

Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification

Part II: Analysis and Extensions

Constantin F. Aliferis

CONSTANTIN.ALIFERIS@NYUMC.ORG

*Center of Health Informatics and Bioinformatics
Department of Pathology
New York University
New York, NY 10016, USA*

Alexander Statnikov

ALEXANDER.STATNIKOV@MED.NYU.EDU

*Center of Health Informatics and Bioinformatics
Department of Medicine
New York University
New York, NY 10016, USA*

Ioannis Tsamardinos

TSAMARD@ICS.FORTH.GR

*Computer Science Department, University of Crete
Institute of Computer Science, Foundation for Research and Technology, Hellas
Heraklion, Crete, GR-714 09, Greece*

Subramani Mani

SUBRAMANI.MANI@VANDERBILT.EDU

*Discovery Systems Laboratory
Department of Biomedical Informatics
Vanderbilt University
Nashville, TN 37232, USA*

Xenofon D. Koutsoukos

XENOFON.KOUTSOUKOS@VANDERBILT.EDU

*Department of Electrical Engineering and Computer Science
Vanderbilt University
Nashville, TN 37212, USA*

Editor: Marina Meila

Abstract

In part I of this work we introduced and evaluated the *Generalized Local Learning* (GLL) framework for producing local causal and Markov blanket induction algorithms. In the present second part we analyze the behavior of GLL algorithms and provide extensions to the core methods. Specifically, we investigate the empirical convergence of GLL to the true local neighborhood as a function of sample size. Moreover, we study how predictivity improves with increasing sample size. Then we investigate how sensitive are the algorithms to multiple statistical testing, especially in the presence of many irrelevant features. Next we discuss the role of the algorithm parameters and also show that Markov blanket and causal graph concepts can be used to understand deviations from optimality of state-of-the-art non-causal algorithms. The present paper also introduces the following extensions to the core GLL framework: parallel and distributed versions of GLL algorithms, versions with false discovery rate control, strategies for constructing novel heuristics for specific domains, and divide-and-conquer *local-to-global learning* (LGL) strategies. We test the generality of the LGL approach by deriving a novel LGL-based algorithm that compares favorably

to the state-of-the-art global learning algorithms. In addition, we investigate the use of non-causal feature selection methods to facilitate global learning. Open problems and future research paths related to local and local-to-global causal learning are discussed.

Keywords: local causal discovery, Markov blanket induction, feature selection, classification, causal structure learning, learning of Bayesian networks

1. Introduction

The present paper constitutes the second part of the study of *Generalized Local Learning* (GLL) which provides a unified framework for discovering local causal structure around a target variable of interest using observational data under broad assumptions. GLL supports local discovery of variables that are direct causes or direct effects of the target and of the Markov blanket of the target. In the first part of the work (Aliferis et al., 2010) we introduced GLL and explained the importance of local causal discovery both for identification of highly predictive and parsimonious feature sets (feature selection problem), and for scaling up causal discovery. We then evaluated GLL instantiations against a plethora of state-of-the-art alternatives in many real, simulated and resimulated data sets. The main conclusions were that GLL algorithms achieved excellent predictivity, compactness and ability to learn local neighborhoods. Moreover, state-of-the-art non-causal feature selection methods often achieve excellent predictivity but are misleading in terms of causal discovery.

In the present paper we provide several extensions to GLL, study its properties, and extend to global graph learning using GLL as the core method. Because of the close relationship with Aliferis et al. (2010) we do not repeat here background material, technical definitions, or algorithm specifications. These are found in Aliferis et al. (2010), Sections 2-4.

The paper is organized as follows: Section 2 studies the empirical convergence of GLL instantiations to the true local neighborhood and to optimal predictivity as a function of sample size. Section 3 studies the effects of multiple statistical testing and the sensitivity of GLL algorithms to large numbers of irrelevant features. Section 4 provides a theoretical analysis of GLL algorithms with respect to determinants of statistical decisions, heuristic efficiency and construction of inclusion heuristic functions, reasons for good performance of direct causes and effects instead of induced Markov blanket, and reduced sensitivity to error estimation problems that affect wrappers and traditional filters. Section 5 covers two algorithmic extensions, parallel processing and False Discovery Rate pre-filtering. Section 6 investigates the use of local learners like GLL for global learning and provides a general local-to-global learning framework. In that section we also derive a new algorithm HHC and compare it to the previously described MMHC, and show the potential of local induction variable ordering for tractability and quality improvements. Section 7 uses causal feature selection theory to shed light on limitations of established and newer feature selection methods and the inappropriateness of causally interpreting their output. Section 8 concludes with a discussion of the findings of the present paper and several open problems. An appendix and an online supplement (<http://www.nyuinformatics.org/downloads/supplements/JMLR2009/index.html>) provide additional results, as well as code and data sets that can be used to replicate the experiments.

2. Empirical Convergence and Comparison of Theoretical to Estimated Markov Blanket

As explained in Aliferis et al. (2010), arguments about the suitability of Markov blanket induction for feature selection for classification are based on large sample results, with convergence of small sample performance to the theoretical optimum being unknown. In the present section we use simulated data sets from published Bayesian networks to produce an empirical evaluation of classification performance convergence with respect to training sample size of two types of classifiers: one that uses the estimated Markov blanket ($MB(T)$) or parents and children set ($PC(T)$) and one that uses the true $MB(T)$ or $PC(T)$ set (obtained from the known generative network). We use polynomial SVMs and KNN to fit each classifier type from three training sample sizes: 200, 500 and 5,000 samples. We note that GLL algorithms provide predictive and optimality guarantees for universal approximator classifiers and SVMs and KNN are used here as exemplars of this class of algorithms. In Aliferis et al. (2010) we also discuss more generally suitable classifiers, distributions and loss functions for GLL instantiations. An independent sample of 5,000 instances is used as evaluation test for classification performance (measured by AUC for binary and proportion of correct classifications for multiclass classification tasks). We use data sets sampled from 9 different Bayesian networks (See Table 15 in the Appendix). For each Bayesian network, we randomly select 10 different targets and generate 5 samples (except for sample size 5,000 where one sample is generated) to reduce variability due to sampling.¹ An independent sample of 5,000 instances is used as evaluation test for classification performance. Several local causal induction algorithms are used (including algorithms that induce direct causes/direct effects, and Markov blankets), and are compared to several non-causal algorithms to obtain reference points for baseline performance: RFE, UAF (univariate association filtering), L0, and LARS-EN (see Table 16 in the Appendix for the list of all algorithms). Classifier parameters (misclassification cost C and degree d for polynomial SVMs and number of neighbors K for KNN) are optimized by nested cross-validation following the same methodology as in Aliferis et al. (2010).

Results are presented in Figure 1 (and more details are given in Tables S19 and S20 of the online supplement). The main conclusions follow. Note that similar patterns are present when KNN is used instead of SVMs (with the only difference that convergence is slightly slower for KNN than for SVMs). For brevity we discuss here the SVM results only.

- (a) Classification performance of the true parents and children and Markov blanket feature sets are not statistically significantly different at the 0.05 alpha level in sample 200 (p-value = 0.1440) and are statistically significantly different for larger samples (p-values = 0.0098 and <0.0001 for sample sizes 500 and 5,000, respectively). The difference in SVM classification performance between using the $PC(T)$ and $MB(T)$ sets however does not exceed 0.02 AUC in favor of the $MB(T)$ set. This means that even when the true $PC(T)$ and $MB(T)$ sets are known in the tested data, fitting classifiers from small data using the $PC(T)$ set is as good as using the $MB(T)$ set. In large sample, $MB(T)$ features have a small predictive advantage over $PC(T)$ features.

1. For networks *Lung_Cancer* and *Gene*, we also add an eleventh target that corresponds to the natural response variable: lung cancer diagnosis and cell cycle state, respectively. For network *Munin* we use only 6 targets because of extreme probability distributions of the majority of variables that do not allow variability in the finite sample of size 500 and even 5000. Because of the same reason, we did not experiment with sample size 200 in the *Munin* network.

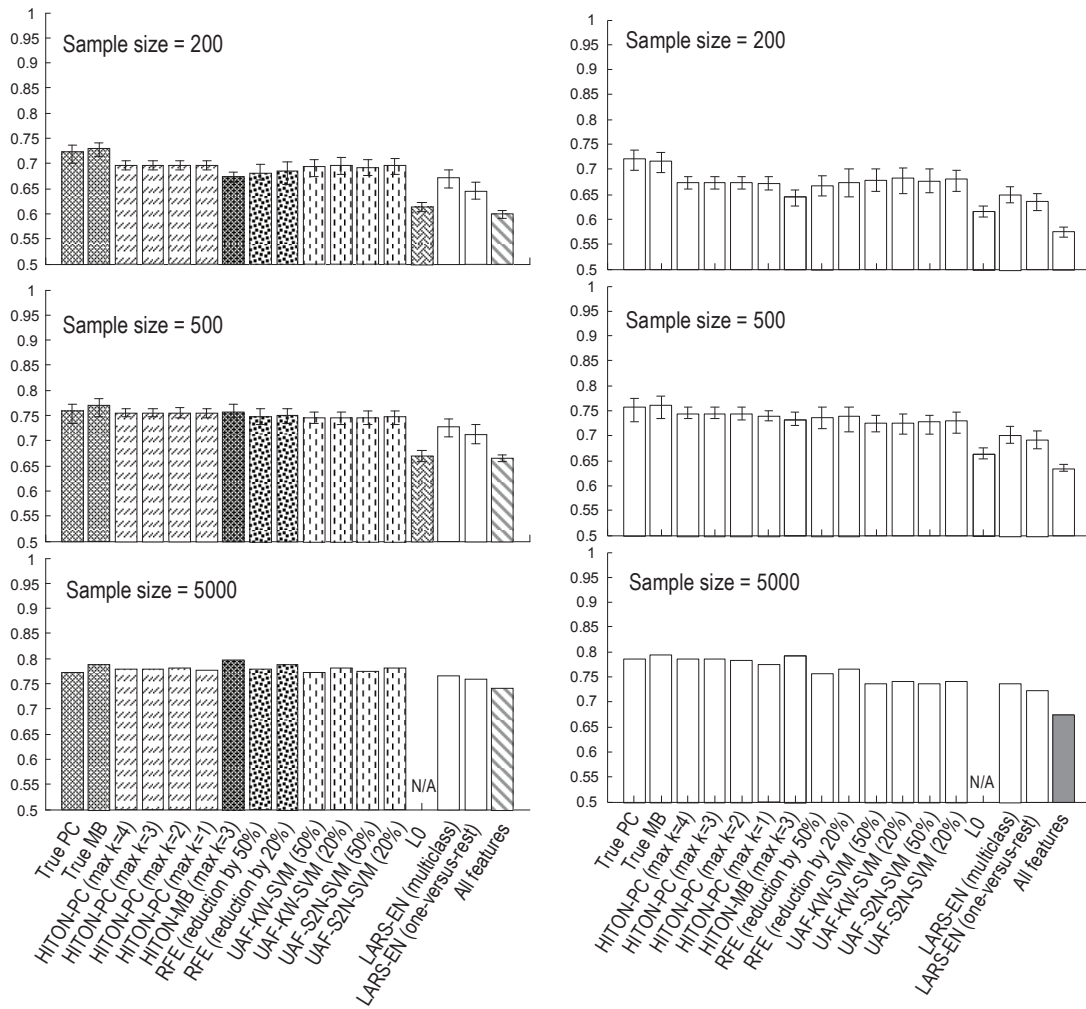


Figure 1: Classification performance of polynomial SVM (left) and KNN (right) classifiers in 9 simulated and resimulated data sets. Results are given for training sample sizes = 200, 500, and 5000. “True-PC” and “True-MB” correspond to the true $PC(T)$ and $MB(T)$ feature sets obtained from the known generative network. The bars denote maximum and minimum performance over multiple training samples of each size (data is available only for sample sizes 200 and 500). The performances reported in the figure are averaged over all data sets, selected targets, and multiple samples of each size. LO did not terminate within the allotted time limit for sample size 5000.

- (b) In small samples, feature selection increases classification performance for all tested classifier types (i.e., both when we know the $PC(T)$ or $MB(T)$ sets and when we estimate them from data) over using all features. This advantage becomes smaller but does not vanish in large sample. The difference in SVM classification performance between an average feature selection method and using all features is statistically significant at the 0.05 alpha level (p-values = <0.0001 , 0.0028 , <0.0001 for sample sizes 200, 500, and 5,000, respectively).

- (c) The true $PC(T)$ or true $MB(T)$ features set when fitted from sample size of 200 has a small (0.02-0.03 AUC/proportion of correct classifications for SVM) advantage over the estimated $PC(T)$ or $MB(T)$ features fitted from small sample. This difference is statistically significant at the 0.05 alpha level with p-values 0.0144 and <0.0001 for the $PC(T)$ and $MB(T)$ classifiers, respectively. Very quickly (as sample size becomes 500), this advantage becomes insignificant (0.01 point of AUC/proportion of correct classifications for SVM) with corresponding p-values 0.4708 and 0.0506 for the $PC(T)$ and $MB(T)$ classifiers, respectively. This implies that predictivity of estimated $MB(T)$ and $PC(T)$ sets converge to the optimal one very quickly with respect to sample size.
- (d) Classifiers for estimated $MB(T)/PC(T)$ sets fitted from small sample and classifiers for the true $MB(T)/PC(T)$ sets fitted from small sample have indistinguishable performance in sample size 500 (as shown in (c) above); then performance increases in sample size 5,000 for both types of classifiers (p-values ranging from <0.0001 to 0.0174 with AUC increases between 0.01 and 0.04). We thus conclude that fitting the right classifier parameters to the identified features is less sample efficient than identifying the right feature set.
- (e) Some of the non-causal feature selection methods (e.g., L0, LARS-EN) tend to compare less favorably in small sample to their large sample performance compared to GLL algorithms.


3. Multiple Statistical Tests and Insensitivity to Irrelevant Variables

In this section we focus our attention to a subtle but an important problem facing many feature and causal discovery algorithms operating in very high dimensional spaces, namely the problem of multiple statistical comparisons, which is exacerbated when many irrelevant features are present. We will show that GLL algorithms have inherent control to false positives due to multiple comparisons while the same is not true for other non-causal feature selection methods tested.

Briefly stated, when conducting n statistical tests with an error type I level α (i.e., statistical significance level, that is probability that a truly null hypothesis is rejected, thus falsely concluding that a statistical difference or association or dependence exists when in reality it does not) it is expected that $\alpha \cdot n$ false positives will occur on average. Consider a common analysis situation in bioinformatics research where a researcher conducts one test per variable (i.e., single nucleotide polymorphism (SNP)) in an assay with 10,000 SNP probes in total. 10,000 such tests need be conducted to see whether univariately each SNP probe is differentially present in two or more phenotype categories. If the researcher uses α equal to 5%, then under the null hypothesis (i.e., all 10,000 SNPs are not truly differentially expressed) the analysis will yield 500 false positive SNP probes. Standard statistical practice involves addressing the problem via one of two basic approaches. The first approach, the classic Bonferroni correction (Casella and Berger, 2002), adjusts the α by replacing it by α/n so that in our example the 5% false positive rate is preserved for each feature selected by the multiple tests. This approach preserves the desired α , but reduces the power to detect statistically significant features (namely the features that are truly differentially expressed and detectable at α but non-detectable at α/n), hence creates false negatives that were not present before the correction. The second approach, False Discovery Rate (FDR) control (Benjamini and Yekutieli, 2001; Benjamini and Hochberg, 1995), trades off false positives and false negatives by ensuring not that each feature passing the chosen p-value threshold preserves the original α , but that from the all features found to be significant (i.e., for which the null hypothesis is rejected) a desired proportion will be false

		<i>Version 1</i> (original network)					<i>Version 2</i> (original network + irrelevant variables)					<i>Version 3</i> (weakened signal + irrelevant variables)					<i>Version 4</i> (only irrelevant variables)				
		max-k parameter																			
Sample size		0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100		1.00	0.99	0.99	0.99	0.99	0.97	0.99	0.98	0.98	0.98	0.63	0.63	0.62	0.62	0.62	0.50	0.50	0.50	0.50	0.50
200		1.00	1.00	0.99	0.98	0.98	0.99	1.00	0.99	0.99	0.99	0.67	0.69	0.67	0.66	0.66	0.51	0.50	0.49	0.50	0.50
500		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.67	0.72	0.73	0.72	0.71	0.50	0.50	0.51	0.49	0.49
1000		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.68	0.74	0.73	0.74	0.72	0.50	0.52	0.51	0.50	0.49
2000		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.69	0.74	0.74	0.74	0.74	0.49	0.50	0.49	0.50	0.49
5000		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.72	0.74	0.74	0.74	0.74	0.51	0.51	0.49	0.49	0.49

		<i>Version 1</i> (original network)					<i>Version 2</i> (original network + irrelevant variables)					<i>Version 3</i> (weakened signal + irrelevant variables)					<i>Version 4</i> (only irrelevant variables)				
		max-k parameter																			
Sample size		0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100		0.95	0.95	0.95	0.95	0.95	0.83	0.92	0.92	0.92	0.92	0.66	0.69	0.69	0.69	0.69	0.50	0.50	0.50	0.50	0.50
200		0.96	0.95	0.95	0.95	0.95	0.89	0.95	0.95	0.95	0.95	0.68	0.77	0.78	0.78	0.78	0.50	0.50	0.50	0.50	0.50
500		0.96	0.96	0.96	0.96	0.96	0.93	0.95	0.95	0.95	0.95	0.71	0.80	0.80	0.80	0.81	0.50	0.51	0.50	0.50	0.50
1000		0.97	0.97	0.97	0.97	0.97	0.94	0.97	0.96	0.96	0.96	0.73	0.82	0.81	0.82	0.82	0.50	0.50	0.50	0.50	0.50
2000		0.97	0.97	0.97	0.97	0.97	0.96	0.97	0.97	0.97	0.97	0.76	0.82	0.82	0.82	0.82	0.50	0.50	0.50	0.50	0.50
5000		0.97	0.98	0.97	0.97	0.97	0.97	0.98	0.97	0.97	0.97	0.81	0.83	0.83	0.83	0.83	0.50	0.50	0.50	0.50	0.50



Low classification performance
High classification performance

Table 1: Classification performance (AUC) of polynomial SVM estimated on 5,000 sample independent testing set for features selected by HITON-PC with parameter $max-k=\{0, 1, 2, 3, 4\}$ on different training sample sizes $\{100, 200, 500, 1000, 2000, 5000\}$. The color of each table cell denotes strength of predictivity with yellow (light) corresponding to low classification performance and red (dark) to high classification performance.


positives on average. In our example, FDR methods may, for example, allow the researcher to ensure that on average no more than 10 out of 100 SNPs selected are false positives. This is highly useful in exploratory analysis of high-dimensional data where subsequent experimentation can sort out false positives easily but where false negatives have high cost.

Constraint-based causal methods employ, in large data sets and depending on connectivity and inclusion heuristic efficiency, many thousands of statistical tests of independence and are thus expected a priori to be particularly sensitive to the multiple testing problem. We note that, rather not obviously at first, testing under the null hypothesis does not only occur when irrelevant features exist but also whenever we test weakly relevant features conditioned on a set of variables that blocks all paths connecting it with the target. Other feature selection methods do not explicitly conduct statistical tests of independence but may also be sensitive to many irrelevant features as we will show. In the present section we first systematically explore empirically and then examine theoretically the degree of sensitivity of GLL algorithms to irrelevant features, how they address the multiple testing problem, and how other feature selection and causal discovery algorithms compare along these dimensions.

