.742 \pm .089$
		CH	$.688 \pm .125$	$.681 \pm .092$	$.688 \pm .121$	$.671 \pm .114$
		CPU time	67.4	41.4	27.0	7.54
	.10	PA	$.668 \pm .162$	$.712 \pm .153$	$.729 \pm .140$	$.740 \pm .167$
		MB	$.774 \pm .099$	$.781 \pm .077$	$.775 \pm .094$	$.773 \pm .088$
		CH	$.686 \pm .120$	$.675 \pm .107$	$.662 \pm .134$	$.676 \pm .124$
		CPU time	71.5	51.6	35.0	8.10
200	.05	PA	$.823 \pm .105$	$.847 \pm .098$	$.825 \pm .095$	$.854 \pm .122$
		MB	$.870 \pm .074$	$.872 \pm .062$	$.873 \pm .082$	$.827 \pm .064$
		CH	$.621 \pm .094$	$.605 \pm .066$	$.657 \pm .122$	$.637 \pm .070$
		CPU time	75.2	59.5	38.6	8.52
	.10	PA	$.831 \pm .099$	$.844 \pm .095$	$.806 \pm .093$	$.871 \pm .111$
		MB	$.876 \pm .071$	$.873 \pm .064$	$.869 \pm .091$	$.821 \pm .066$
		CH	$.626 \pm .101$	$.606 \pm .075$	$.678 \pm .133$	$.657 \pm .095$
		CPU time	79.1	66.5	47.7	8.39
500	.05	PA	$.863 \pm .033$	$.870 \pm .029$	$.832 \pm .047$	$.921 \pm .032$
		MB	$.927 \pm .063$	$.930 \pm .050$	$.939 \pm .070$	$.841 \pm .058$
		CH	$.676 \pm .124$	$.665 \pm .118$	$.743 \pm .111$	$.707 \pm .117$
		CPU time	84.2	74.2	49.6	7.50
	.10	PA	$.862 \pm .031$	$.867 \pm .030$	$.808 \pm .072$	$.917 \pm .031$
		MB	$.932 \pm .065$	$.935 \pm .053$	$.914 \pm .093$	$.839 \pm .063$
		CH	$.685 \pm .127$	$.677 \pm .122$	$.738 \pm .105$	$.727 \pm .125$
		CPU time	88.7	79.4	61.7	8.13

Table 1: Feature selection comparison with Fscore (Mean $\pm$ std); the unit of CPU time is second. This is a simulation study on the LUCAS data set.

2 (b) in Guyon et al. (2008). All of our computations are performed on a computer with CPU 3.0GHz and 2.49 GB RAM. The CPU times for the four data sets are shown in Table 3. Note that the preprocess time for REGED and CINA is long enough, which is mainly due to the discretization method (Aliferis et al., 2003). And the additional requirement of preprocessing time for MARTI is due to the two-stage filter. SIDO needs a relatively shorter preprocess time because the SIS process is simply a correlation computing process.

6. Discussion

For discovering causal and effect features of a target, the POLSL approaches proposed in this paper only try to find the local structure near a target but not to find the whole network, thus they can greatly reduce computational complexity of structural learning. The POLSL approaches are efficient for large causal networks if we are interested only in prediction of a target. We can theoretically show that the approaches can correctly obtain the edges connected to the target and their orientations. Although the Markov blanket of a target is useful for predicting the target without manipulation, it cannot be used for prediction with manipulation, and the MB-based algorithm is incomplete in terms of orientations of the edges connected to the target.

