Journal of Machine Learning Research 1 (2009) 1-48 Submitted 4/00; Published 10/00

Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification. Part I: Algorithms and Empirical Evaluation

Constantin F. Aliferis CONSTANTIN.ALIFERIS@NYUMC.ORG
Center of Health Informatics and Bioinformatics
New York University
New York, NY 10016, USA

Alexander Statnikov STATNIKOV@GMAIL.COM
Center of Health Informatics and Bioinformatics
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

We present an algorithmic framework for learning local causal structure around target variables of interest in the form of direct causes/effects and Markov blankets applicable to very large datasets with relatively small samples. The selected feature sets can be used for causal discovery and classification. The framework (*Generalized Local Learning*, or GLL) can be instantiated in numerous ways, giving rise to both existing state-of-the-art as well as novel algorithms. The resulting algorithms are sound under well-defined sufficient conditions. In a first set of experiments we evaluate several algorithms derived from this framework in terms of predictivity and feature set parsimony and compare to other local causal discovery methods and to state-of-the-art non-causal feature selection methods using real data. A second set of experimental evaluations compares the algorithms in terms of ability to induce local causal neighborhoods using simulated and resimulated data and examines the relation of predictivity with causal induction performance.

Our experiments demonstrate, consistently with causal feature selection theory, that local causal feature selection methods (under broad assumptions encompassing appropriate family of distributions, types of classifiers, and loss functions) exhibit strong feature set

parsimony, high predictivity and local causal interpretability. Although non-causal feature selection methods are often used in practice to shed light on causal relationships, we find that they cannot be interpreted causally even when they achieve excellent predictivity. Therefore we conclude that only local causal techniques should be used when insight into causal structure is sought.

In a companion paper we examine in depth the behavior of GLL algorithms, provide extensions, and show how local techniques can be used for scalable and accurate global causal graph learning.

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

1. Introduction

This paper addresses the problem of how to learn local causal structure around a target variable of interest using observational data. We focus on two specific types of local discovery: (a) identification of variables that are direct causes or direct effects of the target, and (b) discovery of Markov blankets. A Markov Blanket of a variable T is a minimal variable subset conditioned on which all other variables are probabilistically independent of T.

Discovery of local causal relationships is significant because it plays a central role in causal discovery and classification, because of its scalability benefits, and because by naturally bridging causation with predictivity, it provides significant benefits in feature selection for classification. More specifically, solving the local causal induction problem helps understanding how natural and artificial systems work; it helps identify what interventions to pursue in order for these systems to exhibit desired behaviors; under certain assumptions, it provides minimal feature sets required for classification of a chosen response variable with maximum predictivity; and finally local causal discovery can form the basis of efficient algorithms for learning the global causal structure of all variables in the data.

The paper is organized as follows: Section 2 provides necessary background material. The section summarizes related prior work in feature selection and causal discovery; reviews recent results that connect causality with predictivity; explains the central role of local causal discovery for achieving scalable global causal induction; reviews prior methods for local causal and Markov blanket discovery and published applications; finally it introduces the open problems that are the focus of the present report. Section 3 provides formal concepts and definitions used in the paper. Section 4 provides a general algorithmic framework, *Generalized Local Learning (GLL)*, which can be instantiated in many different ways yielding sound algorithms for local causal discovery and feature selection. Section 5 evaluates a multitude of algorithmic instantiations and parameterizations from GLL and compares them to state-of-the-art local causal discovery and feature selection methods in terms of classification performance, feature set parsimony, and execution time in many real datasets. Section 6 evaluates and compares new and state-of-the-art algorithms in terms of ability to induce correct local neighborhoods using simulated data from known networks and resimulated data from real-life datasets. Section 7 discusses the experimental findings and their significance.

The experiments presented here support the conclusion that local structural learning in the form of Markov blanket and local neighborhood induction is a theoretically well-motivated and empirically robust learning framework that can serve as a powerful tool for

data analysis geared toward classification and causal discovery. At the same time several existing open problems offer possibilities for non-trivial theoretical and practical discoveries making it an exciting field of research. A companion paper (part II of the present work) studies the GLL algorithm properties empirically and theoretically, introduces algorithmic extensions, and connects local to global causal graph learning (Aliferis et al., 2009). An online supplement to the present work is available at http://www.dsl-lab.org/supplements/JMLR2008/. In addition to supplementary tables and figures, the supplement provides all software and data needed to reproduce the analyses of the present paper.

2. Background

2.1 Brief review of feature selection and causal discovery research

Variable selection for predictive modeling (also called feature selection) has received considerable attention during the last three decades both in statistics and in machine learning (Guyon and Elisseeff, 2003; Kohavi and John, 1997). Intuitively, variable selection for prediction aims to select only a subset of variables for constructing a diagnostic or predictive model for a given classification or regression task. The reasons to perform variable selection include (a) improving the model predictivity and addressing the curse-of-dimensionality, (b) reducing the cost of observing, storing, and using the predictive variables, and finally, (c) gaining an understanding of the underlying process that generates the data. The problem of variable selection is more pressing than ever, due to the recent emergence of extremely large datasets, sometimes involving tens to hundreds of thousands of variables and exhibiting a very small sample-to-variable ratio. Such datasets are common in gene expression array studies, proteomics, computational biology, text categorization, information retrieval, image classification, business data analytics, consumer profile analysis, temporal modeling, and other domains and data-mining applications.

There are many different ways to define the variable selection problem depending on the needs of the analysis. Often however, the feature selection problem for classification/prediction is defined as identifying the minimum-size subset of variables that exhibit the maximal predictive performance (Guyon and Elisseeff, 2003). Variable selection methods can be broadly categorized into *wrappers* (i.e., heuristic search in the space of all possible variable subsets using a classifier of choice to assess each subsets predictive information), or *filters* (i.e., not using the classifier per se to select features, but instead applying statistical criteria to first select features and then build the classifier with the best features). In addition, there exist learners that perform *embedded variable selection*, i.e., that attempt to simultaneously maximize classification performance while minimizing the number of variables used. For example, shrinkage regression methods introduce a bias into the parameter estimation regression procedure that imposes a penalty on the size of the parameters. The parameters that are close to zero are essentially filtered-out from the predictive model.

A variety of embedded variable selection methods have been recently introduced. These methods are linked to a statement of the classification or regression problem as an optimization problem with specified loss and penalty functions. These techniques usually fall into a few broad classes: One class of methods utilizes the L^2-norm penalty (also known as ridge penalty), e.g. the recursive feature elimination (RFE) method is based on the

L^2-norm formulation of SVM classification problem (Rakotomamonjy, 2003; Guyon et al., 2002). Other methods are based on the L^1-norm penalty (also known as lasso penalty), e.g. feature selection via solution of the L^1-norm formulation of SVM classification problem (Zhu et al., 2004; Fung and Mangasarian, 2004) and penalized least squares with lasso penalty on the regression coefficients (Tibshirani, 1996). A third set of methods is based on convex combinations of the L^1- and L^2-norm penalties, e.g. feature selection using the doubly SVM formulation (Wang et al., 2006) and penalized least squares with elastic net penalty (Zou and Hastie, 2005). A fourth set uses the L^0-norm penalty, e.g. feature selection via approximate solution of the L^0-norm formulation of SVM classification problem (Weston et al., 2003). Finally other methods use other penalties, e.g. smoothly clipped absolute deviation penalty (Fan and Li, 2001).

Despite the recent emphasis on mathematically sophisticated methods such as the ones mentioned, the majority of feature selection methods in the literature and in practice are heuristic in nature in the sense that in most cases it is unknown what consists an optimal feature selection solution *independently of the class of models fitted*, and under which conditions an algorithm will output such an optimal solution.

Typical variable selection approaches also include forward, backward, forward-backward, local and stochastic search wrappers (Guyon and Elisseeff, 2003; Kohavi and John, 1997; Caruana and Freitag, 1994). The most common family of filter algorithms ranks the variables according to a score and then selects for inclusion the top k variables (Guyon and Elisseeff, 2003). The score of each variable is often the univariate (pairwise) association with the outcome variable T for different measures of associations such as the signal-to-noise ratio, the G^2 statistic and others. Information-theoretic (estimated mutual information) scores and multivariate scores, such as the weights received by a Support Vector Machine, have also been suggested (Guyon and Elisseeff, 2003; Guyon et al., 2002). Excellent recent reviews of feature selection can be found in (Guyon et al., 2006a; Guyon and Elisseeff, 2003; Liu and Motoda, 1998).

An emerging successful but also principled filtering approach in variable selection, and the one largely followed in this paper, is based on identifying the Markov blanket of the response ("target") variable T. The Markov blanket of T (denoted as $MB(T)$) is defined as a minimal set conditioned on which all other *measured* variables become independent of T (more details in section 3).

While classification is often useful for *recognizing or predicting the behavior* of a system, in many problem-solving activities one needs to *change the behavior* of the system (i.e., to "manipulate it"). In such cases, knowledge of the causal relations among the various parts of the system is necessary. Indeed, in order to design new drugs and therapies, institutional policies, or economic strategies, one needs to know how the diseased organism, the institution, or the economy work. Often, heuristic methods based on multivariate or univariate associations and prediction accuracy are used to induce causation, e.g., consider as causally "related" the features that have a strong association with T. Such heuristics may lead to several pitfalls and erroneous inductions, as we will show in the present paper. For principled causal discovery with known theoretical properties a causal theory is needed and classification is not, in general, sufficient (Spirtes et al., 2000; Pearl, 2000; Glymour and Cooper, 1999). Consider the classical epidemiologic example of the tar-stained finger of the heavy smoker: it does predict important outcomes (e.g., increased likelihood for heart

attack and lung cancer). However, eliminating the yellow stain by washing the finger does not alter these outcomes. While experiments can help discover causal structure, quite often experimentation is impossible, impractical, or unethical. For example, it is unethical to force people to smoke and it is currently impossible to manipulate most genes in humans in order to discover which genes cause disease and how they interact in doing so. Moreover, the discoveries anticipated due to the explosive growth of biomedical and other data cannot be made in any reasonable amount of time using solely the classical experimental approach where a single gene, protein, treatment, or intervention is attempted each time, since the space of needed experiments is immense. It is clear that computational methods are needed to catalyze the discovery process.

Fortunately, relatively recently (1980s), it was shown that it is possible to soundly infer causal relations from *observational* data in many practical cases (Spirtes et al., 2000; Pearl, 2000; Glymour and Cooper, 1999; Pearl, 1988). Since then, algorithms that infer such causal relations have been developed that can greatly reduce the number of experiments required to discover the causal structure. Several empirical studies have verified their applicability (Tsamardinos et al., 2003b; Spirtes et al., 2000; Glymour and Cooper, 1999; Aliferis and Cooper, 1994).

One of the most common methods to model and induce causal relations is by learning causal Bayesian networks (Neapolitan, 2004; Spirtes et al., 2000; Pearl, 2000). A special, important and quite broad class of such networks is the family of *faithful networks* intuitively defined as those whose probabilistic properties, and specifically the dependencies and independencies, are a direct function of their structure (Spirtes et al., 2000). Cooper and Herskovits were the first to devise a score measuring the fit of a network structure to the data based on Bayesian statistics, and used it to learn the highest score network structure (Cooper and Herskovits, 1992). Heckerman and his colleagues studied theoretically the properties of the various scoring metrics as they pertain to causal discovery (Glymour and Cooper, 1999; Heckerman, 1995; Heckerman et al., 1995). Heckerman also recently showed that Bayesian-scoring methods also assume (implicitly) faithfulness, see Chapter 4 of (Glymour and Cooper, 1999). Another prototypical method for learning causal relationships by inducing causal Bayesian networks is the constraint-based approach as exemplified in the PC algorithm by Spirtes and Glymour (Spirtes et al., 2000). The PC induces causal relations by assuming faithfulness and by performing tests of independence. A network with a structure consistent with the results of the tests of independence is returned. Several other methods for learning networks have been devised subsequently (Chickering, 2003; Moore and Wong, 2003; Cheng et al., 2002a; Friedman et al., 1999b).

There may be many different networks that fit the data equally well, even in the sample limit, and that exhibit the same dependencies and independencies and are thus statistically equivalent. These networks belong to the same Markov equivalence class of causal graphs and contain the same causal edges but may disagree on the direction of some of them, i.e., whether A causes B or vice-versa (Chickering, 2002; Spirtes et al., 2000). An *essential graph* is a graph where the directed edges represent the causal relations on which all equivalent networks agree upon their directionality and all the remaining edges are undirected. Causal discovery by employing causal Bayesian networks is based on the following principles. The PC (Spirtes et al., 2000), Greedy Equivalence Search (Chickering, 2003) and other prototypical or state-of-the-art Bayesian network-learning algorithms provide theoretical guarantees,

that under certain conditions such as faithfulness they will converge to a network that is statistically indistinguishable from the true, causal, data-generating network, if there is such. Thus, if the conditions hold the existence of all and the direction of some of the causal relations can be induced by these methods and graphically identified in the essential graph of the learnt network.

A typical condition of the aforementioned methods is causal sufficiency (Spirtes et al., 2000). This condition requires that for every pair of measured variables all their common direct causes are also measured. In other words, there are no hidden, unmeasured confounders for any pair of variables. Algorithms, such as the FCI, that in some cases can discover causal relationships in the presence of hidden confounding variables and selection bias, have also been designed (see (Spirtes et al., 2000) and Chapter 6 of (Glymour and Cooper, 1999)).

As it was mentioned above, using observational data alone (even a sample of an infinite size), one can infer only a Markov equivalence class of causal graphs, which may be inadequate for causal discovery. For example, it is not possible to distinguish with observational data any of these two graphs that belong to the same Markov equivalence class: $X \rightarrow Y$ and $X \leftarrow Y$. However, experimental data can distinguish between these graphs. For example, if we manipulate X and see no change in the distribution of Y, we can conclude that the data-generative graph is not $X \rightarrow Y$. This principle is exploited by active learning algorithms. Generally speaking, causal discovery with active learning can be described as follows: learn an approximation of a causal network structure from available data (which is initially only observational data), select and perform an experiment that maximizes some utility function, augment data and possibly current best causal network with the result of experiment, and repeat the above steps until some termination criterion is met.

(Cooper and Yoo, 1999) proposed a Bayesian scoring metric that can incorporate both observational and experimental data. Using a similar metric (Tong and Koller, 2001) designed an algorithm to select experiments that reduce the entropy of probability of alternative edge orientations. A similar but more general algorithm has been proposed in (Murphy, 2001) where the expected information gain of a new experiment is calculated and the experiment with the largest information gain is selected. Both above methods were designed for discrete data distributions. (Pournara and Wernisch, 2004) proposed another active learning algorithm that utilizes a loss function defined in terms of the size of transition sequence equivalence class of networks (Tian and Pearl, 2001) and can handle continuous data. (Meganck et al., 2006) have introduced an active learning algorithm that is based on a general decision theoretic framework that allows to assign costs to each experiment and each measurement. It is also worthwhile to mention the GEEVE system of (Yoo and Cooper, 2004) that recommends which experiments to perform to discover gene-regulation pathway. This instance of causal active learning allows to incorporate preferences of the experimenter. Recent work has also provided theoretical bounds and related algorithms to minimize the number of experiments needed to infer causal structure (Eberhardt et al., 2006; Eberhardt et al., 2005).

2.2 Synopsis of theoretical results motivating present research

A key question that has been investigated in the feature selection literature is which family of methods is more advantageous: filters or wrappers. A second one is what are the "relevant" features? The latter question presumably is important because "relevant" features should be important for discovery and so several definitions appeared defining relevancy (Guyon and Elisseeff, 2003; Kohavi and John, 1997). Finally, how can we design optimal and efficient feature selection algorithms? Fundamental theoretical results connecting Markov blanket induction for feature selection and local causal discovery to standard notions of relevance were given in (Tsamardinos and Aliferis, 2003). The latter paper provides a technical account and together with (Spirtes et al., 2000; Pearl, 2000; Kohavi and John, 1997; Pearl, 1988) they constitute the core theoretical framework underpinning the present work. Here we provide a very concise description of the results in (Tsamardinos and Aliferis, 2003) since they partially answer these questions and pave the way to principled feature selection:

1. Relevance cannot be defined independently of the learner and the model-performance metric (e.g., the loss function used) in a way that the relevant features are the solution to the feature selection problem. The quest for a universally applicable notion of relevancy for prediction is futile.
2. Wrappers are subject to the No-Free Lunch Theorem for optimization: averaged out on all possible problems any wrapper algorithm will do as well as a random search in the space of feature subsets. Therefore, there cannot be a wrapper that is a priori more efficient than any other (i.e., without taking into account the learner and model-performance metric). The quest for a universally efficient wrapper is futile as well.
3. Any filter algorithm can be viewed as the implementation of a definition of relevancy. Because of #1, there is no filter algorithm that is universally optimal, independently of the learner and model-performance metric.
4. Because of #2, wrappers cannot guarantee universal efficiency and because of #3, filters cannot guarantee universal optimality and in that respect, neither approach is superior to the other.
5. Under the conditions that (i) the learner that constructs the classification model can actually learn the distribution $P(T|MB(T))$ and (ii) that the loss function is such that perfect estimation of the probability distribution of T is required with the smallest number of variables, the Markov blanket of T is the optimal solution to the feature selection problem.
6. Sound Markov blanket induction algorithms exist for faithful distributions.
7. In faithful distributions and under the conditions of #5, the strongly/weakly/irrelevant taxonomy of variables (Kohavi and John, 1997) can be mapped naturally to causal graph properties. Informally stated, strongly relevant features were defined by (Kohavi and John, 1997) to be features that contain information about the target not found in other variables; weakly relevant features are informative but redundant; irrelevant features are not informative (for formal definitions see section 3). Under the causal interpretation of this taxonomy of relevancy, strongly relevant features are the members of the Markov blanket of the target variable, weakly relevant features are all variables with an undirected path to T which are not themselves members of $MB(T)$, and irrelevant features are variables with no undirected path to the target.

8. Since in faithful distributions the $MB(T)$ contains the direct causes and direct effects of T, and since state-of-the-art $MB(T)$ algorithms output the spouses separately from the direct causes and direct effects, inducing the $MB(T)$ not only solves the feature selection problem but also a form of local causal discovery problem.

Figure 2.1 provides a summary of the connection between causal structure and predictivity.

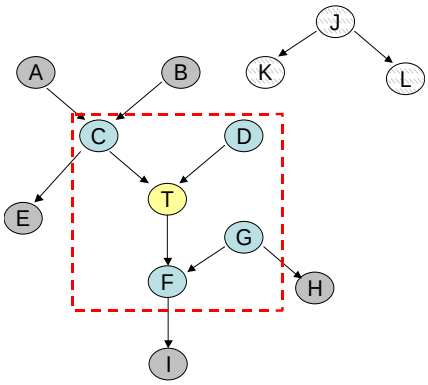


Figure 2.1: Relationship between causal structure and predictivity in faithful distributions. Cyan variables are members of Markov blanket of T. They are depicted inside the red dotted square (i.e., variables that have undirected path to target T and that are predictive of T given the remaining variables which makes them strongly relevant). Markov blanket variables include direct causes of T (C, D), direct effects (F), and "spouses" of T (i.e., direct causes of the direct effects of T) (G). Grey variables are non-members of Markov blanket of T that have undirected path to T. They are not predictive of T given the remaining variables but they are predictive given a subset of the remaining variables (which makes them weakly relevant). Light-gray variables are variables that do not have an undirected path to T. They are not predictive of T given any subset of the remaining variables, thus they are irrelevant.

We will refer to algorithms that perform feature selection by formal causal induction as *causal feature selection* and algorithms that do not as *non-causal*. As highly complementary to the above results we would add the arguments in favor of causal feature selection presented in (Guyon et al., 2007) and recent theoretical (Hardin et al., 2004) and empirical (Statnikov et al., 2006) results that show that under the same sufficient conditions that make Markov blanket the optimal solution to the feature selection and local causal discovery problem, state-of-the-art methods such as ranking features by SVM weights (RFE being a prototypical algorithm (Guyon et al., 2002)) do not return the correct causal neighborhood and are not minimal (i.e., do not solve the feature selection problem) even in the large sample limit.

The above theoretical results also suggest that one should not attempt to define and identify the relevant features for prediction, when discovery is the goal of the analysis. Instead, we argue that a set of features with well-defined *causal* semantics should be identified instead: for example, the $MB(T)$, the set of direct causes and direct effects of T, the set of all (direct and indirect) causes of T, and so on.

We will investigate limitations of prominent non-causal feature selection algorithms in the companion paper (Aliferis et al., 2009).

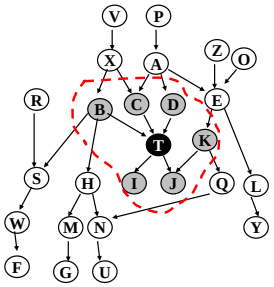
2.3 Methods to speed-up discovery: local discovery as a critical tool for scalability

As appealing as causal discovery may be for understanding a domain, predicting effects of intervention, and pursuing principled feature selection for classification, a major problem up until recent years has been scalability. The PC algorithm is worst-case exponential (Spirtes et al., 2000) and in practical settings it cannot typically handle more than a hundred variables. The FCI algorithm is similarly worst-case intractable (Spirtes et al., 2000) and does not handle more than a couple of dozen of variables practically. Learning Bayesian networks with Bayesian scoring techniques is NP-Hard (Chickering et al., 1994). Heuristic hill-climbing techniques such as the Sparse Candidate Algorithm (Friedman et al., 1999b) do not provide guaranteed correct solutions, neither they are very efficient (they can cope with a few hundred variables at the most in practical applications).

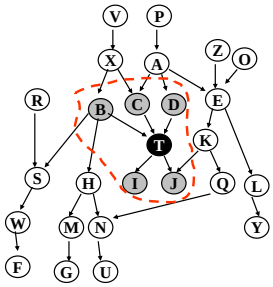
With the advent of massive datasets in biology, medicine, information retrieval, the WWW, finance, economics, and so on, scalability has become a critical requirement for practical algorithms. In early 2000's predictions about the feasibility of causal discovery in high-dimensional data were bleak (Silverstein et al., 2000). A variety of methods to scale up causal discovery have been devised to address the problem:

1. Learn the full graph but focus on special types of distributions;
2. Exploit domain knowledge to speed-up learning;
3. Abandon the effort to learn the full causal graph and instead develop methods that find a portion of the true arcs (not specific to some target variable);
4. Abandon the effort to learn the full causal graph and instead develop methods that learn the local neighborhood of a specific target variable directly;
5. Abandon the effort to learn the fully oriented causal graph and instead develop methods that learn the unoriented graph;
6. Induce constrains of the possible relationships among variables and then learn the full causal graph.

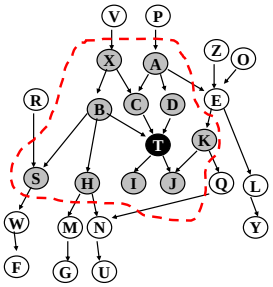
Techniques #1 and #2 were introduced in (Chow and Liu, 1968) for learning tree-like graphs and Naïve-Bayes graphs (Duda and Hart, 1973), while modern versions are exemplified in (i) TAN/BAN classifiers that relax the Naïve-Bayes structure (Cheng and Greiner, 2001; Cheng and Greiner, 1999; Friedman et al., 1997), (ii) efficient complete model averaging of Naïve-Bayes classifiers (Dash and Cooper, 2002), and (iii) algorithm TPDA which restricts the class of distributions so that learning becomes from worst-case intractable to solvable in 4^{th} degree polynomial time to the number of variables (and quadratic if prior knowledge about the ordering of variables is known) (Cheng et al., 2002a). Technique #3 was introduced by (Cooper, 1997) and replaced learning the complete graph by learning only a small portion of the edges (not pre-specified by the user but determined by the discovery method). Techniques $\#4 - 6$ pertain to local learning: Technique #4 seeks to learn the complete causal neighbourhood around a target variable provided by the user (Aliferis et al., 2003a; Tsamardinos et al., 2003b). We emphasize that local learning (technique #4) is not the same as technique #3 (incomplete learning) although inventors of incomplete methods often call them local. Technique #5 abandons directionality and learns only a fully connected but undirected graph by using local learning methods (Tsamardinos et al., 2006; Brown et al., 2005). Often post-processing with additional algorithms can provide



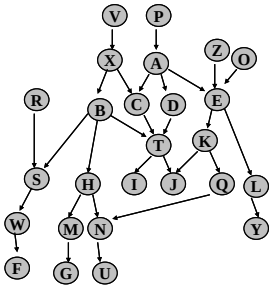
Problem #1: Consider a target variable T and discover Markov Blanket of T.



