Machine Learning Department, School of Computer Science, Carnegie Mellon University (2009)

Learning Directed Graphical Models from Nonlinear and Non-Gaussian Data

Data Analysis Project for Master of Science in Machine Learning

Robert E. Tillman RTILLMAN@CMU.EDU
Carnegie Mellon University
Pittsburgh, PA 15232, United States

Advisor: Peter Spirtes

Abstract

Traditional constraint-based and score-based methods for learning directed graphical models from continuous data have two significant limitations: (i) they require (in practice) assuming dependencies are linear with Gaussian noise; (ii) they cannot distinguish between Markov equivalent structures. More recent structure learning methods avoid both limitations by directly exploiting characteristics of the observed data distribution resulting from nonlinear effects and non-Gaussian noise. We review these methods and focus on the *additive noise model* approach, which while more general than traditional approaches also suffers from two major limitations: (i) it is invertible for certain distribution families, i.e. linear Gaussians, and thus not useful for structure learning in these cases; (ii) it was originally proposed for the two variable case with a multivariate extension that requires enumerating all possible DAGs, which is ususally intractable. To address these two limitations, we introduce *weakly additive noise models*, which extends the additive noise model framework to cases where additive noise models are invertible and noise is not additive. We then provide an algorithm for learning equivalence classes for weakly additive noise models from data which combines a PC style search using recent advances in kernel measures of conditional dependence with greedy local searches for additive noise models. This combined approach provides a more computationally efficient search procedure for when nonlinear dependencies and/or non-Gaussian noise may be present that learns equivalence classes of structures which are often more specific than the Markov equivalence class even in the case of invertible additive noise models and non-additive noise models, addressing the limitations of both traditional structure learning methods and the additive noise model. We evaluate this approach using synthetic data and real climate teleconnection and fMRI data.

Keywords: structure learning, causal discovery, probabilistic graphical models, Bayesian networks, additive noise model, kernel methods

1. Introduction

Learning probabilistic graphical models from data serves two primary purposes: (i) finding compact representations of probability distributions so that probabilistic inference queries can be made efficiently and (ii) modeling unknown data generating mechanisms and predicting causal relationships.

Traditional approaches for learning directed structures from continuous data suffer from two major limitations: (i) in practice, they require assuming that all dependencies are linear with Gaussian noise and (ii) they cannot always identify a unique optimal structure, but rather can only reduce the set of possible structures to an equivalence class of structures which entail the same Markov properties. While the first limitation may not pose a significant problem in many contexts where a linear Gaussian approximation may suffice, there are well known contexts, such as fMRI images, where nonlinear dependencies are common and data may not tend towards Gaussianity. Voortman and Druzdzel (2008) shows that while the accuracy of the well known PC algorithm (Spirtes et al., 2000) for directed structure learning is not significantly affected by violations of the assumption that noise is Gaussian, it is significantly affected by violations of the assumption that dependencies are linear. The second limitation may not be a serious limitation if one is only interested in learning graphical models to do probabilistic inference, since any structure in the equivalence class will yield the same results. However, if one is interested in predicting causal relationships or the effects of intervening on variables, then this does pose a significant limitation since such inferences cannot be made in general using only the equivalence class.

While the linear Gaussian assumption was originally made to simplify the problem of structure learning, it has recently become clear that nonlinearity and non-Gaussianity can actually be a blessing and reveal more information about the true data generating process than Markov relations. New methods have been developed for learning directed structures from continuous data when relationships are linear with non-Gaussian noise and when relationships are (possibly) nonlinear with Gaussian or non-Gaussian noise. These methods can often learn a unique directed structure instead of an equivalence class.

The key objective of this work is to (i) extend these new methods for structure learning so that they are applicable under weaker assumptions and (ii) overcome some of the current limitations with using these methods in practice. After first reviewing the traditional constraint-based and score-based approaches for structure learning, we introduce two of the most popular methods that learn structure by exploiting nonlinear dependencies and non-Gaussian noise, LiNGAM (Shimizu et al., 2006) and the additive noise model approach (Hoyer et al., 2009). We focus specifically on the additive noise model approach, the more general of these approaches, and discuss its two major limitations: (i) the existence of certain distribution families, e.g. linear Gaussians, for which this model cannot be used successfully for structure learning; (ii) the computational costs associated with using this model with more than

two variables. To address both of these limitations, we introduce a new framework for learning directed structures, *weakly additive noise models* (Tillman et al., 2009), which extends the additive noise model framework to the problematic distribution families mentioned above. We then provide an algorithm for learning equivalence classes for weakly additive noise models from data which combines a PC style search using recent advances in kernel measures of conditional dependence with greedy local searches for additive noise models. Combining these two approaches results in a more computationally efficient search procedure for when nonlinear dependencies and/or non-Gaussian noise may be present in data; the procedure learns equivalence classes of structures which are often more specific than the equivalence classes learned by traditional structure learning methods and is useful even with distribution families which are problematic for the additive noise model approach. This addresses both the two major limitations of traditional structure learning methods and the two major limitations of the additive noise model.

Section 2 reviews background in graphical models and introduces some notation; section 3 discusses the traditional approaches to structure learning; section 4 introduces approaches to structure learning which exploit nonlinearity and non-Gaussianity; section 5 introduces the Hilbert space embeddings of distributions framework and the Hilbert Schmidt Independence Criterion (Gretton et al., 2008) nonparametric statistical test of independence which is used in the additive noise model approach and will be useful in later sections; section 6 introduces our weakly additive noise models framework to overcome the limitations of the additive noise model; section 7 introduces our algorithm for learning equivalence classes for weakly additive noise models described above; section 8 provides a detailed description of the conditional independence test that is used by this algorithm; section 9 discusses some related research; section 10 presents experimental results; finally, section 11 offers some conclusions.

2. Probabilistic graphical models

A *directed graph* $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is a set of nodes $\mathcal{V}$ and a set of directed edges $\mathcal{E}$ connecting distinct nodes in $\mathcal{V}$. Two nodes in $\mathcal{V}$ are *adjacent* if they are connected by an edge in $\mathcal{E}$. A *directed path* in $\mathcal{G}$ is a sequence of nodes $V_1, \ldots, V_n$ such that for $1 \leq i < n$, there is a directed edge pointing from V_i to V_{i+1}. If there does not exist a directed path $V_1, \ldots, V_n$ in $\mathcal{G}$ such that V_1 and V_n refer to the same node, then $\mathcal{G}$ is a *directed acyclic graph (DAG)*.