In the first set of experiments we run only semi-interleaved HITON-PC without symmetry correction on two networks and variants. The networks, described in Aliferis et al. (2010), are the *Lung_Cancer* resimulated network and the *Alarm10* network. The former is chosen for its higher

<i>Lung_Cancer</i>	<i>Version 1</i> (original network)					<i>Version 2</i> (original network + irrelevant variables)					<i>Version 3</i> (weakened signal + irrelevant variables)				
	max-k parameter														
	Sample size	0	1	2	3	4	0	1	2	3	4	0	1	2	3
100	3.30	15.30	18.20	18.20	18.20	3.30	15.40	18.40	18.40	18.40	9.40	21.90	23.40	23.40	23.40
200	1.20	7.70	17.70	19.60	19.60	1.20	7.70	17.70	19.60	19.60	4.40	17.50	23.20	23.40	23.40
500	0.80	1.30	5.70	15.10	18.00	0.80	1.30	5.70	15.10	18.00	1.00	4.60	17.50	21.70	21.90
1000	0.30	1.00	1.50	5.40	11.70	0.30	1.00	1.50	5.40	11.70	0.80	1.70	6.60	17.50	19.90
2000	0.30	0.90	1.00	1.80	4.10	0.30	0.90	1.00	1.80	4.10	0.70	1.00	1.80	8.70	15.80
5000	0.00	0.40	1.00	1.10	1.10	0.00	0.40	1.00	1.10	1.10	0.30	0.80	1.00	1.40	4.80

<i>Alarm10</i>	<i>Version 1</i> (original network)					<i>Version 2</i> (original network + irrelevant variables)					<i>Version 3</i> (weakened signal + irrelevant variables)				
	max-k parameter														
	Sample size	0	1	2	3	4	0	1	2	3	4	0	1	2	3
100	1.70	4.10	4.10	4.10	4.10	1.70	4.10	4.20	4.20	4.20	2.20	5.00	5.00	5.00	5.00
200	1.40	3.90	4.00	4.00	4.00	1.40	3.90	4.00	4.00	4.00	1.80	4.50	4.70	4.70	4.70
500	0.40	2.60	2.70	2.70	2.70	0.40	2.60	2.90	3.00	3.00	0.60	3.90	4.40	4.40	4.40
1000	0.10	2.00	2.10	2.10	2.10	0.10	2.00	2.20	2.20	2.20	0.80	3.60	3.90	4.00	4.00
2000	0.00	1.40	1.50	1.50	1.50	0.00	1.40	1.50	1.50	1.50	0.10	3.10	3.60	3.50	3.50
5000	0.00	0.50	1.10	1.20	1.20	0.00	0.50	1.10	1.20	1.20	0.00	1.40	1.70	1.80	1.80



*Small number of false negatives**Large number of false negatives*


Table 2: Number of false negatives in the parents and children set for features selected by HITON-PC with parameter $max-k=\{0,1,2,3,4\}$ on different training sample sizes $\{100,200,500,1000,2000,5000\}$. For Version 4 of the network the parents and children set is empty since there are no relevant variables. The color of each table cell denotes number of false negatives with yellow (light) corresponding to smaller values and red (dark) to larger ones.

connectivity whereas the latter is designed to have lower connectivity. In the *Lung_Cancer* network we focused our attention on the natural target variable; this target has 26 members of the parents and children set and 18 spouses, 14 irrelevant variables, and 741 weakly relevant ones. We created four versions of this network: *Version 1* contains the original network (total number of variables 800). In *Version 2* we augment the original network with 7990 irrelevant variables (total number of variables 8790). *Version 3* is the same as Version 2, except for 10% of values of the target are randomly flipped to weaken the signal (total number of variables 8790). Finally, *Version 4* is same as Version 2, except that there are only irrelevant variables and the target (total number of variables is $8790 - 741 - 18 - 26 = 8005$). The tiled *Alarm10* has also four corresponding versions but its target was chosen randomly and it has only 6 members of the parents and children set and no spouses. In both networks (and their variants) we create irrelevant variables by randomly permuting values of weakly and strongly variables so that the distribution of each variable values is realistic. With these 8 data set versions we can systematically examine the effects of presence of irrelevant variables, strength of predictive signal of features for the target, network connectivity and of the values of the GLL $max-k$ parameter (Aliferis et al., 2010).

We run HITON-PC and build SVM classifiers for all networks and variants, varying sample size and the $max-k$ parameter, and measure AUC, false negatives, false positives that are weakly relevant,

<i>Lung_Cancer</i>	<i>Version 1</i> (original network)					<i>Version 2</i> (original network + irrelevant variables)					<i>Version 3</i> (weakened signal + irrelevant variables)				
	max-k parameter														
	Sample size	0	1	2	3	4	0	1	2	3	4	0	1	2	3
100	65.00	0.80	0.30	0.30	0.30	65.00	0.70	0.40	0.40	0.40	62.40	0.90	0.50	0.50	0.50
200	120.50	3.00	0.10	0.00	0.00	120.50	3.00	0.10	0.00	0.00	85.60	2.90	0.60	0.60	0.60
500	149.00	5.80	0.00	0.10	0.00	149.00	5.80	0.00	0.10	0.00	110.70	4.20	0.40	0.30	0.30
1000	202.90	11.60	0.10	0.00	0.00	202.90	11.60	0.10	0.00	0.00	123.70	5.70	0.00	0.00	0.00
2000	236.10	16.40	0.50	0.10	0.00	236.10	16.40	0.50	0.10	0.00	171.10	12.00	0.40	0.00	0.00
5000	410.40	30.80	2.60	0.10	0.00	410.40	30.80	2.60	0.10	0.00	272.60	20.30	1.10	0.00	0.00

<i>Alarm10</i>	<i>Version 1</i> (original network)					<i>Version 2</i> (original network + irrelevant variables)					<i>Version 3</i> (weakened signal + irrelevant variables)				
	max-k parameter														
	Sample size	0	1	2	3	4	0	1	2	3	4	0	1	2	3
100	22.10	3.70	3.70	3.70	3.70	22.10	2.40	2.40	2.40	2.40	22.50	1.80	1.80	1.80	1.80
200	26.50	0.80	0.80	0.80	0.80	26.50	0.60	0.50	0.50	0.50	25.20	1.30	0.90	0.90	0.90
500	32.20	0.90	0.10	0.10	0.10	32.20	0.80	0.10	0.10	0.10	32.00	1.00	0.20	0.20	0.20
1000	30.20	1.40	0.00	0.00	0.00	30.20	1.30	0.00	0.00	0.00	27.10	0.70	0.10	0.30	0.30
2000	33.50	2.90	0.30	0.30	0.30	33.50	2.80	0.30	0.30	0.30	32.40	1.80	0.60	0.20	0.20
5000	38.00	5.40	0.30	0.20	0.10	38.00	5.30	0.30	0.20	0.10	37.30	3.10	0.20	0.20	0.20



Small number of false positives *Large number of false positives*

Table 3: Number of false positives (within weakly relevant variables) in the parents and children set for features selected by HITON-PC with parameter $max-k=\{0, 1, 2, 3, 4\}$ on different training sample sizes $\{100, 200, 500, 1000, 2000, 5000\}$. For Version 4 of the network there are no weakly relevant variables. The color of each table cell denotes number of false positives with yellow (light) corresponding to smaller values and red (dark) to larger ones.

false positives that are irrelevant and total false positives. To ensure that our results are not affected by variability in small samples, we generate 10 random samples of each size and average results.

Tables 1– 5 provide evidence for the following conclusions:

- (a) Classification performance is mildly or not affected by false positives and false negatives (Table 1). When many false negatives are present, predictivity is compensated by the few remaining strong relevant features plus strongly predictive weakly relevant ones. This implies that classification performance cannot be used to inform us about the presence of false positives/negatives.
- (b) As expected, false negatives are reduced as sample size grows (because power increases), however they also increase as $max-k$ grows, because the number of tests increases as $max-k$ grows and thus overall power decreases (Table 2).
- (c) When no irrelevant features are present, as sample size grows the number of false positives that are weakly relevant increases if $max-k$ is not sufficient to block paths from/to each weakly relevant to/from the target. As $max-k$ increases the false positives decrease to the point that they vanish (Table 3). Overall, both false negatives and false positives vanish given enough

		<i>Version 1</i> (original network)					<i>Version 2</i> (original network + irrelevant variables)					<i>Version 3</i> (weakened signal + irrelevant variables)					<i>Version 4</i> (only irrelevant variables)				
		max-k parameter																			
Sample size		0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100		65.20	0.80	0.30	0.30	0.30	476.60	2.30	1.90	1.90	1.90	551.20	12.60	9.10	9.10	9.10	411.60	12.70	9.80	9.80	9.80
200		122.00	3.00	0.10	0.00	0.00	609.10	4.20	0.10	0.00	0.00	557.20	17.80	3.50	3.60	3.60	488.60	17.30	5.80	5.50	5.50
500		149.20	5.80	0.00	0.10	0.00	595.00	7.90	0.00	0.10	0.00	535.60	17.50	1.30	1.50	1.70	446.00	28.10	6.40	5.00	4.90
1000		203.40	11.60	0.10	0.00	0.00	625.60	13.20	0.10	0.00	0.00	536.90	18.40	0.20	0.30	0.30	422.70	31.20	6.90	5.30	5.10
2000		236.90	16.40	0.50	0.10	0.00	645.10	18.00	0.50	0.10	0.00	579.00	23.10	0.80	0.00	0.00	409.00	31.80	6.10	4.00	4.00
5000		411.10	30.80	2.60	0.10	0.00	813.50	32.50	2.60	0.10	0.00	670.40	32.10	1.10	0.00	0.00	403.10	30.90	6.20	4.70	4.10

		<i>Version 1</i> (original network)					<i>Version 2</i> (original network + irrelevant variables)					<i>Version 3</i> (weakened signal + irrelevant variables)					<i>Version 4</i> (only irrelevant variables)				
		max-k parameter																			
Sample size		0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100		22.10	3.70	3.70	3.70	3.70	414.20	25.40	25.20	25.20	25.20	431.20	28.00	28.20	28.20	28.20	392.10	23.30	23.40	23.40	23.40
200		26.50	0.80	0.80	0.80	0.80	439.40	6.30	4.30	4.30	4.30	453.00	11.60	7.40	7.40	7.40	412.90	19.30	9.70	9.70	9.70
500		32.20	0.90	0.10	0.10	0.10	443.80	4.70	0.90	0.90	0.90	449.90	15.80	4.60	4.10	4.00	411.60	24.40	6.80	6.60	6.60
1000		30.20	1.40	0.00	0.00	0.00	444.30	3.70	0.90	0.60	0.60	427.00	13.30	3.40	3.10	3.00	414.10	22.70	7.20	6.40	6.30
2000		33.50	2.90	0.30	0.30	0.30	415.50	4.40	0.30	0.30	0.30	412.40	11.90	2.40	1.80	1.70	382.00	25.00	8.80	6.50	5.90
5000		38.00	5.40	0.30	0.20	0.10	419.00	6.70	0.40	0.20	0.10	404.40	10.80	1.20	0.50	0.50	381.00	22.90	6.10	5.00	4.90

Small number of false positives
Large number of false positives

Table 4: Number of false positives in the parents and children set for features selected by HITON-PC with parameter $max-k=\{0,1,2,3,4\}$ on different training sample sizes $\{100,200,500,1000,2000,5000\}$. The color of each table cell denotes number of false positives with yellow (light) corresponding to smaller values and red (dark) to larger ones.

sample size and sufficient (but not excessive) $max-k$, (i.e., sample size $\geq 2,000$, $max-k=2$) (Tables 2 and 4).

- (d) When irrelevant features are present, as sample size grows the number of false positives that are weakly relevant increases if $max-k$ is not sufficient to block paths from/to each weakly relevant to/from the target. As $max-k$ increases, the false positives decrease to the point that they vanish (Table 3). False positives due to irrelevant features (Table 5) quickly vanish as $max-k$ becomes 2 or higher and this holds as long as sample size is larger than 200. False negatives are not affected by presence of irrelevant features (Table 2). Thus, overall, with enough sample size and right value of $max-k$, both false negatives and false positives vanish (Tables 2 and 4).
- (e) When the predictive signal is weaker, both false negatives are increased and false positives within weakly relevant variables are decreased for a given sample size (because power is smaller) (Tables 2 and 3). However false positive irrelevant variables (Table 5) are increased. This is due to the fact that fewer features enter the $TPC(T)$ set thus leading to fewer tests that can be performed hence smaller capacity to remove irrelevant false positives. As previously with enough sample and right $max-k$, false positives and negatives are fully eliminated (Tables 2 and 4).
- (f) When the data consists only of irrelevant features, false positives (irrelevant) are reduced as $max-k$ increases for all sample sizes (Table 5). There is a very small persistent residual number of false positives regardless of how small the sample is or how big the $max-k$. These phenom-

		<i>Version 1</i> (original network)					<i>Version 2</i> (original network + irrelevant variables)					<i>Version 3</i> (weakened signal + irrelevant variables)					<i>Version 4</i> (only irrelevant variables)				
		max-k parameter																			
Sample size		0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100		0.20	0.00	0.00	0.00	0.00	411.60	1.60	1.50	1.50	1.50	488.80	11.70	8.60	8.60	8.60	411.60	12.70	9.80	9.80	9.80
200		1.50	0.00	0.00	0.00	0.00	488.60	1.20	0.00	0.00	0.00	471.60	14.90	2.90	3.00	3.00	488.60	17.30	5.80	5.50	5.50
500		0.20	0.00	0.00	0.00	0.00	446.00	2.10	0.00	0.00	0.00	424.90	13.30	0.90	1.20	1.40	446.00	28.10	6.40	5.00	4.90
1000		0.50	0.00	0.00	0.00	0.00	422.70	1.60	0.00	0.00	0.00	413.20	12.70	0.20	0.30	0.30	422.70	31.20	6.90	5.30	5.10
2000		0.80	0.00	0.00	0.00	0.00	409.00	1.60	0.00	0.00	0.00	407.90	11.10	0.40	0.00	0.00	409.00	31.80	6.10	4.00	4.00
5000		0.70	0.00	0.00	0.00	0.00	403.10	1.70	0.00	0.00	0.00	397.80	11.80	0.00	0.00	0.00	403.10	30.90	6.20	4.70	4.10

		<i>Version 1</i> (original network)					<i>Version 2</i> (original network + irrelevant variables)					<i>Version 3</i> (weakened signal + irrelevant variables)					<i>Version 4</i> (only irrelevant variables)				
		max-k parameter																			
Sample size		0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
100		0.00	0.00	0.00	0.00	0.00	392.10	23.00	22.80	22.80	22.80	408.70	26.20	26.40	26.40	26.40	392.10	23.30	23.40	23.40	23.40
200		0.00	0.00	0.00	0.00	0.00	412.90	5.70	3.80	3.80	3.80	427.80	10.30	6.50	6.50	6.50	412.90	19.30	9.70	9.70	9.70
500		0.00	0.00	0.00	0.00	0.00	411.60	3.90	0.80	0.80	0.80	417.90	14.80	4.40	3.90	3.80	411.60	24.40	6.80	6.60	6.60
1000		0.00	0.00	0.00	0.00	0.00	414.10	2.40	0.90	0.60	0.60	399.90	12.60	3.30	2.80	2.70	414.10	22.70	7.20	6.40	6.30
2000		0.00	0.00	0.00	0.00	0.00	382.00	1.60	0.00	0.00	0.00	380.00	10.10	1.80	1.60	1.50	382.00	25.00	8.80	6.50	5.90
5000		0.00	0.00	0.00	0.00	0.00	381.00	1.40	0.10	0.00	0.00	367.10	7.70	1.00	0.30	0.30	381.00	22.90	6.10	5.00	4.90

Small number of false positives *Large number of false positives*

Table 5: Number of false positives (within irrelevant variables) in the parents and children set for features selected by HITON-PC with parameter $max-k=\{0, 1, 2, 3, 4\}$ on different training sample sizes $\{100, 200, 500, 1000, 2000, 5000\}$. The color of each table cell denotes number of false positives with yellow (light) corresponding to smaller values and red (dark) to larger ones.

ena happen because the algorithm needs a sufficient number of elements in the $TPC(T)$ set (i.e., tentative parents and children of T) in order to execute conditional independence tests and remove the false positive irrelevant features.

- (g) The above trends are remarkably consistent in both networks suggesting that different redundancy and connectivity do not affect the above algorithm behavior.

In the second set of experiments we compare empirically in the above two networks (four variants for each as previously) and 6 sample sizes the following algorithms: semi-interleaved HITON-PC, MMPC, a version of HITON-PC where we pre-filter features by Benjamini FDR control (at FDR rate threshold of 5%) (Benjamini and Yekutieli, 2001), the true $PC(T)$ set extracted from the data generating network (denoted as “True-PC” in Table 6), UAF (univariate association filtering) with Bonferroni correction, UAF with Benjamini FDR control, uncorrected UAF, “wrapped” UAF, RFE, and LARS-EN. Tables 6–9 provide support for the following conclusions:

- (h) Due to strength of signal and redundancy of predictors, AUC reaches the theoretical maximum (provided by the generative network) very quickly and for all methods (Table 6).
- (i) When no irrelevant features are present and in the stronger signal setting, simple and FDR-corrected UAF (but not wrapped UAF) has the least false negatives in very small samples (Table 7). As sample size grows all methods reduce their false negatives (Table 7). GLL methods pick up the strongly relevant features without false positives and reach near perfect

Lung_Cancer	Version 1 (original network)					Version 2 (original network + irrelevant variables)					Version 3 (weakened signal + irrelevant variables)					Version 4 (only irrelevant variables)								
	sample size					sample size					sample size					sample size								
	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000
FS method	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
True-PC	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.99	1.00	1.00	1.00	1.00	0.63	0.67	0.67	0.68	0.69	0.72	0.50	0.50	0.50	0.50	0.50	0.50
UAF	1.00	1.00	1.00	1.00	1.00	1.00	0.94	1.00	1.00	1.00	1.00	0.60	0.70	0.74	0.73	0.74	0.74	0.50	0.50	0.50	0.50	0.50	0.50	0.50
UAF+Bonferroni	0.98	1.00	1.00	1.00	1.00	1.00	0.95	1.00	1.00	1.00	1.00	0.61	0.72	0.74	0.74	0.74	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
UAF+FDR	0.99	0.99	1.00	1.00	1.00	1.00	0.98	0.99	1.00	1.00	1.00	0.62	0.67	0.73	0.73	0.74	0.74	0.50	0.49	0.51	0.51	0.49	0.49	0.49
HITON-PC	0.97	0.99	1.00	1.00	1.00	1.00	0.94	0.99	1.00	1.00	1.00	0.61	0.68	0.73	0.73	0.74	0.74	0.49	0.49	0.49	0.49	0.49	0.49	0.49
HITON-PC-FDR	0.99	0.99	1.00	1.00	1.00	1.00	0.95	0.99	1.00	1.00	1.00	0.62	0.67	0.73	0.73	0.74	0.74	0.50	0.49	0.50	0.50	0.50	0.50	0.50
MMPC	0.91	1.00	1.00	1.00	1.00	1.00	0.97	0.98	1.00	1.00	1.00	0.64	0.70	0.72	0.71	0.73	0.74	0.49	0.50	0.50	0.50	0.50	0.50	0.51
LARS-EN	0.97	1.00	1.00	1.00	1.00	1.00	0.97	0.99	1.00	1.00	1.00	0.61	0.65	0.71	0.70	0.73	0.74	0.50	0.50	0.50	0.50	0.50	0.49	0.50
RFE (reduction 50%)	0.95	0.99	1.00	1.00	1.00	1.00	0.96	0.98	1.00	1.00	1.00	0.60	0.68	0.71	0.71	0.74	0.74	0.50	0.50	0.50	0.49	0.50	0.50	0.50
RFE (reduction 20%)	0.98	0.99	1.00	1.00	1.00	1.00	0.97	0.99	1.00	1.00	1.00	0.65	0.69	0.73	0.71	0.74	0.74	0.50	0.50	0.50	0.50	0.50	0.49	0.50
UAF-KW-SVM (50%)	0.95	0.99	1.00	1.00	1.00	1.00	0.96	0.98	1.00	1.00	1.00	0.66	0.71	0.73	0.72	0.74	0.74	0.50	0.50	0.50	0.49	0.49	0.49	0.51
UAF-KW-SVM (20%)	0.94	0.99	1.00	1.00	1.00	1.00	0.94	0.99	1.00	1.00	1.00	0.58	0.64	0.74	0.73	0.74	0.74	0.50	0.50	0.50	0.50	0.50	0.50	0.50
UAF-S2N-SVM (50%)	0.93	0.98	1.00	1.00	1.00	1.00	0.93	0.98	1.00	1.00	1.00	0.58	0.67	0.74	0.73	0.74	0.74	0.50	0.50	0.50	0.49	0.50	0.50	0.50
UAF-S2N-SVM (20%)	0.93	0.98	1.00	1.00	1.00	1.00	0.93	0.98	1.00	1.00	1.00	0.58	0.67	0.74	0.73	0.74	0.74	0.50	0.50	0.50	0.49	0.50	0.50	0.50

Alarm10	Version 1 (original network)					Version 2 (original network + irrelevant variables)					Version 3 (weakened signal + irrelevant variables)					Version 4 (only irrelevant variables)								
	sample size					sample size					sample size					sample size								
	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000
FS method	0.97	0.96	0.97	0.97	0.97	0.97	0.97	0.96	0.96	0.97	0.97	0.97	0.83	0.82	0.83	0.82	0.82	0.83	0.50	0.50	0.50	0.50	0.50	0.50
True-PC	0.95	0.96	0.96	0.97	0.97	0.97	0.83	0.89	0.93	0.94	0.96	0.97	0.66	0.68	0.71	0.73	0.76	0.81	0.50	0.50	0.50	0.50	0.50	0.50
UAF	0.95	0.96	0.97	0.97	0.98	0.98	0.94	0.96	0.97	0.98	0.98	0.81	0.82	0.82	0.82	0.83	0.83	0.50	0.50	0.50	0.50	0.50	0.50	0.50
UAF+Bonferroni	0.95	0.96	0.97	0.97	0.98	0.98	0.95	0.96	0.97	0.97	0.98	0.81	0.82	0.82	0.82	0.83	0.83	0.50	0.50	0.50	0.50	0.50	0.50	0.50
UAF+FDR	0.95	0.95	0.96	0.97	0.97	0.97	0.92	0.95	0.95	0.96	0.97	0.97	0.69	0.78	0.80	0.81	0.82	0.83	0.50	0.50	0.50	0.50	0.50	0.50
HITON-PC	0.94	0.95	0.95	0.97	0.97	0.97	0.94	0.95	0.96	0.96	0.97	0.97	0.80	0.81	0.81	0.81	0.82	0.82	0.49	0.49	0.49	0.49	0.49	0.49
HITON-PC-FDR	0.95	0.95	0.95	0.97	0.97	0.97	0.91	0.95	0.95	0.97	0.97	0.67	0.77	0.80	0.82	0.82	0.83	0.50	0.50	0.50	0.50	0.50	0.50	0.50
MMPC	0.94	0.95	0.97	0.97	0.97	0.97	0.93	0.95	0.96	0.96	0.97	0.97	0.78	0.80	0.81	0.81	0.83	0.82	0.50	0.49	0.50	0.50	0.49	0.49
LARS-EN	0.94	0.95	0.96	0.97	0.97	0.97	0.93	0.94	0.95	0.96	0.96	0.72	0.79	0.81	0.82	0.82	0.82	0.50	0.50	0.50	0.49	0.50	0.50	0.50
RFE (reduction 50%)	0.94	0.95	0.96	0.96	0.97	0.97	0.91	0.95	0.95	0.96	0.97	0.96	0.73	0.79	0.81	0.81	0.82	0.83	0.49	0.50	0.50	0.50	0.50	0.50
RFE (reduction 20%)	0.94	0.95	0.96	0.97	0.97	0.97	0.94	0.95	0.96	0.97	0.97	0.75	0.79	0.81	0.82	0.82	0.83	0.50	0.50	0.50	0.50	0.50	0.50	0.50
UAF-KW-SVM (50%)	0.95	0.96	0.96	0.97	0.97	0.97	0.93	0.95	0.96	0.97	0.97	0.77	0.79	0.82	0.83	0.83	0.83	0.50	0.50	0.50	0.49	0.50	0.50	0.50
UAF-KW-SVM (20%)	0.95	0.96	0.96	0.97	0.97	0.97	0.94	0.95	0.96	0.97	0.97	0.77	0.80	0.81	0.82	0.82	0.83	0.50	0.50	0.50	0.49	0.50	0.50	0.50
UAF-S2N-SVM (50%)	0.95	0.96	0.96	0.97	0.97	0.97	0.94	0.95	0.96	0.97	0.97	0.77	0.80	0.81	0.82	0.82	0.83	0.50	0.50	0.50	0.49	0.50	0.50	0.50
UAF-S2N-SVM (20%)	0.95	0.96	0.96	0.97	0.97	0.97	0.94	0.95	0.96	0.97	0.97	0.77	0.80	0.81	0.82	0.82	0.83	0.50	0.50	0.50	0.49	0.50	0.50	0.50

Low classification performance  High classification performance

Table 6: Classification performance (AUC) of polynomial SVM estimated on 5,000 sample independent testing set for selected features. HITON-PC, HITON-PC-FDR, and MMPC are applied with $max-k=2$. The color of each table cell denotes strength of predictivity with yellow (light) corresponding to low classification performance and red (dark) to high classification performance.