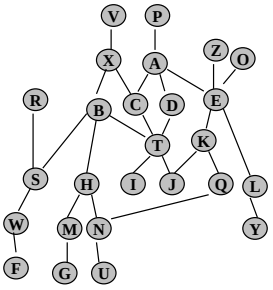
Problem #2: Consider a target variable T and discover Parents and Children of T.



Problem #3: Consider a target variable T and discover regions (e.g., of depth 2 edges) around T.



Problem #4: Discover directed graph.



Problem #5: Discover undirected graph.

Figure 2.2: Five types of causal discovery from local (types 1, 2), to global (4, 5) and intermediate (3). Specialized algorithms that solve type 2 (local causes and effects) can become building blocks for relatively efficiently solving all other types of causal discovery as well (see text for details).

directionality. The latter can also be obtained by domain-specific criteria or experimentation. Finally, technique #6 uses local learning to restrict the search space for full-graph induction algorithms (Tsamardinos et al., 2006; Aliferis and Tsamardinos, 2002b).

In the present paper we explore methods to learn local causal neighborhoods and test them in high-dimensional datasets. In the companion paper (Aliferis et al., 2009) we provide a framework for building global graphs using the local methods. Incomplete learning (technique #3) is not pursued because it is redundant in light of the other (complete) local and global learning approaches. Figure 2.2 provides a visual reference guide to the kinds of causal discovery problems the methods in the present work are able to address by starting from local causal discovery.

2.4 Desiderata for local algorithms, brief review of prior methods for Markov blanket and local neighborhood induction

An ideal local learning algorithm should have three characteristics: (a) well-defined properties, especially broadly applicable conditions that guarantee correctness, (b) good performance in practical distributions and corresponding datasets, including ones with small sample and many features, and finally (c) scalability in terms of running time. We briefly review progress made in the field toward these goals.

Firm theoretical foundations of Bayesian networks were laid down by Pearl and his coauthors (Pearl, 1988). Furthermore, all local learning methods exploit either the constraint-

based framework for causal discovery developed by Spirtes, Glymour, Schienes, Pearl, and Verma and their co-authors (Spirtes et al., 2000; Pearl, 2000; Pearl and Verma, 1991) or the Bayesian search-and-score Bayesian network learning framework introduced by Cooper and Herskovits (Cooper and Herskovits, 1992). The relevant key contributions were covered in sub-section 2.1 and will not be repeated here.

While the above foundations were introduced and developed in the span of at least the last 30 years, local learning is no more than 10 years old. Specialized Markov blanket learning methods were first introduced in 1996 (Koller and Sahami, 1996), incomplete causal methods in 1997 (Cooper, 1997), and local causal discovery methods (for targeted complete induction of direct causes and effects) were first introduced in 2002 and 2003 (Tsamardinos et al., 2003b; Aliferis and Tsamardinos, 2002a). In 1996 Koller et al. introduced a heuristic algorithm for inducing the Markov blanket from data and tested the algorithm in simulated, real text, and other types of data from the UCI repository (Koller and Sahami, 1996). In 1997 Cooper and colleagues introduced and applied the heuristic method K2MB for finding the Markov blanket of a target variable in the task of predicting pneumonia mortality (Cooper et al., 1997). In 1997 Cooper introduced an incomplete method for causal discovery (Cooper, 1997). The algorithm was able to circumvent lack of scalability of global methods by returning a subset of arcs from the full network. To avoid notational confusion we point out that the algorithm was termed LCD (local causal discovery) despite being an *incomplete rather than local* algorithm as local algorithms are defined in the present paper (i.e., focused on some user-specified target variable or localized region of the network). A revision of the algorithm termed LCD2 was presented in (Mani and Cooper, 1999).

In 1999 Margaritis and Thrun introduced the GS algorithm with the intent to induce the Markov blanket for the purpose of speeding up global network learning (i.e., not for feature selection) (Margaritis and Thrun, 1999). GS was the first published sound Markov blanket induction algorithm. The weak heuristic used by GS combined with the need to condition on at least as many variables simultaneously as the Markov blanket size makes it impractical for many typical datasets since the required sample grows exponentially to the size of the Markov blanket. This in turn forces the algorithm to stop its execution prematurely (before it identifies the complete Markov blanket) because it cannot grow the conditioning set while performing reliable tests of independence. Evaluations of GS by its inventors were performed in datasets with a few dozen variables leaving the potential of scalability largely unexplored.

In 2001 Cheng et al. applied the TPDA algorithm (a global BN learner) (Cheng et al., 2002a) to learn the Markov blanket of the target variable in the Thrombin dataset in order to solve a prediction problem of drug effectiveness on the basis of molecular characteristics (Cheng et al., 2002b). Because TPDA could not be run with more than a few hundred variables efficiently, they pre-selected 200 variables (out of 139,351 total) using univariate filtering. Although this procedure in general will not find the true Markov blanket (because otherwise-unconnected with the target spouses can be missed, many true parents and children may not be in the first 200 variables, and many non-Markov blanket members cannot be eliminated), the resulting classifier performed very well winning the 2001 KDD Cup competition.

Friedman et al. proposed a simple Bootstrap procedure for determining membership in the Markov blanket for small sample situations (Friedman et al., 1999a). The Markov

blanket in this method is to be extracted from the full Bayesian network learned by the SCA (Sparse Candidate Algorithm) learner (Friedman et al., 1999b).

In 2002 and 2003 Tsamardinos, Aliferis, et al. presented a modified version of GS, termed IAMB and several variants of the latter that through use of a better inclusion heuristic than GS and optional post-processing of the tentative and final output of the local algorithm with global learners would achieve true scalability to datasets with many thousands of variables and applicability in modest (but not very small) samples (Tsamardinos et al., 2003a; Aliferis et al., 2002). IAMB and several variants were tested both in the high-dimensional Thrombin dataset (Aliferis et al., 2002) and in datasets simulated from both existing and random Bayesian networks (Tsamardinos et al., 2003a). The former study found that IAMB scales to high-dimensional datasets. The latter study compared IAMB and its variants to GS, Koller-Sahami, and PC and concluded that IAMB variants on average perform best in the datasets tested.

In 2003 Tsamardinos and Aliferis presented a full theoretical analysis explaining relevance as defined by Kohavi and John (Kohavi and John, 1997) in terms of Markov blanket and causal connectivity (Tsamardinos and Aliferis, 2003). They also provided theoretical results about the strengths and weaknesses of filter versus wrapper algorithms, the impossibility of a universal definition of relevance, and the optimality of Markov blanket as a solution to the feature selection problem in formal terms. These results were summarized in sub-section 2.2.

The extension of Sparse Candidate Algorithm to create a local-to-global learning strategy was first introduced in (Aliferis and Tsamardinos, 2002b) and led to the MMHC algorithm introduced and evaluated in (Tsamardinos et al., 2006). MMHC was shown in (Tsamardinos et al., 2006) to achieve best-of-class performance in quality and scalability compared to most state-of-the-art global network learning algorithms.

In 2002 Aliferis et al. also introduced parallel and distributed versions of the IAMB family of algorithms (Aliferis et al., 2002). These serve as the precursor of the parallel and distributed local neighborhood learning method presented in the companion paper (Aliferis et al., 2009). The precursor of the GLL framework was also introduced by Aliferis and Tsamardinos in 2002 for the explicit purpose of reducing the sample size requirements of IAMB-style algorithms (Aliferis and Tsamardinos, 2002a).

In 2003 Aliferis et al. introduced algorithm HITON[1] (Aliferis et al., 2003a), and Tsamardinos et al. introduced algorithms MMPC and MMMB (Tsamardinos et al., 2003b). These are the first concrete algorithms that would find sets of direct causes or direct effects and Markov blankets in a scalable and efficient manner. HITON was tested in 5 biomedical datasets spanning clinical, text, genomic, structural and proteomic data and compared against several feature selection methods with excellent results in parsimony and classification accuracy (Aliferis et al., 2003a). MMPC was tested in data simulated from human-derived Bayesian networks with excellent results in quality and scalability. MMMB was tested in the same datasets and compared to prior algorithms such as Koller-Sahami algorithm and IAMB variants with superior results in the quality of Markov blankets. These benchmarking and comparative evaluation experiments provided evidence that the local learning approach held not only theoretical but also practical potential.

1. From the Greek word "Χιτώνν" meaning "cloak", and pronounced <hee tó n>.

HITON-PC, HITON-MB, MMPC, and MMMB algorithms lacked so-called "symmetry correction" (Tsamardinos et al., 2006), however HITON used a wrapping post-processing that at least in principle removed this type of false positives. The symmetry correction was introduced in 2005 and 2006 by Tsamardinos et al. in the context of the introduction of MMHC (Tsamardinos et al., 2006; Tsamardinos et al., 2005). Peña et al. also published work pointing to the need for a symmetry correction in MMPC (Peña et al., 2005b).

HITON was applied in 2005 to understand physician decisions and guideline compliance in the diagnosis of melanomas (Sboner and Aliferis, 2005). HITON has been applied for the discovery of biomarkers in human cancer data using microarrays and mass spectrometry and is also implemented in the GEMS and FAST-AIMS systems for the automated analysis of microarray and mass spectrometry data respectively (Statnikov et al., 2005b; Fananapazir et al., 2005). In a recent extensive comparison of biomarker selection algorithms (Aliferis et al., 2006a; Aliferis et al., 2006b) it was found that HITON outperforms 16 state-of-the-art representatives from all major biomarker algorithmic families in terms of combined classification performance and feature set parsimony. This evaluation utilized 9 human cancer datasets (gene expression microarray and mass spectrometry) in 10 diagnostic and outcome (i.e., survival) prediction classification tasks. In addition to the above real data, resimulation was also used to create two gold standard network structures, one re-engineered from human lung cancer data and one from yeast data. Several applications of HITON in text categorization have been published where the algorithm was used to understand complex "black box" SVM models and convert complex models to Boolean queries usable by Boolean interfaces of Medline (Aphinyanaphongs and Aliferis, 2004), to examine the consistency of editorial policies in published journals (Aphinyanaphongs et al., 2006), and to predict drug-drug interactions (Duda et al., 2005). HITON was also compared with excellent results to manual and machine feature selection in the domain of early graft failure in patients with liver transplantations (Hoot et al., 2005).

In 2003 Frey et al. explored the idea of using decision tree induction to indirectly approximate the Markov blanket (Frey et al., 2003). They produced promising results, however a main problem with the method was that it requires a threshold parameter that cannot be optimized easily. Furthermore, as we show in the companion paper (Aliferis et al., 2009) decision tree induction is subject to synthesis and does not select only the Markov blanket members.

In 2004 Mani et al. introduced BLCD-MB, which resembles IAMB but using a Bayesian scoring metric rather than conditional independence testing (Mani and Cooper, 2004). The algorithm was applied with promising results in infant mortality data (Mani and Cooper, 2004).

A method for learning regions around target variables by recursive application of MMPC or other local learning methods was introduced in (Tsamardinos et al., 2003c). Peña et al. applied interleaved MMPC for learning regions in the domain of bioinformatics (Peña et al., 2005a).

In 2006 Gevaert et al. applied K2MB for the purpose of learning classifiers that could be used for prognosis of breast cancer from microarray and clinical data (Gevaert et al., 2006) . Univariate filtering was used to select 232 genes before applying K2MB.

Other recent efforts in learning Markov blankets include the following algorithms: PCX, which post-processes the output of PC (Bai et al., 2004); KIAMB, which addresses some

violations of faithfulness using a stochastic extension to IAMB (Pea et al., 2007); FAST-IAMB, which speeds up IAMB (Yaramakala and Margaritis, 2005); and MBFS, which is a PC-style algorithm that returns a graph over Markov blanket members (Ramsey, 2006).

2.5 Open problems and focus of paper

The focus of the present paper is to describe state-of-the-art algorithms for inducing direct causes and effects of a response variable or its Markov blanket using a novel cohesive framework that can help in the analysis, understanding, improvement, application (including configuration/parameterization) and dissemination of the algorithms. We furthermore study comparative performance in terms of predictivity and parsimony of state-of-the-art local causal algorithms; we compare them to non-causal algorithms in real and simulated datasets using the same criteria; and show how novel algorithms can be obtained. A second major hypothesis (and set of experiments in the present paper) is that non-causal feature selection methods may yield predictively optimal feature sets while from a causal perspective their output is unreliable. Testing this hypothesis has tremendous implications in many areas (e.g., analysis of biomedical molecular data) where highly predictive variables (biomarkers) of phenotype (e.g., disease or clinical outcome) are often interpreted as being causally implicated for the phenotype and great resources are invested in pursuing these markers for new drug development and other research.

In the second part of our work (Aliferis et al., 2009) we address gaps in the theoretical understanding of local causal discovery algorithms and provide empirical and theoretical analyses of their behavior as well as several extensions including algorithms for learning the full causal graph using a divide-and-conquer local learning approach.

3. Notation and definitions

In the present paper we use Bayesian networks as the language in which to represent data generating processes and causal relationships. We thus first formally define causal Bayesian networks. Recall that in a directed acyclic graph (DAG), a node A is the parent of B (B is the child of A) if there is a direct edge from A to B, A is the ancestor of B (B is the descendant of A) if there is a direct path from A to B. "Nodes", "features", and "variables" will be used interchangeably.

Notation: We will denote variables with uppercase letters X, Y, Z, values with lowercase letters, x, y, z, and sets of variables or values with boldface uppercase or lowercase respectively. A "target" (i.e., response) variable is denoted as T unless stated otherwise.

Definition 1: Conditional Independence. Two variables X and Y are conditionally independent given $\boldsymbol{Z}$, denoted as $I(X, Y|\boldsymbol{Z})$, iff $P(X = x, Y = y|\boldsymbol{Z} = \boldsymbol{z}) = P(X = x|\boldsymbol{Z} = \boldsymbol{z})P(Y = y|\boldsymbol{Z} = \boldsymbol{z})$, for all values $x, y, \boldsymbol{z}$ of $X, Y, \boldsymbol{Z}$ respectively, such that $P(\boldsymbol{Z} = \boldsymbol{z}) > 0$.

Definition 2: Bayesian network $< \boldsymbol{V}, G, J >$. Let $\boldsymbol{V}$ be a set of variables and J be a joint probability distribution over all possible instantiations of $\boldsymbol{V}$. Let G be a directed acyclic graph (DAG) such that all nodes of G correspond one-to-one to members of $\boldsymbol{V}$. We require that for every node $A \in \boldsymbol{V}$, A is probabilistically independent of all non-descendants of A, given the parents of A (i.e., *Markov Condition* holds). Then we call the triplet $< \boldsymbol{V}, G, J >$ a Bayesian network (abbreviated as "BN"), or equivalently a belief network or probabilistic network (Neapolitan, 1990).

Definition 3: Operational criterion for causation. Assume that a variable A can be forced by a hypothetical experimenter to take values a_i. If the experimenter assigns values to A according to a uniformly random distribution over values of A, and then observes $P(B|A = a_i) \neq P(B|A = a_j)$ for some i and j, (and within a time window dt), then variable A is a cause of variable B (within dt).

We note that randomization of values of A serves to eliminate any combined causative influences on both A and B. We also note that universally acceptable definitions of causation have eluded scientists and philosophers for centuries. Indeed the provided criterion is not a proper definition, because it examines one cause at a time (thus multiple causation can be missed), it assumes that a hypothetical experiment is feasible even when in practice this is not attainable, and the notion of "forcing" variables to take values presupposes a special kind of causative primitive that is formally undefined. Despite these limitations, the above criterion closely matches the notion of a Randomized Controlled Experiment which is a de facto standard for causation in many fields of science, and following common practice in the field (Glymour and Cooper, 1999) will serve operationally the purposes of the present paper.

Definition 4: Direct and indirect causation. Assume that a variable A is a cause of variable B according to the operational criterion for causation in definition 3. A is an indirect cause for B with respect to a set of variables $\boldsymbol{V}$, iff A is not a cause of B for some instantiation of values of $\boldsymbol{V} \setminus \{A, B\}$, otherwise A is a direct cause of B.

Definition 5: Causal probabilistic network (a.k.a. causal Bayesian network). A causal probabilistic network (abbreviated as "CPN") $< \boldsymbol{V}, G, J >$ is the Bayesian network $< \boldsymbol{V}, G, J >$ with the additional semantics that if there is an edge $A \rightarrow B$ in G then A directly causes B (for all $A, B \in \boldsymbol{V}$) (Spirtes et al., 2000).

Definition 6: Faithfulness. A directed acyclic graph G is faithful to a joint probability distribution J over variable set $\boldsymbol{V}$ if and only if every independence present in J is entailed by G and the Markov Condition. A distribution J is faithful if and only if there exists a directed acyclic graph G such that G is faithful to J (Spirtes et al., 2000; Glymour and Cooper, 1999).

It follows from the Markov Condition that in a CPN $C =< \boldsymbol{V}, G, J >$ every conditional independence entailed by the graph G is also present in the probability distribution J encoded by C. *Thus, together faithfulness and the causal Markov Condition establish a close relationship between a causal graph G and some empirical or theoretical probability distribution J. Hence we can associate statistical properties of the sample data with causal properties of the graph of the CPN.* The d-separation criterion determines all independencies entailed by the Markov Condition and a graph G.

Definition 7: d-separation, d-connection. A *collider* on a path p is a node with two incoming edges that belong to p. A path between X and Y given a conditioning set $\boldsymbol{Z}$ is open, if (i) every collider of p is in $\boldsymbol{Z}$ or has a descendant in $\boldsymbol{Z}$, and (ii) no other nodes on p are in $\boldsymbol{Z}$. If a path is not open, then it is *blocked*. Two variables X and Y are d-separated given a conditioning set $\boldsymbol{Z}$ in a BN or CPN C if and only if every path between X, Y is blocked (Pearl, 1988).

Property 1: Two variables X and Y are d-separated given a conditioning set $\boldsymbol{Z}$ in a faithful BN or CPN if and only if $I(X, Y|\boldsymbol{Z})$ (Spirtes et al., 2000). It follows, that if they are d-connected, they are conditionally dependent.

Thus, in a faithful CPN, *d*-separation captures *all* conditional dependence and independence relations that are encoded in the graph.

Definition 8: Markov blanket of T, denoted as $MB(T)$. A set $MB(T)$ is a minimal set of features with the following property: for every variable subset $\boldsymbol{S}$ with no variables in $MB(T)$, $I(\boldsymbol{S}, T|MB(T))$. In Pearl's terminology this is called the Markov Boundary (Pearl, 1988).

Property 2: The $MB(T)$ of any variable T in a faithful BN or a CPN is unique (Tsamardinos et al., 2003b) (also directly derived from (Pearl and Verma, 1991; Pearl and Verma, 1990)).

Property 3: The $MB(T)$ in a faithful CPN is the set of parents, children, and parents of children (i.e. "spouses") of T (Pearl, 2000; Pearl, 1988).

Definition 9: Causal sufficiency. For every pair of measured variables in the training data, all their direct common causes are also measured.

Definition 10: Feature selection problem. Given a sample S of instantiations of variable set $\boldsymbol{V}$ drawn from distribution D, a classifier induction algorithm C and a loss function L, find: smallest subset of variables $\boldsymbol{F} \subseteq \boldsymbol{V}$ such that $\boldsymbol{F}$ minimizes expected loss $L(M, D)$ in distribution D where M is the classifier model (induced by C from sample S projected on $\boldsymbol{F}$).

In the above definition, we mean "exact" minimization of $L(M, D)$. In other words, out of all possible subsets of variable set $\boldsymbol{V}$, we are interested in subsets $\boldsymbol{F} \subseteq \boldsymbol{V}$ that satisfy the following two criteria: (i) $\boldsymbol{F}$ minimizes $L(M, D)$ and (ii) there is no subset $\boldsymbol{F}^* \subseteq \boldsymbol{V}$ such that $|\boldsymbol{F}^*| < |\boldsymbol{F}|$ and $\boldsymbol{F}^*$ also minimizes $L(M, D)$.

Definition 11: Wrapper feature selection algorithm. An algorithm that tries to solve the Feature Selection problem by searching in the space of feature subsets and evaluating each one with a user-specified classifier and loss function estimator.

Definition 12: Filter feature selection algorithm. An algorithm designed to solve the Feature Selection problem by looking at properties of the data and not by applying a classifier to estimate expected loss for different feature subsets.

Definition 13: Causal feature selection algorithm. An algorithm designed to solve the Feature Selection problem by (directly or indirectly) inducing causal structure and by exploiting formal connections between causation and predictivity.

Definition 14: Non-causal feature selection algorithm. An algorithm that tries to solve the Feature Selection problem without reference to the causal structure that underlies the data.

Definition 15: Irrelevant, strongly relevant, weakly relevant, relevant feature (with respect to target variable T**)**. A variable set $\boldsymbol{I}$ that conditioned on every subset of the remaining variables does not carry predictive information about T is irrelevant to T. Variables that are not irrelevant are called relevant. Relevant variables are strongly relevant if they are predictive for T given the remaining variables, while a variable is weakly relevant if it is non-predictive for T given the remaining variables (i.e., it is not strongly relevant) but it is predictive given some subset of the remaining variables.

4. A general framework for local learning

In this section we present a formal general framework for learning local causal structure. Such a framework enables a systematic exploration of a family of related but not identical algorithms which can be seen as instantiations of the same broad algorithmic principles encapsulated in the framework. Also, the framework allows us to think about formal conditions for correctness not only at the algorithm level but also at the level of algorithm family. We are thus able to identify two distinct sets of assumptions for correctness: the more general set of assumptions (*admissibility rules*) applies to the generative algorithms and provides a set of flexible rules for constructing numerous algorithmic instantiations each one of which is guaranteed to be correct provided that in addition a more specific and fixed set of assumptions hold (i.e., specific sufficient conditions for correctness of the algorithms that are instantiations of the generative framework).

We consider the following two problems of local learning:

Problem 1: Given a set of variables $\boldsymbol{V}$ following distribution P, a sample D drawn from P, and a target variable of interest $T \in \boldsymbol{V}$: determine the direct causes and direct effects of T.

Problem 2: Given a set of variables $\boldsymbol{V}$ following distribution P, a sample D drawn from P, and a target variable of interest $T \in \boldsymbol{V}$: determine the direct causes, direct effects, and the direct causes of the direct effects of T.

From the work of Pearl, Spirtes, et al. (Spirtes et al., 2000; Pearl, 2000; Pearl, 1988) we know that when the data are observational, causal sufficiency holds for the variables $\boldsymbol{V}$, and the distribution P is faithful to a causal Bayesian network, then the direct causes, direct effects, and direct causes of the direct effects of T, correspond to the parents, children, and spouses of T respectively in that network.

Thus, in the context of the above assumptions, Problem 1 seeks to identify the parents and children set of T in a Bayesian network G faithful to P; we will denote this subset as $PC_G(T)$. There may be several networks that faithfully capture distribution P, however, as we have shown in (Tsamardinos et al., 2003b) (also directly derived from (Pearl and Verma, 1991; Pearl and Verma, 1990)) $PC_G(T) = PC_{G'}(T)$, for any two networks G and G' faithful to the same distribution. So, the set of parents and children of T is unique among all Bayesian networks faithful to the same distribution and so we will drop the superscript and denote it simply as $PC(T)$. Notice that, a node may be a parent of T in one network and a child of T in another, e.g., the graphs $X \leftarrow T$ and $X \rightarrow T$ may both be faithful to the same distribution. However, the set of parents and children of T, i.e., $\{X\}$, remains the same in both networks. Finally, by Theorem 4 in (Tsamardinos et al., 2003b) we know that the Markov blanket $MB(T)$ is unique in all networks faithful to the same distribution. Therefore, under the assumptions of the existence of a causal Bayesian network that faithfully captures P and causal sufficiency of $\boldsymbol{V}$, the problems above can be recast as follows:

Problem 1: Given a set of variables $\boldsymbol{V}$ following distribution P, a sample D drawn from P, and a target variable of interest $T \in \boldsymbol{V}$: determine the $PC(T)$.

Problem 2: Given a set of variables $\boldsymbol{V}$ following distribution P, a sample D drawn from P, and a target variable of interest $T \in \boldsymbol{V}$: determine the $MB(T)$.