If $V_i \to V_j$ is a directed edge in $\mathcal{G}$, then V_i is a *parent* of V_j and V_j is a *child* of V_i. We use $\mathbf{Pa}_{\mathcal{G}}^{V_i}$ to refer to the set of parents of V_i in $\mathcal{G}$ and $\mathbf{Ch}_{\mathcal{G}}^{V_i}$ to refer to the children of V_i in $\mathcal{G}$. The *degree* of a node V_i in $\mathcal{G}$ is the number of edges that are either directed into or out from V_i. A *v-structure (collider)* is a triple $\langle V_i, V_j, V_k \rangle \subseteq \mathcal{V}$ such that $\{V_i, V_k\} \subseteq \mathbf{Pa}_{\mathcal{G}}^{V_j}$. A v-structure is *immoral*, or an *immorality (unshielded collider)*, if there is no edge between V_i and V_k in $\mathcal{G}$.

A joint distribution $\mathcal{P}$ over variables corresponding to nodes in $\mathcal{V}$ is *Markov* with respect to $\mathcal{G}$ if $\mathcal{P}$ can be factored according to the structure of $\mathcal{G}$ as follows:

$$\mathbb{P}_{\mathcal{P}}(\mathcal{V}) = \prod_{V_i \in \mathcal{V}} \mathbb{P}_{\mathcal{P}}\left(V_i \mid \mathbf{Pa}_{\mathcal{G}}^{V_i}\right)$$

$\mathcal{P}$ is *faithful* to $\mathcal{G}$ if every conditional independence that is true in $\mathcal{P}$ is entailed by the above *Markov factorization*. If the Markov factorizations of two distinct DAGs $\mathcal{G}$ and $\mathcal{H}$ entail exactly the same conditional independencies, then $\mathcal{G}$ and $\mathcal{H}$ are said to be *Markov equivalent*. All DAGs which are Markov equivalent have the same adjacencies, but different directed edges (Spirtes et al., 2000).

A *mixed graph* is a graph which consists of both directed and undirected edges. A *partially directed acyclic graph* (PDAG) $\mathcal{H}$ for $\mathcal{G}$ is a mixed graph which represents all of the DAGs that are Markov equivalent to $\mathcal{G}$ (including $\mathcal{G}$). If $V_i \rightarrow V_j$ is a directed edge in $\mathcal{H}$, then all DAGs Markov equivalent to $\mathcal{G}$ have this directed edge; if $V_i - V_j$ is an undirected edge in $\mathcal{H}$, then some DAGs that are Markov equivalent to $\mathcal{G}$ have the directed edge $V_i \rightarrow V_j$, while others have the directed edge $V_i \leftarrow V_j$.

3. Traditional approaches for structure learning

The traditional algorithms for structure learning can be described as either *constraint-based* or *score-based*. Given sample data from a distribution $\mathcal{P}$, constraint-based algorithms use results from a series of conditional independence tests using the sample data to determine the equivalence class of structures that are Markov to $\mathcal{P}$. Score-based algorithms use model selection criteria, e.g. AIC, BIC, to "score" a structure based on its Markov factorization. The goal is to search for a structure (or equivalence class of structures) which maximizes this score.

3.1 Constraint-based methods

The earliest and most straightforward constraint-based structure learning algorithm is the SGS (Spirtes et al., 2000) or IC (Pearl, 2000) algorithm, shown as algorithm 1. This algorithm relies on the fact that certain combinations of conditional independencies and conditional dependencies entail the absence of edges and the existence of immoralities. Once edges have been removed from the complete graph and immoralities have been oriented using conditional independence information, the fourth step uses a set of rules, shown as algorithm 2, often referred to as the "Meek rules" (Meek, 1995) to make any further orientations that can be made based on the fact that no directed cycles are allowed and all immoralities have already been oriented.

While the SGS/IC algorithm can be shown to be correct and complete in the large sample limit (Spirtes et al., 2000), it is neither computationally efficient, since we must search through the entire powerset of $\mathcal{V} \backslash \{V_i, V_j\}$ for all $\{V_i, V_j\} \subseteq \mathcal{V}$ in step 2, nor statistically robust since the results of conditional independence tests with large conditioning sets are generally unreliable. The PC algorithm (Spirtes et al., 2000)

Input : Observed data for variables in $\mathcal{V}$
Output: PDAG $\mathcal{G}$ over nodes $\mathcal{V}$

1 $\mathcal{G} \leftarrow$ the complete undirected graph over the variables in $\mathcal{V}$
2 For $\{V_i, V_j\} \subseteq \mathcal{V}$, if $\exists \mathbf{S} \subseteq \mathcal{V} \backslash \{V_i, V_j\}$, such that $V_i \perp\!\!\!\perp V_j \mid \mathbf{S}$, then remove the $V_i - V_j$ edge from $\mathcal{G}$
3 For $\{V_i, V_j, V_k\} \subseteq \mathcal{V}$ such that $V_i - V_j$ and $V_j - V_k$ are (possibly directed) edges in $\mathcal{G}$, but there is no edge between $V_i - V_k$, if $\nexists \mathbf{S} \subseteq \mathcal{V} \backslash \{V_i, V_j, V_k\}$, such that $V_i \perp\!\!\!\perp V_k \mid \{\mathbf{S} \cup V_j\}$, then orient $V_i \rightarrow V_j \leftarrow V_k$
4 Orient any edges necessary to prevent additional immoralities or cycles using the Meek rules (Meek, 1995)

Algorithm 1: SGS/IC algorithm

Input : Mixed graph $\mathcal{G}$ over nodes $\mathcal{V}$ where only immoralities are oriented
Output: PDAG $\mathcal{G}$ over nodes $\mathcal{V}$

1 **while** $\mathcal{G} \neq \mathcal{G}'$ **do**
2 $\quad \mathcal{G}' \leftarrow \mathcal{G}$
3 $\quad$ For $\{V_i, V_j, V_k\} \subseteq \mathcal{V}$ such that in $\mathcal{G}$, $V_i \rightarrow V_j$ is a directed edge, $V_j - V_k$ is an undirected edge, and V_i and V_k are not adjacent, make $V_j \rightarrow V_k$ a directed edge in $\mathcal{G}$
4 $\quad$ For $\{V_i, V_j\} \subseteq \mathcal{V}$ such that there is a directed path in $\mathcal{G}$ from V_i to V_j, make $V_i \rightarrow V_j$ a directed edge in $\mathcal{G}$
5 $\quad$ For $\{V_i, V_j, V_k, V_l\} \subseteq \mathcal{V}$ such that in $\mathcal{G}$, V_i is adjacent to V_j, V_k, and V_l and $\langle V_j, V_k, V_l \rangle$ is a v-structure, make $V_i \rightarrow V_k$ a directed edge in $\mathcal{G}$
6 **end**