Lung_Cancer	Version 1 (original network)						Version 2 (original network + irrelevant variables)						Version 3 (weakened signal + irrelevant variables)					
	sample size																	
	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000
FS method	3.3	1.2	0.8	0.3	0.3	0.0	3.3	1.2	0.8	0.3	0.3	0.0	9.4	4.4	1.0	0.8	0.7	0.3
UAF	3.3	1.2	0.8	0.3	0.3	0.0	3.3	1.2	0.8	0.3	0.3	0.0	9.4	4.4	1.0	0.8	0.7	0.3
UAF+Bonferroni	13.9	6.1	1.5	1.0	0.9	0.2	17.6	8.4	1.8	1.0	1.0	0.5	24.9	19.9	6.7	2.4	1.0	1.0
UAF+FDR	9.2	2.5	0.9	0.5	0.4	0.0	13.4	4.8	1.3	0.9	0.8	0.0	24.0	16.2	3.5	1.3	1.0	0.8
HITON-PC	18.2	17.7	5.7	1.5	1.0	1.0	18.4	17.7	5.7	1.5	1.0	1.0	23.4	23.2	17.5	6.6	1.8	1.0
HITON-PC-FDR	19.3	18.5	5.7	1.5	1.0	1.0	19.2	18.5	5.7	1.5	1.0	1.0	24.7	23.3	17.9	6.6	1.8	1.0
MMPC	18.5	17.7	5.7	1.5	1.0	1.0	18.9	17.7	5.7	1.5	1.0	1.0	23.4	22.8	17.6	6.6	1.8	1.0
LARS-EN	19.9	14.2	8.8	7.9	3.6	1.0	15.9	18.6	10.0	10.0	3.7	1.6	22.8	21.5	18.3	13.4	9.4	10.7
RFE (reduction 50%)	20.7	15.9	9.4	6.1	4.1	1.0	18.8	14.6	13.3	9.2	3.2	1.6	21.1	15.9	7.6	8.6	14.8	12.8
RFE (reduction 20%)	21.9	17.1	10.5	12.5	4.9	2.6	18.7	18.8	11.0	9.1	3.7	2.3	15.6	18.1	8.3	14.3	16.9	12.3
UAF-KW-SVM (50%)	17.5	16.6	5.9	5.3	1.6	0.7	17.8	15.8	8.6	9.8	5.6	1.5	20.1	14.1	10.9	9.3	8.2	7.3
UAF-KW-SVM (20%)	21.0	18.8	10.5	8.3	2.6	0.7	19.1	18.7	10.7	13.2	6.4	1.2	20.5	14.3	12.4	8.1	6.9	7.2
UAF-S2N-SVM (50%)	20.8	17.1	6.0	7.6	2.5	1.3	17.6	16.7	8.4	7.1	7.0	1.9	16.6	15.4	15.6	11.5	8.3	4.9
UAF-S2N-SVM (20%)	23.1	19.9	9.4	10.5	5.1	1.8	20.5	18.5	10.4	11.3	7.0	0.7	19.4	14.8	15.4	12.3	6.6	5.5

Alarm10	Version 1 (original network)						Version 2 (original network + irrelevant variables)						Version 3 (weakened signal + irrelevant variables)					
	sample size																	
	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000
FS method	1.7	1.4	0.4	0.1	0.0	0.0	1.7	1.4	0.4	0.1	0.0	0.0	2.2	1.8	0.6	0.8	0.1	0.0
UAF	1.7	1.4	0.4	0.1	0.0	0.0	1.7	1.4	0.4	0.1	0.0	0.0	2.2	1.8	0.6	0.8	0.1	0.0
UAF+Bonferroni	4.1	2.7	1.4	1.0	0.5	0.0	4.7	3.2	1.5	1.1	0.7	0.2	5.0	4.4	2.7	1.4	1.0	0.5
UAF+FDR	3.3	2.2	0.8	1.0	0.3	0.0	4.3	2.8	1.4	1.1	0.5	0.0	4.9	3.8	2.4	1.2	0.9	0.2
HITON-PC	4.1	4.0	2.7	2.1	1.5	1.1	4.2	4.0	2.9	2.2	1.5	1.1	5.0	4.7	4.4	3.9	3.6	1.7
HITON-PC-FDR	4.6	4.2	3.2	2.3	1.7	1.0	4.8	4.3	3.2	2.3	1.7	1.0	5.5	4.7	4.4	4.2	3.6	2.1
MMPC	4.1	4.0	3.0	2.4	1.6	1.0	4.3	4.1	3.5	2.4	1.6	1.0	5.0	4.7	4.5	4.2	3.7	2.1
LARS-EN	3.8	3.8	1.7	1.7	1.5	1.4	4.4	4.1	2.5	2.2	1.9	1.4	4.6	4.6	4.6	3.5	2.2	2.0
RFE (reduction 50%)	4.1	3.7	2.1	1.9	2.3	1.5	4.8	4.7	3.2	3.3	2.6	1.8	4.6	4.9	5.2	4.6	4.2	3.6
RFE (reduction 20%)	4.1	3.7	2.4	2.7	2.1	1.8	5.0	4.4	3.4	3.2	2.3	2.0	5.0	5.3	5.0	4.5	3.7	3.3
UAF-KW-SVM (50%)	3.8	3.8	2.2	0.8	0.9	0.4	4.8	3.6	2.4	2.2	1.4	0.1	3.8	4.2	3.4	2.1	2.2	0.8
UAF-KW-SVM (20%)	4.0	3.2	2.4	1.1	0.4	0.0	4.2	3.6	2.4	1.9	1.2	0.0	4.2	4.3	2.7	2.8	1.9	1.2
UAF-S2N-SVM (50%)	3.5	3.6	2.1	1.0	0.8	0.4	4.7	3.8	2.2	2.1	1.5	0.2	5.1	4.4	4.3	3.5	2.7	1.0
UAF-S2N-SVM (20%)	4.3	3.5	2.6	1.3	0.5	0.0	4.9	3.7	2.5	1.9	1.7	0.2	5.0	4.5	3.6	3.0	2.5	1.4




Table 7: Number of false negatives in the parents and children set for selected features. HITON-PC, HITON-PC-FDR, and MMPC are applied with $max-k=2$. For Version 4 of the network the parents and children set is empty since there are no relevant variables. The color of each table cell denotes number of false negatives with yellow (light) corresponding to smaller values and red (dark) to larger ones.

separation (i.e., 1-2 false negatives and zero false positives) at sample size 1,000 and higher (Table 8). No other method simultaneously minimizes false positives and false negatives as GLL.

- (j) In the setting of strong signal with irrelevant features, simple UAF has the least false negatives in very small samples (Table 7) and the largest number of false positives (Table 8).
- (k) When the predictive signal is weaker, false negatives are increased and weakly relevant false positives are decreased for a given sample size compared to the stronger signal case (Tables 7 and 8). Simple UAF is again most sensitive in terms of detecting strongly relevant features in smaller samples until sample size 1,000-2,000 where UAF-Bonferroni and UAF-FDR and GLL match the false negative rates (Table 7). As previously, GLL (with HITON-PC and

Lung_Cancer	Version 1 (original network)						Version 2 (original network + irrelevant variables)						Version 3 (weakened signal + irrelevant variables)					
	sample size																	
	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000
UAF	65.0	120.5	149.0	202.9	236.1	410.4	65.0	120.5	149.0	202.9	236.1	410.4	62.4	85.6	110.7	123.7	171.1	272.6
UAF+Bonferroni	1.8	8.9	33.6	65.5	91.6	160.3	0.6	4.1	21.2	52.5	80.3	134.3	0.1	0.7	4.8	14.9	43.4	83.6
UAF+FDR	9.4	39.3	78.3	130.5	168.6	359.9	2.7	13.6	46.2	82.6	111.8	230.7	0.1	2.3	13.3	33.5	70.8	123.6
HITON-PC	0.3	0.1	0.0	0.1	0.5	2.6	0.4	0.1	0.0	0.1	0.5	2.6	0.5	0.6	0.4	0.0	0.4	1.1
HITON-PC-FDR	0.2	0.0	0.0	0.1	0.3	1.4	0.1	0.1	0.0	0.1	0.3	1.4	0.1	0.6	0.3	0.0	0.3	0.5
MMPC	0.3	0.1	0.0	0.1	0.5	2.7	0.3	0.1	0.0	0.1	0.5	2.7	0.7	0.8	0.4	0.0	0.4	1.1
LARS-EN	7.5	15.7	5.7	3.7	39.2	59.0	4.6	2.1	4.9	1.1	4.0	25.7	5.4	2.9	3.4	4.4	7.2	3.2
RFE (reduction 50%)	0.7	7.1	13.1	22.0	79.1	123.2	3.1	5.5	1.7	5.8	20.3	24.1	82.9	43.5	170.5	108.2	152.6	96.8
RFE (reduction 20%)	0.4	3.2	12.1	3.0	73.1	167.9	4.8	1.3	5.5	1.9	14.0	22.2	141.5	28.1	115.1	18.8	122.6	112.9
UAF-KW-SVM (50%)	2.0	1.5	76.5	6.8	124.9	172.8	1.7	3.3	14.9	2.6	37.7	120.2	8.8	83.0	24.1	257.0	83.5	97.3
UAF-KW-SVM (20%)	0.6	1.1	4.8	2.5	91.4	179.9	1.0	2.1	14.1	0.7	10.3	124.4	6.4	82.5	22.4	137.8	19.1	46.9
UAF-S2N-SVM (50%)	1.3	1.4	43.1	2.7	114.3	139.8	3.5	2.1	7.1	5.0	26.9	109.5	228.9	98.4	25.4	102.6	86.6	180.0
UAF-S2N-SVM (20%)	0.2	0.4	12.7	1.2	70.1	128.1	1.0	1.5	5.3	1.6	22.3	120.8	153.4	117.5	19.5	53.8	93.1	175.8

Alarm 10	Version 1 (original network)						Version 2 (original network + irrelevant variables)						Version 3 (weakened signal + irrelevant variables)					
	sample size																	
	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000	100	200	500	1000	2000	5000
UAF	22.1	26.5	32.2	30.2	33.5	38.0	22.1	26.5	32.2	30.2	33.5	38.0	22.5	25.2	32.0	27.1	32.4	37.3
UAF+Bonferroni	4.4	4.8	7.4	8.6	10.7	14.6	3.3	4.4	6.0	8.0	9.2	13.1	1.5	3.1	4.9	6.7	7.7	10.3
UAF+FDR	5.0	6.2	9.7	10.1	14.3	20.1	3.9	4.8	7.2	8.6	10.7	14.6	1.8	3.8	5.4	7.3	8.7	12.2
HITON-PC	3.7	0.8	0.1	0.0	0.3	0.3	2.4	0.5	0.1	0.0	0.3	0.3	1.8	0.9	0.2	0.1	0.6	0.2
HITON-PC-FDR	0.9	0.5	0.0	0.1	0.1	0.0	0.7	0.4	0.1	0.1	0.1	0.0	0.7	0.6	0.2	0.2	0.2	0.3
MMPC	3.7	0.8	0.2	0.3	0.4	0.1	2.6	0.5	0.2	0.2	0.4	0.1	2.6	0.7	0.3	0.4	0.5	0.3
LARS-EN	20.7	9.4	56.1	24.7	17.2	36.7	3.2	3.0	3.9	4.1	3.9	9.1	1.0	1.6	2.3	3.3	3.4	4.9
RFE (reduction 50%)	16.7	18.6	114.9	68.9	23.7	36.9	2.0	1.3	3.5	2.9	1.5	3.7	19.7	1.4	1.3	1.6	1.9	2.9
RFE (reduction 20%)	11.3	18.1	56.0	9.8	19.7	38.7	2.5	0.9	1.9	2.5	1.7	3.3	11.6	0.9	0.8	1.1	1.5	2.7
UAF-KW-SVM (50%)	13.5	4.0	32.6	51.4	49.7	35.9	3.4	3.4	5.6	5.4	9.1	15.4	13.7	3.7	4.4	5.7	7.6	10.6
UAF-KW-SVM (20%)	5.7	5.4	10.2	42.3	37.5	58.7	3.3	3.1	5.4	5.7	8.8	14.7	5.6	3.3	4.9	5.2	7.3	9.0
UAF-S2N-SVM (50%)	18.6	4.3	72.3	55.0	37.5	38.2	2.0	3.3	8.1	5.9	8.9	14.6	1.4	2.3	2.7	4.2	6.0	9.8
UAF-S2N-SVM (20%)	7.1	4.1	44.6	17.8	38.2	40.1	1.9	3.8	5.0	6.1	8.1	13.1	1.4	2.8	3.2	4.6	6.5	8.8

Small number of false positives
Large number of false positives

Table 8: Number of false positives (within weakly relevant variables) in the parents and children set for selected features. HITON-PC, HITON-PC-FDR, and MMPC are applied with $max-k=2$. For Version 4 of the network there are no weakly relevant variables. The color of each table cell denotes number of false positives with yellow (light) corresponding to smaller values and red (dark) to larger ones.

MMPC performing similarly) achieves excellent false positive rates better than those by FDR not only for weakly relevant but also for irrelevant features.

- (l) HITON-PC augmented with FDR pre-filtering behaves almost identically as regular HITON-PC except for the case with only irrelevant features in the data where HITON-PC without FDR admits a few false positives (Table 9).
- (m) State-of-the-art feature selection methods are prone to select very large numbers of irrelevant features (Table 9).

In conclusion, HITON-PC and by extension GLL algorithms (since the same fundamental mechanisms for variable inclusion and elimination are shared because of the GLL-PC template and admissibility requirements), have a very strong built-in capacity to control for false positives due to multiple comparisons. False positives due to multiple comparisons quickly vanish for $max-k$ 1 or

Lung_Cancer

FS method	Version 1 (original network)					Version 2 (original network + irrelevant variables)					Version 3 (weakened signal + irrelevant variables)					Version 4 (only irrelevant variables)								
	100	200	500	1000	2000	100	200	500	1000	2000	100	200	500	1000	2000	100	200	500	1000	2000				
UAF	0.2	1.5	0.2	0.5	0.8	0.7	411.6	488.6	446.0	422.7	409.0	403.1	488.8	471.6	424.9	413.2	407.9	397.8	411.6	488.6	446.0	422.7	409.0	403.1
UAF+Bonferroni	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.1	0.0
UAF+FDR	0.0	0.1	0.1	0.0	0.1	0.2	0.5	1.1	3.7	4.8	6.7	12.5	0.2	0.7	1.4	3.2	4.8	8.0	0.1	0.0	0.0	0.1	0.1	0.1
HITON-PC	0.0	0.0	0.0	0.0	0.0	0.0	1.5	0.0	0.0	0.0	0.0	0.0	8.6	2.9	0.9	0.2	0.4	0.0	9.8	5.8	6.4	6.9	6.1	6.2
HITON-PC-FDR	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.3	0.1	0.2	0.0	0.1	0.0	0.0	0.1	0.1	0.1
MMPc	0.0	0.0	0.0	0.0	0.0	0.0	1.7	0.0	0.0	0.0	0.0	0.0	8.7	3.4	0.8	0.2	0.4	0.0	8.8	6.4	7.2	6.3	6.7	6.3
LARS-EN	0.4	0.1	0.1	0.0	0.9	0.5	32.8	11.5	29.9	1.4	6.2	52.2	53.2	20.8	32.9	47.3	82.0	39.2	35.9	33.5	69.3	63.0	69.6	38.7
RFE (reduction 50%)	0.0	0.0	0.2	0.5	1.4	1.8	24.8	20.5	4.4	10.7	68.5	78.4	868.9	449.1	1741.5	1132.5	1600.0	1005.6	462.4	1084.1	971.3	918.2	1844.9	223.7
RFE (reduction 20%)	0.0	0.0	0.2	0.0	1.4	3.2	28.3	3.5	21.0	4.9	49.3	78.1	1548.4	252.6	1145.1	183.0	1282.2	1171.0	531.2	106.0	1488.3	103.8	849.0	112.7
UAF-KW-SVM (50%)	0.0	0.0	0.7	0.0	0.5	1.6	1.8	0.2	13.9	0.0	23.5	95.5	56.3	798.3	111.0	2593.7	800.6	801.1	809.3	319.6	1161.8	193.0	1676.1	886.0
UAF-KW-SVM (20%)	0.0	0.0	0.5	0.0	0.5	0.4	1.3	0.1	11.9	0.0	0.0	76.9	47.4	816.0	108.9	1215.2	3.4	12.4	971.0	346.6	1061.1	72.3	1283.0	870.6
UAF-S2N-SVM (50%)	0.0	0.0	0.0	0.0	0.7	0.2	29.8	5.8	3.5	0.0	5.6	49.9	2420.2	911.2	193.5	993.5	803.4	1604.8	676.2	1414.0	1666.8	2540.1	2491.3	1032.9
UAF-S2N-SVM (20%)	0.0	0.0	0.1	0.0	0.3	0.3	7.7	3.3	2.8	0.0	16.5	95.9	1624.5	1077.3	128.1	416.3	805.6	1608.2	1036.1	537.2	819.0	236.2	1279.7	990.0