Problem 1 is geared toward local causal discovery, while Problem 2 is oriented toward causal feature selection for classification. The solutions to these problems can form the basis for solving several other related local discovery problems, such as learning the unoriented set of causal relations (skeleton of a Bayesian network), a region of interest of a given depth of d edges around T, or further analyze the data to discover the orientation of the causal relations.

The *Generalized Local Learning* (GLL) framework consists of two main types of algorithms: GLL-PC (GLL Parent and Children) for Problem 1 and GLL-MB for Problem 2.

4.1 Discovery of the $PC(T)$ set

Identification of the $PC(T)$ set is based on the following theorem in (Spirtes et al., 2000):

Theorem 1: In a faithful BN $< \boldsymbol{V}, G, P >$ there is an edge between the pair of nodes $X \in \boldsymbol{V}$ and $Y \in \boldsymbol{V}$ iff $\neg \mathrm{I}(X, Y | \boldsymbol{Z})$, for all $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, Y\}$.

Any variable X that does have an edge with T belongs to the $PC(T)$. Thus, the theorem gives rise to an immediate algorithm for identifying $PC(T)$: for any variable $X \in \boldsymbol{V} \setminus \{T\}$, and all $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, T\}$, test whether $\mathrm{I}(X, T | \boldsymbol{Z})$. If such a $\boldsymbol{Z}$ exists for which $\mathrm{I}(X, T | \boldsymbol{Z})$, then $X \notin PC(T)$, otherwise $X \in PC(T)$. This algorithm is equivalent to a "localized version" of SGS (Spirtes et al., 2000). The problem of course is that the algorithm is very inefficient because it tests all subsets of the variables and thus does not scale beyond problems of trivial size. The order of complexity is $\mathrm{O}(|\boldsymbol{V}| 2^{|\boldsymbol{V}|-2})$. The general framework presented below attempts to characterize not only the above algorithm but also efficient implementations of the theorem that maintain soundness.

There are several observations that lead to more efficient but still sound algorithms. First notice that, once a subset $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, T\}$ has been found s.t. $\mathrm{I}(X, T | \boldsymbol{Z})$ there is no need to perform any other test of the form $\mathrm{I}(X, T | \boldsymbol{Z}')$: we know that $X \notin PC(T)$. Thus, the sooner we identify good candidate subsets $\boldsymbol{Z}$ that can render the variables conditionally independent from T, the fewer tests will be necessary.

Second, to determine whether $X \in PC(T)$ there is no need to test whether $\neg \mathrm{I}(X, T | \boldsymbol{Z})$ for all subsets $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, T\}$ but only for all subsets $\boldsymbol{Z}' \subseteq Parents_G(T) \setminus \{X\}$ and all $\boldsymbol{Z}' \subseteq Parents_G(X) \setminus \{T\}$ where G is any network faithful to the distribution. To see this, let us first assume that there is no edge between X and T. Notice that either X is a non-descendant of T or T is a non-descendant of X since the network is acyclic and they cannot be both descendants of each other. If X is a non-descendant of T in G, then by the Markov Condition we know that there is a subset $\boldsymbol{Z}$ of $Parents_G(T) = Parents_G(T) \setminus \{X\}$ (the equality because we assume no edge between T and X) such that $\mathrm{I}(X, T | \boldsymbol{Z})$. Similarly, if T is a non-descendant of X in G then there is $\boldsymbol{Z} \subseteq Parents_G(X) \setminus \{T\}$ such that $\mathrm{I}(X, T | \boldsymbol{Z})$. Conversely, if there is an edge $X \rightarrow T$ or $T \rightarrow X$, then the dependence $\neg \mathrm{I}(X, T | \boldsymbol{Z})$ holds for all $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, T\}$ (by the theorem), thus also holds for all $\boldsymbol{Z} \subseteq Parents_G(T) \setminus \{X\}$ or $\boldsymbol{Z} \subseteq Parents_G(X) \setminus \{T\}$. We just proved that:

Proposition 1: In a faithful BN $< \boldsymbol{V}, G, P >$ there is an edge between the pair of nodes $X \in \boldsymbol{V}$ and $T \in \boldsymbol{V}$ iff $\neg \mathrm{I}(X, T | \boldsymbol{Z})$, for all $\boldsymbol{Z} \subseteq Parents_G(X) \setminus \{T\}$ and $\boldsymbol{Z} \subseteq Parents_G(T) \setminus \{X\}$.

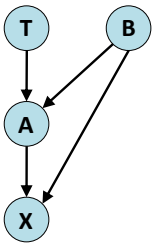


Figure 4.1: $PC(T) = \{A\}, PC(X) = \{A, B\}, X \notin PC(T)$. Notice that, there is no subset of $PC(T)$ that makes T conditionally independent of $X : \neg I(X, T|\emptyset), \neg I(X, T|A)$. However, there is a subset of $PC(X)$ for which X and T become conditionally independent: $I(X, T|\{A, B\})$. The Extended $PC(T)$ (see Definition 16 in this section) is $EPC(T) = \{A, X\}$.

Since the networks in most practical problems are relatively sparse, if we knew the sets $Parents_G(T)$ and $Parents_G(X)$ then the number of subsets that would need to be checked for conditional independence for each $X \in PC(T)$ is significantly smaller: $|2^{|\boldsymbol{V}\setminus\{T,X\}|}| \gg |2^{|Parents_G(X)|}| + |2^{|Parents_G(T)|}|$. Of course, we do not know the sets $Parents_G(T)$ and $Parents_G(X)$ but one could work with any superset of them as shown by the following proposition:

Proposition 2: In a faithful BN $< \boldsymbol{V}, G, P >$ there is an edge between the pair of nodes $X \in \boldsymbol{V}$ and $T \in \boldsymbol{V}$ iff $\neg$I$(X, T|\boldsymbol{Z})$, for all $\boldsymbol{Z} \subseteq \boldsymbol{S}$ and $\boldsymbol{Z} \subseteq \boldsymbol{S'}$, where $Parents_G(X) \setminus \{T\} \subseteq \boldsymbol{S} \subseteq \boldsymbol{V} \setminus \{X, T\}$ and $Parents_G(X) \setminus \{T\} \subseteq \boldsymbol{S'} \subseteq \boldsymbol{V} \setminus \{X, T\}$.

Proof: If there is an edge between the pair of nodes X and T then $\neg$I$(X, T|\boldsymbol{Z})$, for all subsets $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, T\}$ (by Theorem 1) and so $\neg$I$(X, T|\boldsymbol{Z})$ for all $\boldsymbol{Z} \subseteq \boldsymbol{S}$ and $\boldsymbol{Z} \subseteq \boldsymbol{S'}$ too. Conversely, if there is no edge between the pair of nodes X and T, then I$(X, T|\boldsymbol{Z})$, for some $\boldsymbol{Z} \subseteq Parents_G(X) = Parents_G(X) \setminus \{T\} \subseteq \boldsymbol{S}$ or $\boldsymbol{Z} \subseteq Parents_G(T) = Parents_G(T) \setminus \{X\} \subseteq \boldsymbol{S'}$ (by Proposition 1). □

Now, the sets $Parents_G(X)$ and $Parents_G(T)$ depend on the specific network G that we are trying to learn. As we mentioned however, there may be several such statistically equivalent networks among which we cannot differentiate from the data, forming an equivalence class. Thus, it is preferable to work with supersets of $Parents_G(T)$ and $Parents_G(X)$ that do not depend on a specific network member of the class: these supersets are the sets $PC(T)$ and $PC(X)$.

Let us suppose that we have available a superset of $PC(T)$ called $TPC(T)$ (tentative PC). For any node $X \in TPC(T)$ if I$(X, T|\boldsymbol{Z})$ for some $\boldsymbol{Z} \subseteq TPC(T) \setminus \{X, T\}$, then by Proposition 2, we know that X has no edge with T, i.e., $X \notin PC(T)$. So, X should also be removed from $TPC(T)$ to obtain a better approximation of $PC(T)$. If however, $\neg$I$(X, T|\boldsymbol{Z})$ for all $\boldsymbol{Z} \subseteq TPC(T) \setminus \{X, T\}$, then it is still possible that $X \notin PC(T)$ because there may be a set $\boldsymbol{Z} \subseteq PC(X)$ where $\boldsymbol{Z} \nsubseteq PC(T)$ for which I$(X, T|\boldsymbol{Z})$.

Is there actually a case, where X cannot be made independent of T by conditioning on some subset of $PC(T)$? We know that all non-descendants of T can be made independent of T conditioned on a subset of its parents, thus, if there is such an X it has to be a descendant of T. Figure 4.1 shows such a case. These situations are rare in practice as indicated by our empirical results in sections 5 and 6, which implies that by conditioning on all subsets of $TPC(T)$ one will approximate $PC(T)$ quite closely.

Definition 16: We call the Extended $PC(T)$, denoted as $EPC(T)$, the set $PC(T)$ union the set of variables X for which $\neg I(X, T|\boldsymbol{Z})$, for all $\boldsymbol{Z} \subseteq PC(T) \setminus \{X\}$.
The previous results allow us to start building algorithms that operate locally around T in order to find $PC(T)$ efficiently and soundly. Consider first the sketch of the algorithm below:

Algorithm 1:

1. Find a superset $TPC(T)$ of $PC(T)$
2. For each variable $X \in TPC(T)$,
3. If $\exists \boldsymbol{Z} \subseteq TPC(T) \setminus \{X\}$, s.t. $I(X, T|\boldsymbol{Z})$ remove X from $TPC(T)$
4. Return $TPC(T)$

This algorithm will output $TPC(T) \subseteq EPC(T)$. To ensure we end up with the exact $PC(T)$ we can use the following pruning algorithm:

Algorithm 2:

1. For all X in $TPC(T)$ // returned from Algorithm 1
2. If $T \notin TPC(X)$, remove X from $TPC(T)$ // $TPC(X)$ is obtained by running Algorithm 1

In essence, the second algorithm checks for every $X \in TPC(T)$ whether the *symmetrical relation* holds: $T \in TPC(X)$. If the symmetry is broken, we know that $X \notin PC(T)$ since the parents-and-children relation is symmetrical.

What is the complexity of the above algorithms? In Algorithm 1 if step 1 is performed by an Oracle with zero cost, and with $TPC(T)$ equal to $PC(T)$, then the first algorithm requires an order of $O(|\boldsymbol{V}|2^{|PC(T)|})$ tests. The second algorithm will require an order of $O(|\boldsymbol{V}|2^{|PC(X)|})$ tests for each X in $TPC(T)$. Two observations to notice are: (i) the complexity order of the first algorithm depends linearly on the size of the problem $|\boldsymbol{V}|$, exponentially on $|PC(T)|$, which is a structural property of the problem, and how close $TPC(T)$ is to $PC(T)$ and (ii) the second algorithm requires multiple times the time of the first algorithm for minimal returns in quality of learning, i.e., just to take care of the scenario in Figure 4.1 and remove the variables $EPC(T) \setminus PC(T)$ (i.e. X in Figure 4.1).

Since an Oracle is not available the complexity of both algorithms strongly depends on how close approximation of the $PC(T)$ is and how efficiently this approximation is found. The simplest strategy for example is to set $TPC(T) = \boldsymbol{V}$, essentially getting the local version of the algorithm SGS described above. In general any heuristic method that returns a superset of $PC(T)$ is admissible, i.e., it could lead to sound algorithms.

Also notice that in the first algorithm the identification of the members of the $TPC(T)$ (step 1) and the removal of variables from it (step 3) can be interleaved. $TPC(T)$ can grow gradually by one, many variables, or all members of it at a time before it satisfies the requirement that is a superset of $PC(T)$. The requirement for the algorithm to be sound is that, in the end, all tests $I(X, T|\boldsymbol{Z})$ for all subsets $\boldsymbol{Z}$ of $PC(T) \setminus \{X\}$ have been performed.

Given the above, the components of Generalized Local Learning GLL-PC, i.e., an algorithm for $PC(T)$ identification based on the above principles are the following: an *inclusion heuristic function* to prioritize variables for consideration as members of $TPC(T)$ and include them in $TPC(T)$ according to established priority. The second component of the framework is an *elimination strategy*, which eliminates variables from the $TPC(T)$ set. An *interleaving strategy* is the third component and it iterates between inclusion and elimination until a stopping criterion is satisfied. Finally the fourth component is the check that

GLL-PC: High-level pseudocode and main components of Generalized Local Learning - Parents and Children. Returns PC(*T*)

1. $\boldsymbol{U} \leftarrow$ GLL-PC-nonsym(*T*) // first approximate *PC(T)* without symmetry check
2. For all $X \in \boldsymbol{U}$
3. If $T \notin$ GLL-PC-nonsym(*X*) then $\boldsymbol{U} \leftarrow \boldsymbol{U} \setminus \{X\}$ // check for symmetry
4. Return $\boldsymbol{U}$ // true set of parents and children

GLL-PC-nonsym(*T*) // returns a set which is a subset of *EPC*(*T*) and a superset of *PC*(*T*)

1. Initialization
   a. Initialize a set of candidates for the true *PC*(*T*) set: *TPC(T)* $\leftarrow S$, s.t. $S \subseteq \boldsymbol{V} \setminus \{T\}$
   b. Initialize a priority queue of variables to be examined for inclusion in *TPC(T)*: OPEN $\leftarrow \boldsymbol{V} \setminus \{T \cup TPC(T)\}$
2. Apply inclusion heuristic function
   a. Prioritize variables in OPEN for inclusion in *TPC(T)*;
   b. Throw away non-eligible variables from OPEN;
   c. Insert in *TPC(T)* the highest-priority variable(s) in OPEN and remove them from OPEN
3. Apply elimination strategy to remove variables from *TPC(T)*
4. Apply interleaving strategy by repeating steps #2 and #3 until a termination criterion is met
5. Return *TPC(T)*

Figure 4.2: High-level outline and main components (underlined) of GLL-PC algorithm.

the *symmetry requirement* mentioned above is satisfied. See Figure 4.2 for details. The main algorithm calls an internally defined subroutine that induces parents and children of T without symmetry correction (i.e., returns a set which is a subset of $EPC(T)$ and a superset of $PC(T)$). Note that in all references to $TPC(T)$ hereafter, due to generality of the stated algorithms and the process of convergence of $TPC(T)$ to $PC(T)$, $TPC(T)$ stands for just an approximation to $PC(T)$.

Also notice that the term "priority queue" in the schema of Figure 4.2 indicates an abstract data structure that satisfies the requirement that its elements are ranked by some priority function so that the highest-priority element is extracted first. $TPC(T)$ in step 1a of the GLL-PC-nonsym subroutine will typically be instantiated with the empty set when no prior knowledge about membership in $PC(T)$ exists. When the user does have prior knowledge indicating that X is a member of $PC(T)$, $TPC(T)$ can be instantiated to contain X. This prior knowledge may come from domain knowledge, experiments, or may be the result of running GLL-PC on variable X and finding that T is in $PC(X)$ when conducting local-to-global learning (Aliferis et al., 2009; Tsamardinos et al., 2006).

Steps $\#2, 3, 4$ in GLL-PC-nonsym can be instantiated in various ways. Obeying a set of specific rules generates what we call "admissible" instantiations. These admissibility rules are given in Figure 4.3.

Theorem 2: When the following sufficient conditions hold:

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 $\boldsymbol{V}$

any algorithmic instantiation of GLL-PC in compliance with the admissibility rules $\#1-\#3$ above will return the direct causes and direct effects of T. The proof is provided in the Appendix.

We note that the algorithm schema does not address various optimizations and does not address the issue of statistical decisions in finite sample. These will be discussed later. We also note that initialization of $TPC(T)$ in step 1a of the GLL-PC-nonsym function

GLL-PC: Admissibility rules

1. The inclusion heuristic function should respect the following requirement:

   // *Admissibility rule #1*
   All variables $X \in PC(T)$ are eligible for inclusion in the candidate set $TPC(T)$ and each one is assigned a non-zero value by the ranking function. Variables with zero values are discarded and never considered again.

   Note that variables may be re-ranked after each update of the candidate set, or the original ranking may be used throughout the algorithm's operation.

2. The elimination strategy should satisfy the following requirement:

   // *Admissibility rule #2*
   All and only variables that become independent of the target variable T given any subset of the candidate set $TPC(T)$ are discarded and never considered again (whether they are inside or outside $TPC(T)$).

3. The interleaving strategy iterates inclusion and elimination any number of times provided that iterating stops when the following criterion is satisfied:

   //Admissibility rule #3
   At termination no variable outside the set $TPC(T)$ is eligible for inclusion and no variable in the candidate set can be removed at termination.

Figure 4.3: GLL-PC admissibility rules.

Interleaved HITON-PC with symmetry correction
Derived from GLL-PC with following instantiation specifics:

Initialization
$TPC(T) \leftarrow \emptyset$

Inclusion heuristic function
a. Sort in descending order the variables X in OPEN according to their pairwise association with T, i.e., $Assoc(X, T|\emptyset)$.
b. Remove from OPEN variables with zero association with T, *i.e.*, when $I(X, T|\emptyset)$
c. Insert at end of $TPC(T)$ the first variable in OPEN and remove it from OPEN

Elimination strategy
For each $X \in TPC(T)$
If $\exists \mathbf{Z} \subseteq TPC(T) \setminus \{X\}$, s.t. $I(X, T|\mathbf{Z})$ remove X from $TPC(T)$

Interleaving strategy
Repeat
steps #2 and #3 of GLL-PC-nonsym
Until OPEN=$\emptyset$

Figure 4.4: Interleaved HITON-PC with symmetry correction as an instance of GLL-PC.

is arbitrary because correctness (unlike efficiency) of the algorithm is not affected by the initial contents of $TPC(T)$.

We next instantiate the GLL-PC schema to derive two pre-existing algorithms, interleaved HITON-PC with symmetry correction and MMPC with symmetry correction (Tsamardinos et al., 2006; Aliferis et al., 2003a; Tsamardinos et al., 2003b). Figure 4.4 depicts the instantiations needed to obtain interleaved HITON-PC.

The interleaved HITON-PC with symmetry correction algorithm starts with an empty set of candidates, then ranks variables for priority for inclusion in the candidate set by univariate association. It discards variables with zero univariate association. It then accepts each variable into $TPC(T)$. If any variable inside the candidate set becomes independent of

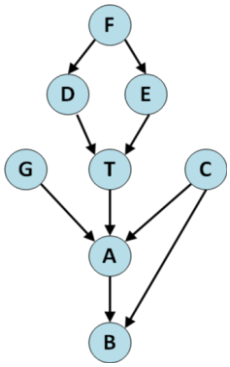


Figure 4.5: Bayesian network used to trace the algorithms.

the response variable T given some subset of the candidate set, then the algorithm removes that variable from the candidate set and never considers it again. In other words, the algorithm attempts to eliminate weakly relevant features from the $TPC(T)$ every time the $TPC(T)$ receives a new member. Iterations of insertion and elimination stop when there are no more variables to examine for inclusion. Once iterating has stopped, the candidate set is filtered using symmetry criterion. Finally, the candidate set is output. Because the admissibility criteria are obeyed, the algorithm is guaranteed to be correct when the assumptions of Theorem 2 hold.

Below we prove that that admissibility rules are obeyed in interleaved HITON-PC with symmetry under the assumptions of Theorem 2:

1. Rule #1 (inclusion) is obeyed because all $PC(T)$ members have non-zero univariate association with T in faithful distributions.
2. Rule #2 (elimination) is directly implemented so it holds.
3. Rule #3 (termination) is obeyed because termination requires empty OPEN and thus eligible variables (i.e., members of $PC(T)$) outside $TPC(T)$ could only be previously discarded from OPEN or $TPC(T)$. Neither case can happen because of admissibility rules $\#1, \#2$ respectively. Similarly all variables in $TPC(T)$ that can be removed are removed because of admissibility rule #2.

A trace of the algorithm is provided below for data coming out of the example BN of the Figure 4.5. We assume that the network is faithful and so the conditional dependencies and independencies can be read off the graph directly using the d-separation criterion. Consider that we want to find parents and children of the target variable T using interleaved HITON-PC with symmetry. Table 4.1 gives a complete trace of step 1 of the instantiated GLL-PC algorithm, i.e. execution of GLL-PC-nonsym subroutine for variable T. The Roman numbers in the table refer to iterations of steps 2 and 3 in GLL-PC-nonsym.

Thus we have $TPC(T) = \{D, E, A, B\}$ by the end of GLL-PC-nonsym subroutine, so $\boldsymbol{U} = \{D, E, A, B\}$ in step 1 of GLL-PC. Next, in steps 2 and 3 we first run GLL-PC-nonsym for all $X \in \boldsymbol{U}$:

- GLL-PC-nonsym$(D) \rightarrow \{T, F\}$
- GLL-PC-nonsym$(E) \rightarrow \{T, F\}$
- GLL-PC-nonsym$(A) \rightarrow \{T, G, C, B\}$
- GLL-PC-nonsym$(B) \rightarrow \{A, C\}$

Step of GLL-PC-nonsym	*Comments*	OPEN	*TPC(T)*
1	Initialize *TPC(T)* and OPEN	*{A, B, C, D, E, F, G}*	∅
2a (I)	Prioritize variables in OPEN for inclusion in *TPC(T)*	*{F, D, E, A, B, G, C}*	∅
2b (I)	Throw away non-eligible members of OPEN (*G* and *C*)	*{F, D, E, A, B}*	∅
2c (I)	Insert in *TPC(T)* the highest-priority variable in OPEN (*F*) and remove it from OPEN	*{D, E, A, B}*	{*F*}
3 (I)	Apply elimination strategy to *TPC(T)*: no effect	*{D, E, A, B}*	*{F}*
2 (II)	Insert the highest-priority variable (*D*) in *TPC(T)* and remove it from OPEN	*{E, A, B}*	*{F, D}*
3 (II)	Apply elimination strategy to *TPC(T)*: no effect	*{E, A, B}*	*{F, D}*
2 (III)	Insert the highest-priority variable (*E*) in *TPC(T)* and remove it from OPEN	*{A, B}*	*{F, D, E}*
3 (III)	Apply elimination strategy to *TPC(T)*: remove *F* since I(*T, F*|*{D,E}*)	*{A, B}*	*{D, E}*
2 (IV)	Insert the highest-priority variable (*A*) in *TPC(T)* and remove it from OPEN	*{B}*	*{D, E, A}*
3 (IV)	Apply elimination strategy to *TPC(T)*: no effect	*{B}*	*{D, E, A}*
2 (V)	Insert the highest-priority variable (*B*) in *TPC(T)* and remove it from OPEN	∅	*{D, E, A, B}*
3 (V)	Apply elimination strategy to *TPC(T)*: no effect	∅	*{D, E, A, B}*
4	Stop interleaving since OPEN = ∅	∅	*{D, E, A, B}*

Table 4.1: Trace of GLL-PC-nonsym(T) during execution of interleaved HITON-PC algorithm.

and then check symmetry requirement. Since $T \notin$ GLL-PC-nonsym(B), the variable B is removed from $\boldsymbol{U}$. Finally, the GLL-PC algorithm returns $\boldsymbol{U} = \{D, E, A\}$ in step 4.

Figure 4.6 shows how algorithm MMPC is obtained from GLL-PC. MMPC is also guaranteed to be sound when the conditions of Theorem 2 hold. Interleaving consists of iterations of just the inclusion heuristic function until OPEN is empty. The heuristic inserts into $TPC(T)$ the next variable F that maximizes the minimum association of variables in OPEN with T given all subsets of $TPC(T)$. In the algorithm, this minimum association of X with T conditioned over all subsets of $\boldsymbol{Z}$ is denoted by $\text{Min}_{\boldsymbol{Z}}\text{Assoc}(X, T|\boldsymbol{Z})$. The intuition is that we accept next the variable that despite our best efforts to be made conditionally independent of T (i.e., conditioned on all subsets of our current estimate $TPC(T)$) is still highly associated with T. The two main differences of the MMPC algorithm from interleaved HITON-PC are the more complicated inclusion heuristic function and the absence of interleaving of the inclusion-exclusion phases before all variables have been processed by the inclusion heuristic function. A set of optimizations and caching operations render the algorithm efficient; for complete details see (Tsamardinos et al., 2006; Tsamardinos et al., 2003b).