n	Dataset	Tscore without / with causal knowledge				Regression with interactions	
		NC-Full	NC-Shrink	Cause-Full	Cause-Shrink	NC-Full	Cause-Shrink
100	0	$.873 \pm .028$	$.876 \pm .035$	$.895 \pm .025$	$.886 \pm .031$	$.799 \pm .039$	$.861 \pm .036$
	1	$.856 \pm .044$	$.868 \pm .045$	$.903 \pm .023$	$.896 \pm .028$	$.765 \pm .054$	$.829 \pm .064$
	2	$.747 \pm .072$	$.725 \pm .078$	$.857 \pm .012$	$.838 \pm .075$	$.659 \pm .057$	$.695 \pm .076$
200	0	$.893 \pm .032$	$.894 \pm .033$	$.895 \pm .025$	$.887 \pm .030$	$.794 \pm .041$	$.874 \pm .051$
	1	$.888 \pm .034$	$.893 \pm .033$	$.903 \pm .023$	$.892 \pm .029$	$.763 \pm .061$	$.853 \pm .065$
	2	$.774 \pm .064$	$.760 \pm .072$	$.857 \pm .012$	$.840 \pm .063$	$.656 \pm .058$	$.741 \pm .063$
500	0	$.911 \pm .013$	$.912 \pm .013$	$.895 \pm .025$	$.884 \pm .035$	$.863 \pm .019$	$.889 \pm .032$
	1	$.910 \pm .012$	$.912 \pm .013$	$.903 \pm .023$	$.894 \pm .026$	$.820 \pm .046$	$.874 \pm .055$
	2	$.795 \pm .032$	$.788 \pm .044$	$.857 \pm .012$	$.843 \pm .059$	$.703 \pm .060$	$.753 \pm .066$

Table 2: Prediction comparison with Tscore (Mean $\pm$ std). NC: no causal knowledge ; Cause: using causal knowledge ; Full: a full logistic regression model; Shrink: Using shrinkage; Interaction: model with interactions. This is a simulation study on the LUCAS data set.

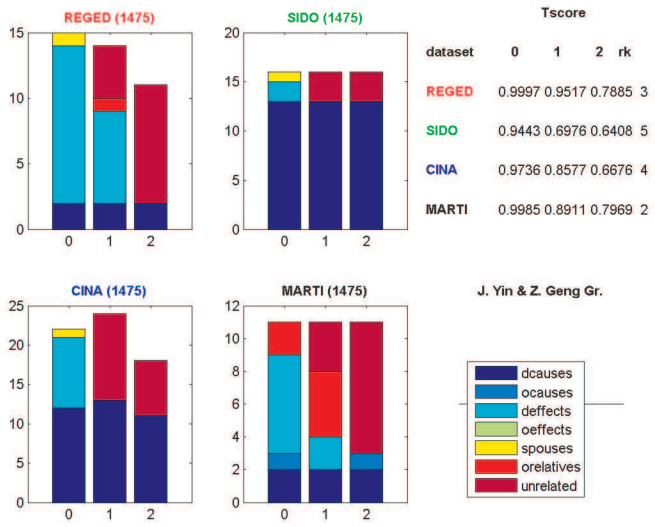

Figure 4: Profile of features selected. Legend: dcause=direct cause, deffect=direct effects, ocauses=other causes (indirect), oeffects=other effects (indirect), spouses=parent of a direct effect, orelatives=other relatives, unrelated=completely irrelevant.

Acknowledgments

We would like to thank the five reviewers for their valuable comments and suggestions. We would appreciate I. Guyon and the competition committee for their encouragement and support to our work. We also thank Xianchao Xie for providing us with his software of

Data set	Preprocessing	Structure Learning	Prediction
REGED	12 hours	15 minutes	5 minutes
SIDO	2 minutes	3 hours	10 minutes
CINA	14 hours	16 hours	10 minutes
MARTI	24 hours	15 minutes	5 minutes

Table 3: CPU times for our results.

structural learning. This research was supported by NSFC 10771007, NBRP 2003CB715900, 863 Project of China (2007AA01Z437), MSRA and MOE-Microsoft Key Laboratory of Statistics and Information Technology of Peking University.

Appendix A.

In this appendix we describe the filtering process in details.