Alarm10

FS method	Version 1 (original network)					Version 2 (original network + irrelevant variables)					Version 3 (weakened signal + irrelevant variables)					Version 4 (only irrelevant variables)								
	100	200	500	1000	2000	100	200	500	1000	2000	100	200	500	1000	2000	100	200	500	1000	2000				
UAF	0.0	0.0	0.0	0.0	0.0	0.0	392.1	412.9	411.6	414.1	382.0	381.0	408.7	422.8	417.9	399.9	380.0	367.1	392.1	412.9	411.6	414.1	382.0	381.0
UAF+Bonferroni	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.1	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.1	0.0
UAF+FDR	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.7	0.8	1.0	1.2	0.4	0.2	1.0	1.0	0.6	1.4	0.0	0.0	0.0	0.2	0.0	0.1
HITON-PC	0.0	0.0	0.0	0.0	0.0	0.0	22.8	3.8	0.8	0.9	0.0	0.1	26.4	6.5	4.4	3.3	1.8	1.0	23.4	9.7	6.8	7.2	8.8	6.1
HITON-PC-FDR	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.2	0.0	0.0	0.3	0.2	0.5	0.2	0.1	0.0	0.0	0.0	0.2	0.0	0.1	0.0
MMPc	0.0	0.0	0.0	0.0	0.0	0.0	29.0	3.4	1.2	1.0	0.0	0.1	37.2	5.9	4.4	3.2	2.8	1.0	26.8	10.4	8.5	7.8	9.3	5.8
LARS-EN	0.0	0.0	0.0	0.0	0.0	0.0	34.2	31.7	32.0	21.3	7.7	54.6	16.3	12.9	19.2	45.2	44.6	38.3	55.6	23.8	32.9	9.8	31.9	73.5
RFE (reduction 50%)	0.0	0.0	0.0	0.0	0.0	0.0	12.2	2.9	32.0	18.7	6.4	25.3	385.9	13.1	5.1	7.3	16.2	33.0	502.6	1819.2	80.6	1940.8	965.3	1090.3
RFE (reduction 20%)	0.0	0.0	0.0	0.0	0.0	0.0	11.7	2.6	8.3	23.3	7.2	20.8	201.2	7.4	2.2	8.5	16.7	30.8	264.8	732.4	735.0	633.0	790.6	237.2
UAF-KW-SVM (50%)	0.0	0.0	0.0	0.0	0.0	0.0	16.2	4.0	1.9	2.0	16.0	39.3	240.9	34.1	3.0	8.6	11.8	29.7	204.9	142.9	447.9	2604.5	811.1	1504.0
UAF-KW-SVM (20%)	0.0	0.0	0.0	0.0	0.0	0.0	15.3	2.1	0.9	5.5	13.6	35.8	74.0	23.4	3.0	3.2	8.8	11.3	219.5	102.5	124.6	1554.9	784.1	1230.1
UAF-S2N-SVM (50%)	0.0	0.0	0.0	0.0	0.0	0.0	3.1	5.2	30.2	2.5	20.0	37.2	9.6	2.7	1.1	2.9	3.3	26.4	615.1	32.7	265.8	1901.0	759.3	721.5
UAF-S2N-SVM (20%)	0.0	0.0	0.0	0.0	0.0	0.0	3.1	7.2	3.9	2.4	14.5	23.1	12.3	17.1	2.9	3.7	4.1	10.1	291.8	772.0	392.7	701.5	120.9	870.6

Small number of false positives

Large number of false positives

Table 9: Number of false positives (within irrelevant variables) in the parents and children set for selected features. HITON-PC, HITON-PC-FDR, and MMPc are applied with $max-k=2$. The color of each table cell denotes number of false positives with yellow (light) corresponding to smaller values and red (dark) to larger ones.

higher *regardless of sample size*. Given enough sample size ($\sim 1,000$ or more in the data tested), and by choosing 5% as the nominal α for all conditioning independence tests executed, the algorithm fully eliminates irrelevant features from its output without incurring a penalty in false negatives, even when irrelevant features are the majority among observed features. Parameter *max-k* controls the false positives due to both weakly relevant and irrelevant features. The false positive rate in this worst-case situation is in the presented experiments $\sim 5/8,000 = 0.000625$ which is much better than what the conservative Bonferroni-adjusted α guarantees, and *without* incurring false negatives (as both Bonferroni and FDR methods do). Both established feature selectors such as variants of UAF and newer ones are very sensitive to irrelevant features and produce large numbers of false positives. Given the attractive characteristics of FDR-augmented HITON-PC, we evaluate it with real data sets in Section 5.

4. Theoretical Analysis of GLL

In the present section we provide a theoretical analysis of the Generalized Local Learning algorithms.

4.1 Determinants of Quality of Statistical Decisions and Computational Tractability.

Parameters *max-k* and *h-ps*

On a rather superficial level when conditioning sets are large enough, statistical tests become less reliable. For example, as explained in Aliferis et al. (2010), cells in contingency tables used to calculate p-values of discrete tests of independence (such as the widely-used G^2 or X^2 test) become scarcely populated and this leads to unreliable test results. This motivates the heuristic practice of considering as unreliable and not executing a test in which the sample size is less than: (“number of cells to be fitted” $\cdot h-ps$), with parameter *h-ps* set to 10 by default in the PC algorithm (Spirtes et al., 2000) and 5 in GLL instantiations. Recall from Aliferis et al. (2010) that *h-ps* stands for “heuristic power size” and denotes the smallest sample size per cell in the contingency table of a reliable conditional test of independence. Moreover, when the conditioning set size is large enough to block all paths between a weakly relevant variable and the target, there is no need to exceed this conditioning set size because the resulting tests are redundant and the operation of the algorithm becomes unnecessarily slow. Thus it seems reasonable that we would wish to restrict the conditioning set size to not exceed this sufficient blocking size. This is accomplished by setting the value of parameter *max-k*. We will see however that *max-k* has a much more elaborate function than simply “trimming away” excessive computations.

In reality things are significantly more complicated because, as first pointed out by Spirtes et al. (2000), statistical reliability of a single test is a misleading concept in the context of complex constraint-based algorithms such as GLL. Standard statistical considerations of the type of testing a hypothesis once do not carry over well to the constraint-based algorithm setting. Similarly, running time is also a complex function of direct or indirect restrictions placed on number of tests and the number of variables with which to build such tests (i.e., the size of $TPC(T)$).

We first explain what happens when running semi-interleaved HITON-PC in faithful distributions (same arguments can be generalized to other GLL-PC and GLL-MB versions). Consider first that in the case of a strongly relevant feature S , when conducting just one test $I(S, T|\emptyset)$ for the purposes of inclusion of S in $TPC(T)$, regardless of how small power is, we should always execute this test because the worst that can happen is that we fail to include S in $TPC(T)$, whereas if we do not

execute the test and assume independence by default, we will surely miss it. In the context of *many tests* however, the notion of single-test reliability for S no longer applies. For example, when we consider a test that has the potential to reject S from $TPC(T)$ (where it was placed previously by a *different* test), by allowing the conditioning test size to grow large, the power is reduced (assuming monotonic association of S through the potentially multiple paths connecting S with T). Hence, we need to preserve the combined power (i.e., combination of individual powers of all tests applied to S) in order to not eliminate S from $TPC(T)$. Although these tests are highly correlated and combined power is larger than the product of powers of the same set of tests performed on independent samples, still the more tests are executed the smaller the combined power and the larger the possibility of falsely eliminating S becomes. The parameter $h-ps$ partially controls power because the larger it is, the smaller number of tests (that would eliminate S) are executed. However $h-ps$ should not be too large either because a strongly relevant S will not be included in $TPC(T)$ in the first place. Parameter $max-k$ also controls in part the number of tests allowed. $Max-k$ does not fully determine the number of tests because it specifies the dimensionality of allowed tests, not their total number. As $max-k$ grows, more tests for eliminating S from $TPC(T)$ are executed, thus the combined power drops. In summary, for a given distribution the number of tests performed is affected by $h-ps$, $max-k$ and the size of $TPC(T)$.

So far the discussion has centered on one type of conditional independence test, that is, tests where the candidate member of $PC(T)$, X , is a strongly relevant feature (type 1). This is the first of four types of conditional tests. The other three are: conditional independence tests where the candidate member of $PC(T)$, X , is a weakly relevant feature and some paths with T are not blocked by the conditioning set (type 2a), conditional independence tests where the candidate member of $PC(T)$, X , is a weakly relevant feature and all paths with T are blocked by the conditioning set (type 2b), and finally conditional independence tests where the candidate member of $PC(T)$, X , is an irrelevant feature (type 3).

The quality of conditional tests of the first type is determined by the *power* of the association of X with T given the conditioning set. Since not one but potentially many such tests are conducted, the combined power of all such tests determines whether X will be selected and stay in the $TPC(T)$ set. For example, variable X (a true member of $PC(T)$) will be considered for inclusion in $TPC(T)$ by HITON-PC with probability = power of detecting $\neg I(X, T)$ given the available sample size and test employed. However for X to stay in $TPC(T)$ until the algorithm terminates, and assuming B, C have entered $TPC(T)$, none of the tests $I(X, T|B)$, $I(X, T|C)$, $I(X, T|\{B, C\})$ must conclude independence. The power of each one of these tests can be lower or higher than the power of $I(X, T)$ and the combined power can quickly diminish, however several mitigating factors prevent this from happening. First, when using linear tests under common distributional assumptions such as multivariate normality, the necessary sample size to achieve desired level of power grows linearly to number of variables in the conditional set. Second, as explained earlier, conditional independence tests of the same variable and T in the same sample are highly correlated. Third, controlling the number of members of $TPC(T)$ by a good heuristic inclusion function reduces the total number of tests; such control occurs indirectly by putting first the true members of $PC(T)$ or members that block many variables. Fourth, the order of executing the tests and constructing conditioning sets is important for reducing the number of tests performed on strongly relevant variables. This is exemplified in semi-interleaved HITON-PC where new entrants in $TPC(T)$ are tested before current $TPC(T)$ members thus if the heuristic inclusion function is a good one, strongly relevant members are tested a smaller number of times at the elimination phase.

Returning our attention to the quality of statistical decisions for weakly relevant variables, we observe that when a conditioning set *does not* block all paths to/from T either for inclusion or for elimination purposes (type 2a), we are sampling under the alternative hypothesis (i.e., there exists association) and the determining factor for failing to reject the weakly relevant feature is the combined power which is determined by the same factors as elaborated for strongly relevant variables previously. The combined probability for rejection may be small for similar reasons as type 1 conditional independence tests (albeit higher than for strongly relevant features due to the fact that under a good inclusion heuristic weakly relevant features enter $TPC(T)$ later than strongly relevant ones and thus more tests are applied on each weakly relevant than on each strongly relevant feature on average).

However, when the conditioning set blocks all paths from/to T (type 2b), *then we sample under the null hypothesis* and the determining factor shifts from the combined power to the *combined* α (i.e., statistical significance). Given that the α for each conditional test is typically low (i.e., 5% or smaller) and that as the number of tests under the null increases, the combined α drops up to exponentially fast, and eliminating weakly relevant features occurs with high probability as the number of applied tests increases. In HITON-PC, the smaller is $h-ps$, the easier it is to include a weakly relevant feature (based on univariate association heuristic), whereas $max-k$ does not affect this function. In terms of rejecting a weakly relevant feature in $TPC(T)$, the larger $max-k$ and the smaller $h-ps$ become, the easier it is to eliminate a weakly relevant feature.

The quality of statistical decisions for type 3 of conditional independence tests, that is for irrelevant variables, is determined by the combined α since we *always* test under the null hypothesis. Because the combined α drops fast as the number of tests applied to each irrelevant variable (and these tests are abundant when even a handful of variables have been admitted in $TPC(T)$), the combined probability for admitting and not rejecting irrelevant variables is exceedingly small. However when no strongly (and thus no weakly) relevant feature exists, conditioning sets inside the $TPC(T)$ set become smaller as irrelevant variables are eliminated from it with the end result of leaving a small number of “residual” irrelevant features in the final output as evidenced in the simulation experiments of Section 3. By pre-filtering variables with an FDR filter (Benjamini and Yekutieli, 2001; Benjamini and Hochberg, 1995), we not only gain the security that if the data consists exclusively of irrelevant variables fewer or no false positives will be returned, but also we can use $max-k$ to control sensitivity and specificity trading weakly relevant false positives for strongly relevant true positives and vice versa (i.e., without worrying about adversely trading off irrelevant features).

Finally, the total number of tests is determined by both parameters $h-ps$ and $max-k$, in a non-monotonic manner. That is, whenever $h-ps$ is extremely large it effectively disallows most tests and the algorithm quickly terminates returning the empty set regardless of $max-k$. For medium/small values of $h-ps$, more tests are executed, more variables enter $TPC(T)$, and many tests are executed before $TPC(T)$ is finalized. $Max-k$ modifies this number by potentially restricting the number of tests. When $h-ps$ is very small, tests are allowed with very large conditioning tests and as long as $max-k$ does not disallow them, the total number of tests grow very large.

		Number of conditional independence tests		Cost of conditional independence tests		Number of false positives (fp) and false negatives (fn)*				
Lung_Cancer	max-k	HITON-PC	MMPC	max-k	HITON-PC	MMPC	max-k	# of fn	# of fp	
	1	4,028	5,683	1	7,257	8,900	1	1	13	
	Target variable #1	2	12,328	14,577	2	33,018	38,892	2	1	0
	Number of members in PC set = 26	3	73,554	77,885	3	277,922	294,211	3	1	0
	4	250,560	259,099	4	1,181,889	1,225,682	4	3	0	
Alarm10	max-k	HITON-PC	MMPC	max-k	HITON-PC	MMPC	max-k	# of fn	# of fp	
	1	457	490	1	545	585	1	1	2	
	Target variable #199	2	470	496	2	608	652	2	1	0
	Number of members in PC set = 6	3	491	521	3	692	752	3	1	0
	4	496	527	4	717	782	4	1	0	

* Results are same for HITON-PC and MMPC for number of false positives and false negatives

Figure 2: Efficiency of HITON-PC versus MMPC.

4.2 Efficiency and Heuristic Robustness of HITON-PC Versus MMPC

Figure 2 presents the number and cost² (proportional to time) of conditional independence tests performed by semi-interleaved HITON-PC versus MMPC in the 2,000-sample data set from the *Alarm10* and *Lung_Cancer* networks. As can be seen, HITON-PC performs fewer tests on average while achieving the same performance as MMPC. We notice that the max-min association heuristic closely reflects the logic behind the combined probability for error for the weakly relevant features. MMPC when testing under the alternative hypothesis (i.e., strongly relevant features, or unblocked weakly relevant ones) requires measuring all relevant associations, whereas HITON requires just the univariate ones *for inclusion purposes*. However semi-interleaved HITON tries to eliminate the newly included variable immediately upon inclusion and thus effectively conducts a similar number of tests as MMPC. Both algorithms when testing under the null hypothesis (irrelevant or fully-blocked weakly relevant features) on average execute the same number of tests. The max-min association inclusion heuristic is a priori more prone to basing its decisions for inclusion in $TPC(T)$ on less statistically reliable criteria. This is because the more associations are considered and the larger the conditioning sets are, the higher variance in the minimum association estimates is expected, making the maximum of such associations over all variables considered more prone to sampling error (i.e., it is likely to be overfitted to the sample). Because of better robustness of the univariate association relative to the weakest association over many conditional associations true members of $PC(T)$ may enter the $TPC(T)$ set earlier. However both HITON-PC and MMPC exhibit similar performance in real and simulated data sets, demonstrating that the theoretical problem with max-min association is in practice very rare.

4.3 Synthesis and Problems for Inclusion Heuristics; Constructing New Inclusion Heuristics

A problem when inducing local neighborhoods and particularly Markov blankets is that of *information synthesis*. The problem consists of a variable X that is not in $PC(T)$ having higher association (univariate or conditional on some subsets) with T than members of $PC(T)$ (for a concrete example see Figure 13). We will call such variables, *synthesis variables*. Synthesis variables were identified

2. The cost of a conditional independence test is calculated as the number of variables participating in it (excluding target variable). For example, univariate tests have cost = 1, tests with conditioning on two variables have cost = 3.

as major problems for algorithms such as IAMB (Tsamardinos and Aliferis, 2003; Tsamardinos et al., 2003a) or GS (Margaritis and Thrun, 1999) that induce Markov blankets and do so by conditioning in their inclusion phase on all variables in the tentative $MB(T)$. Because of the requirement to condition on all variables in the tentative $MB(T)$, the sample requirements grow exponentially fast to the size of the tentative $MB(T)$ and thus it is absolutely imperative to keep out of it synthesis variables since they unnecessarily increase the sample requirements to the point that the algorithm may need to stop executing conditional independence tests (and either halt or output the tentative $MB(T)$ as best but flawed estimate of the true $MB(T)$).

With regards to GLL algorithms, most efficient operation is achieved when the variables that alone or in combination have the property that block the largest fraction of weakly relevant variables, enter first in $TPC(T)$ (even if they are not strongly relevant themselves). Synthesis variables may or may not have this property, so synthesis may or may not be a problem for a specific GLL algorithm based on characteristics of the specific data in hand.

Construction of new inclusion heuristics may be required in difficult cases where the univariate and max-min heuristics do not work well leading to very slow processing time and very large $TPC(T)$ sets, in order to make operation of local learning tractable. In practice, both the univariate and max-min association heuristics work very well with real and simulated data sets, so we do not pursue here implementation and testing new heuristics in artificial problems, although we recognize the possibility of such need in future problematic data distributions. We outline here, in broad strokes, general strategies for creating new inclusion heuristics for such cases:

1. *Random heuristic search informed by standard heuristic values.* This strategy is based on using one of the usual heuristics to rank candidate variables and making selection decisions based on random selection of a candidate variable with probability proportional to the original heuristic value. This enables using the older heuristic as a starting point but allowing occasionally deviations from it to explore the possibility that lower-ranked candidates may have better potential as blocking variables. A simulated-annealing determination of probability of selection (or other efficient stochastic search algorithms) can be pursued as well.
2. *Constructing new heuristic functions by observing blocking capability* (in terms of candidate variables blocked by conditioning sets in which V is a member) or *probability of a variable V to remain in $TPC(T)$* . The empirical observations can be collected from a variety of tractable sources: either from a single incomplete run of the algorithm (i.e., without waiting to terminate), or in other data sets characteristic of the domain, or in multiple runs on smaller (randomly chosen) subsets of the original feature set. The new heuristic function F can be constructed as the conditional probability:

$$F(V_i) = P(V_i \in TPC(T) | h(V_i))$$

where $h(V_i)$ is the original heuristic value of variable V_i , or the proportion of candidates blocked by a conditioning set containing V_i :

$$F(V_i) = \sum_{k=1}^M N_k(V_i) / M$$

where $N_k(V_i)$ is the number of candidate variables blocked by a conditioning set that contains variable V_i in trial k .

3. *Exploiting known domain structure.* When properties of the causal structure of the data generating structure and/or distributional characteristics are known, one can use this information alone or in conjunction with the previous two strategies to derive more efficient heuristics.

We note that developing an inclusion heuristic that leads to efficient execution of GLL is not always feasible since the very problem of finding the features with direct edges with the target is intractable in the worst case (e.g., consider a graph that is fully connected). In some cases, as we will show in Section 6, *it is possible to transform an intractable local learning problem into a tractable one by employing a global learning strategy (i.e., exploiting asymmetries in connectivity).*

4.4 Inductive Bias of GLL

Informally the inductive bias of GLL is that it seeks a balance of false negatives for strongly relevant variables with false positives for weakly relevant and irrelevant variables. The main regulating parameters (for standard inclusion heuristics, elimination and interleaving strategies) are h -ps and max -k. In practice, the algorithms tested in our work to date reveal higher sensitivity to max -k and thus at first approximation we treat optimization of this parameter as having higher priority. Smaller max -k empirically decreases false negatives and increases false positives overall. Larger max -k increases the false negatives and decreases the false positives. GLL in moderate to large samples achieves small numbers of false negatives and small numbers of false positives. In very small samples GLL prefers false positive errors than false negative ones when max -k is small. This occurs because given *some* evidence in favor of $PC(T)$ membership (provided by lower-dimensional and thus more sample efficient) tests of a variable X but *no reliable proof* to the contrary (provided by omitted higher-dimensional and thus unreliable tests), the algorithm outputs X as member of $PC(T)$. A similar behavior exists for the $MB(T)$ versions (with respect to $MB(T)$ membership). Notice that as max -k grows many more tests can be executed provided that a liberal h -ps is chosen, and these tests can be used to eliminate both weakly relevant as well as strongly relevant features in $TPC(T)$. The choice of a more liberal h -ps default value in GLL (compared to the more stringent value in the published implementation of PC algorithm) allows a more effective control of the tradeoff between false positives and false negatives in small samples by changing values of max -k.

By contrast, the SGS and PC algorithms (Spirtes et al., 2000) given *no evidence* in favor of membership of X in $PC(T)$ and *no reliable proof* to the contrary, assumes that X has a common edge with T . IAMB (Tsamardinos and Aliferis, 2003; Tsamardinos et al., 2003a) to the contrary, given *some* reliable evidence in favor of a variable X belonging to $MB(T)$ but *no reliable proof* to the contrary, outputs X as member of $MB(T)$ if X is in the tentative Markov blanket $TMB(T)$ and is agnostic with respect to membership in $MB(T)$ if X is outside $TMB(T)$. Bayesian scoring methods in small samples are dominated by their priors and typically they prefer sparse networks which lead to fewer false positives and more false negatives.

4.5 Reasons for Good Performance of Non-Symmetry Corrected Algorithms

The empirical evaluations in part I of this work (Aliferis et al., 2010) have shown that the addition of symmetry correction adds little to quality, while it detracts from computational efficiency. Evidently very often $EPC(T) \approx PC(T)$ in real-life distributions and targets of interest. In addition, due to imperfect power to detect and return strongly relevant features, applying symmetry correction leads to reduced power and increased false negatives.