Below we prove that admissibility rules are obeyed in MMPC with symmetry under the assumptions of Theorem 2:

1. Rule #1 (inclusion) is obeyed because all $PC(T)$ members have non-zero conditional association with T in faithful distributions.
2. Rule #2 (elimination) is directly implemented so it holds.
3. Rule #3 (termination) is obeyed because termination requires empty OPEN and thus eligible variables (i.e., members of $PC(T)$) outside $TPC(T)$ could only be previously

discarded from OPEN or $TPC(T)$. Neither case can happen because of admissibility rules #1, #2 respectively. Similarly all variables in $TPC(T)$ that can be removed are removed because of admissibility rule #2.

We now introduce a new algorithm, semi-interleaved HITON-PC with symmetry correction, see Figure 4.7. Semi-interleaved HITON-PC operates like interleaved HITON-PC with one major difference: it does not perform a full variable elimination in $TPC(T)$ with each $TPC(T)$ expansion. On the contrary, once a new variable is selected for inclusion, it attempts to eliminate it and if successful it discards it without further attempted eliminations. If it is not eliminated, it is added to the end of the $TPC(T)$ and new candidates for inclusion are sought. Because the admissibility criteria are obeyed the algorithm is guaranteed to be correct under the assumptions of Theorem 2.

MMPC with symmetry correction

Derived from GLL-PC with following instantiation specifics:

Initialization

TPC(T) ← ∅

Inclusion heuristic function

a. Sort in descending order the variables *X* in OPEN according to $\text{Min}_{\mathbf{Z}}\text{Assoc}(X, T|\mathbf{Z})$ for $\mathbf{Z} \subseteq TPC(T)\backslash\{X\}$
b. Remove from OPEN variables *X* with zero association with *T*, given some $\mathbf{Z} \subseteq TPC(T)\backslash\{X\}$
c. Insert at end of *TPC(T)* the first variable in OPEN and remove it from OPEN

Elimination strategy

If OPEN=∅
  For each $X \in TPC(T)$
    If $\exists\, \mathbf{Z} \subseteq TPC(T)\backslash\{X\}$, s.t. $I(X, T|\mathbf{Z})$ remove *X* from *TPC(T)*

Interleaving strategy

Repeat
  steps #2 and #3 of GLL-PC-nonsym
Until OPEN=∅

Figure 4.6: MMPC with symmetry correction as an instance of GLL-PC.

Below we prove that admissibility rules are obeyed in semi-interleaved HITON-PC with symmetry under the assumptions of Theorem 2:

1. Rule #1 (inclusion) is obeyed because all $PC(T)$ members have non-zero univariate association with T in faithful distributions.
2. Rule #2 (elimination) is directly implemented so it holds.
3. Rule #3 (termination) is obeyed because termination requires empty OPEN and thus eligible variables (i.e., members of $PC(T)$) outside $TPC(T)$ could only be previously discarded from OPEN or $TPC(T)$. Neither case can happen because of admissibility rules #1, #2 respectively. Similarly all variables in $TPC(T)$ that can be removed are removed because of admissibility rule #2.

A trace of the algorithm is provided below for data coming out of the example faithful BN of the Figure 4.5. Consider that we want to find parents and children of the target variable T using semi-interleaved HITON-PC with symmetry. Table 4.2 gives a complete trace of step 1 of the instantiated GLL-PC algorithm, i.e. execution of GLL-PC-nonsym subroutine for variable T. The Roman numbers in the table refer to iterations of steps 2 and 3 in GLL-PC-nonsym.

Thus we have $TPC(T) = \{D, E, A, B\}$ by the end of GLL-PC-nonsym subroutine, so $\boldsymbol{U} = \{D, E, A, B\}$ in step 1 of GLL-PC. Next, in steps 2 and 3 we first run GLL-PC-nonsym for all $X \in \boldsymbol{U}$:

Semi-Interleaved HITON-PC with symmetry correction

Derived from GLL-PC with following instantiation specifics:

Initialization

TPC(T) ← ∅

Inclusion heuristic function

a. Sort in descending order the variables X in OPEN according to their pairwise association with T, i.e., *Assoc(X, T|∅)*.
b. Remove from OPEN variables with zero association with T, *i.e.*, when I*(X, T|∅)*
c. Insert at end of *TPC(T)* the first variable in OPEN and remove it from OPEN

Elimination strategy

If OPEN=∅

For each $X \in TPC(T)$

If $\exists \mathbf{Z} \subseteq TPC(T) \setminus \{X\}$, s.t. I($X, T|\mathbf{Z}$) remove X from *TPC(T)*

Else

X ← last variable added to *TPC(T)* // in step 2 of GLL-PC-nonsym

If $\exists \mathbf{Z} \subseteq TPC(T) \setminus \{X\}$, s.t. I($X, T|\mathbf{Z}$) remove X from *TPC(T)*

Interleaving strategy

Repeat

steps #2 and #3 of GLL-PC-nonsym

Until OPEN=∅

Figure 4.7: Semi-interleaved HITON-PC with symmetry correction as an instance of GLL-PC.

Step of GLL-PC-nonsym	*Comments*	OPEN	*TPC(T)*
1	Initialize *TPC(T)* and OPEN	*{A, B, C, D, E, F, G}*	∅
2a (I)	Prioritize variables in OPEN for inclusion in *TPC(T)*	*{F, D, E, A, B, G, C}*	∅
2b (I)	Throw away non-eligible members of OPEN (*G* and *C*)	*{F, D, E, A, B}*	∅
2c (I)	Insert in *TPC(T)* the highest-priority variable in OPEN (*F*) and remove it from OPEN	*{D, E, A, B}*	{*F*}
3 (I)	Apply elimination strategy to *TPC(T)*: no effect	*{D, E, A, B}*	*{F}*
2 (II)	Insert the highest-priority variable (*D*) in *TPC(T)* and remove it from OPEN	*{E, A, B}*	*{F, D}*
3 (II)	Apply elimination strategy to *TPC(T)*: no effect	*{E, A, B}*	*{F, D}*
2 (III)	Insert the highest-priority variable (*E*) in *TPC(T)* and remove it from OPEN	*{A, B}*	*{F, D, E}*
3 (III)	Apply elimination strategy to *TPC(T)*: No effect	*{A, B}*	*{F, D, E}*
2 (IV)	Insert the highest-priority variable (*A*) in *TPC(T)* and remove it from OPEN	*{B}*	*{F, D, E, A}*
3 (IV)	Apply elimination strategy to *TPC(T)*: no effect	*{B}*	*{F, D, E, A}*
2 (V)	Insert the highest-priority variable (*B*) in *TPC(T)* and remove it from OPEN	∅	*{F, D, E, A, B}*
3 (V)	Apply elimination strategy to *TPC(T)*: remove *F* since I(*T, F|{D,E}*)	∅	*{D, E, A, B}*
4	Stop interleaving since OPEN = ∅	∅	*{D, E, A, B}*

Table 4.2: Trace of GLL-PC-nonsym(T) during execution of semi-interleaved HITON-PC algorithm.

- GLL-PC-nonsym(D) $\rightarrow \{T, F\}$
- GLL-PC-nonsym(E) $\rightarrow \{T, F\}$
- GLL-PC-nonsym(A) $\rightarrow \{T, G, C, B\}$
- GLL-PC-nonsym(B) $\rightarrow \{A, C\}$

and then check symmetry requirement. Since $T \in$ GLL-PC-nonsym(B), the variable B is removed from $\boldsymbol{U}$. Finally, the GLL-PC algorithm returns $\boldsymbol{U} = \{D, E, A\}$ in step 4.

4.2 Discovery of the $MB(T)$ set

As mentioned in section 3 the $MB(T)$ contains all information sufficient for the determination of the conditional distribution of $T : P(T|MB(T)) = P(T|\boldsymbol{V} \setminus \{T\})$ and further, it coincides with the parents, children and spouses of T in any network faithful to the distribution (if any) under causal sufficiency. The previous subsection described a general family of algorithms to obtain the $PC(T)$ set, and so in order to find the $MB(T)$ one needs in addition to $PC(T)$, to also identify the spouses of T.

First notice that, approximating $MB(T)$ with $PC(T)$ and missing the spouse nodes may in theory discard very informative nodes. For example, suppose that X and T are two uniformly randomly chosen numbers in $[0, 1]$ and that $Y = \min(1, X + T)$. Then, the only faithful network representing the joint distribution is $X \rightarrow Y \leftarrow T$, where X is the spouse of T. In predicting T, the spouse node X may reduce the uncertainty completely: conditioned on Y, T may become completely determined (when both X and T are less than 0.5). Thus, it theoretically makes sense to develop algorithms that identify the spouses in addition to the $PC(T)$, even though later in section 5 we empirically determine that within the scope of distributions and problems tried, the $PC(T)$ resulted in feature subsets almost as predictive as the full $MB(T)$. In the companion paper (Aliferis et al., 2009) we also provide possible reasons explaining the good performance of $PC(T)$ versus $MB(T)$ for classification in practical tasks.

The theorem on which the algorithms in this family are based to discover the $MB(T)$ is the following:

Theorem 3: In a faithful BN $< \boldsymbol{V}, G, P >$, if for a triple of nodes X, T, Y in $G, X \in PC(Y)$, $Y \in PC(T)$, and $X \notin PC(T)$, then $X \rightarrow Y \leftarrow T$ is a subgraph of G iff $\neg I(X, T|\boldsymbol{Z} \cup \{Y\})$, for all $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, T\}$ (Spirtes et al., 2000).

We distinguish two cases: (i) X is a spouse of T but it is also a parent or child, e.g., $X \rightarrow T \rightarrow Y$ and also $X \rightarrow Y$. In this case, we cannot use the theorem above to identify Y as a collider and X as a spouse. But at the same time we do not have to: $X \in PC(T)$ and so it will be identified by GLL-PC. (ii) $X \in MB(T) \setminus PC(T)$ in which case we can use the theorem to locally discover the subgraph $X \rightarrow Y \leftarrow T$ and determine that X should be included in $MB(T)$.

We now introduce the GLL-MB in Figure 4.8. The admissibility requirement is simply to use an admissible GLL-PC instantiation.

For the identification of $PC(T)$ any method of GLL-PC can be used. Also, in step 5a we know such a $\boldsymbol{Z}$ exist since $X \notin PC(T)$ (by Theorem 1); this $\boldsymbol{Z}$ has been previously determined and is cached during the call to GLL-PC.

Theorem 4: When the following sufficient conditions hold

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;

GLL-MB: Generalized Local Learning - Markov Blanket

1. *PC(T)* ← GLL-PC*(T)* // obtain *PC*(*T*) by running *GLL-PC* for variable *T*
2. For every variable $Y \in PC(T)$
   PC(*Y*) ← GLL-PC(*Y*) // obtain *PC*(*Y*) for every member *Y* of *PC*(*T*)
3. *TMB(T)* ← *PC(T)* // initialize *TMB(T)* with *PC*(*T*) members
4. $\boldsymbol{S} \leftarrow \{\cup_{Y \in PC(T)} PC(Y)\} \setminus \{PC(T) \cup \{T\}\}$ // these are the potential spouses
5. For every variable $X \in \boldsymbol{S}$
   a. Retrieve a subset $\boldsymbol{Z}$ s.t. I(*X*, *T* | $\boldsymbol{Z}$) // subset was identified and stored in steps 1 and 2
   b. For every variable $Y \in PC(T)$ *s.t.* $X \in PC(Y)$ // *Y* is a potential common child of *T* and *X*
   c. If $\neg$I(*X*, *T*| $\boldsymbol{Z} \cup \{Y\}$) // *X* is a spouse
   d. Insert *X* into *TMB(T)*
6. Optionally: Eliminate from *TMB(T)* predictively redundant members using a backward wrapper approach.
7. Return *TMB*(*T*)

Figure 4.8: GLL-MB: Generalized Local Learning - Markov Blanket algorithm

c. Causal sufficiency in $\boldsymbol{V}$

any algorithmic instantiation of GLL-MB in compliance with the admissibility rule will return $MB(T)$ (with no need for step 6). The proof is provided in the Appendix.

A new Markov blanket algorithm, semi-interleaved HITON-MB, can be obtained by instantiating GLL-MB (Figure 4.8) with the semi-interleaved HITON-PC algorithm with symmetry correction for GLL-PC.

Semi-interleaved HITON-MB is guaranteed to be correct under the assumptions of Theorem 4, hence the only proof of correctness needed is the proof of correctness for semi-interleaved HITON-PC with symmetry (which was provided earlier).

A trace of the semi-interleaved HITON-MB algorithm for data coming out of the example faithful BN of the Figure 4.5 follows below. Please refer to Figure 4.8 for step numbers. Consider that we want to find Markov blanket of T. In step 1, we find $PC(T) = \{D, E, A\}$. Then in step 2 we find $PC(X)$ for all $X \in PC(T)$:

- $PC(D) = \{T, F\}$,
- $PC(E) = \{T, F\}$,
- $PC(A) = \{T, G, C, B\}$,

In step 3 we initialize $TMB(T) \leftarrow \{D, E, A\}$. The set $\boldsymbol{S}$ in step 4 contains the following variables: $\{F, G, C, B\}$. In step 5 we loop over all members of $\boldsymbol{S}$ to find spouses of T. Let us consider each variable separately:

- Loop for $X = F$: In step 5a we retrieve a subset $\boldsymbol{Z} = \{D, E\}$ that renders $X = F$ independent of T. In step 5b we loop over all potential common children of F and T, i.e. $Y = D$ and $Y = E$. When we consider $Y = D$, we find that $X = F$ is independent of T given $\boldsymbol{Z} \cup \{Y\} = \{D, E\}$ and thus do not include F in $TMB(T)$ in step 5d. When we consider $Y = E$, we also do not include F in $TMB(T)$ in step 5d.
- Loop for $X = G$: In step 5a we retrieve a subset $\boldsymbol{Z} = \emptyset$ that renders $X = G$ independent of T. In step 5b we loop over all potential common children of G and T, i.e. variable $Y = A$. We find that $X = G$ is dependent on T given $\boldsymbol{Z} \cup \{Y\} = \{A\}$ and thus include G in $TMB(T)$ in step 5d.
- Loop for $X = C$: In step 5a we retrieve a subset $\boldsymbol{Z} = \emptyset$ that renders $X = C$ independent of T. In step 5b we loop over all potential common children of C and T, i.e. variable $Y = A$. We find that $X = C$ is dependent on T given $\boldsymbol{Z} \cup \{Y\} = \{A\}$ and thus include C in $TMB(T)$ in step 5d.

- Loop for $X = B$: In step 5a we retrieve a subset $\boldsymbol{Z} = \{A, C\}$ that renders $X = B$ independent of T. In step 5b we loop over all potential common children of B and T, i.e. variable $Y = A$. We find that $X = B$ is independent of T given $\boldsymbol{Z} \cup \{Y\} = \{A, C\}$ and thus do not include G in $TMB(T)$ in step 5d.

By the end of step 5, we have $TMB(T) = \{D, E, A, G, C\}$. Notice that it is the true $MB(T)$. In step 6 we perform wrapping to remove members of $TMB(T)$ that are redundant for classification. Let us assume that we used a backward wrapping procedure that led to removal of variable G, for example because omitting this variable does not increase classification loss. Thus, we have $TMB(T) = \{D, E, A, C\}$ in step 7 when the algorithm terminates.

The above algorithm specifications and proofs demonstrate that it is relatively straightforward to derive correct algorithms and prove their correctness using the GLL framework. It is also straightforward to derive relaxed versions (for example non-symmetry corrected versions of interleaved and semi-interleaved HITON and MMPC) which trade-off correctness for improved tractability.

4.3 Computational complexity

The complexity of all algorithms presented depends on the time for the tests of independence and measures of associations. For the G^2 test of independence for discrete variables, for example, we use in reported experiments an implementation linear to the sample size and exponential to the number of variables in the conditional set. However, because the latter number is small in practice, tests are relatively efficient. Faster implementations exist that only take time $n \log(n)$ to the number n of training instances, independent of the size of the conditioning set. Also, advanced data structures (Moore and Wong, 2003) can be employed to improve the time complexity (see (Tsamardinos et al., 2006) for details on the implementation of the tests). In reported experiments we also implement the measure of association $\text{Assoc}(X, T|\boldsymbol{Z})$ to be the negative *p-value* returned by the test $I(X, T|\boldsymbol{Z})$ and so it takes exactly the same time to compute as a test of independence. In the following discussion, we consider the complexity of the algorithms in terms of the number of tests and measures of association they perform.

The number of tests of the GLL-PC algorithm in Figure 4.2 depends on several factors. These are the inclusion heuristic efficiency in approximating the $PC(T)$, the time required by the inclusion heuristic, and the size of the $PC(T)$ which is a structural property of the problem to solve. Interleaved-HITON-PC (algorithm in Figure 4.4) for example, will sort the variables using $|\boldsymbol{V}|$ measures of associations. Subsequently, it will perform a test $I(X, T|\boldsymbol{Z})$ for all subsets of the largest $TPC(T)$ in any iteration of interleaving of the inclusion-exclusion steps. With appropriate caching a test will never have to be repeated. Thus, assuming the largest size of the $TPC(T)$ is in the order of the $PC(T)$, the complexity of the GLL-PC-nonsym subroutine is $O(|\boldsymbol{V}|2^{|PC(T)|})$. In step 3, it will execute the GLL-PC-nonsym subroutine again for all $X \in TPC(T)$. Assuming each neighborhood of X is about the same as the $PC(T)$, when checking the symmetry condition, the algorithm will perform another $O(|\boldsymbol{V}||PC(T)|2^{|PC(T)|})$tests.

To identify $MB(T)$ by the GLL-MB algorithm in Figure 4.8 we first need to initialize subset $\boldsymbol{S}$. Assuming all neighborhoods are about the same size (i.e. equal to $|PC(T)|$), the total complexity to find the set $\boldsymbol{S}$ is $O(|\boldsymbol{V}||PC(T)|^2 2^{|PC(T)|})$ since we call GLL-PC

for each member of the $PC(T)$. In fact, several optimizations can reduce this order to $O(|\boldsymbol{V}||PC(T)|2^{|PC(T)|})$ but we will not elaborate further in this paper. In step 5, in the worst case we perform a single test for each node in $\boldsymbol{S}$ and each node in $PC(T)$ for a total of at most $O(|PC(T)|^2)$ tests (the subset $\boldsymbol{Z}$ in step 5a is cached and retrieved). So the order of the algorithm is $O(|\boldsymbol{V}||PC(T)|^2 2^{|PC(T)|})$ tests given the structural assumptions above.

All other algorithmic instantiations of the template in this section have similar complexity.

At this point it is worth noting a number of polynomial approximation algorithms in the literature that increase efficiency without sacrificing quality to a large degree. The identification of a subset $\boldsymbol{Z}$ in step 3 of the GLL-PC-nonsym subroutine as described in algorithm instantiations of GLL-PC is a step exponential to the size of the $TPC(T)$; however, one could attempt to discover it in a greedy fashion, for example by starting with the empty set and adding to $\boldsymbol{Z}$ the variable decreasing the association with T the most. These ideas started with the TPDA algorithm (Cheng et al., 2002a) and were further explored in (Brown et al., 2005). Similar improvements can be applicable to inclusion strategy.

For the above analysis we assumed that all tests $I(X, T|\boldsymbol{Z})$ can or should be performed and return the correct results. However, in the next sub-section we discuss how the statistical decisions of independence or dependence are made; these decisions severely affect the complexity of the algorithms as well.

4.4 Dealing with statistical decisions

The quality of the algorithms in practice highly depends on their ability to statistically determine whether $I(X, T|\boldsymbol{Z})$ or $\neg I(X, T|\boldsymbol{Z})$ (equivalently whether $\text{Assoc}(X, T|\boldsymbol{Z})$ is zero or non-zero) for a pair of variables X and T and a set of variables Z. The test $I(X, T|\boldsymbol{Z})$ is implemented as a statistical hypothesis test with null hypothesis H_0: X and T are independent given $\boldsymbol{Z}$. A *p-value* corresponding to this test statistic's distribution expresses the probability of seeing the same or more extreme (i.e., indicative of dependence) test statistic values when sampling from distributions where H_0 is true. If the *p-value* is lower than a given threshold (i.e., significance level "alpha") α, then we consider the independence hypothesis to be improbable and reject it. Thus, for a sufficiently low *p-value* we accept *dependence*. If however, the *p-value* is not low enough to provide confidence in rejecting H_0 then there are two possibilities:

a) H_0 actually holds, i.e., the variables are indeed conditionally independent.
b) H_0 does not hold, the variables are conditionally dependent but we cannot confidently reject H_0.

The reasons for b) are that either the dependence is weak relatively to the available sample to be detected (in order words, we have low probability to reject the null hypothesis H_0 when it does not hold, i.e. low statistical power), or we are using the wrong statistical test for this type of dependency. In essence, we would like to distinguish between the following cases:

a) $I(X, T|\boldsymbol{Z})$ holds with high-probability
b) $\neg I(X, T|\boldsymbol{Z})$ holds with high-probability
c) Undetermined case given the available sample

To deal with case c) in our implementations we take the following approach, introduced by (Spirtes et al., 2000): we consider that we are facing case c) if there is no sufficient

power according to a *reliability criterion*. In our implementations this criterion depends on parameter *h-ps*. The criterion dictates that if and only if we have at least *h-ps* sample instances per number of cells (i.e., number of parameters to be estimated) in the contingency tables for the discrete statistical tests then the test is reliable.

Once a test is deemed unreliable an algorithm needs to decide how to handle the corresponding statistical decision. For example, the PC algorithm for global causal discovery (Spirtes et al., 2000) considers that given no other evidence, all variables are dependent with each other. That is, a pair of variables is always connected by an edge in the graph unless a subset $\boldsymbol{Z}$ is discovered that renders them conditionally independent.

The implementations of GLL instantiations in the present paper do not perform an unreliable test either. However, ignoring unreliable tests with 0-order conditioning test (i.e., univariate tests) is equivalent to assuming $I(X, T|\boldsymbol{Z})$ whereas ignoring unreliable tests with higher-order conditioning test (i.e., conditioning sets with 1 or more conditioning variables) is equivalent to assuming $\neg I(X, T|\boldsymbol{Z})$ as far as this unreliable test is concerned (because the final judgment on independence, is deferred to reliable, typically lower-order tests). Thus, given no evidence of dependence, we assume the unreliable tests to return $I(X, T|\boldsymbol{Z})$. The different treatment of the PC implementation leads to problems as discussed in (Tsamardinos et al., 2006) pointing to the importance of this implementation aspect of the algorithms.

Another practical implementation issue arises when prior knowledge, experiments, or domain substantive knowledge ensures that a variable X is in $PC(T)$ or that X is not in $PC(T)$. In such cases the algorithm can be modified to "lock" X inside or outside $TPC(T)$ respectively in order to avoid the possibility that errors in statistical decisions will counter previously validated knowledge and possibly propagate more statistical decision errors.

In addition to *h-ps*, a second restriction on the conditioning set size is provided by parameter *max-k*. This parameter places an absolute limit on the number of elements in a conditioning set size, *without reference to available sample size*. As such *max-k* participates in the reliability judgment but also restricts the computational complexity of the algorithms by trading off computational complexity for fit to data.

Specifically first consider that more variables than the actual $PC(T)$ could be output by the algorithm. A variable X that becomes independent of T only when we condition on $\boldsymbol{Z}$, with $|\boldsymbol{Z}| > max\text{-}k$ could enter the $TPC(T)$ and will not be removed afterwards. For example, if $max\text{-}k = 1$, then variable F in Figure 4.5 cannot be d-separated from T given any $\boldsymbol{Z}$ with $|\boldsymbol{Z}| \leq 1$. Thus, the reliability criterion may increase the number of tests performed, since these depend on the size of the $TPC(T)$. On the other hand, the criterion forces certain tests not to be performed, specifically those whose conditioning set $\boldsymbol{Z}$ size is larger than *max-k*. Thus, since only $\binom{TPC(T)}{max-k}$ subsets are tested out of all possible $2^{|TPC(T)|}$ ones, the complexity of the algorithm GLL-PC-nonsym now becomes $O(|\boldsymbol{V}||TPC(T)|^{max-k})$, i.e., polynomial of order *max-k*.

The parameters *h-ps* and *max-k* are user-specified or, alternatively, optimized automatically by cross-validation, or optimized for a whole domain. The role and importance of these two parameters, especially with respect to quality of statistical decisions, is explored in detail in the companion paper (Aliferis et al., 2009). Finally, because the quality of statistical decisions is not addressed in the proofs of correctness provided earlier, it was im-

plicitly assumed that whenever sufficient sample size is provided to the algorithms statistical decisions are reliable.