Algorithm 2: Meek rules

is a greedy version of SGS/IC which avoids these problems. Instead of searching all subsets of $\mathcal{V}\backslash\{V_i, V_j\}$ for a set $\mathbf{S}$ such that $V_i \perp\!\!\!\perp V_j \mid \mathbf{S}$ in step 2, PC initially considers only the set $\mathbf{S} = \emptyset$ for all V_i and V_j pairs and then iteratively increases the cardinality of such sets $\mathbf{S}$ that are considered. At each iteration, a set $\mathbf{S}$ is only considered if it consists of nodes adjacent to either V_i or V_j. When $\nexists V_k \in \mathcal{V}$ with degree in $\mathcal{G}$ greater than the current value of $|\mathbf{S}|$, step 2 is complete. The remaining steps of PC are the same as SGS/IC. PC provably discovers the correct PDAG in the large sample limit when the Markov, faithfulness, and causal sufficiency, i.e. there are no unmeasured common causes of two or more measured variables, assumptions hold (Spirtes et al., 2000). The partial correlation based Fisher Z-transformation test, which assumes dependencies are linear with Gaussian noise, is used for conditional independence testing when PC is used with continuous variables.

The PC algorithm is one of many constraint-based algorithms that are used for structure learning. For instance, there are other constraint-based algorithms that may be used if one wishes to drop the causal sufficiency assumption (Spirtes et al., 2000; Hoyer et al., 2008b; Pellet and Elisseeff, 2009), if one is interested in learning cyclic structures (Richardson, 1996; Lacerda et al., 2008), and for doing structure learning in high dimensions (Tsamardinos et al., 2006; Kalisch and Bühlmann, 2007).

3.2 Score-based methods

Score-based algorithms treat structure learning as a hill-climbing problem where one searches for a structure which maximizes some chosen score function. Typically, a score functions is chosen which decomposes according to graphical structure. BIC, shown below where m is the sample size, and d is the number of parameters in the model, is such a score.

$$\begin{aligned} BIC(\mathcal{G}|\mathcal{V}, \theta) &= \log \mathbb{P}(\mathcal{V}|\mathcal{G}, \theta) - \frac{d}{2} \log m \\ &= \log \prod_{V_i \in \mathcal{V}} \mathbb{P}\left(\mathcal{V}|\mathbf{Pa}_{\mathcal{G}}^{V_i}, \theta\right) - \frac{d}{2} \log m \\ &= \sum_{V_i \in \mathcal{V}} \log \mathbb{P}\left(\mathcal{V}|\mathbf{Pa}_{\mathcal{G}}^{V_i}, \theta\right) - \frac{d}{2} \log m \end{aligned}$$

Notice that the BIC score is simply the data loglikelihood penalized to select sparse structures. If the penalty is removed, then this score becomes the so-called *Bayesian score*, which is maximized by the complete graph.

The GES algorithm (Chickering, 2002) is a score-based structure learning algorithm which learns the correct PDAG in the large sample limit under the Markov, faithfulness, and causal sufficiency assumptions when used with a score function that is *locally consistent* (Chickering, 2002), or has the following two properties:

1. Adding an edge that eliminates an independence constraint that is not entailed by the Markov factorization of the correct structure increases the score

2. Adding an edge that does not eliminate an independence constraint that is not entailed by the Markov factorization of the correct structure decreases the score

The BIC score is locally consistent (Chickering, 2002). GES searches over the space of PDAGs beginning with the empty graph. It consists of two stages where edges are added or removed and certain edges may be reversed until the score function is maximized.

4. Exploiting nonlinearity and non-Gaussianity for structure learning

Two popular approaches have emerged recently which rely on nonlinear and non-Gaussian characteristics of the observed data distribution to learn structure. The *LiNGAM* family of algorithms require linear dependencies with non-Gaussian noise and use Independent Components Analysis (ICA) (Hyvärinen and Oja, 2000) to learn structure. The *additive noise model* approach requires either nonlinearity or non-Gaussianity to learn structure using nonparametric regression and independence methods. Both methods can be used to identify a unique DAG rather than only its Markov equivalence class. We describe each below.

4.1 LiNGAM

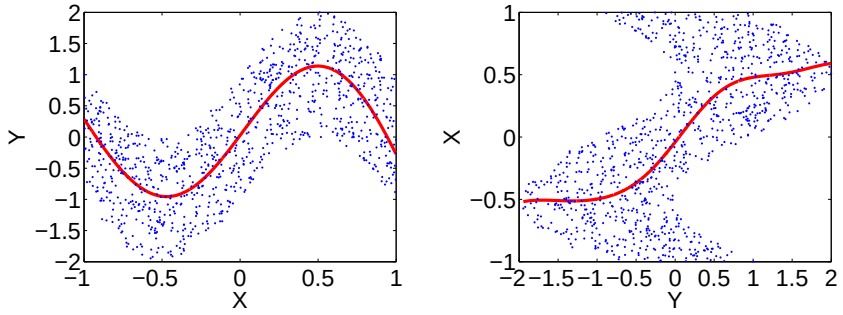
In the LiNGAM (Shimizu et al., 2006) framework, each variable is assumed to linear function of its parents in the correct DAG plus non-Gaussian noise. For instance, for $V_1, \ldots, V_n$, if V_2 and V_3 are the only parents of V_1, then $V_1 = a + bV_2 + cV_3 + \epsilon_1$ where a and b and c are real valued and ϵ_1 represents the non-Gaussian disturbance term for V_1. If the data is preprocessed by subtracting out the mean of each variable, we get the following system of equations, where $\mathbf{V} = \langle V_1, \ldots, V_n \rangle$, $\mathbf{e} = \langle \epsilon_1, \ldots, \epsilon_n \rangle$, and $\mathbf{B}$ is a matrix of coefficients (setting coefficients to zero for variables which are not parents):

$$\mathbf{V} = \mathbf{BV} + \mathbf{e}$$

Solving for $\mathbf{V}$, we can obtain the following matrix $\mathbf{A}$:

$$\mathbf{V} = \mathbf{BV} + \mathbf{e} = (\mathbf{I} - \mathbf{B})^{-1}\mathbf{e} = \mathbf{Ae}$$

The standard ICA procedure can be used to estimate this $\mathbf{A}$ matrix, i.e. the *mixing matrix*, subject to a reordering of the independent components, i.e. the noise terms, provided at least $n - 1$ of the noise terms are non-Gaussian. The LiNGAM discovery algorithm shows how the correct reordering may be obtained from the ICA estimate of $\mathbf{A}$, using the insight that since the correct structure is assumed to be a DAG, $\mathbf{A}$ must be lower triangular after some permutation of its rows and columns. After this reordering is obtained, additional statistical tests are applied to prune coefficients close to zero. The correct DAG is then obtained by noticing that non-zero coefficients correspond to parents of nodes.