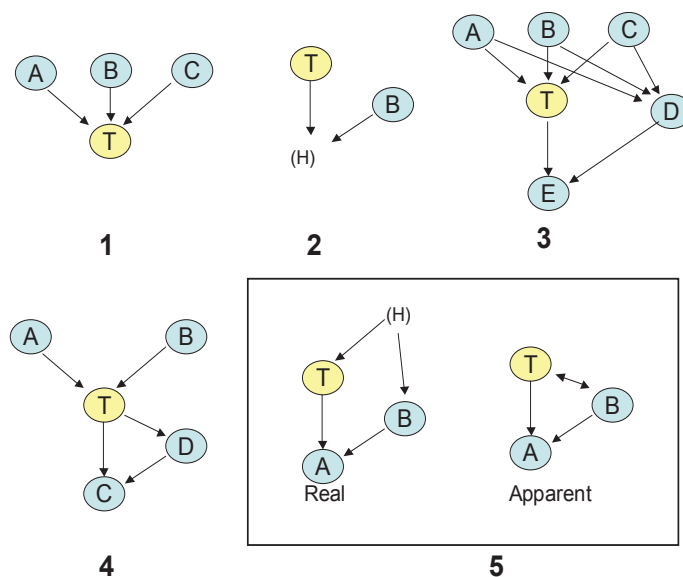


Figure 3: Scenarios explaining good empirical performance of $PC(T)$ set for classification.

4.6 Reasons for Good Performance of the $PC(T)$ Set Instead of the $MB(T)$ Set for Classification

According to the theoretical results summarized in Aliferis et al. (2010), under broad assumptions spouses are needed for optimal classification performance. Given that in the majority of data sets tested in Aliferis et al. (2010) as well as the experiments in Section 2 of the present paper, when the set of parents and children is used instead of $MB(T)$ it produces equal or almost equal performance, more compact feature sets and faster feature selection times than inducing the full $MB(T)$ (i.e., both $PC(T)$ and $MB(T)$ estimated under the same assumptions of the theory that predicts that $MB(T)$ is needed for optimal feature selection). In this sub-section we provide likely explanations for the empirically excellent performance of substituting the set $PC(T)$ in place of $MB(T)$ for classification (apart from the obvious possibility that spouses may be much fewer and with smaller predictive value than parents and children). Figure 3 describes visually five plausible scenarios explaining the phenomenon.

The first scenario corresponds to the situation whereby the target variable T does not have children (and thus no spouses) by virtue of domain constraints. Such situations happen when the target variable is a variable preceded in time by all other variables (e.g., patient outcome on the basis of earlier observations); or when naturally the target variable cannot have children (e.g., the target being meaning category of a text document as a function of patterns of presence/absence of words in the text). The second scenario describes the situation where a child is not observed (hidden) in the data set and thus the spouse B cannot be made informative for the target and thus it can neither be detected nor can it enhance a classifier built from the data. The third scenario describes the situation where a spouse has connecting paths to the target but these cannot be blocked simultaneously because of small sample size and/or choice of $max-k$. Hence GLL-PC could admit the spouse D as a member of $PC(T)$. The fourth scenario simply shows a case where a spouse is also a child (or parent) and thus will be a member of $PC(T)$ as well as $MB(T)$. Finally the fifth

scenario shows that an unmeasured variable may make a spouse appear as having a direct edge to or from the target (and thus are detectable by GLL-PC).

We note that in practical data analysis and evaluations when both $PC(T)$ and $MB(T)$ are induced and are found to have similar classification performance, typically $MB(T)$ is much larger than $PC(T)$. However this may be a reflection of the inductive bias of GLL which prefers to admit potential false positives if they cannot be shown for sample size reasons to be independent of the target.

Finally note that explanations #1, 2, 3, and 4 are special cases of the assumptions of the Markov blanket induction theory and thus they do not refute these assumptions (whereas #5 violates causal sufficiency). In the discussion section we consider additional situations with violations of GLL assumptions.

4.7 Error Estimation Problems in Wrapping and Standard Filters Due to Small Sample Size. GLL Filtering is Less Sensitive to Error Estimation Difficulties and Robust to Small Samples

Wrapping has been praised as a feature selection methodology for its ability to tailor the feature selection to the inductive bias of the classifier(s) of choice as well as to the loss function of interest (Kohavi and John, 1997). Occasionally, this property will work against the analysis (see Section 7 for example for how it can jeopardize causal discovery). On the other hand, wrapping has been criticized for its very large computational cost as well as on the grounds that it is subject to No Free Lunch Theorem limitations (i.e., a priori all wrappers are equally good, making it hard to find the right wrapper for the distribution, loss function and classifier(s) of interest) (Tsamardinos and Aliferis, 2003). In the present section we explain what we believe is perhaps the most serious practical shortcoming of wrapping feature selection methods, namely that *they rely on error estimation procedures that are often unreliable because of small sample sizes*. The difficulties that will be presented here help explain the sometimes poor performance of some of the feature selection algorithms in the evaluation part (Aliferis et al., 2010). In contrast, we will show that GLL filtering is resistant to these problems.

Recall that the critical point when applying error estimators is to have a sufficiently small variance and to be unbiased or to correct for any bias, as for example is the case of the (biased) Bootstrap estimator. Consider an idealized example where a greedy (steepest-descent) backward selection wrapper algorithm is applied on faithful data that contains 5 irrelevant features I_1, \dots, I_5 and one strongly relevant feature S .

Assume that in reality the optimal feature set consisting of only the strongly relevant feature S gives a predictor model with true error measured by AUC is 0.75 in the large sample (i.e., in the distribution where the data is sampled from). For all practical unbiased error estimators, because of variability in the estimates of error due to small sample sizes, and because of potential sensitivity of the classifier employed to irrelevant features, some subsets that contain S will have error estimates in small sample situations that are larger and some smaller than the true AUC of 0.75. The backward wrapping starts by eliminating one variable at a time producing feature sets and corresponding predictor models and by eliminating the feature that decreases error the most relative to the starting model that contains all features. As a result, a feature set can be chosen, not because the error is truly decreased if we remove any more features, but because the error estimates vary and the backward wrapper (naively) does not take this into account. If the wrapper is configured

Action	Decision	Notes
Rank variables according to univariate association with target T	S (association = 0.8) I_1 (association = 0.3) I_2 (association = 0.1) I_3 (association = 0.1) I_4 (association = 0.05) I_5 (association = 0.0)	Some associations of irrelevant variables are non-zero due to sampling variation
Test S for inclusion: $\neg I(S, T)$	Admit S in $TPC(T)$	Assuming S is a strong predictor of the target, the power of the univariate test will be sufficient to reject independence
Test I_1 for inclusion: $I(I_1, T)$	Eliminate I_1	Test will be correct with probability $1-\alpha$ (typically 0.95)
Test I_2 for inclusion: $I(I_2, T)$	Eliminate I_2	Test will be correct with probability $1-\alpha$ (typically 0.95)
Test I_3 for inclusion: $\neg I(I_3, T)$	Consider I_3	Assume we were unlucky and had a false positive
Test I_3 for inclusion: $I(I_3, T S)$	Eliminate I_3	Test will be correct with probability $1-\alpha$ (typically 0.95). Very unlikely (probability = 0.0025) that I_3 will pass through second test
Test I_4 for inclusion: $I(I_4, T)$	Eliminate I_4	Test will be correct with probability $1-\alpha$ (typically 0.95)
Test I_5 for inclusion: $I(I_5, T)$	Eliminate I_5	Test will be correct with probability $1-\alpha$ (typically 0.95)
Test S for final elimination: no test to be made	Accept S	
Return $\{S\}$ as final output		

Table 10: Trace of semi-interleaved HITON-PC without symmetry correction (i.e., GLL-PC-nonsym subroutine) showing insensitivity to error estimation difficulties that affect wrappers.

to employ statistical significance tests each time it compares estimates of error between pairs of feature sets and corresponding classifiers, because statistical tests of error estimate differences are often underpowered (which is another manifestation of the large variance in error estimates) such tests will often fail to reveal true differences. Thus the wrapper can falsely conclude that two models have same error when in reality they do not. This will entail choosing wrongly the smallest of the two and eliminating valuable features. Also due to multiple comparisons, such an algorithm will falsely conclude for a proportion of feature sets that a difference in predictor model performance is statistically significant thus continuing removal of relevant features when they should not be removed.

We emphasize that this problem is not present in wrapper methods only. In traditional feature ranking methods, the above problem is also present but often ignored in the sense that many studies on feature ranking algorithms produce a performance-to-feature-number plot, with performance estimated on a single data set. However the practical data analysis problem of how to select a specific number of features that achieves at most some desired error is left unspecified and in fact subject to the same error estimation difficulty that applies to wrapping. Moreover, in recent algorithms such as RFE, the problem is acknowledged implicitly in the applied examples provided by the authors of the method, since feature sets are reduced by for example 50% in each iteration of the

algorithm creating a new subset of features examined by cross-validation by the algorithm (Guyon et al., 2002). This is done to reduce overfitting of selected feature set to the data because of the large variability of error estimates. As evidenced by the evaluations presented in Aliferis et al. (2010), it is possible to improve on traditional wrapping, ranking and RFE selection by applying statistical tests of difference of error estimates, or by increasing/decreasing the granularity of feature selection (i.e., proportion of features removed at each iteration). Still the produced feature sets are not optimal in parsimony. The numbers of strongly relevant, weakly relevant and irrelevant features is not critical to the existence of the problem, neither is the type of wrapper (forward, backward, forward-backward, GA, etc.) as long as some basic requirements are met: error estimation is not perfect but subject to sampling variability due to small sample, and enough features exist in data for enough error estimate comparisons to be spurious.

Contrary to the above, GLL filtering relies little on error estimation³ and uses robust mechanisms to control false negatives and false positives separately for strongly relevant, weakly relevant and irrelevant features respectively. In Table 10 we give a concrete demonstration of how semi-interleaved HITON-PC (without symmetry correction for simplicity) is less prone to errors in the same example. The critical observation is for an irrelevant feature to enter $TPC(T)$ and stay in it, it has to survive multiple (i.e., $2^{|TPC(T)|}$) tests of conditional independence and each such test has probability $1 - \alpha$ to leave the irrelevant feature in $TPC(T)$. The total probability of failing to reject the irrelevant variable thus grows up to exponentially small to the number of tests performed and is independent of the sample size. In our simplified example with just one strongly irrelevant feature inside $TPC(T)$, each irrelevant feature has probability of entering and staying in $TPC(T)$ of at most $\alpha^2 = 0.0025$. This is true regardless of whether sample size is 10,000 samples or just 10 samples.

5. Algorithmic Extensions to GLL

In the present section we introduce algorithmic extensions to the Generalized Local Learning algorithms: parallel and distributed local learning and FDR pre-filtering.

5.1 Parallel and Distributed Local Learning

Following ideas for parallelizing the IAMB algorithm for $MB(T)$ estimation (Aliferis et al., 2002), we introduce a coarse-grain parallelization of GLL-PC that addresses two problems: (a) the data does not fit into fast memory (RAM), and (b) even if the data fits, we wish to speedup execution time by parallel processing. We allow for the possibility that the user may have access to just one node or, alternatively, may have access to several nodes arranged in a parallel cluster. The algorithm presented can return $PC(T)$ and can run with any instantiation of GLL-PC. The algorithm is designed to be correct provided that no symmetry correction is required (i.e., in distributions where $EPC(T) \equiv PC(T)$). Correct parallel/distributed versions in distributions where symmetry correction is needed can also be obtained as can algorithms that parallelize $MB(T)$ induction. In the present paper we only discuss parallel GLL-PC without symmetry correction because of its concep-

3. Notice that some reliance on error estimation exists in domains where a suitable $max-k$ and α are not known and need be optimized by cross-validation. The corresponding number of parameterizations is very small however (typically at the order of 10 combined parameter configurations) and thus error estimation is less likely to lead the algorithm astray. The same is true for the optional wrapping step in GLL-MB which selects features from a highly reduced set compared to the original feature set (notice that this wrapping step is seldom needed in practice and is reserved for higher sample settings).

Chunked Parallel GLL-PC Algorithm (not symmetry corrected)

Input: Dataset D , target variable T , desired number of data chunks ch .

1. Split the data D into ch arrays C_i of equal size, such that each array contains a non-overlapping subset of the variables plus T .
2. For all i , compute $ChunkPC_i(T) \leftarrow \text{GLL-PC-nonsym}(T, C_i)$
3. $L \leftarrow \text{GLL-PC-nonsym}(T, \cup_i ChunkPC_i(T))$
4. Return L and exit

Figure 4: Chunked Parallel GLL-PC algorithm (not symmetry corrected).

tual and implementation simplicity and speed, because it can be used for both causal discovery and prediction, and because as demonstrated empirically (Aliferis et al., 2010), many real distributions behave consistently with being “symmetrical” (i.e., $EPC(T) \equiv PC(T)$).

Chunked Parallel GLL-PC algorithm (not symmetry corrected): This algorithm assumes that one has access to several nodes and that the data can fit to the available memory once distributed, while it may or may not fit to a single node. Initially the algorithm divides the input data D into ch chunks C_i such that every C_i includes all cases, but only a subset V_i of the variable set V plus T . For simplicity we assume that each chunk has an equal number of features (that can be determined, for example, by the maximum size that can be processed in fast memory or the number of available computer nodes in a parallel implementation). Variations where unequal variable allocations are employed can be easily obtained in similar fashion. Then GLL-PC-nonsym is run on each chunk (as indicated by the extra input argument C_i) returning $ChunkPC_i(T)$ (i.e., parents and children of T in chunk C_i). Next, GLL-PC-nonsym is run on one node with the union $\cup_i ChunkPC_i(T)$, it obtains a local neighborhood L , and terminates by outputting L . Figure 4 gives the parallel GLL-PC high-level pseudo-code. Step #2 is the parallel step.

We note that a potential problem with chunked GLL-PC is that the tentative neighborhood in some chunk(s) may grow very large (up to the size of the chunk in the worst case) while the true neighborhood across all variables may be very small. This creates the possibility of overflow both in the sense of data not fitting in a single node and in the sense of not having enough sample size to perform reliable statistical inferences.

Theorem 1 *Chunked parallel GLL-PC without symmetry correction is sound given the sufficient conditions for soundness of GLL-PC and the requirement that in the generating distribution P , $PC(T)$ is the same as the Extended $PC(T)$ (see definition of $EPC(T)$ in Aliferis et al. 2010).*

Proof In each chunk, GLL-PC-nonsym will identify all true members of $PC(T)$ that are in the chunk (because these can never be rendered independent of T , according to Theorem 1 in Aliferis et al. 2010) and some false positives which cannot be eliminated without conditioning on $PC(T)$ members that belong to another chunk. Thus in step #3, GLL-PC-nonsym is executed on a superset of $PC(T)$. By definition, all non-members of $PC(T)$ can be rendered independent of T conditioned on some subset of $PC(T)$ as long as $PC(T) \equiv EPC(T)$. Since $PC(T) \equiv EPC(T)$, the identified $PC(T)$ will be correct. ■

The complexity of Chunked Parallel GLL-PC without symmetry correction is in the worst case exponentially slower than running GLL-PC on all data. This is because the complexity of GLL-PC

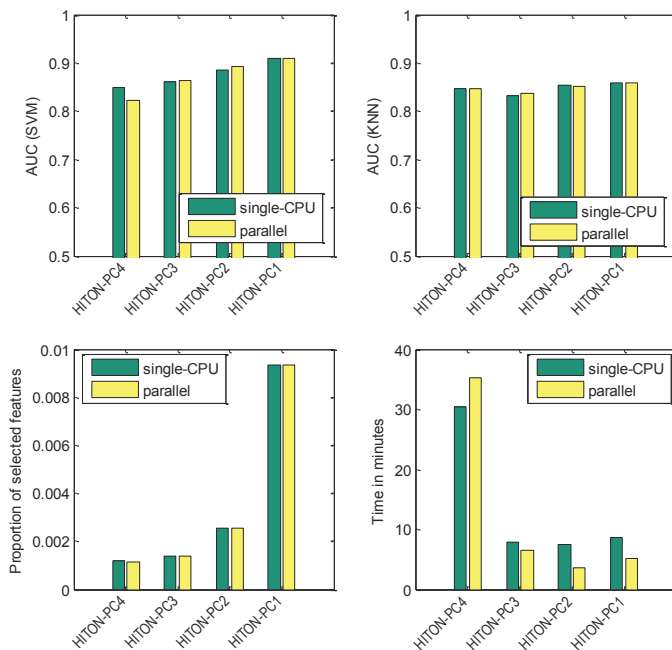


Figure 5: Results of application of single-CPU and parallel versions of semi-interleaved HITON-PC on the four largest real data sets (*Ohsumed*, *ACPJ_Etiology*, *Thrombin*, and *Nova*). Average results over 4 data sets are shown. The following versions of HITON-PC are used: HITON-PC4 ($max-k=4$, $\alpha=0.05$), HITON-PC3 ($max-k=3$, $\alpha=0.05$), HITON-PC2 ($max-k=2$, $\alpha=0.05$), HITON-PC1 ($max-k=1$, $\alpha=0.05$).

is worst-case exponential to the size of $TPC(T)$ and while $TPC(T)$ in all data can be very small, in some chunks $TPC(T)$ can be as large as the chunk itself. When however local neighborhoods in each chunk are smaller than the global $TPC(T)$ and since GLL-PC is worst-case exponential, the algorithm can also be exponentially faster than running GLL-PC on all data. This is in sharp contrast with parallel IAMB where both the speedup is linear to the number of chunks in the best case (upper bound on the speed-up factor is ch) and worst-case running time is a small constant multiple of running the algorithm on all data (Aliferis et al., 2002).

Chunked Distributed GLL: When we run the algorithm with data already distributed, the data splitting and transfer step #1 (as well as associated transfer cost) is omitted. Typically we will need to link the distributed data using a suitable common key. For example consider a large organization wishing to analyze data in order to find determinants of production costs overall many and geographically dispersed branches, each with its own local data set and different recorded features. An appropriate key might be time label of observations. Another example is hospital patient data distributed among numerous local databases in different units and labs of the hospital, where patient id is a suitable key.

Chunked GLL with single CPU: This variant assumes access to one CPU only and addresses the problem of data not fitting in the fast memory. By processing parts of the data sequentially and obtaining a small superset of $PC(T)$ each time, a much larger data set than what fits in fast memory can be analyzed.

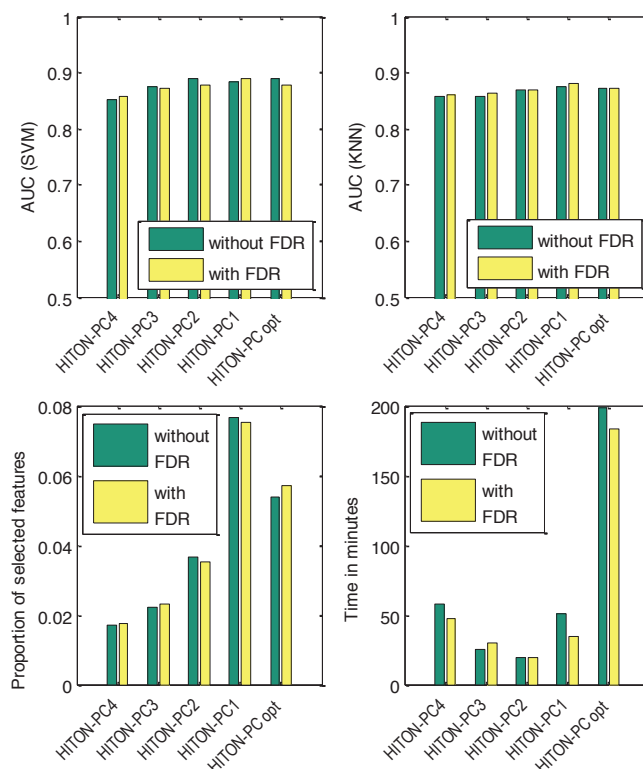


Figure 6: Results of application of semi-interleaved HITON-PC with and without FDR correction on 13 real data sets. Average results over the data sets are shown. The following versions of HITON-PC are used: HITON-PC4 ($max-k=4$, $\alpha=0.05$), HITON-PC3 ($max-k=3$, $\alpha=0.05$), HITON-PC2 ($max-k=2$, $\alpha=0.05$), HITON-PC1 ($max-k=1$, $\alpha=0.05$), HITON-PC opt ($max-k$ and α are optimized over values $\{1, 2, 3, 4\}$ and $\{0.05, 0.01\}$, respectively, by cross-validation to maximize SVM classification performance).

We now apply a parallel version of semi-interleaved HITON-PC on the four largest real data sets (*Ohsumed*, *ACPJ_Etiology*, *Thrombin*, and *Nova*) of the empirical evaluation in Aliferis et al. (2010). We use 10 CPU's on the ACCRE cluster described in Aliferis et al. (2010). As can be seen in Figure 5 the parallel version achieves the same parsimony and classification performance as the single-CPU application with speedup for three out of four versions of HITON-PC (see Figure 5). P-values from the permutation test of the null hypothesis that single-CPU and parallel GLL-PC algorithms achieve the same performance are 0.7468 (for SVM classification), 0.4950 (for KNN classification), 0.2408 (for proportion of selected features), and 0.6374 (for running time in minutes). We note that running times for HITON-PC algorithm in this subsection are less than in the remainder of the paper because these experiments were executed on the most recent version of the ACCRE cluster.