A recent treatment that specifically addresses the role of statistical decisions in finite sample is presented in (Tsamardinos and Brown, 2008a). In this work, a bound of the p-value of the existence of an edge is provided; the bound can be used to control the False Discovery Rate of the identification of the $PC(T)$ or all the edges in a network.

5. Comparative evaluation of local causal and non-causal feature selection algorithms in terms of feature selection parsimony and classification accuracy

In the present section we examine the ability of GLL algorithms to discover compact sets of features with as high classification performance as possible for each dataset and compare them with other local causal structure discovery methods as well as non-causal feature selection methods.

In order to avoid bias in error estimation we apply nested N-fold cross-validation. The inner loop is used to try different parameters for the feature selection and classifier methods while the outer loop tests the best configuration on an independent test set. Details are given in (Statnikov et al., 2005b; Dudoit and van der Laan, 2003; Scheffer, 1999).

All experiments discussed in this section and elsewhere in this paper were conducted on ACCRE (Advanced Computing Center for Research and Education) High Performance Computing system at Vanderbilt University. The ACCRE system consists of 924 x86 processors (the majority of which 2 GHz) and 668 PowerPC processors (2.2 GHz) running 32 and 64-bit Linux OS. The overall computational capacity of the cluster is approximately 6 TFLOPS. For preliminary and exploratory experiments we utilized a smaller cluster of eight 3.2 GHz x86 processors.

The evaluated algorithms are listed in the Appendix Table A.1. They were chosen on the basis of prior independently published results showing their state-of-the-art performance and applicability to the range of domains represented in the evaluation datasets. We compare several versions of GLL, including parents and children (PC) and Markov blanket (MB) inducers. Whenever we refer to HITON-PC algorithm in this paper, we mean semi-interleaved HITON-PC without symmetry correction, unless mentioned otherwise. Also, other GLL algorithms evaluated do not have symmetry correction unless mentioned otherwise. Finally, unless otherwise noted, GLL-MB does not implement a wrapping step.

Table A.2 in the Appendix presents the evaluation datasets. The datasets were chosen on the basis of being representative of a wide range of problem domains (biology, medicine, economics, ecology, digit recognition, text categorization, and computational biology) in which feature selection is essential. These datasets are challenging since they have a large number of features with small-to-large sample sizes. Several datasets used in prior feature selection and classification challenges were included. All datasets have a single binary target variable.

To perform imputation in datasets with missing values, we applied a non-parametric nearest neighbor method (Batista and Monard, 2003). Specifically, this method imputes each missing value of a variable with the present value of the same variable in the most similar instance according to Euclidian distance metric. Discretization in non-sparse con-

tinuous datasets was performed by a univariate method (Liu et al., 2002) implemented in *Causal Explorer* (Aliferis et al., 2003b). For a given continuous variable, the method considers many binary and ternary discretization thresholds (by means of a sliding window) and chooses the one that maximizes statistical association with the target variable. In sparse continuous datasets, discretization was performed by assigning value 1 to all non-zero values. All variables in each dataset were also normalized to be in [0, 1] range to facilitate classification by SVM and KNN. All computations of statistics for the preprocessing steps were performed based on training data only to ensure unbiased classification error estimation. Statistical comparison between algorithms was done using two-sided permutation test (with 10,000 permutations) at 5% alpha level (Good, 2000). The null hypothesis of this test is that algorithms perform the same.

Both polynomial SVMs and KNN were used for building classifiers from each selected feature set. In complementary experiments, the *native* classifier for each one of several feature selection methods (LARS-EN, L0, and RFVS) was used and its performance was compared against classifiers induced by SVMs and KNN. For SVMs, the misclassification cost C and kernel degree d were optimized over values [1, 10, 100] and [1, 2, 3, 4], respectively. For KNN, the number of nearest neighbors k was optimized over values [1,...,min(1000, number of instances in the training set)]. All optimization was conducted in nested cross-validation using training data only, while the testing data was used only once to obtain an error estimate for the final classifier. We used the libSVM implementation of SVM classifiers (Fan et al., 2005) and our own implementation of KNN.

We note that use of SVMs and KNN does not imply that GLL methods are designed to be filters for these two algorithms only, or that the algorithm comparison results narrowly apply to these two classifiers. Rather as explained in section 2.2, GLL algorithms provide performance guarantees as long as the classifier used has universal approximator properties. SVMs and KNN are two exemplars of practical and scalable such methods in wide use. We also emphasize that selecting features with a wrapper or embedded feature selection method that *is not* SVM or KNN specific is not affected by the inductive bias mismatch because such mismatch is affecting performance only when the classifier used is "handicapped" relative to the native classifier (Tsamardinos and Aliferis, 2003; Kohavi and John, 1997). We provide experimental data substantiating this point in the Appendix Table A.3 (and Table S1 in the online supplement) where we compare classification performance of RFVS, LARS-EN, and L0 with features selected by each corresponding method to the classification performance of SVMs and KNN using the same features. It is shown that SVM predictivity matches, whereas KNN predictivity compares favorably, with the classifiers that are native to each feature selector. On the other hand, the choice of SVMs and KNN provides several advantages to the research design of the evaluation: (a) the same classifiers can be used with all datasets removing a possible confounder in the evaluation; (b) they can be used without feature selection (i.e., full variable set) to give a reference point of predictivity under no feature selection (that in practice is as good as empirically optimal predictivity especially when using SVMs); (c) they can be used when sample size is smaller than number of variables; (d) prior evidence suggests that they are suitable classifiers for the domains; (e) they can be executed in tractable time using nested cross-validation as required by our protocol.

In all cases when an algorithm had not terminated within 2 days of single-CPU time per run on a single training set (including optimization of the feature selector parameters) and in order to make the experimental comparison feasible with all methods and datasets in the study, we deemed it to be impractical and terminated it. While the practicality of spending more than two days of single-CPU time on a single training set can be debated, we believe that use of slower algorithms in practice is problematic due to the following reasons: (i) in the context of N-fold cross-validation the total running time is at least N times longer (i.e., >20 days single-CPU time); (ii) the analyst does not know whether the algorithm will terminate within a reasonable amount of time, and (iii) when quantification of uncertainty about various parameters (e.g., estimating variance in error estimates via bootstrapping) is needed the analysis becomes prohibitive regardless of analyst flexibility and computational resources. When comparing a pair of algorithms we consider only the datasets where both algorithms terminate within the allotted time.

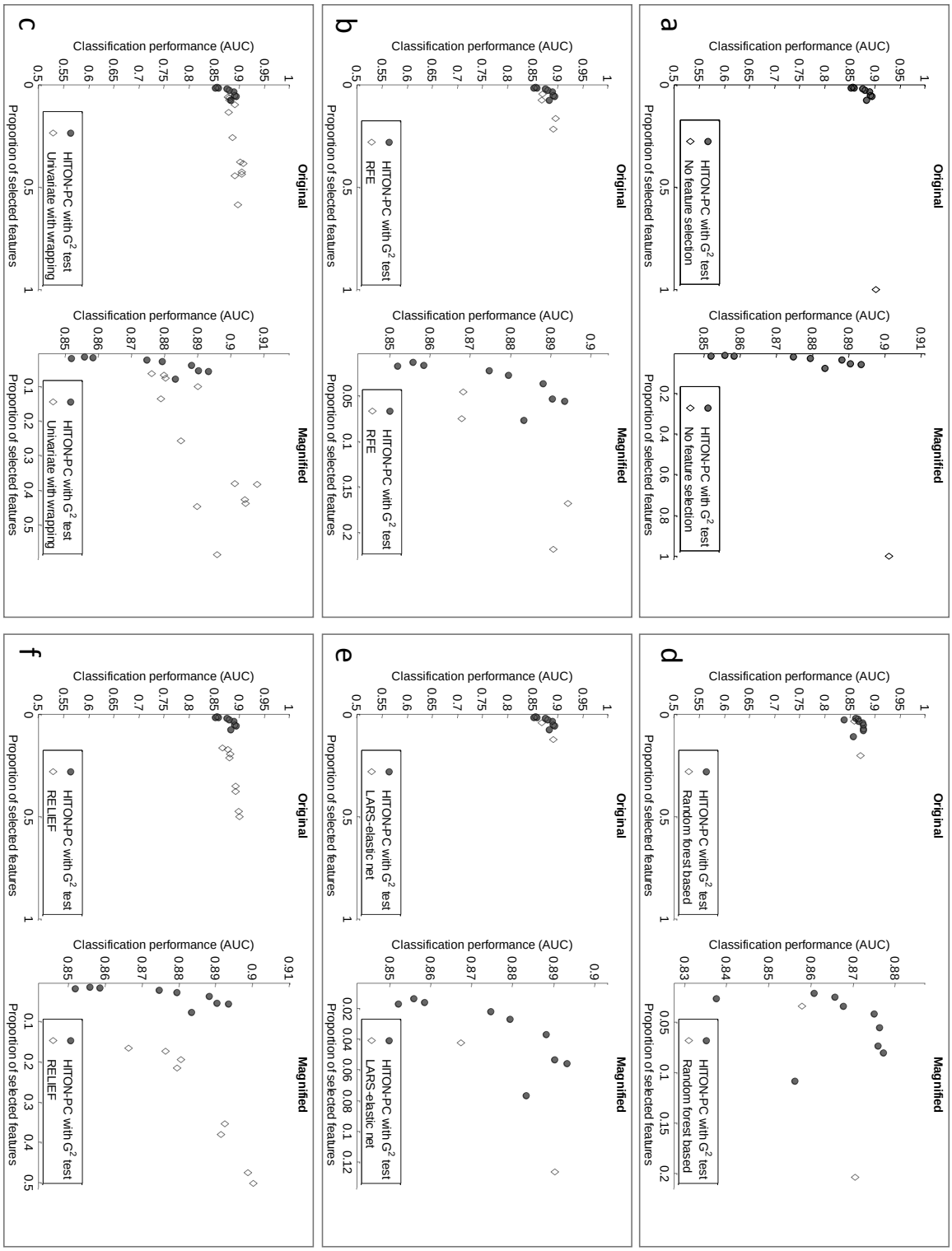
We evaluate the algorithms using the following metrics:

1. Number of features selected;
2. Proportion of features selected relative to the original number of features (i.e. prior to feature selection);
3. Classification performance measured as area under ROC curve (AUC) (Fawcett, 2003);
4. Feature selection time in minutes[2].

5.1 Causal feature selection returns more compact feature sets than non-causal feature selection

Figure 5.1 compares each evaluated algorithm to semi-interleaved HITON-PC with G^2 test as a reference performance for GLL, in the two-dimensional space defined by proportion of selected features and classification performance by SVM (results for KNN are similar and are available in Table S5 in the online supplement). As can be seen in the figure (and also in Figure S1 of the online supplement), GLL algorithms typically return much more compact sets than other methods. More compact results are provided by versions that induce the PC set rather than the MB for obvious reasons. Out of GLL methods, the most compact sets are returned when the Z-test is applicable (continuous data) compared to G^2 test (discrete or discretized data). As seen in Tables S2-S3 in the online supplement, depending on the parameterization of GLL, compactness varies. However, regardless of configuration, both GLL and other local causal methods (i.e., IAMB, BLCD-MB, FAST-IAMB, K2MB) with the exception of Koller-Sahami are typically more compact than non-causal feature selection methods (i.e., univariate methods with backward wrapping, RFE, RELIEF, Random Forest-based Variable Selection, L0, and LARS-EN). Forward stepwise selection and some configurations of LARS-EN, Random Forest-based Variable Selection, and RFE are often very parsimonious, however their parsimony varies greatly across datasets. Notice that whenever an algorithm variant employed statistical comparison among feature sets (in particular non-causal ones), it improved compactness (Figure S1 and Tables S2-S3 in the

2. In all cases we used the implementations provided by the authors of methods, or state-of-the-art implementations, and thus reported time should be considered representative of what practitioners can expect in real-life with equipment and data similar to the ones used in the present study. However, we note that running times should be interpreted as indicative only since numerous implementation details and possible optimizations as well as computer platform discrepancies can affect results.


a
b
c
d
e
f
Original
Magnified
Classification performance (AUC)
Proportion of selected features
HITON-PC with G2 test
No feature selection
RFE
Univariate with wrapping
Random forest based
LARS-elastic net
RELIEF

g
Original
Magnified
HITON-PC with G2 test
L0-norm
Classification performance (AUC)
Proportion of selected features
h
HITON-PC with G2 test
Forward stepwise selection
HITON-PC with G2 test
Fonward step. selection
i
HITON-PC with G2 test
Koller-Sahami
j
HITON-PC with G2 test
IAMB and variants
k
HITON-PC with G2 test
K2MB and BLCD-MB
l
HITON-PC with G2 test
HITON-PC with Z-test

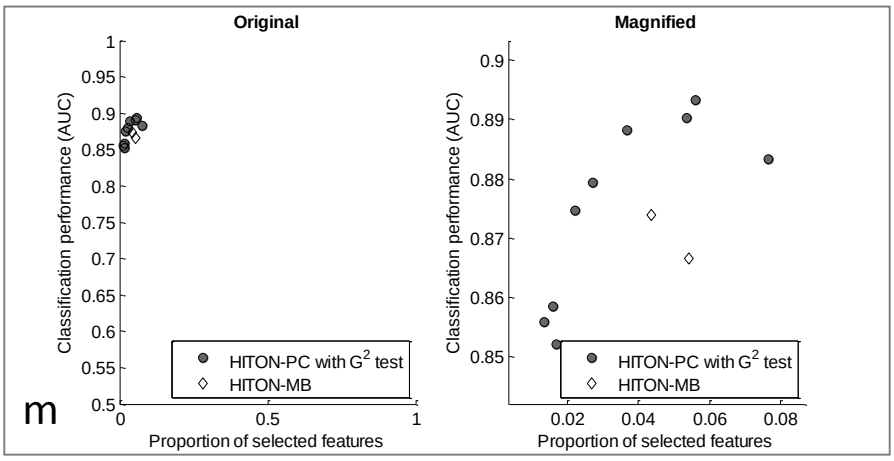


Figure 5.1: (**continued from previous two pages**): Comparison of each algorithmic family with semi-interleaved HITON-PC with G^2 test. HITON-PC is executed with 9 different configurations: $\{max\text{-}k = 1,\ \alpha = 0.05\}$, $\{max\text{-}k = 2, \alpha = 0.05\}$, $\{max\text{-}k = 3, \alpha = 0.05\}$, $\{max\text{-}k = 4, \alpha = 0.05\}$, $\{max\text{-}k = 1, \alpha = 0.01\}$, $\{max\text{-}k = 2, \alpha = 0.01\}$, $\{max\text{-}k = 3, \alpha = 0.01\}$, $\{max\text{-}k = 4, \alpha = 0.01\}$, and a configuration that selects one of the above parameterizations by nested cross-validation. Results shown are averaged across all real datasets where both HITON-PC with G^2 test and an algorithmic family under consideration are applicable and terminate within 2 days of single-CPU time per run on a single training set. Multiple points for each algorithm correspond to different parameterizations/configurations. See Appendix Table A.1 for detailed list of algorithms. The left graph has x-axis (proportion of selected features) ranging from 0 to 1 and y-axis (classification performance AUC) ranging from 0.5 to 1. The right graph has the same data, but the axes are magnified to see the details better.

online supplement). Table 5.1 gives statistical comparisons of compactness between one reference GLL algorithm (semi-interleaved HITON-PC with G^2 test and cross-validation-based optimization of the algorithm parameters) and 43 non-GLL algorithms and variants (including no feature selection). In 21 cases the GLL reference method gives statistically significantly more compact sets compared to all other methods, in 16 cases parsimony is not statistically distinguishable, and in 6 cases HITON-PC gives less compact feature sets. These 6 cases correspond strictly to non-GLL causal feature selection algorithms and at the expense of severe predictive suboptimality (0.06 to 0.10 AUC) relative to the reference GLL method (see Tables S4-S5 in the online supplement).

5.2 Compactness versus classification performance

Compactness is only one of the two requirements for solving the feature selection problem. A maximally compact algorithm that does not achieve optimal predictivity does not solve the feature selection problem. Figure 5.1 examines the trade-off of compactness and SVM predictivity (results for KNN are similar and available in Table S5 in the online supplement). The best possible point for each graph is at the upper left corner. For ease of visualization the results are plotted for each algorithmic family separately. To avoid overfitting and to examine robustness of various methods to parameterization we did not select the best performing configuration, but plotted all of them. Notice that some algorithms did not run on all 13 real datasets (i.e., algorithms with Fisher's Z-test are applicable only to

	Predicitivity		*Reduction*	
Feature selection method	P-value	Nominal winner	P-value	Nominal winner
No feature selection	0.1890	Other	**<0.0001**	HITON-PC
RFE: 4 variants	0.9754	Other	**0.0046**	HITON-PC
	0.8030	Other	**0.0042**	HITON-PC
	0.1312	HITON-PC	0.3634	HITON-PC
	0.1008	HITON-PC	0.6816	Other
UAF-KruskalWallis-SVM: 4 variants	0.2248	Other	**0.0028**	HITON-PC
	0.0098	Other	**0.0004**	HITON-PC
	1.0000	HITON-PC	0.1414	HITON-PC
	0.3232	HITON-PC	0.3998	HITON-PC
UAF-Signal2Noise-SVM: 4 variants	0.0710	Other	**0.0018**	HITON-PC
	0.0752	Other	**0.0030**	HITON-PC
	0.4420	HITON-PC	0.7850	HITON-PC
	0.2820	HITON-PC	0.6604	HITON-PC
UAF-Neal-SVM: 4 variants	0.5046	Other	**<0.0001**	HITON-PC
	0.9782	HITON-PC	**<0.0001**	HITON-PC
	0.6980	HITON-PC	**0.0044**	HITON-PC
	0.3806	HITON-PC	**0.0186**	HITON-PC
Random Forest Variable Selection: 2 variants	0.6064	HITON-PC	0.3252	HITON-PC
	0.5050	HITON-PC	0.1338	Other
LARS-Elastic Net: 2 variants	1.0000	Other	0.1112	HITON-PC
	0.0832	HITON-PC	0.5216	Other
RELIEF: 8 variants	0.2032	Other	**<0.0001**	HITON-PC
	0.9362	Other	**<0.0001**	HITON-PC
	0.4388	Other	**0.0014**	HITON-PC
	0.8432	Other	**0.0010**	HITON-PC
	0.4290	HITON-PC	**0.0108**	HITON-PC
	0.3114	HITON-PC	0.0518	HITON-PC
	0.4424	HITON-PC	0.0706	HITON-PC
	0.2748	HITON-PC	**0.0404**	HITON-PC
L0-norm	**0.0258**	HITON-PC	0.1942	HITON-PC
Forward Stepwise Selection	**0.0028**	HITON-PC	0.2758	Other
Koller-Sahami: 6 variants	0.7506	HITON-PC	**<0.0001**	HITON-PC
	0.6234	HITON-PC	**<0.0001**	HITON-PC
	0.6278	HITON-PC	**<0.0001**	HITON-PC
	<0.0001	HITON-PC	**<0.0001**	Other
	0.1278	HITON-PC	0.3856	HITON-PC
	0.1236	HITON-PC	**<0.0001**	HITON-PC
IAMB: 3 variants	**<0.0001**	HITON-PC	**<0.0001**	Other
	<0.0001	HITON-PC	**<0.0001**	Other
	<0.0001	HITON-PC	0.1202	Other
K2MB	**<0.0001**	HITON-PC	**<0.0001**	Other
BLCD-MB	**<0.0001**	HITON-PC	**<0.0001**	Other
FAST-IAMB	**<0.0001**	HITON-PC	**<0.0001**	Other

Table 5.1: Statistical comparison via permutation test (Good, 2000) of 43 non-GLL algorithms (including no feature selection) to the reference GLL algorithm (semi-interleaved HITON-PC with G^2 test and cross-validation-based optimization of the algorithm parameters by SVM classifier) in terms of SVM predictivity and parsimony. Each non-GLL algorithm compared to HITON-PC in each row is denoted by "Other". Bolded p-values are statistically significant at 5% alpha.

continuous data, while some algorithms did not terminate within 2 days of single-CPU time per run on a single training set). For such cases, we plotted results only for datasets where the algorithms were applicable and the results for HITON-PC correspond to the same datasets. As can be seen, GLL algorithms that induce PC sets dominate both other causal and non-causal feature selection algorithms. This is also substantiated in Table 5.1 (and Table S7 in the online supplement that provides results for KNN classifier) that gives statistical comparisons of predictivity between the reference GLL algorithm and all 43 non-GLL algorithms and variants (including no feature selection). In 9 cases the GLL reference method gives statistically significantly more predictive sets compared to all other methods, in 33 cases predictivity is not statistically distinguishable, and in 1 case GLL gives less predictive feature sets (however the magnitude of the GLL suboptimal predictivity is only 0.018 AUC on average, whereas the difference in compactness is more than 33% features selected on average).

The overall performance patterns of combined predictivity and parsimony are highly consistent with Markov blanket induction theory (section 2.2) which predicts maximum compactness and optimal classification performance when using the MB. Different instantiations of the GLL method give different trade-offs between predictivity and parsimony (details and statistical comparisons to the reference method are provided in online supplement Tables S2-S6 and S8).

In the companion paper (Aliferis et al., 2009), we examine in detail conditions under which PC induction can give optimal classification performance (the empirical illustration is provided in Figure 5.1(m)). The comparison of HITON-PC with G^2 test and HITON-PC with Z-test reveals that both statistics perform similarly, while the latter (where it is applicable) does not require discretization of continuous data that can simplify data analysis significantly (see Figure 5.1(l) and statistical comparisons in Table S9 in the online supplement). In Table S10 of the online supplement we provide statistical comparisons of non-GLL causal feature selection methods in terms of predictivity and parsimony. K2MB, BLCD-MB, IAMB, and FAST-IAMB rather unexpectedly perform statistically indistinguishably in terms of predictivity and parsimony. Since BLCD-MB differs from K2MB by an additional backward elimination step, this implies that this step rarely results in elimination of features in the real datasets tested.

5.3 Analysis of running times

Table S6 in the online supplement gives detailed running times for all feature selection experiments. Major observations include that: (i) univariate methods, RELIEF, RFE, LARS-EN are in general the fastest ones, (ii) Koller-Sahami is probably the slowest method since it does not terminate on several datasets within the allotted time limit, (iii) FAST-IAMB is two orders of magnitude faster than IAMB on the average, and (iv) GLL algorithms are practical for very high-dimensional data (e.g., in the Thrombin dataset with $> 100{,}000$ features GLL-PC requires 10 to 52 minutes single-CPU time depending on fixed-parameter configuration, and less than 3 hours when GLL-PC is automatically optimized by cross-validation).

In conclusion, the GLL reference algorithm dominates most feature selection methods in predictivity and compactness. Some non-GLL causal methods are more parsimonious

than the reference GLL method at the expense of severe classification suboptimality. One univariate method exhibits slightly higher predictivity but with severe disadvantage in parsimony. No feature selection method achieves equal or better compactness with equal or better classification performance than GLL.

6. Comparative evaluation of Markov blanket induction, local causal neighborhood and other non-causal algorithms for local structure discovery

In the present section we study the ability of GLL algorithms to discover local causal structure (in the form of parent and children sets and Markov blankets) and compare them with other local structure discovery methods as well as non-causal feature selection. While many researchers apply feature selection techniques strictly to improve the cost and effectiveness of classification, in many fields researchers routinely apply feature selection in order to gain insights *about the causal structure of the domain*. A frequently encountered example is in bioinformatics where a plethora of feature selection methods are applied in high-throughput genomic and proteomic data to discover biomarkers suitable for new drug development, personalizing medical treatments, and orienting subsequent experimentation (Zhou et al., 2002; Li et al., 2001; Holmes et al., 2000; Eisen et al., 1998). It is thus necessary to test the appropriateness of various feature selection techniques for causal discovery, not just classification.

In order to compare the performance of the tested techniques for causal discovery, we simulate data from known Bayesian networks and also use resimulation, whereby real data is used to elicit a causal network and then data is simulated from the obtained network (see Table A.4 in the Appendix). For each 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[3]. An independent sample of 5,000 instances is used for evaluation of classification performance.