Forward model Backward model

Figure 1: Forward and backward model regression estimates for nonlinear non-Gaussian example

The original LiNGAM model has been extended to cases where latent variables may be present (Hoyer et al., 2008a) and where directed cycles may be present (Lacerda et al., 2008).

4.2 Additive noise model

In the additive noise model approach to structure learning (Hoyer et al., 2009), we assume each variable V_i can be written as a (possibly nonlinear) smooth function $f(\cdot)$ of its parents in the correct DAG $\mathcal{G}$ plus an additive noise term ϵ_i with an arbitrary distribution:

$$V_i = f\left(\mathbf{Pa}_{\mathcal{G}}^{V_i}\right) + \epsilon_i$$

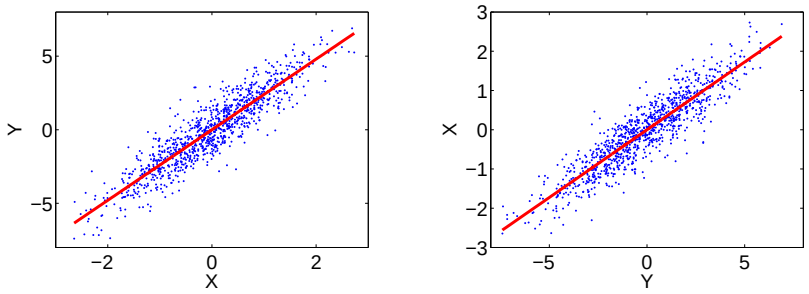
We also assume that the additive noise components are mutually independent:

$$\mathbb{P}(\epsilon_1, \ldots, \epsilon_n) = \prod_{i=1}^{n} \mathbb{P}(\epsilon_i)$$

Hoyer et al. (2009) proposed this approach originally for the two variables. Assume $X \rightarrow Y$ is the correct structure $\mathcal{G}$ and we have the following functional forms:

$$\begin{aligned} X &= \epsilon_X & \epsilon_X &\sim \mathit{Unif}(-1, 1) \\ Y &= \sin(\pi X) + \epsilon_Y & \epsilon_Y &\sim \mathit{Unif}(-1, 1) \end{aligned}$$

We will refer to $Y = f(X) + \epsilon_Y$ as the *forward model* and $X = g(Y) + \epsilon_X$ as the backward model. We can estimate the forward and backward models by nonparametrically regressing Y on X and X on Y, respectively. Figure 1 shows the resulting regressions for the above nonlinear non-Gaussian example. If we use the resulting regressions to estimate the residuals, we find that $\hat{\epsilon}_Y \perp\!\!\!\perp X$, but $\hat{\epsilon}_X \not\perp\!\!\!\perp Y$, which is clear from figure 1. This provides a means for distinguishing when $X \rightarrow Y$ is the



Forward model Backward model

Figure 2: Forward and backward model regression estimates for linear Gaussian example

correct structure from when $Y \to X$ is the correct structure using observational data: we simply do the nonparametric regressions for both directions and then if we find independence for one direction and dependence for the other direction, we accept the direction where we find independence. However, consider the following case where $f(\cdot)$ is linear and the noise components are both Gaussian:

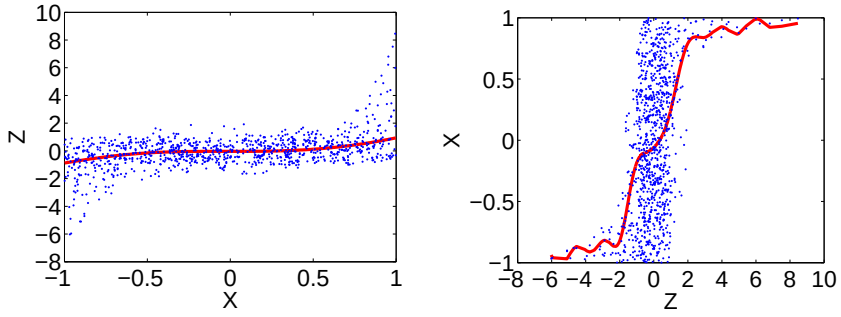
$$\begin{aligned} X &= \epsilon_X & \epsilon_X &\sim \mathcal{N}(0,1) \\ Y &= 2.4X + \epsilon_Y & \epsilon_Y &\sim \mathcal{N}(0,1) \end{aligned}$$

Figure 2 shows the regression estimates for the forward and backward models for the above linear Gaussian example. In this case, we find $\hat{\epsilon}_Y \perp\!\!\!\perp X$ and $\hat{\epsilon}_X \perp\!\!\!\perp Y$ so we have no means for distinguishing whether $X \to Y$ or $Y \to X$ is the correct structure. When this happens, we say that the additive noise model is *invertible*. Hoyer et al. (2009); Zhang and Hyvärinen (2009a) show, however, that in the two variable case, the additive noise model is not invertible whenever $f(\cdot)$ is nonlinear or $f(\cdot)$ is linear and the noise components' densities are non-Gaussian (plus a few special cases [see Zhang and Hyvärinen (2009a) for details]).

Another identifiability limitation of this approach is that it is not closed under marginalization of intermediary variables when $f(\cdot)$ is nonlinear. For example, assume $X \to Y \to Z$ is the correct structure, but we observe only X and Z. Then we should expect to learn $X \to Z$. Consider the following case:

$$\begin{aligned} X &= \epsilon_X & \epsilon_X &\sim Unif(-1,1) \\ Y &= X^3 + \epsilon_Y & \epsilon_Y &\sim Unif(-1,1) \\ Z &= Y^3 + \epsilon_Z & \epsilon_Z &\sim Unif(0,1) \end{aligned}$$

Figure 3 shows the regression estimates for the forward and backward models with X and Z for the above example. In this case, we observe $\hat{\epsilon}_X \not\perp\!\!\!\perp Z$ and $\hat{\epsilon}_Z \not\perp\!\!\!\perp X$ so we



Forward model Backward model

Figure 3: Forward and backward model regression estimates for linear Gaussian example

cannot distinguish $X \rightarrow Z$ from $Z \rightarrow X$.