5.2 FDR pre-Filtering

As explained in Section 3, in simulated and resimulated data sets with weak-signal/small sample and in all-irrelevant features situations, removing features using false discovery rate control can

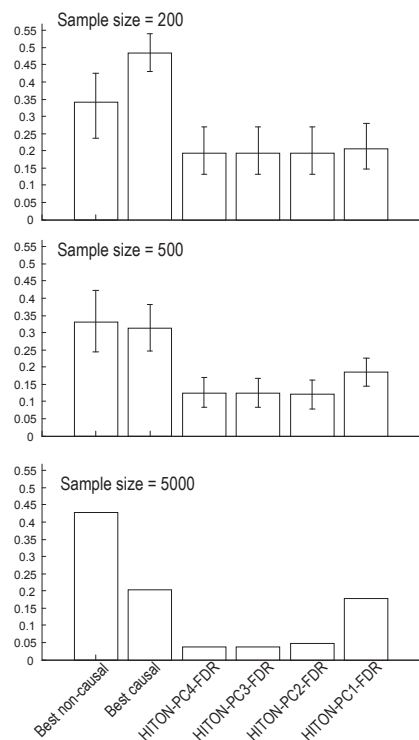


Figure 7: Graph distances averaged over all 9 simulated and resimulated data sets, all selected targets in each data set, and multiple samples of a given size. The following versions of semi-interleaved HITON-PC with FDR correction are used: HITON-PC4-FDR ($max-k=4$, $\alpha=0.05$), HITON-PC3-FDR ($max-k=3$, $\alpha=0.05$), HITON-PC2-FDR ($max-k=2$, $\alpha=0.05$), and HITON-PC1-FDR ($max-k=1$, $\alpha=0.05$). “Best causal” is the best causal feature selection algorithm among techniques that do not incorporate FDR. “Best non-causal” is the best non-causal feature selection algorithm. See Aliferis et al. (2010) for a detailed list of algorithms.

improve the number of false positives in HITON-PC and MMPC. We applied HITON-PC with FDR pre-filtering in all real data sets of Aliferis et al. (2010). As can be seen in Figure 6, this enhancement does not entail improvements in parsimony, classification performance or running time in the data sets tested. P-values from the permutation test of the null hypothesis that GLL-PC algorithms with and without FDR correction achieve the same performance are 0.5254 (for SVM classification), 0.3698 (for KNN classification), 0.9426 (for proportion of selected features), and 0.3776 (for running time in minutes). Since however the algorithm exhibits small sensitivity to false positives due to multiple comparisons when many irrelevant features are expected and few relevant features are present, we recommend pre-filtering with FDR. Alternatively, if one gets a few variables combined with error estimates consistent with uninformative classifier, then re-running standard GLL with FDR pre-processing can be tried.

When evaluating local causal discovery performance in the simulated data of Aliferis et al. (2010), semi-interleaved HITON-PC with FDR pre-processing achieves dramatically better performance than other algorithms including other HITON and MMPC variants with respect to graph

LGL: Local-to-Global Learning

1. Find $PC(X)$ for every variable X in the data using an admissible instantiation of GLL-PC and prioritizing which variables to induce $PC(X)$ for, according to a prioritization strategy.
2. Piece together the undirected skeleton from the local GLL-PC results.
3. Use any desired arc orientation scheme to orient edges.

Figure 8: Local-to-Global Learning (LGL) algorithmic schema.

MMHC Global Learning Algorithm

1. Find $PC(X)$ for every variable X in data using MMPC (without symmetry correction) and lexicographic prioritization.
2. Piece together the undirected skeleton using an “OR rule” (an edge exists between A and B iff A is in $PC(B)$ or B is in $PC(A)$).
3. Use greedy steepest-ascent TABU search and BDeu score to orient edges.

Figure 9: MMHC global learning algorithm as an instance of LGL.

HHC Global Learning Algorithm

1. Find $PC(X)$ for every variable X in data using semi-interleaved HITON-PC (without symmetry correction) and lexicographic prioritization.
2. Piece together the undirected skeleton using an “OR rule” (an edge exists between A and B iff A is in $PC(B)$ or B is in $PC(A)$).
3. Use greedy steepest-ascent TABU search and BDeu score to orient edges.

Figure 10: HHC global learning algorithm as an instance of LGL.

distance score, which indicates average causal proximity to the target of the returned variables. Specifically, in large sample ($N=5,000$) HITON-PC with FDR correction achieves up to 5-fold reduction in the graph distance score relative to the best non-FDR filtered causal algorithm and up to 9-fold reduction compared to the best non-causal algorithm. In small sample ($N=200$) the reduction in both cases is 2-fold. P-values from the permutation test of the null hypothesis that the best non-causal algorithm performs the same as the average HITON-PC with FDR correction are <0.0001 for sample sizes 200, 500, and 5,000. P-values for comparison with the best causal algorithm are <0.0001 , 0.0030, and <0.0001 for sample sizes 200, 500, and 5000, respectively. See Figure 7. This improvement incurs only a very small decrease in sensitivity as evidenced by small concurrent increases in false negatives.

6. Spanning Local to Global Learning

In the present section we investigate the use of local learning methods (such as GLL) for global learning in a divide-and-conquer fashion. We remind that a major motivation for pursuing local causal learning methods is scaling up causal discovery and causal feature selection as explained in Aliferis et al. (2010). Although similar concepts can be used for region learning, we will not address this type of discovery problem here. The main points of the present section are that (a) the local-to-global framework can be instantiated in several ways with excellent empirical results; (b) an important previously unnoticed factor is the variable order in which to execute local learning, and (c) trying to use non-causal feature selection in order to facilitate global learning (instead of causal local learning) is not as a promising strategy as previously thought.

6.1 General Concepts

A precursor to the main idea behind the local-to-global learning approach can be found in SCA (Friedman et al., 1999), where a heuristic approximation of the local causes of every variable constrains the space of search of the standard greedy search-and-score Bayesian algorithm for global learning increasing thus computational efficiency. Given powerful methods for finding local neighborhoods, provided by the GLL framework, one can circumvent the need for uniform connectivity (as well as user knowledge of that connectivity) and avoid the application of inefficient heuristics employed in SCA thus improving on quality and speed of execution. Figure 8 provides the general algorithmic schema term LGL (for local-to-global learning). Steps #1-3 can be instantiated in numerous ways. If an admissible GLL-PC (as defined in Section 4 of Aliferis et al. 2010) is used in step #1, and step #2 is consistent with the results of GLL-PC for all variables, and a sound orientation scheme in step #3, then the total algorithm is trivially sound under the assumptions of correctness of GLL-PC. These are the admissibility requirements for the LGL template. It follows that:

Proposition 1 *Under the following sufficient conditions we obtain correctly oriented causal graph with any admissible instantiation of LGL:*

- a. *There is a causal Bayesian network faithful to the data distribution P ;*
- b. *The determination of variable independence from the sample data D is correct;*
- c. *Causal sufficiency in V .*

The recently-introduced algorithm MMHC is an instance of the LGL framework (Tsamardinos et al., 2006). Figure 9 shows how MMHC instantiates LGL. MMHC is not sound with respect to orientation because greedy steepest-ascent search is not a sound search strategy for search-and-score global learning. Despite being theoretically not sound the algorithm works very well in practice and in an extensive empirical evaluation it was shown to outperform in speed and quality several state-of-the-art algorithms (Greedy Search, GES, OR, PC, TPDA, and SCA) (Tsamardinos et al., 2006).

6.2 A New Instantiation of LGL: HHC

To demonstrate the generality and robustness of the LGL framework we provide here as an instantiation of LGL, a new global learning algorithm termed HHC (see Figure 10), and compare it empirically to the state-of-the-art MMHC algorithm. We also show that the two algorithms are not identical in edge quality or computational efficiency, with the new algorithm being at least as good on average as MMHC.

Table 11 presents results for missing/extra edges in undirected skeleton, number of statistical tests for construction of skeleton, structural Hamming distance (SHD), Bayesian score, and execution time on 9 of the largest data sets used for the evaluation of MMHC. Since the data sets were simulated from known networks, the algorithm output can be compared to the true structure. As can be seen, in all 9 data sets, HHC performs equally well with MMHC in terms of SHD and Bayesian score. In 8 out of 9 data sets it performs from 10% to 50% fewer tests, and in one data set (*Link*) it performs >10 times the tests performed by MMHC resulting in running 35% slower in terms of execution time. Because MMHC was found to be superior to a number of other algorithms for the

HHC

	Dataset								
	<i>Child10</i>	<i>Insurance10</i>	<i>Alarm10</i>	<i>Hailfinder10</i>	<i>Pigs</i>	<i>Munin</i>	<i>Lung_Cancer</i>	<i>Gene</i>	<i>Link</i>
Extra edges in learned skeleton	95	143	176	1265	276	36	621	601	1456
Missing edges in learned skeleton	25	149	165	359	0	257	91	6	439
Structural Hamming distance for DAG	101	297	344	728	4	273	187	72	1150
Bayesian score for DAG	-188.61	-229.02	-178.56	-738.77	-496.11	-33.14	-559.43	-651.36	-337.74
Number of statistical tests for skeleton construction	28,879	52,757	82,543	217,490	134,244	733	859,348	401,779	7,931,044
Time for building skeleton (in minutes)	0.74	1.59	2.47	8.05	3.98	0.23	24.40	12.32	537.72
Total time for running algorithm (in minutes)	1.21	3.32	6.80	24.84	14.33	0.47	181.97	60.14	563.46

MMHC

	Dataset								
	<i>Child10</i>	<i>Insurance10</i>	<i>Alarm10</i>	<i>Hailfinder10</i>	<i>Pigs</i>	<i>Munin</i>	<i>Lung_Cancer</i>	<i>Gene</i>	<i>Link</i>
Extra edges in learned skeleton	71	128	184	1220	281	38	567	557	1541
Missing edges in learned skeleton	25	148	164	352	0	258	88	4	396
Structural Hamming distance for DAG	100	296	346	725	4	275	191	69	1145
Bayesian score for DAG	-188.95	-229.03	-179.09	-738.80	-496.11	-33.12	-559.01	-651.12	-337.62
Number of statistical tests for skeleton construction	32,980	67,943	90,117	243,571	177,278	1,023	1,360,493	451,364	644,055
Time for building skeleton (in minutes)	0.81	1.99	2.49	12.81	5.45	0.38	55.16	12.23	382.93
Total time for running algorithm (in minutes)	1.42	3.79	5.21	29.54	13.11	0.46	451.70	51.84	415.69

Table 11: Comparison of HHC and MMHC global learning algorithms. Both algorithms were executed on a random sample of size 1000, using default parameters of MMHC as implemented in *Causal Explorer* (i.e., G^2 test for conditional independence, $\alpha = 0.05$, $max-k = 10$, Dirichlet weight = 10, BDeu priors).

data sets tested, HHC’s better performance over MMHC in 8 out of 9 data sets (in terms of number of statistical tests for skeleton construction) and similar performance in 9 out of 9 data sets (in terms

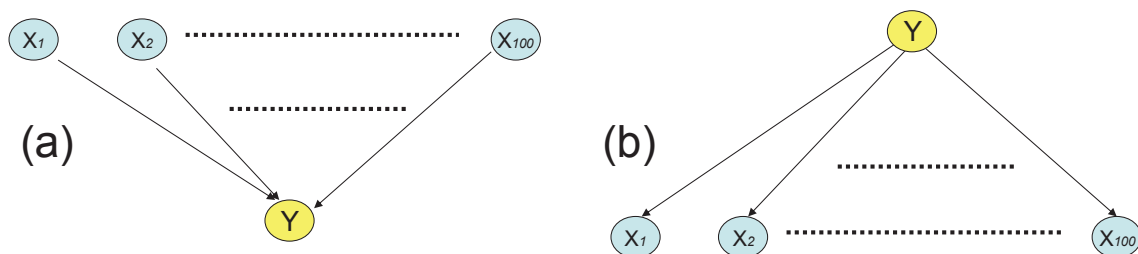


Figure 11: Two examples where the variable ordering for local learning can make execution of the LGL algorithm from quadratic to exponential-time.

of quality metrics) translates also to excellent performance of HHC relative to Greedy Search, GES, OR, PC, TPDA, and SCA (Tsamardinos et al., 2006).

6.3 Importance of Variable Prioritization for Quality and Efficiency

An important parameter of local-to-global learning previously unnoticed in algorithms such as SCA and MMHC is the ordering of variables when executing the local causal discovery variable-by-variable (i.e., not in parallel). We will assume that results are shared among local learning runs of GLL-PC, that is when we start learning $PC(X)$ by GLL-PC rather than starting with an empty $TPC(X)$ set, we start with all variables $Y: X \in PC(Y)$. This constitutes a sound instantiation of the GLL-PC algorithm template as explained in Aliferis et al. (2010). Figure 11 gives two extreme examples where the right order can “make-or-break” an LGL algorithm.

In Figure 11(a) it is straightforward (and left to the reader to verify) that an order of local learning $\langle X_1, X_2, \dots, X_{100}, Y \rangle$ without symmetry correction (the latter being a reasonable choice as we have seen) requires a quadratic number of conditional independence tests (CITs) for the unoriented graph to be correctly learned. However, the order of local learning $\langle Y, X_1, X_2, \dots, X_{100} \rangle$ requires up to an exponential number of CITs as $max-k$ and sample are allowed to grow without bounds. Even with modest $max-k$ values, the number of CITs is higher-order polynomial and thus intractable. Even when Y is not in the beginning but as long as a non-trivial number of X ’s are after it in the ordering, the algorithm will be intractable or at least very slow. The latter setting occurs in the majority of runs of the algorithm with random orderings.

In Table 12 we provide data from a simulation experiment showing the above in concrete terms and exploring the effects of limited sample and connectivity at the same time. As can be seen, under fixed sample, running HHC with order from larger to smaller connectivity, as long as the sample is enough for the number of parents to be learned (i.e., number of parents is ≤ 20), increases run time by more than 100-fold. However because sample is fixed, as the number of parents grows the number of conditional independence tests equalizes between the two strategies because CITs that have too large conditioning sets for the fixed sample size are not executed. Although the number of CITs is self-limiting under these conditions, quality (in terms of number of missing edges, that is, number of undiscovered parents of T) drops very fast as the number of parents increases. The random ordering strategy trades off quality for execution time with the wrong (larger-to-smaller connectivity) ordering, however in all instances the right ordering offers better quality and 2 to 100-fold faster execution than random ordering.

Number of parents of Y	order from low-to-high connectivity			random order (average results over 10 orders)			order from high-to-low connectivity		
	extra edges	missing edges	CITs	extra edges	missing edges	CITs	extra edges	missing edges	CITs
10	2	0	63	2	0	2,461	2	0	4,325
20	4	0	233	4.7	5.2	26,203	5	7	29,774
30	12	0	526	12	12.4	41,499	11	21	9,020
40	13	0	904	16.4	20.1	51,269	19	33	5,626
50	22	7	1,428	28.8	30	16,828	34	43	4,149
60	29	7	2,001	32.9	35.7	36,950	38	54	3,862
70	41	19	2,773	45.7	37.9	24,456	55	63	4,464
80	58	28	3,652	65.4	55.1	12,630	70	74	5,023
90	66	35	4,634	72.3	57.6	16,718	87	85	5,592
100	77	44	5,594	88.7	80	16,266	96	94	7,229

Table 12: Results of simulation experiment with HHC algorithm. The graphical structure is depicted on Figure 11(a). HHC was run on a random sample of size 1,000 with G^2 test for conditional independence, $\alpha=0.05$, $max-k = 5$, Dirichlet weight = 10, BDeu priors.

A more dramatic difference exists for the structure in Figure 11(b) where Y is a parent of all X 's. Here the number of tests required to find the parent (Y) of each X_i is quadratic to the number of variables with the right ordering (low-to-high connectivity) whereas an exponential number is needed with the wrong ordering (large-to-small connectivity). Because the sample requirements are constant to the number of children of Y , quality is affected very little and there is no self-restricting effect of the number of CITs, opposite to what holds for causal structure in Figure 11(a). Hence the number of CITs grows exponentially larger for the large-to-small connectivity ordering versus the opposite ordering and a similar trend is also present for the average random ordering in full concordance with our theoretical expectations. See Table 13 for results of related simulation experiments.

These results show that in some cases, *it is possible to transform an intractable local learning problem into a tractable one by employing a global learning strategy (i.e., by exploiting asymmetries in connectivity)*. Thus the variable order in local-to-global learning may have promise for substantial speedup and improved quality in real-life data sets (assuming the order of connectivity is known or can be estimated). However the optimal order is a priori unknown for some domain. Can we use local variable connectivity as a proxy to optimal order in real data? The next experiment assumes the existence of an oracle that gives the true local connectivity for each variable. The experiment examines empirically the effect of three orders (low-to-high connectivity, lexicographical (random) order, and high-to-low connectivity order) on the quality of learning and number of CITs in the MMHC evaluation data sets. It also compares the sensitivity of HHC to order.

As can be seen in Figure 12, the order does have an effect on computational efficiency however not nearly as dramatic in the majority of these more realistic data sets compared to the simpler structures of Figure 11. An exception is the *Link* data set in which low-to-high connectivity allows HHC to run 17 times faster than lexicographical (random) order and 27 times faster than high-to-low connectivity order. For the majority of cases, running these algorithms with lexicographical (i.e.,

Number of children of Y	order from low-to-high connectivity			random order (average results over 10 orders)			order from high-to-low connectivity		
	extra edges	missing edges	CITs	extra edges	missing edges	CITs	extra edges	missing edges	CITs
10	1	0	106	1	0	2,342	1	0	4,366
20	11	0	489	9.7	0	141,148	9	0	377,448
30	18	0	1,173	16.8	0	2,321,030	17	0	5,020,400
40	24	0	1,968	-	-	-	-	-	-
50	33	0	3,190	-	-	-	-	-	-
60	48	0	5,031	-	-	-	-	-	-
70	53	0	6,899	-	-	-	-	-	-
80	71	0	8,939	-	-	-	-	-	-
90	76	0	11,448	-	-	-	-	-	-
100	95	0	14,677	-	-	-	-	-	-

Table 13: Results of simulation experiment with HHC algorithm. The graphical structure is depicted on Figure 11(b). HHC was run on a random sample of size 1,000 with G^2 test for conditional independence, $\alpha=0.05$, $max-k=5$, Dirichlet weight = 10, BDeu priors. Empty cells correspond to experiments when the algorithm did not terminate within 10,000,000 CITs.

random) order is very robust and does not affect quality adversely but affects run time and number of CITs to a small degree (details in Table S21 in the online supplement).

Thus, while connectivity affects which variable order is optimal in LGL algorithms, ranking by local connectivity does not exactly correspond to the optimal order. Figure S3 in the online supplement shows the number of CITs plotted against true local connectivity in each one of the 9 data sets used in this section. Related to the above, Figure S4 in the supplement also shows the distribution of true local connectivity in each data set. Consistent trends indicating the shape of the distributions by which the degree of local connectivity may determine an advantage of orderings low-to-high to high-to-low connectivity are not apparent in these data sets.

We hypothesize that more robust criteria for the effect of variable ordering in LGL algorithms can be devised. For example, the number or total cost of CITs required to locally learn the neighborhood of each variable. Such criteria are also more likely to be available or to be approximated well during practical execution of an algorithm than true connectivity. A variant of HHC, algorithm HHC-OO (standing for HHC with optimal order) (Aliferis and Statnikov, 2008) orders variables dynamically according to heuristic approximations to the total number of CITs for each variable. We also conjecture that the strategy for piecing together the local learning results strongly interacts with the local variable ordering to determine the tradeoff between the quality and efficiency of LGL algorithms. Evaluation of these hypotheses is outside the scope of the present paper.