First stage of the filtering process

We use a regression model for each of gene expression features, in which calibrant features are treated as explanatory variables and a gene expression as a response variable. For the jth observation (microarray), let $x_j^{(i)}$ denote a centralized observed value of the ith feature, $s_j^{(i)}$ the latent true value of the ith feature and $\epsilon_j^{(i)}$ the noise. Suppose

$$x_j^{(i)} = s_j^{(i)} + \epsilon_j^{(i)}, \tag{2}$$

for $i = 1, \ldots, F$ ($F = 999$ for MARTI) and $j = 1, \ldots, n$ ($n = 500$ for MARTI), where $s_j^{(i)}$ is independent of $\epsilon_j^{(i)}$. To remove noise, some calibrant features spread regularly across the microarray and they have mean zero. Let y_{kj} denote the kth calibrate feature of the jth observation for $k = 1, \ldots, c$ ($c = 25$ for MARTI) and $Y_j = (y_{1j}, \ldots, y_{cj})$. We assume that the noise at spot i has the following model related to noise at c calibrant spots

$$\epsilon_j^{(i)} = f^{(i)}(\beta^{(i)}, Y_j) + e_j^{(i)}, \tag{3}$$

where $f^{(i)}(\cdot)$ is a known function (we used a linear one), $\beta^{(i)}$ is an unknown parameter vector and $e_j^{(i)}$ is a residue with mean zero which is independent of Y_j. From (2) and (3) we have

$$x_j^{(i)} = f^{(i)}(\beta^{(i)}, Y_j) + (e_j^{(i)} + s_j^{(i)}).$$

We treat $e_j^{(i)} + s_j^{(i)}$ as an error with mean 0 which is independent of Y. Using the least squares method, we can get estimates $\hat{\beta}^{(i)}$, $\hat{\epsilon}_j^{(i)} = f^{(i)}(\hat{\beta}^{(i)}, Y_j)$, and $\hat{s}_j^{(i)} = x_j^{(i)} - \hat{\epsilon}_j^{(i)}$.

Second stage of the filtering process

We treat each microarray separately and thus we omit subscript j. We locally filter the residual noise of each corrected feature $\hat{s}^{(i)}$ using features near the spot i. Suppose that the model for the ith corrected feature is

$$\hat{s}^{(i)} = r^{(i)} + \eta^{(i)}, \tag{4}$$

where $r^{(i)}$ and $\eta^{(i)}$ denote the true latent value and the residual noise respectively, and $r^{(i)}$ is independent of $\eta^{(i)}$. Let $Z^{(j)} = (z_1^{(j)}, z_2^{(j)})$ denote the geometric coordinate of spot j relative to the origin spot i, which is a pair of integers. Define the neighbor area of spot i as $\Omega^{(i)} = \{j : j \neq i, |Z^{(j)} - Z^{(i)}| \leq L\}$ where $|\cdot|$ denotes a distance and L is the upper bound (a user chosen constant) of the distance between spot i and any spot j in the neighbor area. Assume that $\eta^{(j)}$ has a polynomial surface in the neighbor area

$$\eta^{(j)} = g^{(i)}(\alpha^{(i)}, Z^{(j)}) + \xi^{(j)} \tag{5}$$

for $j \in \Omega^{(i)} \cup \{i\}$, where $g^{(i)}$ is a known function (we used a quadratic one), $\alpha^{(i)}$ is an unknown parameter vector, and $\xi^{(j)}$ is an error term with mean zero. From (4) and (5) we have

$$\hat{s}^{(j)} = g^{(i)}(\alpha^{(i)}, Z^{(j)}) + (r^{(j)} + \xi^{(j)})$$

for $j \in \Omega^{(i)}$. Treating $(r^{(j)} + \xi^{(j)})$ as an error term with mean zero, we first find the model and remove 'outliers' to keep informative signals of features. Then using estimates $\hat{\alpha}^{(i)}$, we obtain $\hat{\eta}^{(i)}$ by (4), and finally we get $\hat{r}^{(i)}$ by (3), which is the estimate of the ith feature to be used for prediction modeling.

Appendix B.

In this appendix we prove theorems presented in Section 3.