In order to avoid overfitting of the results to the method used to induce the causal network, an algorithm with different inductive bias is used than the algorithms tested. In our case we use SCA (Friedman et al., 1999b). We note that SCA has greatly different inductive bias from the GLL variants and thus the comparison (provided that the causal generative model is a Bayesian network) is not unduly biased toward them, while still allowing induction of a credible causal graphical model. Specifically, the inductive biases of the two methods can be described as follows: SCA performs global, heuristically constrained, Bayesian search-and-score, greedy TABU iterative search for a Bayesian network that has maximum-a-posteriori probability given the data under uninformative prior on all possible network structures. GLL algorithms induce a local causal neighborhood, under the distributional assumption of faithfulness and causal sufficiency, employing statistical tests of conditional independence, and preferring to assume a variable is in the local neighborhood

3. 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.

whenever a conditional test is not applicable due to small sample (provided that a univariate association exists, otherwise independence is the default) in order to minimize false negative risk of losing a true member and overall risk of false positives and false negatives if true network is not dense. More about the inductive bias of GLL can be found in (Aliferis et al., 2009).

We obtained two resimulated networks as follows: (a) *Lung_Cancer* network: We randomly selected 799 genes and a phenotype target (cancer versus normal tissue indicator) from human gene expression data of (Bhattacharjee et al., 2001). Then we discretized continuous gene expression data and applied SCA to elicit network structure. (b) *Gene* network: It was obtained from a subset of variables of yeast gene expression data of (Spellman et al., 1998) that contained 800 randomly selected genes and a target variable denoting cell cycle state. Continuous gene expression data was also discretized and SCA was applied to learn network. This research design follows (Friedman et al., 2000).

Furthermore, we note that additional factors not captured in the simulation or resimulation process make real-life discovery potentially harder than in our experiments. Such factors include for example, deviations of faithfulness, existence of temporal and cellular aggregation effects, unmeasured variables, and various measurement, normalization, and noise artifacts. However evaluations with simulated and resimulated data yield comparative performances that are still highly informative since if a method cannot induce the correct structure from relatively easier settings, it is unlikely that in harder real-life situations it will perform any better. In other words successful causal structure discovery performance in simulated and resimulated networks represents at a minimum "gate-keeper" level performance that will filter the more promising from the less promising methods (Spirtes et al., 2000). Finally, as (Spirtes et al., 2000) note the behavior of constraint-based algorithms is particularly complex and theoretical analyses are very difficult to perform. The same is true for several other modern feature selection methods. Hence, simulation experiments are necessary in order to gain a deeper understanding of the strengths and limitations of many state-of-the-art algorithms. The evaluated algorithms are provided in Appendix Table A.5.

We evaluate the algorithms using the following metrics:

1. *Graph distance.* This metric calculates the average shortest unoriented graph distance of each variable returned by an algorithm to the local neighborhood of target, normalized by the average such distance of all variables in the graph. The rationale is to normalize the score to allow for comparisons across datasets and to correct the score for randomly choosing variables. The score is a non-negative number and has the following interpretation: value 0 means that current feature set is a subset of the true local neighborhood of the target, values less than 1 are better than random selection in the specific network, values equal to 1 are as good as random selection in the specific network and values higher than 1 are worse than random selection. The metric is computed using Dijkstra's shortest path algorithm.
2. *Euclidean distance from the perfect sensitivity and specificity (in the ROC space)* for discovery of local neighborhood of the target variable. This is computed as in (Tsamardinos et al., 2003b) and provides a loss function-neutral combination of sensitivity and specificity.
3. *Proportion of false positives and proportion of false negatives.*

4. *Classification performance using polynomial SVM and KNN classifiers* with parameters optimized by nested cross-validation (misclassification cost C and kernel degree d for SVMs and number of nearest neighbors k for KNN) on an independently sampled test dataset with large sample (n=5000). The performance is measured by AUC (Fawcett, 2003) on binary tasks and proportion of correct classifications on multiclass tasks.
5. *Feature selection time in minutes.* All caveats regarding interpretation of running times stated in section 5 apply here as well.

We note that the causal discovery evaluations emphasize *local* discovery of direct causes and direct effects and this choice is supported by several reasons. First, in many domains searching for direct causes and effects is natural (e.g., biological pathway discovery). Second, for non-causal feature selection methods, a natural causal interpretation of their output is being among the direct causes and direct effects (or the Markov blanket) of the target. Consider for example clustering or differential gene expression in bioinformatics where if *Gene1* clusters with *Gene2*, or if *Gene3* is more strongly differentially expressed with respect to some phenotype than *Gene4* then *Gene1* and *Gene2* are interpreted to be members of the same pathway (i.e., in close proximity in the gene regulatory/causal network), and *Gene 3* is interpreted to be more likely to determine the phenotype than *Gene4*. Similar interpretations abound for other non-causal feature selection methods. We notice that if a method is locally causally inconsistent then it is very unlikely that it will be globally causally consistent either. The logic of this argument is that algorithms either return global or local causal knowledge. If an algorithm outputs a global causal graph and this is incorrect, then this implies that locally it will be wrong for at least some variables. Conversely, if the global graph is correct then locally it is correct as well. If algorithm B outputs a correct local causal set (e.g., direct causes and direct effects) then we can "piece together" these sets and obtain a correct global graph. Finally, if an algorithm outputs an incorrect non-empty local causal set, this implies that B returns non-causes as direct causes or remote causes as direct causes (and the same for effects). Thus, it is not possible to construct the full causal graph strictly from knowledge provided by the algorithm. As a result, local causal consistency is necessary for global consistency as well.

A second reason for focusing on local causal discovery is that it is much harder in practice than indirect causal discovery in highly interconnected causal networks. In our bioinformatics example, because cancer affects many pathways, it is trivial to find genes affected by cancer, since a large proportion (e.g., half) of the measured genes are expected to be affected. However, it is vastly harder to find the chain of events that leads from occurrence of cancer to *Gene1* becoming under- or over-expressed. In such settings, discovery of remote causation is not particularly hard, neither it is particularly interesting. Conversely, when one has a locally correct causal discovery algorithm as elucidated in section 2, global causal learners can be relatively easily constructed.

Finally, in our evaluations we do not examine quality of causal orientation of the algorithms output for several reasons: First, while GLL algorithms' output can be oriented by constraint-based or other post-processing, non-causal feature selection methods do not readily admit orientation. Second, orientation is not needed when target T is a terminal variable as is often the case in the real data. Third, oriented local causal discovery is harder than unoriented one (Ramsey et al., 2006), and it makes sense to examine the ability of

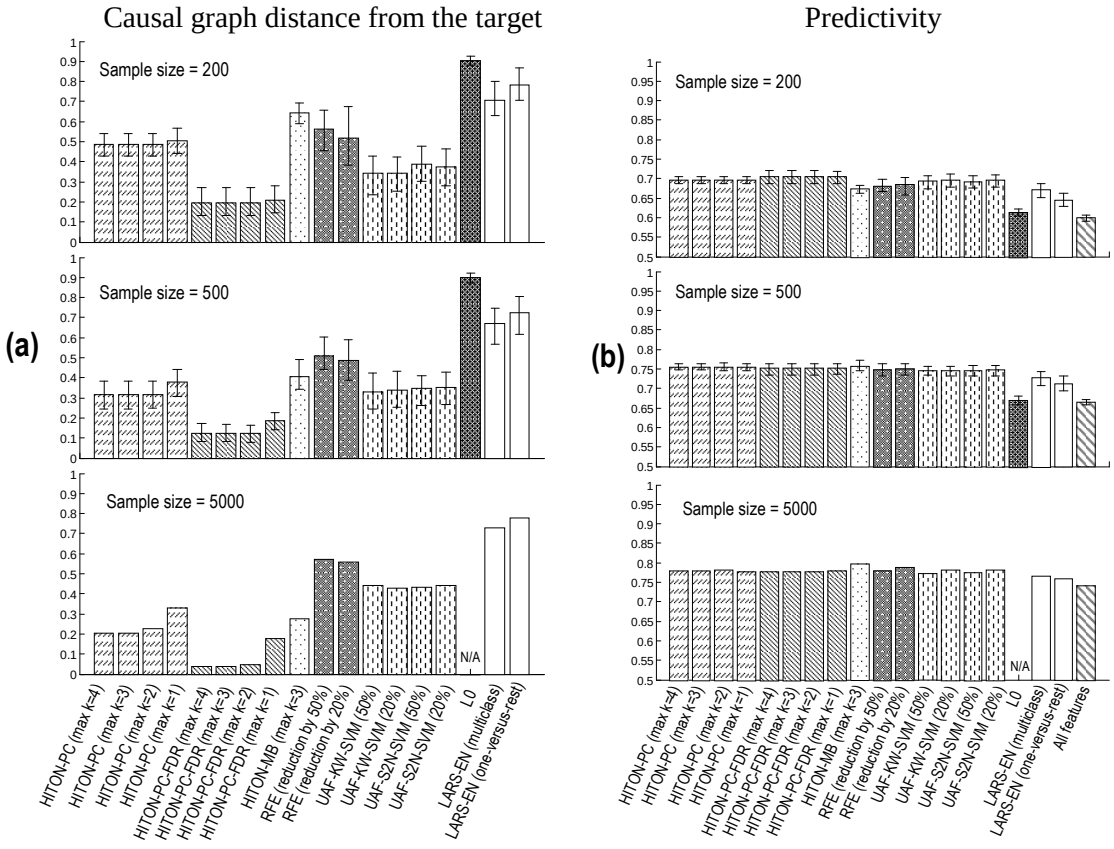


Figure 6.1: Performance of feature selection algorithms in 9 simulated and resimulated datasets: (a) *graph distance*, (b) *classification performance of polynomial SVM classifiers*. The smaller is causal graph distance and the larger is classification performance, the better is the algorithm. The results are given for training sample sizes = 200, 500, and 5000. 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 metrics reported in the figure are averaged over all datasets, selected targets, and multiple samples of each size. L0 did not terminate within 2 days (per target) for sample size 5000. Please see text for more details.

the feature selection algorithms for causal discovery in tasks of incremental difficulty, especially since as we will see most of the non-causal algorithms do not perform well even when seeking unoriented causality. Fourth, orientation information can be obtained subsequently by experiments or knowledge-based post-processing and in many practical settings it is not the primary obstacle to causal discovery.

6.1 Superiority of causal over non-causal feature selection methods for causal discovery

Causal methods achieve, consistently under a variety of conditions and across all metrics employed, superior causal discovery performance than non-causal feature selection methods in our experiments. Figures 6.1(a) and 6.2 compare semi-interleaved HITON-PC to HITON-MB, RFE, UAF, L0, and LARS-EN in terms of graph distance and for different sample sizes. Other GLL instantiations such as Interleaved-HITON-PC, MMPC, and Interleaved-MMPC perform similarly to HITON-PC (data in Table S12 in the online supplement). We apply

Sample size 200

	Child10	Insurance10	Alarm10	Hailfinder10	Pigs	Link	Lung_Cancer	Gene	*Average*
HITON-PC (max k=4)	0.43	0.41	0.42	0.83	0.41	0.44	0.44	0.50	0.48
HITON-PC (max k=3)	0.43	0.41	0.42	0.83	0.41	0.44	0.44	0.50	0.48
HITON-PC (max k=2)	0.43	0.41	0.42	0.83	0.41	0.44	0.44	0.50	0.48
HITON-PC (max k=1)	0.45	0.42	0.42	0.83	0.41	0.46	0.53	0.50	0.50
HITON-PC-FDR (max k=4)	0.29	0.15	0.24	0.18	0.10	0.17	0.24	0.18	0.19
HITON-PC-FDR (max k=3)	0.29	0.15	0.24	0.18	0.10	0.17	0.24	0.18	0.19
HITON-PC-FDR (max k=2)	0.29	0.15	0.24	0.18	0.10	0.17	0.24	0.18	0.19
HITON-PC-FDR (max k=1)	0.29	0.15	0.24	0.18	0.10	0.17	0.34	0.18	0.21
HITON-MB (max k=3)	0.70	0.68	0.50	0.99	0.49	0.66	0.50	0.64	0.64
RFE (reduction of features by 50%)	0.58	0.38	0.50	0.71	0.52	0.45	0.75	0.59	0.56
RFE (reduction of features by 20%)	0.57	0.46	0.54	0.65	0.46	0.30	0.63	0.54	0.52
UAF-KruskalWallis-SVM (50%)	0.45	0.27	0.32	0.50	0.26	0.34	0.34	0.26	0.34
UAF-KruskalWallis-SVM (20%)	0.43	0.32	0.38	0.55	0.27	0.29	0.29	0.22	0.34
UAF-Signal2Noise-SVM (50%)	0.47	0.31	0.44	0.47	0.33	0.35	0.46	0.27	0.39
UAF-Signal2Noise-SVM (20%)	0.44	0.35	0.40	0.56	0.28	0.29	0.44	0.25	0.38
L0	0.95	0.93	0.83	0.97	0.99	0.83	0.82	0.92	0.90
LARS-EN (for multiclass response)	0.67	0.70	0.64	0.79	0.78	0.66	0.64	0.78	0.71
LARS-EN (one-versus-rest)	0.83	0.68	0.67	0.92	0.89	0.70	0.67	0.89	0.78

Sample size 500

	Child10	Insurance10	Alarm10	Hailfinder10	Pigs	Link	Munin	Lung_Cancer	Gene	*Average*
HITON-PC (max k=4)	0.23	0.26	0.32	0.57	0.27	0.33	0.24	0.28	0.32	0.31
HITON-PC (max k=3)	0.23	0.26	0.32	0.57	0.27	0.33	0.24	0.28	0.32	0.31
HITON-PC (max k=2)	0.23	0.26	0.32	0.57	0.27	0.33	0.24	0.29	0.32	0.32
HITON-PC (max k=1)	0.24	0.28	0.37	0.57	0.34	0.39	0.24	0.52	0.45	0.38
HITON-PC-FDR (max k=4)	0.09	0.08	0.20	0.13	0.02	0.11	0.29	0.14	0.07	0.12
HITON-PC-FDR (max k=3)	0.09	0.08	0.20	0.13	0.02	0.11	0.29	0.13	0.07	0.12
HITON-PC-FDR (max k=2)	0.09	0.08	0.20	0.13	0.02	0.11	0.29	0.11	0.07	0.12
HITON-PC-FDR (max k=1)	0.09	0.11	0.23	0.13	0.08	0.12	0.29	0.40	0.22	0.19
HITON-MB (max k=3)	0.28	0.34	0.37	0.85	0.30	0.43	0.35	0.34	0.38	0.41
RFE (reduction of features by 50%)	0.63	0.51	0.61	0.53	0.37	0.40	0.26	0.70	0.56	0.51
RFE (reduction of features by 20%)	0.54	0.48	0.69	0.53	0.41	0.39	0.26	0.58	0.49	0.49
UAF-KruskalWallis-SVM (50%)	0.37	0.27	0.42	0.49	0.21	0.39	0.34	0.27	0.24	0.33
UAF-KruskalWallis-SVM (20%)	0.40	0.27	0.41	0.48	0.26	0.40	0.30	0.26	0.25	0.34
UAF-Signal2Noise-SVM (50%)	0.40	0.27	0.42	0.51	0.22	0.45	0.29	0.33	0.22	0.35
UAF-Signal2Noise-SVM (20%)	0.42	0.30	0.43	0.51	0.23	0.43	0.30	0.32	0.24	0.35
L0	0.98	0.97	0.93	0.98	0.99	0.87	0.53	0.87	0.97	0.90
LARS-EN (for multiclass response)	0.67	0.71	0.70	0.75	0.78	0.68	0.33	0.60	0.79	0.67
LARS-EN (one-versus-rest)	0.70	0.74	0.74	0.91	0.90	0.77	0.30	0.62	0.82	0.72

Sample size 5000

	Child10	Insurance10	Alarm10	Hailfinder10	Pigs	Link	Munin	Lung_Cancer	Gene	*Average*
HITON-PC (max k=4)	0.13	0.16	0.25	0.35	0.20	0.19	0.04	0.23	0.30	0.20
HITON-PC (max k=3)	0.13	0.16	0.25	0.35	0.20	0.19	0.04	0.23	0.30	0.20
HITON-PC (max k=2)	0.13	0.17	0.25	0.33	0.22	0.19	0.04	0.36	0.33	0.23
HITON-PC (max k=1)	0.18	0.27	0.29	0.33	0.30	0.42	0.04	0.63	0.50	0.33
HITON-PC-FDR (max k=4)	0.00	0.03	0.10	0.10	0.00	0.08	0.04	0.00	0.00	0.04
HITON-PC-FDR (max k=3)	0.00	0.03	0.10	0.10	0.00	0.08	0.04	0.00	0.00	0.04
HITON-PC-FDR (max k=2)	0.00	0.05	0.10	0.10	0.00	0.08	0.04	0.08	0.00	0.05
HITON-PC-FDR (max k=1)	0.01	0.17	0.14	0.11	0.16	0.16	0.04	0.55	0.23	0.18
HITON-MB (max k=3)	0.17	0.20	0.28	0.38	0.27	0.30	0.20	0.33	0.35	0.28
RFE (reduction of features by 50%)	0.63	0.64	0.58	0.59	0.40	0.90	0.28	0.66	0.48	0.57
RFE (reduction of features by 20%)	0.58	0.58	0.69	0.54	0.54	0.92	0.22	0.50	0.43	0.56
UAF-KruskalWallis-SVM (50%)	0.37	0.37	0.62	0.55	0.42	0.69	0.38	0.39	0.20	0.44
UAF-KruskalWallis-SVM (20%)	0.37	0.40	0.60	0.54	0.27	0.59	0.41	0.42	0.24	0.43
UAF-Signal2Noise-SVM (50%)	0.46	0.35	0.65	0.54	0.43	0.67	0.24	0.31	0.25	0.43
UAF-Signal2Noise-SVM (20%)	0.39	0.42	0.58	0.51	0.31	0.60	0.39	0.50	0.25	0.44
LARS-EN (for multiclass response)	0.67	0.85	0.65	0.87	0.74	0.75	0.52	0.71	0.79	0.73
LARS-EN (one-versus-rest)	0.71	0.86	0.74	0.84	0.95	0.80	0.48	0.74	0.88	0.78

Figure 6.2: Causal graph distance results for training sample sizes = 200, 500 and 5000. The results reported in the figure are averaged over all selected targets. Lighter cells correspond to smaller (better) values of graph distance; darker cells correspond to larger (worse) values of graph distance. L0 did not terminate within 2 days (per target) for sample size 5000.

HITON-PC as is and also with a variable pre-filtering step such that only variables that pass a test of univariate association with the target at 5% False Discovery Rate (FDR) threshold are input into the algorithm (Benjamini and Yekutieli, 2001; Benjamini and Hochberg, 1995). Motivation and analysis of incorporating FDR in GLL is provided in (Aliferis et al., 2009).

As can be seen, in all samples HITON-PC variants return features closely localized near the target while HITON-MB requires relatively larger sample size to localize well. The distance is smaller as sample size grows. Methods such as univariate filtering localize features well in some datasets and badly in others. As sample size grows, localization of univariate filtering deteriorates. Methods L0, and LARS-EN exhibit a *reverse-localization* bias (i.e., preferentially select features *away* from the target). Performance of RFE varies greatly across datasets in its ability to localize features and this is independent of sample size. A "bull's eye" plot for *Insurance10* dataset is provided in Figure 6.3. A localization example for *Insurance10* dataset is shown in Figure 6.4. The presented visualization examples are representative of the relative performance of causal versus non-causal algorithms. Table 6.1 provides p-values (via a permutation test at 5% alpha) for the differences of localization among algorithms.

Tables S13-S16 and Figure S2(a)-(d) in the online supplement compare the same algorithms in terms of (a) Euclidian distance from the point of perfect sensitivity and specificity, (b) proportion of false negatives, (c) proportion of false positives, and (d) running time in minutes. Consistent with the results presented in the main text, local causal discovery algorithms strongly outperform non-causal feature selection methods in ability to find the direct causes and effects of the target variable.

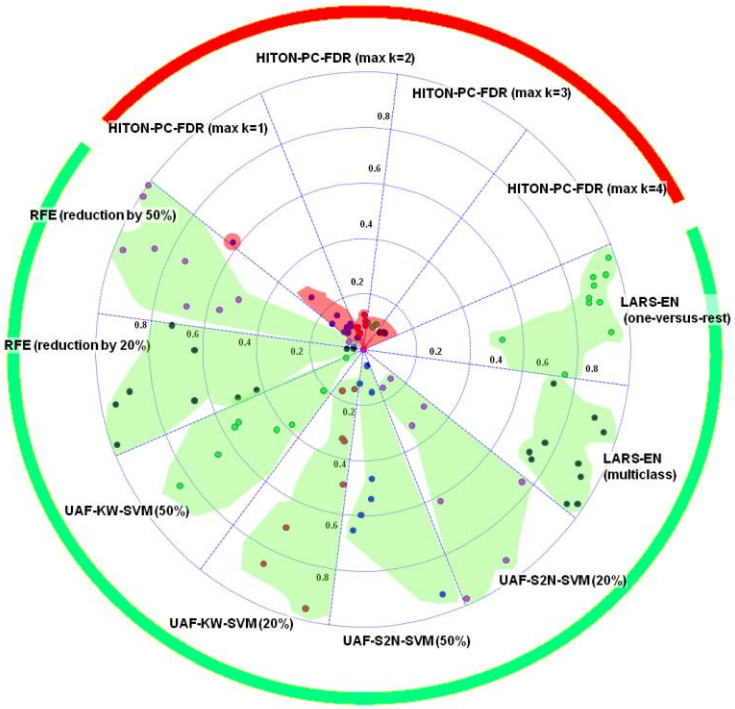


Figure 6.3: Visualization of graph distances for *Insurance10* network and sample size 5000 by "bull's eye" plot. For each method, results for 10 randomly selected targets are shown. The closer are points to the origin, the better is ability for local causal discovery. Results for GLL method HITON-PC-FDR are highlighted with red; results for baseline methods are highlighted with green.

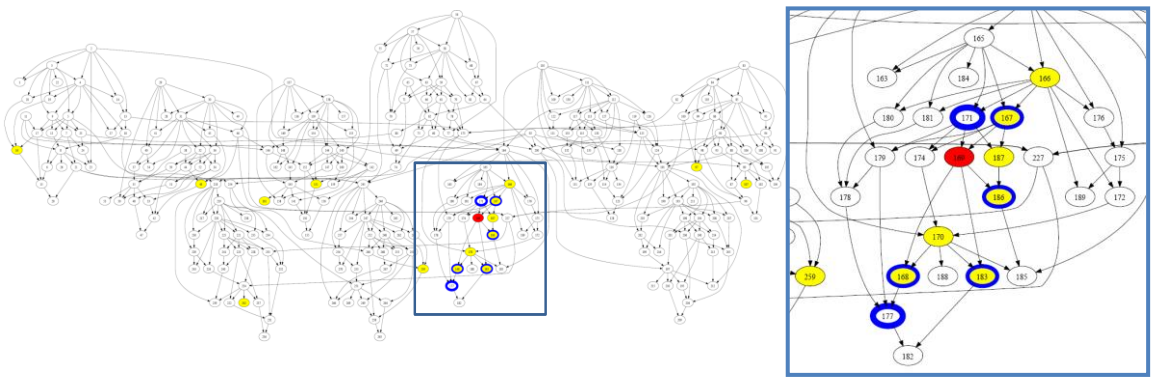


Figure 6.4: An example of poor localization by a baseline method and good localization by a GLL method. *Left*: Graph of the adjacency matrix of *Insurance10* network. Target variable is shown with red. HITON-PC discovers all 5 members of the parents and children set and a false positive variable #177 that is located close to the true neighborhood (discovered variables are shown with blue bolded circles). RFE discovers 4 out of 5 members of the PC set and introduces many false positives scattered throughout the network (discovered variables are shown with yellow circles). *Right*: A magnified area of the *Insurance10* network close to the target variable.