6.4 Using non-Causal Feature Selection for Global Learning

In recent years several researchers have proposed that because modern feature selection methods can deal with large dimensionality/small sample data sets, they could also be used to speed up or approximate large scale causal discovery (e.g., Kohane et al. 2003 use univariate feature selection to build so-called “relevance networks”), or hybrid methods can be employed that use feature selection

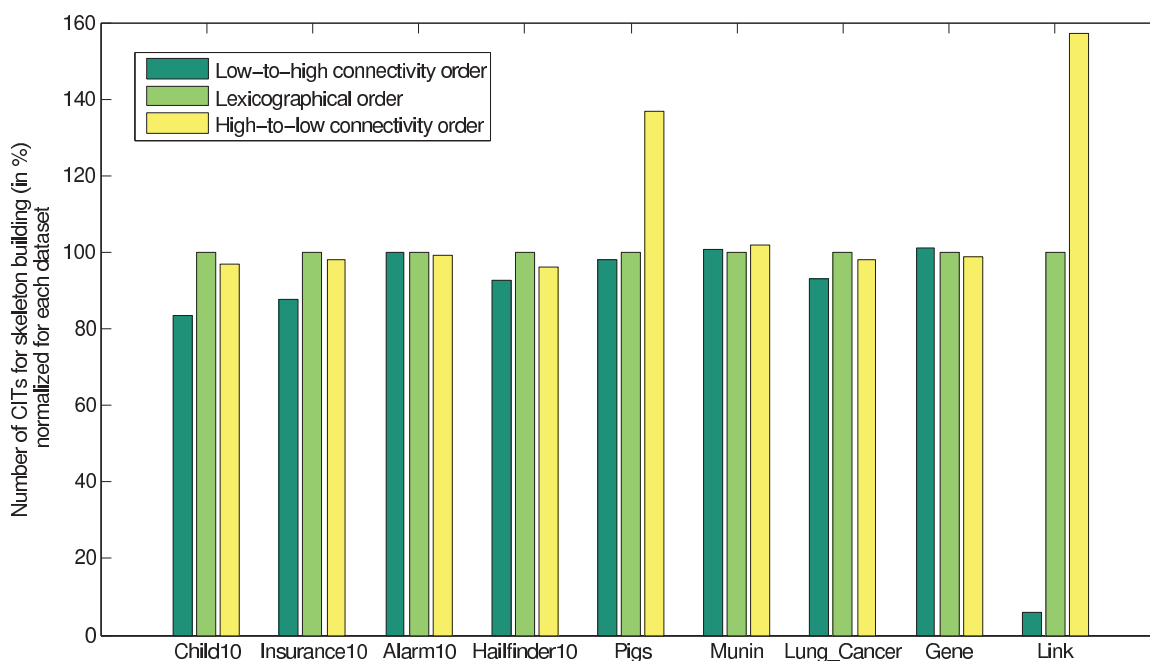


Figure 12: Number of CITs required for skeleton construction during execution of HHC expressed as % points and normalized within each data set to lexicographical order. Data for three orderings of variables is shown on the figure: low-to-high connectivity, lexicographical, and high-to-low connectivity orders. HHC was executed with same parameters as in Table 11. More detailed results are provided in Table 11 and Table S21 in the online supplement.

as a pre-processing to build a skeleton and then an orientation algorithm like Greedy Search in the spirit of MMHC and LGL (Schmidt et al., 2007). The results of Aliferis et al. (2010) contradict this postulate because they show that non-causal feature selection does not give locally correct results. However it is still conceivable that orientation-and-repair post-processing algorithms (e.g., with Bayesian search-and-score) can still provide a high quality final causal graph. We test this hypothesis by examining several such hybrid methods using respectively RFE, LARS-EN and UAF post-processed by Greedy TABU Bayesian search-and-score. We use simulated data sets from 5 out of 9 Bayesian networks employed earlier in the present section. This is because the other 4 networks cannot be used for reliable training and testing of the underlying classifier since they have several variables with very unbalanced distributions. As shown in Table 14, the hypothesis is not corroborated by the experimental results. In particular, Greedy Search with feature selection-based skeleton, exhibits substantial drops in quality of the returned networks (measured by structural hamming distance Tsamardinos et al., 2006) and typically more than one order of magnitude longer running times compared to HHC with lexicographical (random) variable ordering. On the basis of these findings, which are consistent with the results in Aliferis et al. (2010), we do not find encouraging evidence that non-causal feature selection can be used as an adjunct to global causal discovery. Strong evidence exists however in favor of using principled local causal methods instead, within the frameworks of LGL.

	<i>Child10</i>				<i>Pigs</i>				<i>Hailfinder10</i>			
	<i>RFE</i>	<i>LARS</i>	<i>UAF</i>	<i>HHC</i>	<i>RFE</i>	<i>LARS</i>	<i>UAF</i>	<i>HHC</i>	<i>RFE</i>	<i>LARS</i>	<i>UAF</i>	<i>HHC</i>
Extra edges in learned skeleton	2078	7558	3014	95	2262	29570	5593	276	6424	40948	7904	1265
Missing edges in learned skeleton	26	8	20	25	2	0	0	0	461	211	325	359
Structural Hamming distance for DAG	121	117	135	101	76	102	7	4	796	756	733	728
Bayesian score for DAG	-190.0	-189.1	-189.8	-188.61	-497.2	-496.8	-496.4	-496.11	-740.5	-736.4	-737.4	-738.77
Time for building skeleton (in minutes)	41.63	43.57	44.97	0.74	348.44	184.47	355.59	3.98	572.13	365.45	581.34	8.05
Total time for running algorithm (in minutes)	43.23	48.52	47.05	1.21	361.15	265.07	373.54	14.33	603.62	503.63	612.63	24.84

	<i>Gene</i>				<i>Lung Cancer</i>			
	<i>RFE</i>	<i>LARS</i>	<i>UAF</i>	<i>HHC</i>	<i>RFE</i>	<i>LARS</i>	<i>UAF</i>	<i>HHC</i>
Extra edges in learned skeleton	4039	55384	9834	621	7469	38753	12486	601
Missing edges in learned skeleton	47	8	28	91	120	24	78	6
Structural Hamming distance for DAG	125	156	115	187	220	139	175	72
Bayesian score for DAG	-658.3	-653.1	-655.1	-559.43	-562.4	-555.6	-560.1	-651.36
Time for building skeleton (in minutes)	737.99	513.12	783.97	24.40	493.84	377.85	563.46	12.32
Total time for running algorithm (in minutes)	784.54	912.33	890.63	181.97	708.77	1096.19	855.18	60.14

Table 14: Results for hybrid methods using RFE, LARS-EN and UAF.

7. Using Causal Graphs and Markov Blanket Theory as a Conceptual Analysis Framework for Feature Selection Methods

In the present section we show that by adopting a causal structural perspective founded on the theoretical results outlined in Aliferis et al. (2010), several strengths and weaknesses and general performance characteristics of non-causal feature selection algorithms become apparent and our empirical findings in Aliferis et al. (2010) can be better understood. We review several established and state-of-the-art methods both from a feature selection perspective (e.g., does the algorithm exhibit false positives and false negatives relative to minimal feature set that yields optimal predictivity?) and from a causal discovery perspective (is the output of the algorithm causally sound?). With respect to the latter for reasons elucidated in Aliferis et al. (2010), we focus on localization of causal inferences (i.e., whether the feature selection output is locally causally correct), and when this is not obtained, we examine whether some other useful causal inference can be made.

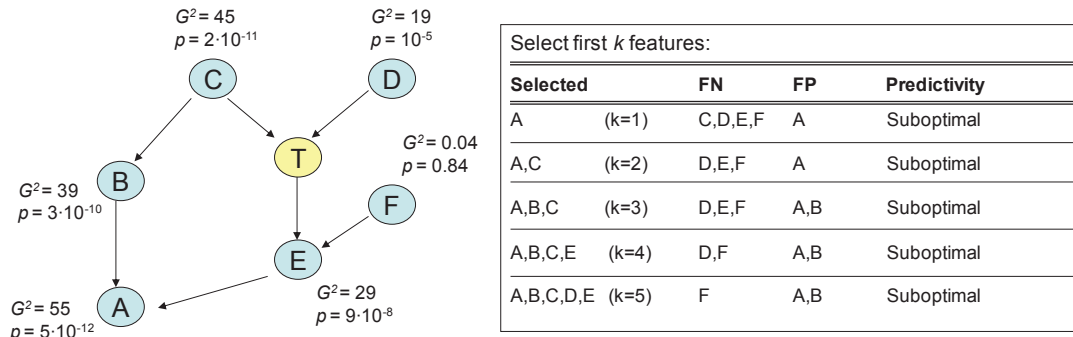


Figure 13: Limitations of univariate feature selection explained using a causal graph perspective. Strength of univariate association with the target variable T is measured in a fixed sample of size 10,000 by the negative p -value of a G^2 -test and depicted next to each variable.

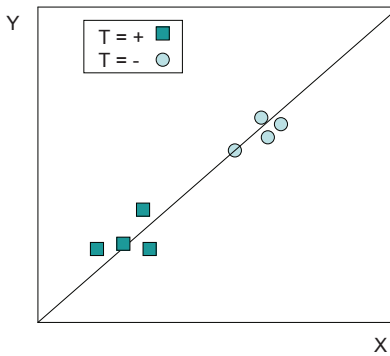


Figure 14: Example showing that Principal Component Analysis yields redundant features.

7.1 Univariate Association Filtering

Figure 13 shows the causal structure of a data-generating process. The causal structure is parameterized as shown in Appendix Figure 19. This structure and parameterization entails that $association(B, T) < association(C, T)$. Because of *synthesis of information along two paths* however, $association(A, T) > association(C, T)$ and $association(A, T) > association(E, T)$. The example illustrates that from the feature selection perspective the optimal predictor set (i.e., the Markov blanket) for predicting or classifying the target T is $\{C, D, E, F\}$. However, because univariate associations of non- $MB(T)$ members can be higher than those of members, false positives are incurred when selecting features using univariate association-based filters. Furthermore, spouses without connecting path to the target will have zero univariate association and thus will not be selected at all by univariate filtering. The embedded table shows the false positives and false negatives (relative to the gold standard set $MB(T)$) at each possible threshold for variable inclusion. In all cases predictivity is suboptimal.

From the causal discovery perspective, the example makes evident that non-causally relevant features such as A and B can be selected with higher ranking than causally relevant ones such as D and E . Association synthesis thus forbids an interpretation of the higher-ranked causal variables as more direct causes (or effects) than lower-ranked features even when all of them are causal. Worse yet, even without synthesis, an arbitrarily large number of non-causal features can be selected before

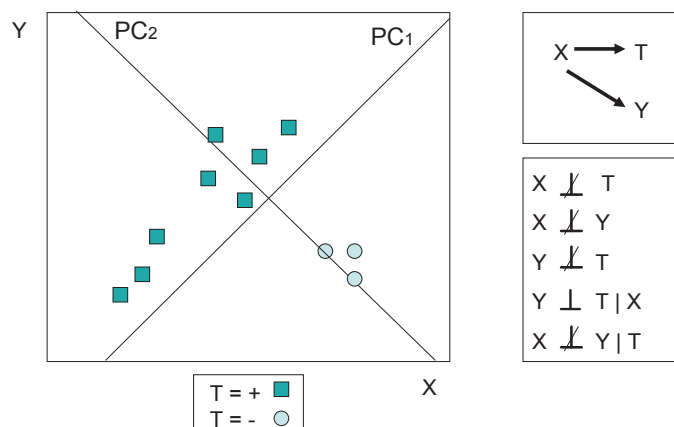


Figure 15: Example showing that Principal Component Analysis yields locally causally inconsistent results.

truly causal ones are selected. To see why this is the case consider that between C and B there may be arbitrarily many variables arranged in a chain so that their association with T is larger than that of both true cause D and true effect E .

7.2 Principal Component Analysis

As can be seen in Figure 14, the principal component defined by the diagonal ($Y - X = 0$) perfectly separates the two target classes and will be chosen by a PCA procedure since it explains maximum proportion of variance in the data. While projecting the original data on this single dimension reduces dimensionality of the classification problem, from the perspective of finding the original features that are important and non-redundant the method leads to false positives (since the coefficients of both Y and X are equal in the depicted Principal Component, indicating that both features are deemed equally necessary).

The example in Figure 15 shows that PCA is not sound for causal discovery. As shown in the figure, X is a direct cause of T and Y is not causal for T but confounded by X . Application of causal learning via the usual assumptions and procedures reveals that X is a direct cause or effect of T and that Y is not directly causally linked with T (the requisite conditional independence tests are depicted). However, an optimal procedure for Principal Component classification will select the second principal component PC_2 which achieves perfect classification. However both X and Y have equal coefficients in each principal component. Hence PCA may select both redundant features and non-causal features.

7.3 Feature Selection Using SVM Weights

A fundamental weakness of the maximum-gap inductive bias, as employed in SVMs, is its local causal inconsistency. Consider a scenario (Figure 16) similar to the previous sub-section where we wish to discover the direct causes of a response variable T , from observations about variables X, Y, T . Assume for simplicity that T is a terminal variable and thus X and Y precede it in time. For example, T can be a clinical phenotype and X, Y can be gene expression values. The causal process that generates the data is seen in the upper right corner of Figure 16. As can be seen in the

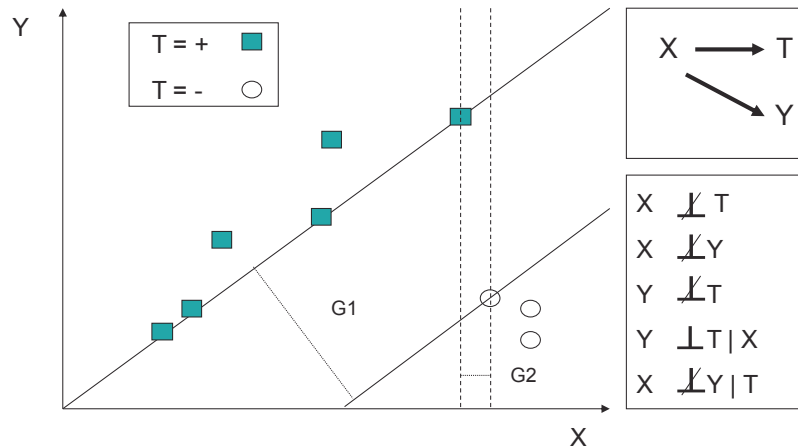


Figure 16: Example showing that SVM weight-based feature selection yields locally causally inconsistent results and redundant features.

left part of the figure, the SVM classifier can perfectly predict T using X and Y as predictors. In doing so it prefers the classifier with gap $G1$ to the classifier with smaller gap $G2$. The preferred classifier assigns non-zero (and in fact equal) weights to both X , Y thereby admitting Y in the local causal neighborhood if selected variables are interpreted causally. However, X renders Y independent from T and not vice versa. More generally, in distributions where the Causal Markov Condition holds, SVMs will occasionally fail to detect that Y is not a local cause of T . Sound causal discovery algorithms do not face this problem, however. In addition, the preference for maximum gap classifier biases in favor of assigning non-zero weights to redundant features (Y in the example).

On the positive side, theoretical results show that SVMs in the large sample will assign zero weights to irrelevant variables (Hardin et al., 2004). Despite this theoretical good property, in the experiments of Aliferis et al. (2010) it was found that in realistic finite sample weights of irrelevant variables are non-zero. In the work of Statnikov et al. (2006) it was found that weights of irrelevant features occasionally exceed those of weakly relevant features and furthermore that SVM weights are also susceptible to assigning larger weights to synthesis features rather than direct causes and effects.

7.4 Wrapping

One of the widely-cited advantages of wrapping as a feature selection method is that it allows to tailor the selection of features to the inductive bias of the classifier (Kohavi and John, 1997). We show here how this property when combined with rich connectivity may yield causally misleading results. Consider the generative process of Figure 17. The target T is a quadratic function of its true causes A , B . Variables X , Y are effects of A , B respectively with similar non-linear functional relationships. A causal discovery procedure such as HITON-PC given enough sample and a suitable statistical test of independence will discover $\{A, B\}$ as the correct set of direct causes and direct effects. Consider however a practitioner who attacks the problem of learning a good classifier for T and reducing the necessary feature set using wrapping instead. If, as would normally be the case, the analyst starts with a simpler model class before proceeding to consider more complex

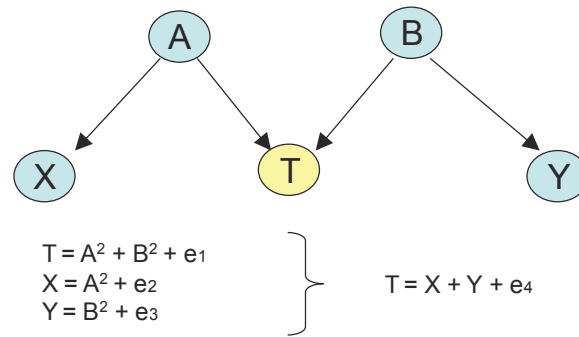


Figure 17: Example showing that wrapping, by tailoring feature selection to the classifier inductive bias may produce causally misleading results.

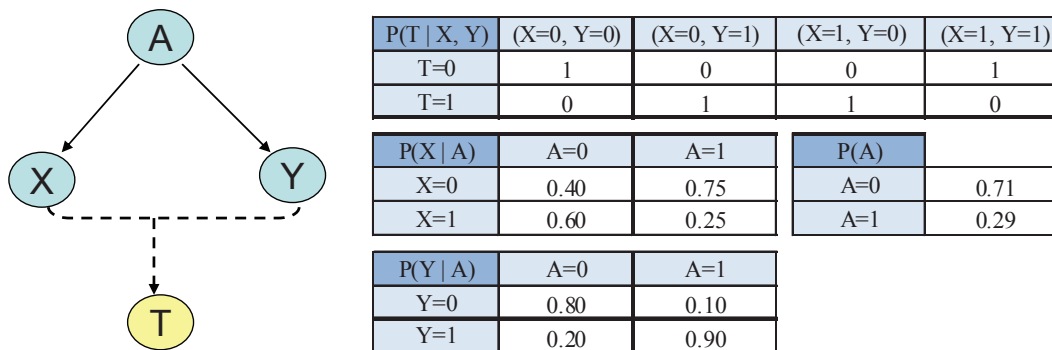


Figure 18: Example showing that connectivity may mitigate violations of faithfulness. Dashed line indicates a highly non-linear function (XOR). The left part shows the causal structure, while the right part shows its parameterization.

ones, assuming that noise components e_2 , and e_3 are small enough then the linear classifier would perform very well with $\{X, Y\}$ as predictors and a wrapper tailored to the linear inductive bias would eliminate A and B .

In small networks with a few variables and limited connectivity the above possibility is small, however in large networks with thousands of variables and rich connectivity as well as with massive information redundancy (e.g., biological networks) such “variable replacement” is entirely feasible and thus tailoring feature selection to a classifier’s inductive bias (as wrapping does) can be an obstacle to sound causal discovery.

7.5 Connectivity and Priors Compensating for Violations of Faithfulness - Learning XOR Parents Using Univariate Association in GLL and Other Algorithms

A violation of faithfulness where constraint-based algorithms are expected to fail is when the target is an extremely non-linear function of its parents. A prototypical example is when T is the parity (XOR) of its parents A and B . Conventional wisdom, based on the truth table of the XOR function, dictates that first-order effects are zero and, as a result, the parents cannot be detected by the inclusion heuristic of the algorithm (i.e., HITON-PC or MMPC). As shown in Figure 18 however,

connectivity among variables can mitigate this difficulty. In the figure, variables X and Y can have non-zero univariate association with T , even though in textbook descriptions of parity where parents are unconnected and with 50% prior probability each for being 0 or 1, univariate association vanishes. An example parameterization that allows for this effect is given in the figure as well. This counter-intuitive phenomenon occurs because when X and Y are common effects of A , knowing the value of X is informative about A and thus about Y . Therefore the joint values of $\{X, Y\}$ are constrained and this creates univariate association of X and Y with T . Similarly, conditional association of X with T given Y is non zero. The phenomenon is not restricted to parity (or other extremely non-linear) functions in which the parity parents are connected in the network. Figure 20 in the Appendix shows an example where skewed priors on the unconnected parity parents X, Y lead to non-zero univariate association of X and Y with the target T .

The phenomenon described in this sub-section does not only apply to GLL algorithms but extends to other feature selectors as well. For example, the success of univariate filtering as feature selector, which has been documented in many domains (Guyon et al., 2006), can in part be explained via connectivity effects that allow univariate association to detect complex non-linear relationships of selected features with the target variable.

The discussion in this section is complemented by analysis of embedded feature selection in decision tree induction and of RELIEF in the online supplement Figures S5 and S6 (omitted here due to space limitations). It is shown that these algorithms can admit false positives and false negatives both predictively and causally with respect to the target variable neighborhood.

8. Discussion and Open Problems

In this section we present a thorough discussion of results, outline open problems and future directions, and provide a conclusion.

8.1 Discussion of Results

The algorithms presented, and their applied evaluation and theoretical analysis clarify many of the initially open questions discussed in Aliferis et al. (2010) and point to several new research directions. We showed that in empirical tests with 9 simulated data sets, GLL convergence to optimal performance is very fast with respect to sample size both in the sense of producing feature sets that have equal predictivity as the true $MB(T)$ and $PC(T)$ sets, and in the sense of achieving near optimal predictivity even at moderate samples sizes. These results corroborate the empirically good performance of GLL instantiations in real data sets (Aliferis et al., 2010).

An unexpected and important finding was that *GLL algorithms exhibit strong intrinsic control of false positives due not only to weakly relevant but also due to irrelevant features*. This control is empirically better in the tested data sets than what formal state-of-the-art FDR control provides except in the rare case when the data consists exclusively of irrelevant features. In Statnikov et al. (2010) we show that GLL can discover differentially expressed genes when the sample size is so small that FDR does not yield any gene. The same cannot be said for other feature selection methods that were found to be particularly prone to false positives due to both irrelevant and weakly relevant features. On the other hand, it needs to be noted that classical FDR methods do not control at all weakly relevant false positives (as GLL does). A simple pre-filtering of GLL algorithms with an FDR control method eliminates false positives in all cases tested and yields the best algorithm for

local causal learning among tested algorithms. We expect that other algorithms for example PC and MMHC will benefit from such an FDR prefiltering as well.

Within the GLL framework both the $max-k$ and $h-ps$ parameters control the false positives and false negatives tradeoff, through control of combined power and combined significance levels. We examined via targeted experiments and theoretical discussion the complex determination of quality of statistical decisions in GLL algorithms (aspects of which are shared by previous global constraint-based algorithms). Having two parameters to control quality of statistical decisions confers advantages since they can regulate different aspects of such decisions, and trade-off statistical quality with computational complexity.