Zhang and Hyvärinen (2009a) shows the additive noise model approach can be generalized to more than two variables. To test whether a DAG is compatible with the data, we regress each variable on its parents and test whether the resulting residuals are mutually independent. Using this method naively to identify the correct DAG is impractical, however, even for a few variables, since we must check all possible DAGs, and the set of all possible DAGs grows superexponentially with the number of variables, e.g. there are $\approx 4.2 \times 10^{18}$ DAGs with 10 nodes.

Since we do not assume linear dependencies nor Gaussian noise in this framework, a sufficiently powerful nonparametric independence test must be used to test independence of the residuals. Typically, the Hilbert Schmidt Independence Criterion (Gretton et al., 2008) is used, which we describe in the next section.

5. Hilbert Schmidt Independence Criterion

The Hilbert Schmidt Independence Criterion (HSIC) (Gretton et al., 2008) is a powerful and computationally efficient kernel-based nonparametric test for independence of two random variables X and Y that is used with the additive noise model approach for structure learning. Before describing this test, we review reproducing kernel Hilbert spaces.

Let $k(\cdot, \cdot)$ be a kernel mapping $\mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}$ and let $\mathcal{H}_{\mathcal{A}}$ be a Hilbert space of functions from $\mathcal{A}$ to $\mathbb{R}$. $\mathcal{H}_{\mathcal{A}}$ is a *reproducing kernel Hilbert space* (RKHS) for which $k(\cdot, \cdot)$ is the *reproducing kernel* if the following holds for all $f(\cdot) \in \mathcal{H}_{\mathcal{A}}$ and $a \in \mathcal{A}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}_{\mathcal{A}}}$ denotes the inner product in $\mathcal{H}_A$:

$$\langle f(\cdot), k(a, \cdot) \rangle_{\mathcal{H}_{\mathcal{A}}} = f(a)$$

We may treat $k(a, \cdot)$ as a mapping of a to the feature space $\mathcal{H}_{\mathcal{A}}$. Then, for $a, a' \in \mathcal{A}$, we can computer inner products $\langle k(a, \cdot), k(a', \cdot)\rangle_{\mathcal{H}_{\mathcal{A}}}$ efficiently in this high dimensional space by simply evaluating $k(a, a')$. This property of RKHSs is often referred to as the *reproducing property*. Now assume we choose a kernel that is symmetric positive definite, e.g. for $a_1, \ldots a_n \in \mathcal{A}$ and $r_1, \ldots, r_n \in \mathbb{R}$, the following holds:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} r_i r_j k(a_i, a_j) \geq 0$$

Most popular kernels are in fact symmetric positive definite, e.g. Gaussian, polynomial. The Moore-Aronszajn theorem shows that all such kernels are reproducing kernels that uniquely define corresponding RKHSs (Aronszajn, 1950). Thus, in order to take advantage of the reproducing property, we need only ensure that we choose a symmetric positive definite kernel.

Let the random variables X and Y have domains, specifically, separable metric spaces, $\mathcal{X}$ and $\mathcal{Y}$, respectively. Let $k(\cdot, \cdot)$ be a symmetric positive definite kernel mapping $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $l(\cdot, \cdot)$ a symmetric positive definite kernel mapping $\mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. Then $k(\cdot, \cdot)$ and $l(\cdot, \cdot)$ are the reproducing kernels for RHKSs $\mathcal{H}_{\mathcal{X}}$ and $\mathcal{H}_{\mathcal{Y}}$. We define the *mean map* μ_X for X and its empirical estimator, an embedding which maps *distributions* (as opposed to points) to elements in RHKSs, as follows:

$$\mu_X = \mathbb{E}_X[k(x, \cdot)] \qquad \hat{\mu}_X = \frac{1}{m}\sum_{i=1}^{m} k(x_i, \cdot)$$

Similarly, we have μ_Y and $\hat{\mu}_Y$ for Y. If $k(\cdot, \cdot)$ is *characteristic*, e.g. Gaussian and Laplace kernels, then the mean map is injective (Gretton et al., 2007; Fukumizu et al., 2008; Sriperumbudur et al., 2008) so distinct probability distributions are mapped to distinct elements in $\mathcal{H}_{\mathcal{X}}$ and $\mathcal{H}_{\mathcal{Y}}$. We also define a second mapping, the *cross covariance* $\mathcal{C}_{XY}$ for X and Y as follows, using $\otimes$ to denote the tensor product:

$$\mathcal{C}_{XY} = ([k(x, \cdot) - \mu_X] \otimes [l(y, \cdot) - \mu_Y])$$

The HSIC measure of the dependence of X and Y is simply the squared Hilbert-Schmidt norm of this operator:

$$\mathbb{H}_{XY} = \|\mathcal{C}_{XY}\|^2_{HS}$$

Gretton et al. (2008) shows that $\mathbb{H}_{XY} = 0$ if and only if $X \perp\!\!\!\perp Y$ when $k(\cdot, \cdot)$ and $l(\cdot, \cdot)$ are characteristic kernels. For m paired i.i.d. samples for X and Y, we estimate $\mathbb{H}_{XY}$ by first constructing $m \times m$ *Gram matrices* K and L for X and Y, respectively, i.e. for $1 \leq i \leq m$ and $1 \leq j \leq m$, $k_{ij} = k(x_i, x_j)$ and $l_{ij} = l(y_i, y_j)$. We then form the *centered Gram matrices* $\tilde{K}$ and $\tilde{L}$ as follows, where I_m is the $m \times m$ identity matrix, and $\mathbf{1}_m$ is a column vector of m ones:

$$H = I_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^\top \qquad \tilde{K} = HKH \qquad \tilde{L} = HLH$$

The estimator for $\mathbb{H}_{XY}$ is computed simply by finding the trace, denoted $tr(\cdot)$, of the product of centered Gram matrices (Gretton et al., 2008):

$$\hat{\mathbb{H}}_{XY} = \frac{1}{m^2} tr\left(\tilde{K}\tilde{L}\right)$$

We have two approaches for determining the significance threshold for a level-α statistical test using HSIC. The first is the permutation approach, shown as algorithm 3, where we compute $\hat{\mathbb{H}}_{XY}$ for multiple random assignments (p) of the Y samples to X and use the $1-\alpha$ quantile of the resulting empirical distribution over $\hat{\mathbb{H}}_{XY}$. The second approach is to estimate the threshold using a Gamma distribution. Gretton et al. (2008) shows that $m\hat{\mathbb{H}}_{XY} \sim Gamma(\alpha, \beta)$, where α and β are computed using the first two moments of the estimator $\hat{\mathbb{H}}_{XY}$ under the null distribution (for details see Gretton et al. (2008)).