Proof of Theorem 1. Define $W = \{T\} \cup PCD(T)$. In algorithm Local-Graph, $PCD(X)$ is obtained for each $X \in W$, and V contains all of them. For two nodes u and v, either u is not a descendant of v or v is not a descendant of u. A node is d-separated from its non-descendant by its parent set. Then two nodes u and v in W are not adjacent if and only if they are d-separated by a subset S_{uv} of V. Thus algorithm Local-Graph can correctly find all edges between nodes in W. If there is a pattern $u - T - v$ and T is not contained in the separator S_{uv}, then we can discover a v-stricture $u \to T \leftarrow v$. Thus we have proven Theorem 1.

Proof of Theorem 2. In the PCD-by-PCD algorithm, we find $PCD(T)$ and set $canV = PCD(T)$ where $canV$ denotes a list of nodes whose PCDs will be found at the latter steps.

At step 2 we repeatedly find $PCD(X)$ for X in $canV$ at step 2 (b). Let V denote the set of variables whose PCD has been found. Suppose that the algorithm for finding $PCD(X)$ is correct, such as the algorithm MMPC (Tsamardinos et al., 2006).

At step 2 (d), we can correctly obtain an undirected edge $X - Z$ if we have that both $Z \in PCD(X)$ and $X \in PCD(Z)$. For both Z and X in V, we have obtained $PCD(Z)$ and $PCD(X)$. At step 2 (d), we only need to treat $Z \in (V \cap PCD_X)$ since $Z \notin PCD(X)$ implies no edge $X - Z$, and every other pair of Z and Z' contained in V has been treated at the previous step 2 (d) when Z' or Z entered in V.

At step 2 (e), we try to discover v-structures which contain X as a node since all the undirected edges obtained newly at step 2 (d) contain X and other v-structures without X have been discovered at the previous step 2 (e).

At step 2 (f), we try to orient undirected edges via v-structures obtained newly at step 2 (e).

At step 2 (g), we add nodes of $PCD(X)$ to the end of $canV$.

Finally, we discuss the stop rule. The condition (1) means that all edges connecting T have been oriented and thus the algorithm can stop. The condition (2) means that there is no more node whose PCD needs to be found, which implies other nodes disconnecting T. The condition (3) means that we have found PCDs for all nodes and thus we cannot orient some edges connecting T. If the algorithm stops by the condition (3), we have found the global skeleton graph of the underlying causal network and all v-structures, and thus we obtained the Markov equivalence class. If the algorithm stops by the condition (2), then the underlying causal network is not connected, and we have found the skeleton graph and all v-structures in the connected component. ■

References

C. Aliferis, I. Tsamardinos and A. Statnikov. *'Causal Explorer: A Probabilistic Network Learning Toolkit for Biomedical Discovery.'* The 2003 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences METMBS'03, June 23-26, 2003.

J. Fan and J. Lv. Sure independence screening for ultra-high dimensional feature space. To appear in *J. R. Statist. Soc.* B 70, 849-911, 2008.

I. Guyon, C. Aliferis, G. Cooper, A. Elisseeff, J. Pellet, P. Spirtes and A. Statnikov. Design and analysis of the causation and prediction challenge, in: JMLR: Workshop and conference Proceedings 1-16, 2008.

S. Lauritzen. *Graphical Models*. Clarendon Press, London, 1996.

C. Meek. Causal inference and causal explanation with background knowledge, in: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, 403-410, Morgan Kaufmann, San Francisco, 1995.

M. Y. Park and T. Hastie. L_1-regularization path algorithm for generalized linear models. *J. R. Statist. Soc.* B 69, 659-677, 2007.

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

J. Ramsey. A PC-style Markov blanket search for high dimensional datasets. Technical Report No. CMU-PHIL-177, 2006.

P. Spirtes, C. Glymour and R. Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, the second edition, 2000.

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

X. Xie and Z. Geng. A recursive method for structural learning of directed acyclic graphs. *J Machine Learning Research*, 9, 459-483, 2008.