Our efforts to explain the good predictive performance of the estimated $PC(T)$ set compared to the estimated $MB(T)$ set focused on producing explanations consistent with sufficient assumptions for Markov blanket optimality so that the good performance of the $PC(T)$ set would not be wrongly construed as entailing rejection of the theoretical assumptions, or as inability to infer the correct $MB(T)$ when the assumptions hold in the data. This is because both the results of our simulated experiments in Aliferis et al. (2010) as well as previously published experiments (Tsamardinos et al., 2003b) show that GLL algorithms estimate very well the $MB(T)$ and $PC(T)$ sets.

We also used a causal graph point of view and Markov blanket concepts to understand a variety of non-causal feature selection algorithms. This approach *provides a cohesive and fresh perspective into the behavior of several algorithms for feature selection*. We made this point by showing that the theory readily reveals why prominent feature selection methods exhibit many false positives and why they cannot be used for sound causal discovery. This complements the findings of Aliferis et al. (2010) that demonstrate empirical feature selection and causal discovery suboptimality for many state-of-the-art non-causal feature selection methods.

We discussed in detail a fundamental statistical weakness of wrapping, namely that it is prone to errors due to imperfect error estimation. This is especially the case when sample size is small whereby practical unbiased error estimators have large variance. The same problem applies implicitly to widely-used feature selection approaches such as ranking by univariate association and selecting the first k features. We showed why GLL algorithms are less sensitive to this shortcoming. In general our results show that GLL instantiations are robust enough to apply across a wide variety of domains.

Established feature selection criteria in statistics such as the AIC (Akaike Information Criterion) bare some resemblance to Markov blanket feature selection in the sense that AIC does not require classification error estimation. Specifically, AIC balances the number of features (parameters) with the likelihood of the data given a model: $AIC = 2k - 2\log(L)$, where k is the number of parameters and L is the likelihood function. Model selection is driven by optimizing AIC. A critical difference however is that Markov blanket induction does not require a generative model of the data to be calculated (but relies on conditional independence tests). Given that inducing a generative model is in general harder than finding features that cannot be rendered independent of the target, and given that many recent powerful classifiers do not build generative models (e.g., SVMs) it follows that the Markov blanket induction approach has a corresponding advantage over AIC. Markov blanket induction is less model-dependent than AIC for the same reason. Note that similarly the GLL algorithms by not attempting to induce edge directionality (a task harder than edge detection, Ramsey et al., 2006) except when absolutely necessary they avoid incurring errors in edge detection produced by false conclusions about directionality (since one type of discovery affects the other). As

a result, Markov blanket induction via the GLL framework has advantages over eliciting Markov blankets by using methods that require global or local orientation.

The extensive evaluation of GLL algorithms in Aliferis et al. (2010) shows that the sufficient conditions stated in the proofs for correctness are likely to hold often, or that violations may be small. In some cases we showed that the algorithms may not fail when the assumptions are violated. Due to the critical role of non-faithfulness as a major source of possible failure we discuss it here in more detail. Faithfulness is violated in a variety of situations (Spirtes et al., 2000), notably in practice when (a) extremely non-linear or deterministic functions exist, when (b) causality cannot be localized, and when (c) variables share the same information for a response (target variable). Practical examples, respectively, are extreme epistasis in genetics, non-local causation in quantum mechanics, and gene-phenotype information redundancy in gene expression microarrays. For many additional reasons see Spirtes et al. (2000) and Meek (1995).

However, we showed that even in prototypical non-faithful functions such as XOR, the existence of unbalanced priors or the existence of connectivity among XOR parent variables of the target can make such parent variables visible again to the GLL algorithms as well as other feature selectors (e.g., univariate association filtering). We believe that this finding may have broad implications of which we mention a few. First, it explains in part the success of univariate feature selection methods in many domains since univariate filtering can pick up features that are involved in extremely non-linear functions. Second, other algorithms that are typically thought to not be able to learn such functions, such as Genetic Algorithms (Sharpe, 2000) in many situations may be able to do just that. In addition, to the extent that biological systems have evolved by evolutionary processes similar to genetic algorithms, truly extreme epistatic functions may not be as rare as previously thought. Recent proposals that suggest that such functions (i.e., biological systems) can be learned (i.e., evolved) by GAs (i.e., by evolution) through multiple objective optimization may be too pessimistic (Lenski et al., 2003). Third, previous postulates that randomized experiments (e.g., in biology, medicine and psychology) because they examine one causal factor at a time are thus unable to detect parity-like functions, may also be pessimistic (Aliferis and Cooper, 1998).

Returning to non-local causality, we point out that cognitively it is advantageous to modularize causal knowledge in order to reduce the connectivity of causal graphs and thus to control learning complexity (as well as to increase ability to store and process such knowledge with limited cognitive resources). We may thus be facing in both natural as well as artificial systems a selection bias (relative to all possible theoretical distributions) where causal systems and models of those are highly modular because it is easier to create and handle such systems and their models. Indeed in most known macroscopic causal processes (e.g., biological pathways, medicine, engineering, economics, social networks) causal systems are highly modular and thus local.

For all of the above reasons faithfulness is a very reasonable a priori, and powerful in practice, distributional assumption. At the same time at least some violations can be tolerated well by causal algorithms that are designed to use it and existing research addresses violations systematically, for example extensions of standard causal discovery algorithms capable of addressing target information equivalency (Statnikov, 2008).

The exploration of parallel and distributed techniques in the present paper showed that *GLL is amenable to parallelized and distributed local causal discovery and feature selection*. We established empirically the potential of parallelization for speeding up processing time without loss of quality. The presented parallel algorithm can also be used for distributed feature selection and causal discovery in a principled manner. Many more algorithms (namely that induce Markov blankets and

admit symmetry correction when needed) can be constructed following the approach introduced in parallel and distributed IAMB for Markov blanket induction (Aliferis et al., 2002). In contrast to parallel IAMB however, parallel GLL-PC can be exponentially faster (or slower) than induction in the full data. This is a very interesting future research direction.

In exploring the transition from local-to-global strategies we showed that the local-to-global learning framework LGL can be instantiated in several ways. We examined one new instantiation of local-to-global learning, algorithm HHC. Although in most real data tested a random variable order is as good as perfectly-informed ordering by local connectivity, we showed in the present paper something previously unnoticed, namely that in some cases the right order of local neighborhood learning can entail exponential time vs. low-order polynomial time execution of local-to-global algorithms. This finding has a subtle implication: if the right ordering can be found for local learning, the resulting global learning of all variables can be faster than the local learning targeted at just one variable. Thus, just as local learning can speed up global learning the reverse may also be true.

On the other hand, our results showed that the idea that non-causal feature selection methods could help in addressing scalability of formal causal algorithms may be misplaced in light of the failure of non-causal feature selection methods to induce causality and given that highly scalable and sound methods such as GLL algorithms do exist. Several tested algorithms where non-causal feature selection is used to elicit a skeleton which is then oriented and refined by formal causal global methods are very slow and typically produce lower-quality graphs than LGL instantiations relying on sound local causal methods.

8.2 Open Problems and Future Directions

The results presented in Aliferis et al. (2010) and in the present paper merely scratch the surface of causal feature selection algorithms, local causal learning, and local-to-global learning. We briefly discuss here a few salient opportunities for moving this exciting area forward.

An assumption that is probably too strong for soundness of $MB(T)$ induction is that of causal sufficiency. For example, we conjecture without formal proof, that the algorithms should attain soundness even if the causal sufficiency is localized among the target and the members of its Markov blanket. Even when this local causal sufficiency is violated, predictive optimality among measured variables may not be compromised in many practical situations (although the usual causal interpretation of the found features is affected). Characterizing localized versions of faithfulness and causal sufficiency is an area that is likely to give a better understanding of existing algorithms and possibly lead to improvements. Examining and dealing with the effects of temporal aggregation, sampling (e.g., cellular) aggregation, feedback loops, and limited local causality on feasibility of local causal discovery will be helpful in determining the space of practical usefulness of the GLL framework.

A previously underemphasized important parameter for false negatives control is the order of conditional independence tests used for elimination (i.e., part of the elimination strategy in the GLL-PC schema). In general, the earlier time that strongly relevant variables are being examined for elimination, the better the chances for avoiding a false negative conditional independence test result since the combined power is larger. This is accomplished implicitly in HITON-PC and MMHC by using heuristics that include strongly relevant features first in $TPC(T)$ and then in both semi-interleaved HITON-PC and MMHC, where new candidates are considered for elimination *first* and where conditioning sets are constructed with stronger candidates for $PC(T)$ *first*. Systematic study

of such prioritization schemes may yield performance benefits over existing GLL instantiations. Other areas that may yield improved performance is selective or full model averaging to address instability of $MB(T)$ estimation in small samples and optimizing alpha thresholds and FDR thresholds either for a domain or a data set, possibly separately for each variable.

In general, the treatment of determination of unreliable tests by means of the heuristic rule and parameter $h-ps$ in GLL instantiations can be improved by incorporating formal power-size analysis whenever possible. More broadly, removing the requirement for a uniform sample size requirement across independence tests of same order (but different response function) is likely to yield improved algorithms. Other statistical issues such as improved statistical handling of structural zeros for discrete statistics, improved statistical tests that combine discrete and continuous data, handling “forced” covariates (i.e., variables that need to remain in $TPC(T)$ or $TMB(T)$ so that a particular effect is controlled for) are also worth exploring. Related to proper statistical testing is the issue of optimal discretization, not for classification as has been explored before in the literature, but for causal discovery (for a study toward that direction see Fu 2005). Other statistical extensions are to adapt the GLL method for survival analysis, or other time-to-event analyses without discretizing outcomes and with ability to handle observation censoring.

Exploitation of prior knowledge and development of methods to exploit prior causal knowledge (e.g., variable ordering, forced edges, forbidden edges, known size of local neighborhoods, known directionalities/structure and degree of connectivity, etc.) may yield greatly improved methods. Comparisons of knowledge-enhanced to pure data-driven instantiations will then be very informative.

An obvious possibility not examined in the present work is using GLL methods for regression. Another natural line of future research is to study situations where a loss function does not require exact knowledge of the conditional probability $P(T|MB(T))$ in which a promising strategy is to use a wrapping post-processing step to remove unnecessary features thus tailoring the final feature set to a loss function less stringent than the ones that typically guarantee soundness for GLL-MB algorithms.

Different distributional assumptions, for example monotone DAG faithfulness to make GLL and LGL algorithms faster (for a first attempt see Brown et al. 2005) may provide algorithms that tradeoff well quality for speed in specific domains.

Although we did not address the issue in this work, post-processing the results of GLL and LGL output using algorithms that detect hidden variables and orient edges is an obvious direction for research.

The study of convergence behavior of GLL and of false discovery rate control were either empirical or qualitative in the present paper. Derivation of mathematical analyses of convergence to the optimal $MB(T)$ and optimal classifier (as function of sample size), of effects of synthesis, of how common synthesis is, of combined power and alpha for specific distributions will be very interesting, especially as other components of the framework (for example handling of unreliable tests) are also formalized.

Developing methods that handle efficiently very large neighborhoods with hundreds of features and small sample size, as well as developing methods for special-purpose causal structures (e.g., genome-wide association studies) is also an area where significant improvements can be made.

The skeleton phase of LGL is a form of dynamic programming and this explains its efficiency and soundness and probably leaves reduced opportunity for dramatic efficiency improvements. One possible avenue would be the exploration of different strategies for linking together the local skele-

P(T C, D)	(D=0, C=0)	(D=0, C=1)	(D=1, C=0)	(D=1, C=1)
T=0	0.55	0.45	0.48	0.45
T=1	0.45	0.55	0.52	0.55

P(E T, F)	(T=0, F=0)	(T=0, F=1)	(T=1, F=0)	(T=1, F=1)
E=0	0.6	0.4	0.55	0.55
E=1	0.4	0.6	0.45	0.45

P(A B, E)	(B=0, E=0)	(B=0, E=1)	(B=1, E=0)	(B=1, E=1)
A=0	0.90	0.03	0.04	0.03
A=1	0.03	0.90	0.03	0.03
A=2	0.03	0.04	0.90	0.04
A=3	0.04	0.03	0.03	0.90

P(C)	
C=0	0.50
C=1	0.50

P(F)	
F=0	0.50
F=1	0.50

P(D)	
F=0	0.50
F=1	0.50

P(B C)	C=0	C=1
B=0	0.98	0.02
B=1	0.02	0.98

Figure 19: Parameterization of the network in Figure 13.

ton results (step #2 in LGL schema). Both MMHC and HHC use an “OR” strategy but many alternative approaches can be devised. Furthermore, the edge orientation step may be greatly improved over the use of greedy search-and-score. Numerous other obvious instantiations of LGL (for instance combining GLL-PC versions with global algorithms such as GES, and TPDA) can also be implemented with substantial potential for good empirical performance. Moreover, methods to automatically identify optimal variable prioritization for local learning can yield improvements in certain distributions and we outlined related research directions in Section 6.3.

Finally, extending the framework to address broader definitions of feature selection is particularly important. Examples include finding: all sets that give desired trade-off between feature number and predictivity; all sets with smallest cost that give highest predictivity (i.e., when different observation costs apply for each variable); and all sets that optimize arbitrary multi-attribute utility/loss functions.

8.3 Conclusions

The empirical and theoretical results presented in the present paper and its companion paper (Aliferis et al., 2010) support the notion that local causal learning in the form of Markov blanket and local neighborhood induction is a theoretically well-motivated and empirically robust learning methodology as embodied in the Generalized Local Learning framework. Generalized Local Learning yields algorithms with excellent performance in data analysis geared toward classification and causal discovery. Local-to-global learning strategies have the potential to enhance large-scale causal discovery. Several existing open problems offer possibilities for non-trivial theoretical and practical discoveries, making this an exciting field of research.

Appendix A.

This Appendix provides additional tables and figures referenced in the paper.

Bayesian network	Number of variables	Training samples	Number of selected targets
<i>Child10</i>	200	5 x 200, 5 x 500, 1 x 5000	10
<i>Insurance10</i>	270	5 x 200, 5 x 500, 1 x 5000	10
<i>Alarm10</i>	370	5 x 200, 5 x 500, 1 x 5000	10
<i>Hailfinder10</i>	560	5 x 200, 5 x 500, 1 x 5000	10
<i>Munin</i>	189	5 x 500, 1 x 5000	6
<i>Pigs</i>	441	5 x 200, 5 x 500, 1 x 5000	10
<i>Link</i>	724	5 x 200, 5 x 500, 1 x 5000	10
<i>Lung_Cancer</i>	800	5 x 200, 5 x 500, 1 x 5000	11
<i>Gene</i>	801	5 x 200, 5 x 500, 1 x 5000	11

Table 15: Simulated and resimulated data sets used for experiments. The *Lung_Cancer* network is resimulated from human lung cancer gene expression data (Bhattacharjee et al., 2001) using the SCA algorithm (Friedman et al., 1999). The *Gene* network is resimulated from yeast cell cycle gene expression data (Spellman et al., 1998) using SCA algorithm. More details about data sets are provided in Tsamardinos et al. (2006).

HITON-PC (max k=4)	Interleaved MMPC (max k=2)
HITON-PC (max k=3)	Interleaved MMPC (max k=1)
HITON-PC (max k=2)	HITON-MB (max k=3)
HITON-PC (max k=1)	MMMB (max k=3)
Interleaved HITON-PC (max k=4)	RFE (reduction of features by 50%)
Interleaved HITON-PC (max k=3)	RFE (reduction of features by 20%)
Interleaved HITON-PC (max k=2)	UAF-KruskalWallis-SVM (50%)
Interleaved HITON-PC (max k=1)	UAF-KruskalWallis-SVM (20%)
MMPC (max k=4)	UAF-Signal2Noise-SVM (50%)
MMPC (max k=3)	UAF-Signal2Noise-SVM (20%)
MMPC (max k=2)	L0
MMPC (max k=1)	LARS-EN (for multiclass response)
Interleaved MMPC (max k=4)	LARS-EN (one-versus-rest)
Interleaved MMPC (max k=3)	

Table 16: Algorithms used in local causal discovery experiments with simulated and resimulated data.

References

- C. F. Aliferis and G. F. Cooper. Aspects of modeling with mtbn's. *Technical report CBMI 1998-3, Center for Biomedical Informatics, University of Pittsburgh*, 1998.
- C. F. Aliferis and A. Statnikov. Dynamic ordering-based global learning. *Technical report DSL-08-02*, 2008.

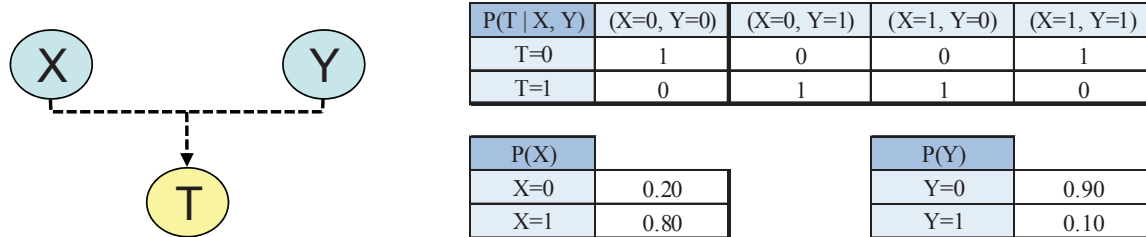


Figure 20: In this example, $T = XOR(X, Y)$. The priors of X and Y are given in the table. Both X and Y have very strong univariate association with T despite being XOR parents and in the absence of connectivity.

- C. F. Aliferis, I. Tsamardinos, and A. Statnikov. Large-scale feature selection using markov blanket induction for the prediction of protein-drug binding. *Technical Report DSL 02-06*, 2002.
- C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification. part i: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11:171–234, 2010.
- Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.
- Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Ann. Statist.*, 29(4):1165–1188, 2001.
- A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker, and M. Meyerson. Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proc. Natl. Acad. Sci. U.S.A.*, 98(24):13790–13795, Nov 2001.
- L. E. Brown, I. Tsamardinos, and C. F. Aliferis. A comparison of novel and state-of-the-art polynomial bayesian network learning algorithms. *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*, 2005.
- G. Casella and R. L. Berger. *Statistical Inference*. Thomson Learning, Australia, 2nd edition, 2002.
- N. Friedman, I. Nachman, and D. Pe’er. Learning bayesian network structure from massive datasets: the “sparse candidate” algorithm. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 1999.
- 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):389–422, 2002.

- I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh. *Feature Extraction: Foundations and Applications*. Springer-Verlag, Berlin, 2006.
- D. Hardin, I. Tsamardinos, and C. F. Aliferis. A theoretical characterization of linear svm-based feature selection. *Proceedings of the Twenty First International Conference on Machine Learning (ICML)*, 2004.
- I. S. Kohane, A. T. Kho, and A. J. Butte. *Microarrays for an Integrative Genomics*. MIT Press, Cambridge, Mass, 2003.
- R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2): 273–324, 1997.
- R. E. Lenski, C. Ofria, R. T. Pennock, and C. Adami. The evolutionary origin of complex features. *Nature*, 423(6936):139–144, May 2003.
- D. Margaritis and S. Thrun. Bayesian network induction via local neighborhoods. *Advances in Neural Information Processing Systems*, 12:505–511, 1999.
- C. Meek. Strong completeness and faithfulness in bayesian networks. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 411–418, 1995.
- J. Ramsey, J. Zhang, and P. Spirtes. Adjacency-faithfulness and conservative causal inference. *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, 2006.
- M. Schmidt, A. Niculescu-Mizil, and K. Murphy. Learning graphical model structure using l_1 -regularization paths. *Proceedings of the Twenty-Second National Conference on Artificial Intelligence (AAAI)*, 2007.
- O. J. Sharpe. *Towards a Rational Methodology for Using Evolutionary Search Algorithms*. PhD thesis, University of Sussex, 2000.
- P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Mol.Biol Cell*, 9(12):3273–3297, Dec 1998.
- P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, Prediction, and Search*, volume 2nd. MIT Press, Cambridge, Mass, 2000.
- A. Statnikov. Algorithms for discovery of multiple markov boundaries: Application to the molecular signature multiplicity problem. *Ph.D.Thesis, Department of Biomedical Informatics, Vanderbilt University*, 2008.
- A. Statnikov, D. Hardin, and C. F. Aliferis. Using svm weight-based methods to identify causally relevant and non-causally relevant variables. *Proceedings of the NIPS 2006 Workshop on Causality and Feature Selection*, 2006.

- A. Statnikov, J. Feig, E. Fisher, and C.F. Aliferis. Novel bioinformatics methods for discovery of complex molecular signatures, pathways, and biomarkers in very small sample situations. *Submitted*, 2010.
- I. Tsamardinos and C. F. Aliferis. Towards principled feature selection: relevancy, filters and wrappers. *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics (AI & Stats)*, 2003.
- I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Algorithms for large scale markov blanket discovery. *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pages 376–381, 2003a.
- I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Time and sample efficient discovery of markov blankets and direct causal relations. *Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 673–678, 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.