	Sample size = 200		Sample size = 500		Sample size = 5000	
Comparison	P-value	Nominal winner	P-value	Nominal winner	P-value	Nominal winner
average semi-interleaved HITON-PC with G2 test vs. HITON-MB	**<0.0001**	HITON-PC	**0.0042**	HITON-PC	**0.0472**	HITON-PC
average semi-interleaved HITON-PC with G2 test vs. average RFE	0.2594	HITON-PC	**0.0076**	HITON-PC	**<0.0001**	HITON-PC
average semi-interleaved HITON-PC with G2 test vs. average UAF	**0.0078**	UAF	0.6788	HITON-PC	**0.0086**	HITON-PC
average semi-interleaved HITON-PC with G2 test vs. L0	**<0.0001**	HITON-PC	**<0.0001**	HITON-PC		N/A
average semi-interleaved HITON-PC with G2 test vs. average LARS-EN	**<0.0001**	HITON-PC	**<0.0001**	HITON-PC	**<0.0001**	HITON-PC
average semi-interleaved HITON-PC-FDR with G2 test vs. HITON-MB	**<0.0001**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR
average semi-interleaved HITON-PC-FDR with G2 test vs. average RFE	**<0.0001**	HITON-PC-FDR	**0.0028**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR
average semi-interleaved HITON-PC-FDR with G2 test vs. average UAF	**<0.0001**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR
average semi-interleaved HITON-PC-FDR with G2 test vs. L0	**<0.0001**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR		N/A
average semi-interleaved HITON-PC-FDR with G2 test vs. average LARS-EN	**<0.0001**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR	**<0.0001**	HITON-PC-FDR

Table 6.1: Statistical comparison between semi-interleaved HITON-PC with G^2 test (with and w/o FDR correction) and other methods in terms of graph distance. Bolded p-values are statistically significant at 5% alpha.

6.2 Classification performance is misleading for causal discovery

Despite causally wrong outputs (i.e., failing to return the Markov blanket or parents and children set), several non-causal feature selection methods achieve comparable classification performance with causal algorithms in the simulated data. Figure 6.1(b) (and Tables S17-S18 and Figure S2(e) in the online supplement) shows the average AUC and proportion of

correct classifications. This phenomenon is related to information redundancy of features in relation to the target in non-sparse causal processes. In addition, it is facilitated by the relative insensitivity of state-of-the-art classifiers to irrelevant and redundant features. *Good classification performance is thus greatly misleading as a criterion for quality of causal hypotheses* generated by non-causal feature selection algorithms.

In conclusion, the results in the present section strongly undermine the hope that non-causal feature selection methods can be used as good heuristics for causal discovery. The idea that non-causal feature selection can be used for causal discovery should be viewed with caution (Guyon et al., 2007). Whole research programs are, in many domains, built on experiments motivated by causal hypotheses that were generated by non-causal feature selection results (Zhou et al., 2002; Li et al., 2001; Holmes et al., 2000; Eisen et al., 1998) and this seems an unfortunate and inadvisable practice, in light of existence of principled causal algorithms. On the other hand, generalized local learning algorithms in simulated and resimulated experiments show great potential for local causal discovery.

7. Discussion

7.1 Main findings

Our experimental evaluation shows that GLL algorithms typically attain the theoretically expected benefits of strong feature set parsimony without loss of performance relative to the best classification attained by any method used in the experiments. The wide range of datasets and algorithms used shows that the sufficient conditions stated in the proofs for correctness for GLL are likely to hold and/or that violations may be small or well tolerated.

The second major result from our experiments is that we showed that use of non-causal feature selection methods for learning causality although very widespread, is generally inadvisable. We used resimulated and simulated data and showed that causally-motivated feature selection methods connect local causal discovery with feature selection for classification consistent with recent theoretical work. Feature selection algorithms that are not causal have a tendency to return highly predictive feature sets that are scattered all over the network, or that are in the periphery of the network, and cannot be otherwise interpreted in a way that makes useful and consistent causal sense. We strongly caution practitioners to use principled causal discovery algorithms whenever available and to not substitute causal discovery methods with predictive/non-causal feature selection ones for reasons of convenience or due to non familiarity with such methods. Practical software widely exists that can be used to apply state-of-the-art causal methods including the methods studied in the present paper that is available for download from the online supplement.

Finally, the theoretical framework that is based in large part on faithfulness and other assumptions summarized in sections 2 and 3 is a valuable frame of reference both conceptually and algorithmically. However, we do not consider it to be an absolute and immutable measure by which to judge all new and existing algorithms. Our data shows that algorithms that are not deemed correct under the more general assumptions of the framework (e.g., algorithms that do not employ symmetry correction, or algorithms that use $PC(T)$ instead of $MB(T)$ for feature selection for classification) offer in many real datasets same predictive quality and better computational tractability than the sound algorithms. This is a reflection of several factors. One of them is the existence of distributions that are

special classes of faithful ones and are easier to analyze (e.g., where symmetry correction is not required, or in other words where $EPC(T) = PC(T)$). A second factor is mitigating circumstances for violations of assumptions (Aliferis et al., 2009). A third factor is that practical implementations of sound algorithms are statistically imperfect (in other words, a theoretical assumption that conveniently leads to a proof of correctness, for example that a conditional test of independence is correct, does not entail immediate or flawless practical feasibility since all such tests admit errors in practice). An alternative set of assumptions for correctness may require vaguely sufficient sample size disregarding the practical difficulty of determining whether in any given analysis this requirement is met. As a result, practical implementations may claim soundness without being demonstrably sound in applied settings. We address the small-sample behavior of GLL algorithms with empirical analysis in the companion paper (Aliferis et al., 2009).

7.2 Limitations and open problems

A possible critique of the present work is that Markov blanket features may not work well with a plethora of classifiers, distributions and loss functions. Indeed, a feature selector that is uniformly optimal is not attainable as shown by the results in (Tsamardinos and Aliferis, 2003), and several (possibly infinite) conceivable classifiers will fail to capture the information in the selected features. Our focus was to examine if the GLL framework has merit in the sense of whether GLL instantiations when applied and compared to reasonable state-of-the-art baseline feature selectors in many complex datasets from typical analysis domains and with practical classifiers, loss function and sample sizes, yield good performance consistent with the theoretical claims of GLL.

Another possibility we would like to address is that best predictivity achieved in our experiments for each dataset may not be optimal since some classifier other than SVMs and KNN may yield better predictivity. We believe that this possibility is remote for the following reason: Evidence from earlier published work where we have applied instances of GLL with classifiers such as ANNs, Decision Trees, Simple Bayes, as well as SVMs and KNN supports that the choice of classifier matters very little in practice and similar predictivity/parsimony patterns as the ones reported here were found (Aliferis et al., 2003a). On the other hand, the use of SVMs and KNN as classifiers uniformly across our experiments confers many benefits explained in section 5. To further support the use of these classifiers we provide additional experimental results in Appendix Table A.3 where we use features extracted from embedded or wrapper-based feature selectors (L0, RFVS, LARS-EN) and compare SVMs and KNN to classifiers native to the above embedded and wrapper-based methods. We found that SVMs and KNN achieve predictivity comparable to the classifiers from the aforementioned feature selectors.

Additional strong evidence in favor of our conclusions that GLL algorithms yield highly predictive and parsimonious feature sets is given by the simulated and resimulated data experiments where both the data-generative model and optimal feature sets are known. In those experiments the true Markov blanket is directly given by the model and does constitute the gold standard for the smallest and optimally informative feature set for common loss functions in the sense that *it contains all information available for predicting the target.*

The experiments showed that the GLL algorithms identify this Markov blanket very well and better than the baseline comparison algorithms.

Although the GLL framework and the studied instantiations and implementations are theoretically well motivated and empirically robust in many practical data analysis domains, as demonstrated in our experiments, as with all machine learning methods they should be expected to not perform well in quality or efficiency in certain distributions. Such distributions may include cases where the Markov blanket is very large and thus the combinatorics of the elimination phase makes it too slow. Another case can be when extreme non-linearities render the $PC(T)$ members "invisible" to the algorithm (because univariate association with the target is zero). Another possibility for hurting efficiency arises when excessive synthesis of information exists such that the true members of $PC(T)$ are not considered before other weakly relevant variables enter the $TPC(T)$. Also when certain types of deterministic relationships exist or more broadly target information equivalence (i.e., special types of violations of faithfulness), many Markov blankets may exist and the algorithms will return a predictively optimal feature set but both causal localization and optimal parsimony may be lost (Statnikov, 2008). The practical importance of these possibilities needs to be assessed domain-by-domain.

Some of the adverse situations described in the limitations sub-section can be addressed by relaxing the algorithm operation (e.g., for very large Markov blankets the analyst can set $max\text{-}k$ to a very small number and achieve faster execution but incur some false positives). In some domains, violation of assumptions are mitigated by other factors (e.g., (Aliferis et al., 2009) describes how connectivity can make extremely epistatic parents visible to the algorithms). These and other situations constitute open research areas and very recent research efforts attempt to address these issues. For example, (Statnikov, 2008) provides algorithms that address multiplicity of Markov blankets and (Tsamardinos and Brown, 2008b) introduce a method for kernel mapping of extremely non-linear functions to a faithful feature space that can be used to do feature selection via GLL in the transformed feature space.

Although the emphasis of the present work was in classification, Markov blanket theory applies equally well to regression and thus the GLL framework can be used for regression problems as well. An empirical analysis of performance of regression-oriented GLL instantiations and comparisons to state-of-the-art methods were not pursued here however.

7.3 Further problems addressed in the companion paper

While the theory motivating local learning and especially Markov blanket induction for feature selection has wide implications, it is far from complete. To begin with, all theoretical arguments to-date apply to the large sample case. While the theory implies that the large-sample Markov blanket and the corresponding classifiers fitted from large sample, are predictively optimal, it is not known to what extend learning from small samples affects the optimality of Markov blanket based feature selection. More specifically, it is not clear how often in small samples and real-life distributions the true Markov blanket (i.e., obtained from the data-generative process) gives an optimal classifier when the latter is fitted from small samples with state-of-the-art classifiers. Similarly, we do not know whether the estimated Markov blanket gives an optimal classifier when the latter is fitted from small samples or

even when it is fitted from the large sample. Related to the above for practical applications, we do not know how fast is convergence of the estimated Markov blanket/classifier to true Markov blanket/optimal classification as a function of sample size, for the available state-of-the-art Markov blanket inducing algorithms. In the second part of our work (Aliferis et al., 2009) we examine these issues. We also provide explanations why counter-intuitively relaxed versions of some algorithms that trade-off computational efficiency for theoretical soundness tend to outperform sound versions in some domains. Moreover, we systematically study the factors that influence the quality and number of statistical decisions, explain the inductive bias of the algorithms, show how non-causal feature selection methods can be understood in light of Markov blanket induction theory, and address divide-and-conquer local to global causal graph learning strategies.

Appendix A.

Proof of Theorem 2: Consider the algorithm in Figure 4.2. First notice, that as we mentioned above, when conditions (a) and (c) hold the direct causes and direct effects of T will coincide with the parents and children of T in the causal Bayesian network G that faithfully captures the distribution (Spirtes et al., 2000). As we have shown in section 4 and in (Tsamardinos et al., 2003b), the $PC_G(T) = PC(T)$ is unique in all networks faithfully capturing the distribution.

First we show that the algorithm will terminate, that is that the termination criterion of admissibility rule #3 will be met. The criterion requires that no variable eligible for inclusion will fail to enter $TPC(T)$ and that no variable that can be eliminated from $TPC(T)$ is left inside. Indeed because (a) due to admissibility rule #1 all eligible variables in OPEN are identified, (b) $\boldsymbol{V}$ is finite and OPEN instantiated to $\boldsymbol{V} \setminus \{T\}$, and (c) termination will not happen before all eligible members of OPEN are moved from OPEN to $TPC(T)$, the first part of the termination criterion will be satisfied. The second part of the termination criterion will also be satisfied because of admissibility rule #2 which examines for removal all variables and discards the ones that can be removed.

Lemma 1: The output of GLL-PC-nonsym $TPC(T)$ is such that: $PC(T) \subseteq TPC(T) \subseteq EPC(T)$.

Proof: Let us assume that $X \in PC(T)$ and show that $X \in TPC(T)$ by the end of GLL-PC-nonsym. By admissibility rule #3, X will never fail to enter $TPC(T)$ by the end of GLL-PC-nonsym. By Theorem 1, for all $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X\}$, $\neg I(X, T|\boldsymbol{Z})$ and so the elimination strategy because of admissibility rule #2 will never remove X from $TPC(T)$ by the end of GLL-PC-nonsym.

Now, let us assume that $X \in TPC(T)$ by the end of GLL-PC-nonsym and show that $X \in EPC(T)$. Let us assume the opposite, i.e., that $X \notin EPC(T)$ and so by definition $I(X, T|\boldsymbol{Z})$, for some $\boldsymbol{Z} \subseteq PC(T)\setminus\{X\}$. By the same argument as in the previous paragraph, we know that at some point before termination of the algorithm, in step 4, $TPC(T)$ will contain the $PC(T)$. Since $X \notin EPC(T)$, the elimination strategy will find that $I(X, T|\boldsymbol{Z})$, for some $\boldsymbol{Z} \subseteq PC(T) \setminus \{X\}$ and remove X from $TPC(T)$ contrary to what we assumed. Thus, $X \in EPC(T)$ by the end of GLL-PC-nonsym. □

Lemma 2: If $X \in EPC(T) \setminus PC(T)$, then $T \notin EPC(X) \setminus PC(X)$

Proof: Let us assume that $X \in EPC(T) \setminus PC(T)$. For every network G faithful to the

distribution P $Parents_G(T) \subseteq PC_G(T) = PC(T)$. X has to be a descendant of T in every network G faithful to the distribution because if it is not a descendant, then there is a subset $\boldsymbol{Z}$ of T's parents s.t., $I(X, T|\boldsymbol{Z})$ (by the Markov Condition). Since $X \in EPC(T) \setminus PC(T)$, we know that by definition $\neg I(X, T|\boldsymbol{Z})$, for all $\boldsymbol{Z} \subseteq PC(T) \setminus \{X\}$. By the same argument, if also $T \in EPC(X) \setminus PC(X)$, T would have to be a descendant of X in the every network G which is impossible since the networks are acyclic. So, $T \notin EPC(X) \setminus PC(X)$. □

Let us assume that $X \in PC(T)$. By Lemma 1, $X \in TPC(T)$ by the end of GLL-PC-nonsym. Since also $T \in PC(X)$, substituting X for T, we also have that by the end of GLL-PC-nonsym, $T \in TPC(X)$. So, X will not be removed from $\boldsymbol{U}$ by the symmetry requirement of GLL-PC either, and will be in the final output of the algorithm.

Conversely, let us assume that $X \notin PC(T)$ and show $X \notin \boldsymbol{U}$ at termination of algorithm GLL-PC. If X never enters $TPC(T)$ by the inclusion heuristic, the proof is done. Similarly, if X enters but is later removed from $TPC(T)$ by the exclusion strategy, the proof is done too. So, let us assume that X enters $TPC(T)$ at some point and by the end of GLL-PC-nonsym(T) is not removed by the exclusion strategy. By Lemma 1, we get that by the end of GLL-PC-nonsym, $X \in EPC(T)$ and since we assumed $X \notin PC(T)$, we get that $X \in EPC(T) \setminus PC(T)$. By Lemma 2, we get that $T \notin EPC(X) \setminus PC(X)$. Since also $T \notin PC(X)$, we get that $T \notin EPC(X)$. Step 3 of GLL-PC will thus eliminate X from $\boldsymbol{U}$. □

Proof of Theorem 4: Since we assume faithful Bayesian networks, d-separation in the graph of such a network is equivalent to independence and can be used interchangeably (Spirtes et al., 2000).

If $X \in MB(T)$, we show $X \in TMB(T)$ in the end. If $X \in MB(T)$ and $X \in PC(T)$, it will be included in the $TMB(T)$ in step 3, will not be removed afterwards and will be included in the final output.

If $X \in MB(T) \setminus PC(T)$ then X will be included in $\boldsymbol{S}$ since if X is a spouse of T, there exists Y (by definition of spouse) s.t., $X \in PC(Y)$, $Y \in PC(T)$ and $X \notin PC(T)$. For that Y, by Theorem 3 we know that $\neg I(X, T|\boldsymbol{Z} \cup \{Y\})$, for all $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, T\}$ and so the test in step 5c will succeed and X will be included in $TMB(T)$ in the end.

Conversely, if $X \notin MB(T)$ we show that $X \notin TMB(T)$ by the end of the algorithm. Let $\boldsymbol{Z}$ be the subset in step 5a, s.t., $I(X, T|\boldsymbol{Z})$ (i.e. $\boldsymbol{Z}$ d-separates X and T). Then, $\boldsymbol{Z}$ blocks all paths from X to T. For the test in step 5c to succeed a node Y must exist that opens a new path, previously closed by $\boldsymbol{Z}$, from X to T. Since by conditioning on an additional node a path opens, Y has to be a collider (by the d-separation definition) or a descendant of a collider on a path from X to T. In addition, this path must have length two edges since all nodes in $\boldsymbol{S}$ are the parents and children of the $PC(T)$ but without belonging in $PC(T)$. Thus, for the test in step 5c to succeed there has to be a path of length two from X to T with a collider in-between, i.e., X has to be a spouse of T. Since $X \notin MB(T)$ the test will fail for all Y and $X \notin TMB(T)$ by the end of the algorithm. □

Method	*Additional Information*	*Reference*
No feature selection		
RFE (recursive feature elimination SVM-based method)	reduction by 50% at each iteration, best performing feature subset is returned	(Guyon et al., 2002)
	reduction by 20% at each iteration, best performing feature subset is returned	
	reduction by 50% at each iteration, statistically same as best performing feature subset is returned	
	reduction by 20% at each iteration, statistically same as best performing feature subset is returned	
UAF-KruskalWallis-SVM (univariate ranking by Kruskal-Wallis statistic and feature selection with SVM backward wrapper)	reduction by 50% at each iteration, best performing feature subset is returned	(Statnikov et al., 2005a; Hollander and Wolfe, 1999)
	reduction by 20% at each iteration, best performing feature subset is returned	
	reduction by 50% at each iteration, statistically same as best performing feature subset is returned	
	reduction by 20% at each iteration, statistically same as best performing feature subset is returned	
UAF-Signal2Noise-SVM (univariate ranking by signal-to-noise statistic and feature selection with SVM backward wrapper)	reduction by 50% at each iteration, best performing feature subset is returned	(Guyon et al., 2006b; Statnikov et al., 2005a; Furey et al., 2000)
	reduction by 20% at each iteration, best performing feature subset is returned	
	reduction by 50% at each iteration, statistically same as best performing feature subset is returned	
	reduction by 20% at each iteration, statistically same as best performing feature subset is returned	
UAF-Neal-SVM (univariate ranking by Radford Neal's statistic and feature selection with SVM backward wrapper)	reduction by 50% at each iteration, best performing feature subset is returned	Chapter 10 in (Guyon et al., 2006a)
	reduction by 20% at each iteration, best performing feature subset is returned	
	reduction by 50% at each iteration, statistically same as best performing feature subset is returned	
	reduction by 20% at each iteration, statistically same as best performing feature subset is returned	
Random Forest Variable Selection (RFVS)	best performing feature subset is returned	(Diaz-Uriarte and Alvarez de Andres, 2006; Breiman, 2001)
	statistically same as best performing feature subset is returned	
LARS-Elastic Net (LARS-EN)	best performing feature subset is returned	(Zou and Hastie, 2005)
	statistically same as best performing feature subset is returned	
RELIEF (with backward wrapping by SVM)	Number of neighbors = 1, reduction by 50% at each iteration, best performing feature subset is returned	(Kononenko, 1994; Kira and Rendell, 1992)
	Number of neighbors = 1, reduction by 20% at each iteration, best performing feature subset is returned	
	Number of neighbors = 5, reduction by 50% at each iteration, best performing feature subset is returned	
	Number of neighbors = 5, reduction by 20% at each iteration, best performing feature subset is returned	
	Number of neighbors = 1, reduction by 50% at each iteration, statistically same as best performing feature subset is returned	
	Number of neighbors = 1, reduction by 20% at each iteration, statistically same as best performing feature subset is returned	
	Number of neighbors = 5, reduction by 50% at each iteration, statistically same as best performing feature subset is returned	
	Number of neighbors = 5, reduction by 20% at each iteration, statistically same as best performing feature subset is returned	
L0-norm		(Weston et al., 2003)

Table A.1: Algorithms used in evaluation on real datasets. Whenever statistical comparison was performed inside a wrapper, we used a non-parametric method by (DeLong et al., 1988). The only exception is Random Forest-based Variable Selection (RFVS), where we used a method recommended by its authors (Diaz-Uriarte and Alvarez de Andres, 2006). For GLL algorithms (i.e., variants of HITON-PC, HITON-MB, MMPC, MMMB) we experimented with both G^2 and Fisher's Z-test whenever the latter was applicable.

Method	*Additional Information*	*Reference*
Forward Stepwise Selection	using SVM classifier for wrapping	(Caruana and Freitag, 1994)
Koller-Sahami (with backward wrapping by SVM)	k=0, best performing feature subset is returned	(Koller and Sahami, 1996)
	k=1, best performing feature subset is returned	
	k=2, best performing feature subset is returned	
	k=0, statistically same as best performing feature subset is returned	
	k=1, statistically same as best performing feature subset is returned	
	k=2, statistically same as best performing feature subset is returned	
IAMB	G^2 test and α=0.05	(Tsamardinos and Aliferis, 2003; Tsamardinos et al., 2003a)
	G^2 test and α=0.01	
	mutual information criterion with threshold=0.01	
K2MB		(Cooper et al., 1997; Cooper and Herskovits, 1992)
BLCD-MB		(Mani and Cooper, 2004)
FAST-IAMB	G^2 test and α=0.05	(Yaramakala and Margaritis, 2005)
HITON-PC (semi-interleaved)	max k=4 and α=0.05	Novel algorithm
	max k=3 and α=0.05	
	max k=2 and α=0.05	
	max k=1 and α=0.05	
	max k=4 and α=0.01	
	max k=3 and α=0.01	
	max k=2 and α=0.01	
	max k=1 and α=0.01	
	max k and α selected by cross-validation	
Interleaved HITON-PC	max k=4 and α=0.05	(Aliferis et al., 2003a)
	max k=3 and α=0.05	
	max k=2 and α=0.05	
	max k=1 and α=0.05	
	max k=4 and α=0.01	
	max k=3 and α=0.01	
	max k=2 and α=0.01	
	max k=1 and α=0.01	
	max k and α selected by cross-validation	
MMPC	max k=4 and α=0.05	(Tsamardinos et al., 2006; Tsamardinos et al., 2003b)
	max k=3 and α=0.05	
	max k=2 and α=0.05	
	max k=1 and α=0.05	
	max k=4 and α=0.01	
	max k=3 and α=0.01	
	max k=2 and α=0.01	
	max k=1 and α=0.01	
	max k and α selected by cross-validation	
Interleaved MMPC	max k=4 and α=0.05	Novel algorithm
	max k=3 and α=0.05	
	max k=2 and α=0.05	
	max k=1 and α=0.05	
	max k=4 and α=0.01	
	max k=3 and α=0.01	
	max k=2 and α=0.01	
	max k=1 and α=0.01	
	max k and α selected by cross-validation	
HITON-MB (semi-interleaved)	max k=3 and α=0.05	Novel algorithm
	max k=3 and α=0.01	
MMMB	max k=3 and α=0.05	(Tsamardinos et al., 2003b)
	max k=3 and α=0.01	

Dataset name	*Domain*	*Num. variables*	*Num. samples*	*Target*	*Data type*	*Cross-val. design*	*Discretization applied*	*Notes*	*Reference*
Infant_ Mortality	Clinical	86	5,337	Died within the first year	Discrete	1-fold cross-val.	Already discrete	Imputed by nearest neighbor method	(Mani and Cooper, 1999)
Ohsumed	Text	14,373	5,000	Relevant to neonatal diseases	Continuous	1-fold cross-val.	Word absent/present		(Joachims, 2002)
ACPJ_ Etiology	Text	28,228	15,779	Relevant to etiology	Continuous	1-fold cross-val.	Word absent/present		(Aphinyanaphongs et al., 2006)
Lymphoma	Gene expression	7,399	227	3-year survival: dead vs. alive	Continuous	10-fold cross-val.	Binary/ternary univariate; used window sizes 10,15, 20, 25, 30 for ternary		(Rosenwald et al., 2002)
Gisette	Digit recognition	5,000	7,000	Separate 4 from 9	Continuous	1-fold cross-val.	Pixel present/absent	Used original training & validation sets only	*NIPS 2003 Feature Selection Challenge* (Guyon et al., 2006a)
Dexter	Text	19,999	600	Relevant to corporate acquisitions	Continuous	10-fold cross-val.	Word absent/present	Used original training & validation sets only	*NIPS 2003 Feature Selection Challenge* (Guyon et al., 2006a)
Sylva	Ecology	216	14,394	Ponderosa pine vs. everything else	Continuous & discrete	1-fold cross-val.	Binary/ternary univariate; used window sizes 1000, 1500, 2000, 2500, 3000 for ternary	Used original training & validation sets only	*WCCI 2006 Performance Prediction Challenge*
Ovarian_ Cancer	Proteomics	2,190	216	Cancer vs. normals	Continuous	10-fold cross-val.	Binary/ternary univariate; used window sizes 10, 15, 20, 25, 30 for ternary		(Conrads et al., 2004)
Thrombin	Drug discovery	139,351	2,543	Binding to thrombin	Discrete (binary)	1-fold cross-val.	Already discrete		*KDD Cup 2001*
Breast_ Cancer	Gene expression	17,816	286	Estrogen-receptor positive (ER+) vs. ER-	Continuous	10-fold cross-val.	Binary/ternary univariate, used window sizes 10, 15, 20, 25, 30 for ternary		(Wang et al., 2005)
Hiva	Drug discovery	1,617	4,229	Activity to AIDS HIV infection	Discrete (binary)	1-fold cross-val.	Already discrete	Used original training & validation sets only	*WCCI 2006 Performance Prediction Challenge*
Nova	Text	16,969	1,929	Separate politics from religion topics	Discrete (binary)	1-fold cross-val.	Already discrete	Used original training & validation sets only	*WCCI 2006 Performance Prediction Challenge*
Bankruptcy	Financial	147	7,063	Personal bankruptcy	Continuous & discrete	1-fold cross-val.	Binary/ternary univariate, used window sizes 1000, 1500, 2000, 2500, 3000 for ternary	Imputed by nearest neighbor method	(Foster and Stine, 2004)

Table A.2: Real datasets used in evaluation of predictivity and compactness.