Input : Paired i.i.d. samples $(x_1, y_1), \ldots, (x_m, y_m)$, $\tilde{K}$, α
Output: Significance threshold t

for $i = 1, \ldots, p$ **do**
  Let $y'_1, \ldots, y'_m$ be a random permutation of $y_1, \ldots, y_m$
  Construct the centered Gram matrix $\tilde{L}'$ for $y'_1, \ldots, y'_m$
  $\hat{\mathbb{H}}^{(i)}_{XY} \leftarrow \frac{1}{m^2} tr\left(\tilde{K}\tilde{L}'\right)$
end
$t \leftarrow 1-\alpha$ quantile of the empirical distribution of $\hat{\mathbb{H}}^{(1)}_{XY}, \ldots \hat{\mathbb{H}}^{(p)}_{XY}$

Algorithm 3: Permutation approach for determining significance threshold

6. Weakly additive noise models

In this section, we extend the additive noise model framework to account for cases where additive noise models are invertible as well as most cases where noise is not additive. This will allow us to develop a new search algorithm (section 7) which addresses the two major limitations of traditional constraint-based and score-based structure learning methods as well as the two major limitations of the additive noise model discussed in section 4.2.

First we define a new class of models, *weakly additive noise models*.

Definition 1 *Let $\mathcal{P}$ be a probability distribution over $\mathcal{V}$ that is Markov to a DAG $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$. Then, $\psi = \left\langle V_i, \mathbf{Pa}^{V_i}_{\mathcal{G}} \right\rangle$ is a* local additive noise model *for $\mathcal{P}$ contained in $\mathcal{G}$ if the functional form of V_i can be expressed as $V_i = f\left(\mathbf{Pa}^{V_i}_{\mathcal{G}}\right) + \epsilon$, where*

1. *$f(\cdot)$ is an arbitrary (possibly nonlinear) smooth function*

2. ϵ *has an arbitrary density*

3. $\epsilon \perp\!\!\!\perp \mathbf{Pa}_{\mathcal{G}}^{V_i}$

Definition 2 *Let $\mathcal{P}$ be a probability distribution over $\mathcal{V}$, $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ a DAG, and Ψ a set of local additive noise models. Then, $\mathcal{M} = \langle \mathcal{G}, \Psi \rangle$ is a* weakly additive noise model *for $\mathcal{P}$ if*

1. *$\mathcal{P}$ is Markov to $\mathcal{G}$*

2. *$\psi \in \Psi$ if and only if ψ is a local additive noise model for $\mathcal{P}$ contained in $\mathcal{G}$*

3. *$\forall \left\langle V_i, \mathbf{Pa}_{\mathcal{G}}^{V_i} \right\rangle \in \Psi$, there does not exist a $V_j \in \mathbf{Pa}_{\mathcal{G}}^{V_i}$ and an arbitrary directed graph $\mathcal{G}'$ (not necessarily related to $\mathcal{P}$) such that $V_i \in \mathbf{Pa}_{\mathcal{G}'}^{V_j}$ and $\left\langle V_j, \mathbf{Pa}_{\mathcal{G}'}^{V_j} \right\rangle$ is a local additive noise model for $\mathcal{P}$ contained in $\mathcal{G}'$*

When we assume a data generating process has a weakly additive noise model representation, we assume only that there are no cases where $X \to Y$ is contained in the true structure and there exists a functional form for X and Y such that we can express $X = f(Y) + \epsilon_X$ with $\epsilon_X \perp\!\!\!\perp Y$, but no such form such that we can express $Y = f(X) + \epsilon_Y$ with $\epsilon_Y \perp\!\!\!\perp X$. In other words, the data cannot appear as though it admits an additive noise model representation, but only in the incorrect direction. This representation is still appropriate when additive noise models are invertible, and when noise is not additive: such cases only lead to weakly additive noise models which express greater underdetermination of the true data generating process. Even in such cases, however, this representation will be *at least* as rich as the PDAG representation, since we can always infer Markov properties regardless of the function forms of the variables in the true data generating process. Furthermore, the combination of Markov properties and certain local additive noise models may entail further characteristics of the data generating process so for many cases this representation may be as rich as the additive noise model representation is in ideal cases even if only a few local additive noise models are identifiable.

We now define the notion of distribution-equivalence for weakly additive noise models.

Definition 3 *A weakly additive noise model $\mathcal{M} = \langle \mathcal{G}, \Psi \rangle$ is* distribution-equivalent *to $\mathcal{N} = \langle \mathcal{G}', \Psi' \rangle$ if and only if*

1. *$\mathcal{G}$ and $\mathcal{G}'$ are Markov equivalent*

2. *$\psi \in \Psi$ if and only if $\psi \in \Psi'$*

Distribution-equivalence for weakly additive noise models defines what can be discovered about the true data generating mechanism using observational data. We now define a new structure to partition the structural representations of data generating processes which instantiate distribution-equivalent weakly additive noise models.

Definition 4 *A* weakly additive noise partially directed acyclic graph *(WAN-PDAG) for* $\mathcal{M} = \langle \mathcal{G}, \Psi \rangle$ *is a mixed graph* $\mathcal{H} = \langle \mathcal{V}, \mathcal{E} \rangle$ *such that for* $\{V_i, V_j\} \subseteq \mathcal{V}$,

1. $V_i \rightarrow V_j$ *is a directed edge in* $\mathcal{H}$ *if and only if* $V_i \rightarrow V_j$ *is a directed edge in* $\mathcal{G}$ *and in all* $\mathcal{G}'$ *such that* $\mathcal{N} = \langle \mathcal{G}', \Psi' \rangle$ *is distribution-equivalent to* $\mathcal{M}$
2. $V_i - V_j$ *is an undirected edge in* $\mathcal{H}$ *if and only if* $V_i \rightarrow V_j$ *is a directed edge in* $\mathcal{G}$ *and there exists a* $\mathcal{G}'$ *and* $\mathcal{N} = \langle \mathcal{G}', \Psi' \rangle$ *distribution-equivalent to* $\mathcal{M}$ *such that* $V_i \leftarrow V_j$ *is a directed edge in* $\mathcal{G}'$

We now get the following results.

Lemma 5 *Let* $\mathcal{M} = \langle \mathcal{G}, \Psi \rangle$ *and* $\mathcal{N} = \langle \mathcal{G}', \Psi' \rangle$ *be weakly additive noise models. If* $\mathcal{M}$ *and* $\mathcal{N}$ *are distribution equivalent, then* $\forall \left\langle V_i, \mathbf{Pa}_{\mathcal{G}}^{V_i} \right\rangle \in \Psi$, $\mathbf{Pa}_{\mathcal{G}}^{V_i} = \mathbf{Pa}_{\mathcal{G}'}^{V_i}$ *and* $\mathbf{Ch}_{\mathcal{G}}^{V_i} = \mathbf{Ch}_{\mathcal{G}'}^{V_i}$.

Proof Since $\mathcal{M}$ and $\mathcal{N}$ are distribution-equivalent, $\mathbf{Pa}_{\mathcal{G}}^{V_i} = \mathbf{Pa}_{\mathcal{G}'}^{V_i}$ since $\left\langle V_i, \mathbf{Pa}_{\mathcal{G}}^{V_i} \right\rangle$ is a local additive noise model for $\mathcal{M}$ and $\mathcal{N}$. Now since $\mathcal{G}$ and $\mathcal{G}'$ Markov equivalent, $\mathcal{G}$ and $\mathcal{G}'$ have the same adjacencies. Thus, $\forall V_j \in \mathbf{Ch}_{\mathcal{G}}^{V_i}$, since V_j is adjacent to V_i in $\mathcal{G}'$ and $V_j \notin \mathbf{Pa}_{\mathcal{G}'}^{V_i}$, it must be the case that $V_j \in \mathbf{Ch}_{\mathcal{G}'}^{V_i}$. ■

Before stating the next theorem, we must discuss the *extended Meek rules*. Meek (1995) gives an additional rule to use with the Meek rules whenever "background knowledge" about the true structure is available, i.e. any information other than adjacencies and immoralities:

- For $\{V_i, V_j, V_k, V_l\} \subseteq \mathcal{V}$ such that in $\mathcal{G}$, $V_i \rightarrow V_j$ and $V_j \rightarrow V_k$ are directed edges, V_k and V_l are adjacent, V_l and V_i are adjacent, and V_j and V_l are adjacent, make $V_l \rightarrow V_j$ a directed edge in $\mathcal{G}$

We will use the *extended Meek rules* to refer to the Meek rules with this additional rule. Meek (1995) proves that the extended Meek rules are correct and complete subject to any additional background knowledge.

Theorem 6 *The WAN-PDAG for* $\mathcal{M} = \langle \mathcal{G}, \Psi \rangle$ *is constructed by*

1. *adding all directed and undirected edges in the PDAG instantiated by* $\mathcal{M}$
2. $\forall \left\langle V_i, \boldsymbol{Pa}_{\mathcal{G}}^{V_i} \right\rangle \in \Psi$, *directing all* $V_j \in \boldsymbol{Pa}_{\mathcal{G}}^{V_i}$ *as* $V_j \rightarrow V_i$ *and all* $V_k \in \boldsymbol{Ch}_{\mathcal{G}}^{V_i}$ *as* $V_i \rightarrow V_k$
3. *applying the extended Meek rules, treating orientations made using* Ψ *as background knowledge*

Proof 1 is correct since all structures that are Markov equivalent to $\mathcal{G}$ will have the same adjacencies and directed edges as $\mathcal{G}$. 2 is correct by lemma 5. 3 is correct by the correctness and completeness results given in Meek (1995). ■

WAN-PDAGs can used to identify the same information about the data generating mechanism as additive noise models, when additive noise models are identifiable, but provide a more powerful representation of uncertainty and can be used to discover more information when additive noise models are unidentifiable. In the next section, we describe an efficient algorithm for learning WAN-PDAGs from data.

7. The Kernel PC (kPC) algorithm

We now describe the Kernel PC (kPC) algorithm[1] for learning WAN-PDAGs from data. kPC consists of two stages. The first stage simply involves using the PC algorithm with a nonparametric conditional independence, rather than the Fisher Z-transformation test (since we do not want to assume linear dependences or Gaussian noise), to learn a PDAG. We use HSIC whenever unconditional independence needs to be tested and an extension of HSIC to the conditional case, discusses in section 8, whenever conditional independence needs to be tested. The remainder of this section discusses the second stage, which is a "PC-style" search for noninvertible local additive noise models using the PDAG resulting from the first stage.

The motivation for searching for noninvertible local additive noise models in the PDAG rather attempting to fit an additive noise model for the entire structure (other than consistency with the framework in section 6) comes from noticing that (i) it may be more efficient to do so and require fewer tests since orientations implied by a noninvertible local additive noise model may imply further orientations and (ii) we may find more total orientations by considering local additive noise models, e.g. if all relations are linear and only one variable has a non-Gaussian noise term. The basic strategy used to search for noninvertible local additive noise models is a "PC-style" greedy search where we look for undirected edges in the current mixed graph (starting with the PDAG) that are adjacent to the fewest other undirected edges. If these edges can be oriented by discovering a noninvertible local additive noise model, then we make these orientations, apply the extended Meek rules using these orientations as background knowledge, and continue iterating until no more edges can be oriented. Algorithm 4 provides pseudocode for this second stage. Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ be the resulting PDAG and $\forall V_i \in \mathcal{V}$, let $\mathbf{U}_{\mathcal{G}}^{V_i}$ denote the nodes connected to V_i in $\mathcal{G}$ by an undirected edge at the current iteration of the algorithm. We get the following results.