Feature subset	*Classifier*	Infant_ Mortality	Ohsumed	ACPJ_ Etiology	Lymphoma	Gisette	Dexter	Sylva	Ovarian_ Cancer	Thrombin	Breast_ Cancer	Hiva	Nova	Bankruptcy
LARS-EN (w/o stat. comp.)	SVM	0.88	0.80	0.89	0.60	0.99	0.98	1.00	0.98	0.89	0.92	0.73	0.96	0.95
	LARS-EN	0.88	0.81	0.88	0.60	1.00	0.98	1.00	0.99	0.89	0.92	0.77	0.94	0.94
LARS-EN (with stat. comp.)	SVM	0.86	0.77	0.82	0.57	0.99	0.98	1.00	0.96	0.85	0.94	0.62	0.96	0.95
	LARS-EN	0.87	0.78	0.82	0.57	1.00	0.97	0.99	0.96	0.90	0.94	0.69	0.93	0.94
L0	SVM	0.82	0.72	0.84	0.60	0.99	0.97	1.00	0.97	0.81	0.91	0.68	0.96	T
	L0	0.81	0.72	0.87	0.58	0.99	0.97	1.00	0.96	0.81	0.91	0.69	0.95	T
RFVS (w/o stat. comp.)	SVM	0.82	T	T	0.61	T	0.98	1.00	0.97	T	0.93	0.74*	T	0.96
	RF	0.84	T	T	0.63	T	0.98	1.00	0.97	T	0.91	0.78	T	0.97
RFVS (with stat. comp.)	SVM	0.86	T	T	0.61	T	0.98	1.00	0.96	T	0.93	0.68*	T	0.97
	RF	0.78	T	T	0.63	T	0.98	1.00	0.97	T	0.92	0.75	T	0.97

Table A.3: Classification performance (AUC) for polynomial SVMs and classifiers native to LARS-EN, L0, and RFVS feature selection algorithms induced with features selected by the latter three methods. In cells marked with "T", the corresponding feature selection method did not terminate within the allotted time.

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

Table A.4: Simulated and resimulated datasets used for experiments. *Lung_Cancer* network is resimulated from human lung cancer gene expression data (Bhattacharjee et al., 2001) using SCA algorithm (Friedman et al., 1999b). *Gene* network is resimulated from yeast cell cycle gene expression data (Spellman et al., 1998) using SCA algorithm. More details about datasets are provided in (Tsamardinos et al., 2006).

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

Table A.5: Algorithms used in local causal discovery experiments with simulated and resimulated data.

References

C. F. Aliferis and G. F. Cooper. (1994) An evaluation of an algorithm for inductive learning of Bayesian belief networks using simulated data sets. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI).*

C. F. Aliferis, A. Statnikov, E. Kokkotou, P. P. Massion and I. Tsamardinos. (2006a) Local regulatory-network inducing algorithms for biomarker discovery from mass-throughput datasets. *Technical Report DSL 06-05.*

C. F. Aliferis, A. Statnikov and P.P. Massion (2006b) Pathway induction and high-fidelity simulation for molecular signature and biomarker discovery in lung cancer using microarray gene expression data. *Proceedings of the 2006 American Physiological Society Conference "Physiological Genomics and Proteomics of Lung Disease".*

C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani and X. D. Koutsoukos. (2009) Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification. Part II: Analysis and Extensions. *Journal of Machine Learning Research.*

C. F. Aliferis and I. Tsamardinos. (2002a) Algorithms for large-scale local causal discovery and feature selection in the presence of small sample or large causal neighborhoods. *Technical Report DSL 02-08.*

C. F. Aliferis and I. Tsamardinos. (2002b) Using Local Causal Induction to Improve Global Causal Discovery: Enhancing the Sparse Candidate Set. *Technical Report DSL 02-04.*

C. F. Aliferis, I. Tsamardinos and A. Statnikov. (2002) Large-scale feature selection using Markov blanket induction for the prediction of protein-drug binding. *Technical Report DSL 02-06.*

C. F. Aliferis, I. Tsamardinos and A. Statnikov. (2003a) HITON: a novel Markov blanket algorithm for optimal variable selection. *AMIA 2003 Annual Symposium Proceedings*, 21-25.

C. F. Aliferis, I. Tsamardinos, A. Statnikov and L. E. Brown. (2003b) Causal Explorer: a causal probabilistic network learning toolkit for biomedical discovery. *Proceedings of the 2003 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS)*.

Y. Aphinyanaphongs and C. F. Aliferis. (2004) Learning Boolean queries for article quality filtering. *Medinfo 2004*, 11, 263-267.

Y. Aphinyanaphongs, A. Statnikov and C. F. Aliferis. (2006) A comparison of citation metrics to machine learning filters for the identification of high quality MEDLINE documents. *J. Am. Med. Inform. Assoc.*, 13, 446-455.

X. Bai, C. Glymour, R. Padman, J. Ramsey, P. Spirtes and F. Wimberly. (2004) PCX: Markov Blanket Classification for Large Data Sets with Few Cases. *Technical Report, Center for Automated Learning and Discovery*.

G. E. A. P. A. Batista and M. C. Monard. (2003) An Analysis of Four Missing Data Treatment Methods for Supervised Learning. *Applied Artificial Intelligence*, 17, 519-533.

Y. Benjamini and Y. Hochberg. (1995) Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57, 289-300.

Y. Benjamini and D. Yekutieli. (2001) The control of the false discovery rate in multiple testing under dependency. *Ann. Statist*, 29, 1165-1188.

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. (2001) Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proc. Natl. Acad. Sci. U. S. A*, 98, 13790-13795.

L. Breiman. (2001) Random forests. *Machine Learning*, 45, 5-32.

L. E. Brown, I. Tsamardinos and C. F. Aliferis. (2005) A comparison of novel and state-of-the-art polynomial Bayesian network learning algorithms. *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*.

R. Caruana and D. Freitag. (1994) Greedy attribute selection. *Proceedings of the Eleventh International Conference on Machine Learning*, 28-36.

J. Cheng and R. Greiner. (1999) Comparing Bayesian network classifiers. *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI)*, 101-107.

J. Cheng and R. Greiner. (2001) Learning Bayesian Belief Network Classifiers: Algorithms and System. *Proceedings of 14th Biennial conference of the Canadian society for computational studies of intelligence*.

J. Cheng, R. Greiner, J. Kelly, D. Bell and W. Liu. (2002a) Learning Bayesian networks from data: an information-theory based approach. *Artificial Intelligence*, 137, 43-90.

J. Cheng, C. Hatzis, H. Hayashi, M. A. Krogel, S. Morishita, D. Page and J. Sese. (2002b) KDD Cup 2001 report. *ACM SIGKDD Explorations Newsletter*, 3, 47-64.

D. M. Chickering. (2002) Learning equivalence classes of bayesian-network structures. *Journal of Machine Learning Research*, 2, 445-498.

D. M. Chickering. (2003) Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3, 507-554.

D. M. Chickering, D. Geiger and D. Heckerman. (1994) Learning Bayesian networks is NP-hard. *Technical Report MSR-TR-94-17*.

C. Chow and C. Liu. (1968) Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14, 462-467.

T. P. Conrads, V. A. Fusaro, S. Ross, D. Johann, V. Rajapakse, B. A. Hitt, S. M. Steinberg, E. C. Kohn, D. A. Fishman, G. Whitely, J. C. Barrett, L. A. Liotta, E. F. Petricoin, III and T. D. Veenstra. (2004) High-resolution serum proteomic features for ovarian cancer detection. *Endocr. Relat Cancer*, 11, 163-178.

G. F. Cooper. (1997) A Simple Constraint-Based Algorithm for Efficiently Mining Observational Databases for Causal Relationships. *Data Mining and Knowledge Discovery*, 1, 203-224.

G. F. Cooper, C. F. Aliferis, R. Ambrosino, J. Aronis, B. G. Buchanan, R. Caruana, M. J. Fine, C. Glymour, G. Gordon and B. H. Hanusa. (1997) An evaluation of machine-learning methods for predicting pneumonia mortality. *Artificial Intelligence in Medicine*, 9, 107-138.

G. F. Cooper and E. Herskovits. (1992) A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9, 309-347.

G. F. Cooper and C. Yoo. (1999) Causal Discovery from a Mixture of Experimental and Observational Data. *Proceedings of Uncertainty in Artificial Intelligence*, 116-125.

D. Dash and G. F. Cooper. (2002) Exact model averaging with naive Bayesian classifiers. *Proc. 19th Int. Conf. Machine Learning (ICML)*, 91-98.

E. R. DeLong, D. M. DeLong and D. L. Clarke-Pearson. (1988) Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 44, 837-845.

R. Diaz-Uriarte and S. Alvarez de Andres. (2006) Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7, 3.

R. O. Duda and P. E. Hart. (1973) *Pattern classification and scene analysis*. Wiley, New York.

S. Duda, C. F. Aliferis, R. Miller, A. Statnikov and K. Johnson. (2005) Extracting drug-drug interaction articles from MEDLINE to improve the content of drug databases. *AMIA 2005 Annual Symposium Proceedings*, 216-220.

S. Dudoit and M. J. van der Laan. (2003) Asymptotics of cross-validated risk estimation in model selection and performance assessment. *UC Berkeley Division of Biostatistics Working Paper Series*, 126.

F. Eberhardt, C. Glymour and R. Scheines. (2005) On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables. *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, 178-183.

F. Eberhardt, C. Glymour and R. Scheines. (2006) N-1 Experiments Suffice to Determine the Causal Relations Among N Variables. *Innovations in Machine Learning: Theory And Applications*.

M. B. Eisen, P. T. Spellman, P. O. Brown and D. Botstein. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. U. S. A,* 95, 14863-14868.

J. Fan and R. Li. (2001) Variable Selection Via Nonconcave Penalized Likelihood and Its Oracle Properties. *Journal of the American Statistical Association*, 96, 1348-1361.

R. E. Fan, P. H. Chen and C. J. Lin. (2005) Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 6, 1918.

N. Fananapazir, M. Li, D. Spentzos and C. F. Aliferis. (2005) Formative evaluation of a prototype system for automated analysis of mass spectrometry data. *AMIA 2005 Annual Symposium Proceedings*, 241-245.

T. Fawcett. (2003) ROC Graphs: Notes and Practical Considerations for Researchers. *Technical Report, HPL-2003-4, HP Laboratories.*

D. P. Foster and R. A. Stine. (2004) Variable Selecion in Data Mining: Building a Predictive Model for Bankruptcy. *Journal of the American Statistical Association*, 99, 303-314.

L. Frey, D. Fisher, I. Tsamardinos, C. F. Aliferis and A. Statnikov. (2003) Identifying Markov blankets with decision tree induction. *Proceedings of the Third IEEE International Conference on Data Mining (ICDM).*

N. Friedman, D. Geiger and M. Goldszmidt. (1997) Bayesian network classifiers. *Machine Learning*, 29, 131-163.

N. Friedman, M. Goldszmidt and A. Wyner. (1999a) Data analysis with Bayesian networks: A bootstrap approach. *Proceedings of Uncertainty in Artificial Intelligence (UAI)*, 206-215.

N. Friedman, M. Linial, I. Nachman and D. Peér. (2000) Using Bayesian networks to analyze expression data. *J Comput. Biol.*, 7, 601-620.

N. Friedman, I. Nachman and D. Pe'er. (1999b) Learning Bayesian network structure from massive datasets: the "Sparse Candidate" algorithm. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI).*

G. M. Fung and O. L. Mangasarian. (2004) A Feature Selection Newton Method for Support Vector Machine Classification. *Computational Optimization and Applications*, 28, 185-202.

T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer and D. Haussler. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16, 906-914.

O. Gevaert, S. F. De, D. Timmerman, Y. Moreau and M. B. De. (2006) Predicting the prognosis of breast cancer by integrating clinical and microarray data with Bayesian networks. *Bioinformatics*, 22, e184-e190.

C. N. Glymour and G. F. Cooper. (1999) *Computation, causation, and discovery.* AAAI Press, Menlo Park, Calif.

P. I. Good. (2000) *Permutation tests: a practical guide to resampling methods for testing hypotheses.* Springer, New York.

I. Guyon, C. F. Aliferis and A. Elisseeff. (2007) Causal Feature Selection. In Liu,H. and Motoda,H. (eds), *Computational Methods of Feature Selection.* Chapman and Hall.

I. Guyon, S. Gunn, M. Nikravesh and L. A. Zadeh. (2006a) *Feature extraction: foundations and applications.* Springer-Verlag, Berlin.

I. Guyon and A. Elisseeff. (2003) An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, 1157-1182.

I. Guyon, J. Li, T. Mader, P. A. Pletscher, G. Schneider and M. Uhr. (2006b) Feature selection with the CLOP package. Technical report, 2006, http://clopinet.com/isabelle/Projects/ETH/TM-fextract-class.pdf.

I. Guyon, J. Weston, S. Barnhill and V. Vapnik. (2002) Gene selection for cancer classification using support vector machines. *Machine Learning*, 46, 389-422.

D. Hardin, I. Tsamardinos and C. F. Aliferis. (2004) A theoretical characterization of linear SVM-based feature selection. *Proceedings of the Twenty First International Conference on Machine Learning (ICML)*.

D. Heckerman. (1995) A tutorial on learning with Bayesian networks. *Technical Report MSR-TR-95-06*.

D. Heckerman, D. Geiger and D. M. Chickering. (1995) Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20, 197-243.

M. Hollander and D. Wolfe. (1999) *Nonparametric statistical methods*. Wiley, New York, NY, USA.

J. H. Holmes, D. R. Durbin and F. K. Winston. (2000) The learning classifier system: an evolutionary computation approach to knowledge discovery in epidemiologic surveillance. *Artif. Intell. Med.*, 19, 53-74.

N. Hoot, I. Feurer, C. W. Pinson and C. F. Aliferis. (2005) Modelling liver transplant survival: comparing techniques of deriving predictor sets. *Journal of Gastrointestinal Surgery*, 9, 563.

T. Joachims. (2002) *Learning to classify text using support vector machines*. Kluwer Academic Publishers, Boston.

K. Kira and L. A. Rendell.(1992) A practical approach to feature selection. *Proceedings of the Ninth International Workshop on Machine Learning*, 249-256.

R. Kohavi and G. H. John. (1997) Wrappers for feature subset selection. *Artificial Intelligence*, 97, 273-324.

D. Koller and M. Sahami. (1996) Toward optimal feature selection. *Proceedings of the International Conference on Machine Learning*, 1996.

I. Kononenko. (1994) Estimating attributes: Analysis and extensions of RELIEF. *Proceedings of the European Conference on Machine Learning*, 171-182.

L. Li, C. R. Weinberg, T. A. Darden and L. G. Pedersen. (2001) Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics*, 17, 1131-1142.

H. Liu, F. Hussain, C. L. Tan and M. Dash. (2002) Discretization: an enabling technique. *Data Mining and Knowledge Discovery*, 6, 393- 423.

H. Liu and H. Motoda. (1998) *Feature extraction, construction and selection: a data mining perspective*. Kluwer Academic, Boston.

S. Mani and G. F. Cooper. (1999) A Study in Causal Discovery from Population-Based Infant Birth and Death Records. *Proceedings of the AMIA Annual Fall Symposium*, 319.

S. Mani and G. F. Cooper. (2004) Causal discovery using a Bayesian local causal discovery algorithm. *Medinfo 2004*, 11, 731-735.

D. Margaritis and S. Thrun. (1999) Bayesian network induction via local neighborhoods. *Advances in Neural Information Processing Systems*, 12, 505-511.

S. Meganck, P. Leray and B. Manderick. (2006) Learning Causal Bayesian Networks from Observations and Experiments: A Decision Theoretic Approach. *Modeling Decisions in Artificial Intelligence, LNCS*, 58-69.

A. Moore and W. K. Wong. (2003) Optimal reinsertion: a new search operator for accelerated and more accurate Bayesian network structure learning. *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, 552-559.

K. P. Murphy. (2001) Active learning of causal Bayes net structure. *Technical Report, University of California, Berkeley.*

R. E. Neapolitan. (1990) *Probabilistic reasoning in expert systems: theory and algorithms.* Wiley, New York.

R. E. Neapolitan. (2004) *Learning Bayesian networks. Pearson Prentice Hall*, Upper Saddle River, NJ.

J. Pearl. (1988) *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann Publishers, San Mateo, California.

J. Pearl. (2000) *Causality: models, reasoning, and inference.* Cambridge University Press, Cambridge, U.K.

J. Pearl and T. Verma. (1991) A Theory of Inferred Causation. *Principles of Knowledge Representation and Reasoning: Proceedings of Second International Conference*, 441-452.

J. Pearl and T. S. Verma. (1990) Equivalence and synthesis of causal models. *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 220-227.

J. Peña, J. Bjorkegren and J. Tegner. (2005a) Growing Bayesian network models of gene networks from seed genes. *Bioinformatics*, 21, 224-229.

J. Peña, J. Bjorkegren and J. Tegner. (2005b) Scaslable, efficient and correct learning of Markov boundaries under the faithfulness assumption. *Proceedings of the Eighth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty.*

J. Peña, R. Nilsson, J. Björkegren and J. Tegnér. (2007) Towards scalable and data efficient learning of Markov boundaries. *International Journal of Approximate Reasoning*, 45, 211-232.

I. Pournara and L. Wernisch. (2004) Reconstruction of gene networks using Bayesian learning and manipulation experiments. *Bioinformatics*, 20, 2934-2942.

A. Rakotomamonjy. (2003) Variable selection using SVM-based criteria. *Journal of Machine Learning Research*, 3, 1357-1370.

J. Ramsey. (2006) A PC-style Markov blanket search for high-dimensional datasets. *Technical Report, CMU-PHIL-177, Carnegie Mellon University, Department of Philosophy.*

J. Ramsey, J. Zhang and P. Spirtes. (2006) Adjacency-Faithfulness and Conservative Causal Inference. *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI).*

A. Rosenwald, G. Wright, W. C. Chan, J. M. Connors, E. Campo, R. I. Fisher, R. D. Gascoyne, H. K. Muller-Hermelink, E. B. Smeland, J. M. Giltnane, E. M. Hurt, H. Zhao, L. Averett, L. Yang, W. H. Wilson, E. S. Jaffe, R. Simon, R. D. Klausner, J. Powell, P. L. Duffey, D. L. Longo, T. C. Greiner, D. D. Weisenburger, W. G. Sanger, B. J. Dave, J. C. Lynch, J. Vose, J. O. Armitage, E. Montserrat, A. Lopez-Guillermo, T. M. Grogan, T. P. Miller, M. LeBlanc, G. Ott, S. Kvaloy, J. Delabie, H. Holte, P. Krajci, T. Stokke and L. M. Staudt. (2002) The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *N. Engl. J Med.*, 346, 1937-1947.

A. Sboner and C. F. Aliferis. (2005) Modeling clinical judgment and implicit guideline compliance in the diagnosis of melanomas using machine learning. *AMIA 2005 Annual Symposium Proceedings*, 664-668.

T. Scheffer. (1999) Error estimation and model selection. Ph.D.Thesis, Technischen Universität Berlin, School of Computer Science.

C. Silverstein, S. Brin, R. Motwani and J. Ullman. (2000) Scalable Techniques for Mining Causal Structures. *Data Mining and Knowledge Discovery*, 4, 163-192.

P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein and B. Futcher. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol Cell*, 9, 3273-3297.

P. Spirtes, C. N. Glymour and R. Scheines. (2000) *Causation, prediction, and search.* MIT Press, Cambridge, Mass.

A. Statnikov. (2008) Algorithms for Discovery of Multiple Markov Boundaries: Application to the Molecular Signature Multiplicity Problem. Ph.D. Thesis, Department of Biomedical Informatics, Vanderbilt University.

A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin and S. Levy. (2005a) A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21, 631-643.

A. Statnikov, D. Hardin and C. F. Aliferis. (2006) 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.*

A. Statnikov, I. Tsamardinos, Y. Dosbayev and C. F. Aliferis. (2005b) GEMS: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. *Int. J. Med. Inform.*, 74, 491-503.

J. Tian and J. Pearl. (2001) Causal Discovery from Changes: A Bayesian Approach. *UCLA Cognitive Systems Laboratory, Technical Report (R-285).*

R. Tibshirani. (1996) Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58, 267-288.

S. Tong and D. Koller. (2001) Active learning for structure in Bayesian networks. *Proceedings of the International Joint Conference on Artificial Intelligence*, 17, 863-869.

I. Tsamardinos and C. F. Aliferis. (2003) Towards principled feature selection: relevancy, filters and wrappers. *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics (AI & Stats).*

I. Tsamardinos, C. F. Aliferis and A. Statnikov. (2003a) Algorithms for large scale Markov blanket discovery. *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 376-381.

I. Tsamardinos, C. F. Aliferis and A. Statnikov. (2003b) 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)*, 673-678.

I. Tsamardinos, C. F. Aliferis, A. Statnikov and L. E. Brown. (2003c) Scaling-up Bayesian network learning to thousands of variables using local learning technique. *Technical Report DSL 03-02.*

I. Tsamardinos, Brown L.E. and C. F. Aliferis. (2005) The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. *Technical report DSL-05-01.*

I. Tsamardinos and L. E. Brown. (2008a) Bounding the False Discovery Rate in Local Bayesian Network Learning. *Proceedings of the Twenty Third National Conference on Artificial Intelligence (AAAI).*

I. Tsamardinos and L. E. Brown. (2008b) Markov Blanket-Based Variable Selection in Feature Space. *Technical report DSL-08-01.*

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

L. Wang, J. Zhu and H. Zou. (2006) The Doubly Regularized Support Vector Machine. *Statistica Sinica*, 16, 589-615.

Y. Wang, J. G. Klijn, Y. Zhang, A. M. Sieuwerts, M. P. Look, F. Yang, D. Talantov, M. Timmermans, M. E. Meijer-van Gelder, J. Yu, T. Jatkoe, E. M. Berns, D. Atkins and J. A. Foekens. (2005) Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet*, 365, 671-679.

J. Weston, A. Elisseeff, B. Scholkopf and M. Tipping. (2003) Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3, 1439-1461.

S. Yaramakala and D. Margaritis. (2005) Speculative Markov Blanket Discovery for Optimal Feature Selection. *Proceedings of the Fifth IEEE International Conference on Data Mining*, 809-812.

C. Yoo and G. F. Cooper. (2004) An evaluation of a system that recommends microarray experiments to perform to discover gene-regulation pathways. *Artif. Intell. Med.*, 31, 169-182.

X. Zhou, M. C. J. Kao and W. H. Wong. (2002) Transitive functional annotation by shortest-path analysis of gene expression data. *Proceedings of the National Academy of Sciences*, 99, 12783-12788.

J. Zhu, S. Rosset, T. Hastie and R. Tibshirani. (2004) 1-norm support vector machines. *Advances in Neural Information Processing Systems (NIPS).*

H. Zou and T. Hastie. (2005) Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B(Statistical Methodology)*, 67, 301-320.