Lemma 7 *If an edge is oriented in the second stage of kPC, it is implied by a non-invertible local additive noise model.*

1. MATLAB code may be obtained from http://www.andrew.cmu.edu/∼rtillman/kpc

Input : PDAG $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$
Output: WAN-PDAG $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$

$s \leftarrow 1$
while $\max_{V_i \in \mathcal{V}} \left| \boldsymbol{U}_{\mathcal{G}}^{V_i} \right| \geq s$ **do**
  foreach $V_i \in \mathcal{V}$ *such that* $\left| \boldsymbol{U}_{\mathcal{G}}^{V_i} \right| = s$ *or* $\left| \boldsymbol{U}_{\mathcal{G}}^{V_i} \right| < s$ *and* $\boldsymbol{U}_{\mathcal{G}}^{V_i}$ *was updated* **do**
    $s' \leftarrow s$
    while $s' > 0$ **do**
      foreach $\boldsymbol{S} \subseteq \boldsymbol{U}_{\mathcal{G}}^{V_i}$ *such that* $|\boldsymbol{S}| = s'$ *and* $\forall S_k \in \boldsymbol{S}$, *orienting* $S_k \rightarrow V_i$, *does not create an immorality* **do**
        Nonparametrically regress V_i on $\mathbf{Pa}_{\mathcal{G}}^{V_i} \cup \mathbf{S}$ and compute the residual $\hat{\epsilon}_{i\mathbf{S}}$
        if $\hat{\epsilon}_{i\boldsymbol{S}} \perp\!\!\!\perp \boldsymbol{S}$ *and* $\nexists V_j \in \boldsymbol{S}$ *and* $\boldsymbol{S'} \subseteq \boldsymbol{U}_{\mathcal{G}}^{V_j}$ *such that. regressing* V_j *on* $\boldsymbol{Pa}_{V_j}^{\mathcal{G}} \cup \boldsymbol{S'} \cup V_i$ *results in the residual* $\hat{\epsilon}_{j\boldsymbol{S'} \cup \{V_i\}} \perp\!\!\!\perp \boldsymbol{S'} \cup \{V_i\}$ **then**
          $\forall S_k \in \mathbf{S}$, orient $S_k \rightarrow V_i$, and $\forall U_l \in \mathbf{U}_{\mathcal{G}}^{V_i} \backslash \mathbf{S}$ orient $V_i \rightarrow U_l$
          Apply the extended Meek rules
          $\forall V_m \in \mathcal{V}$, update $\mathbf{U}_{\mathcal{G}}^{V_m}$, set $s' = 1$, and break
        end
      end
      $s' \leftarrow s' - 1$;
    end
  end
  $s \leftarrow s + 1$
end

Algorithm 4: Second Stage of kPC

Proof If the condition at line 8 of kPC is true then $\left\langle V_i, \mathbf{Pa}_{\mathcal{G}}^{V_i} \cup \mathbf{S} \right\rangle$ is a noninvertible local additive noise model. All $U_l \in \mathbf{U}_{\mathcal{G}}^{V_i} \backslash \mathbf{S}$ must be children of V_i by lemma 5 and the extended Meek rules are correct by Meek (1995). Thus, all orientations made at lines 9-10 are correct.. ■

Lemma 8 *Suppose $\psi = \langle V_i, \mathbf{W} \rangle$ is a noninvertible local additive noise model. Then kPC will make all orientations implied by ψ.*

Proof Let $\tilde{\mathbf{S}} = \mathbf{W} \backslash \mathbf{Pa}_{V_i}^{\mathcal{G}}$ for $\mathbf{Pa}_{V_i}^{\mathcal{G}}$ at the current iteration. kPC must terminate with $s > |\tilde{\mathbf{S}}|$ since $|\tilde{\mathbf{S}}| \leq |\mathbf{U}_{\mathcal{G}}^{V_i}|$ so $\mathbf{S} = \tilde{\mathbf{S}}$ at some iteration. Since $\left\langle V_i, \mathbf{Pa}_{\mathcal{G}}^{V_i} \cup \tilde{\mathbf{S}} \right\rangle$ is a noninvertible local additive noise model, line 8 is satisfied so all edges connected to V_i are oriented. ■

Theorem 9 *Assume data is generated according to some weakly additive noise model $\mathcal{M} = \langle \mathcal{G}, \Psi \rangle$. Then kPC will return the WAN-PDAG instantiated by $\mathcal{M}$ assuming perfect conditional independence information, Markov, faithfulness, and causal sufficiency.*

Proof The PC algorithm is correct and complete with respect to the PDAG learned (Spirtes et al., 2000). Orientations made due to the discovery of noninvertible local additive noise models are correct by lemma 7 and all such orientations that can be made are made by lemma 8. Since the Meek rules, which are correct and complete by Meek (1995), are invoked after each orientation made due to the discovery of some noninvertible local additive noise model, they are invoked after all such orientations are made. ■

8. Kernel-based conditional independence

We now define a kernel-based conditional dependence measure similar to HSIC (Fukumizu et al., 2008) (see also (Sun et al., 2007, Section 2.2) for a related quantity with a different normalization) that we use in the first stage of kPC for conditional independence testing.

Let $X, Y, \mathcal{X}, \mathcal{Y}, k(\cdot,\cdot), l(\cdot,\cdot), \mathcal{H}_{\mathcal{X}}, \mathcal{H}_{\mathcal{Y}}, \tilde{K}$, and $\tilde{L}$ be defined as in section 5. Let Z be a conditioning variable with domain $\mathcal{Z}$, $m(\cdot,\cdot)$ be a reproducing kernel which defines RKHS $\mathcal{H}_{\mathcal{Z}}$, and $\tilde{M}$ be the corresponding centered Gram matrix for an i.i.d. sample $z_1, \ldots, z_m$ (paired with $(x_1, y_1), \ldots, (x_m, y_m)$). Let $\ddot{X} = (X, Z)$ and $\ddot{Y} = (Y, Z)$ be the *extended* random variables for X and Y. The *conditional cross covariance* (Fukumizu et al., 2008) $\mathcal{C}_{XY|Z}$ of X and Y given Z is defined as follows:

$$\mathcal{C}_{XY|Z} = \mathcal{C}_{\ddot{X}Z} \mathcal{C}_{ZZ}^{-1} \mathcal{C}_{Z\ddot{Y}}$$

Just as HSIC measures the dependence of X and Y by taking the squared Hilbert Schmidt norm of $\mathcal{C}_{XY}$, we measure the conditional dependence of X and Y given Z by taking the squared Hilbert Schmidt norm of $\mathcal{C}_{XY|Z}$:

$$\mathbb{H}_{XY|Z} = \|\mathcal{C}_{XY|Z}\|^2_{HS}$$

It follows from (Fukumizu et al., 2008, Theorem 3) that if $k(\cdot,\cdot)$, $l(\cdot,\cdot)$, and $m(\cdot,\cdot)$ are characteristic kernels, then $\mathbb{H}_{XY|Z} = 0$ if and only if $X \perp\!\!\!\perp Y|Z$. Fukumizu et al. (2008) provides the empirical estimator, where ϵ is a regularizer:

$$\hat{\mathbb{H}}_{XY|Z} = \frac{1}{m^2} tr(\tilde{K}\tilde{L} - 2\tilde{K}\tilde{M}(\tilde{M}+\epsilon I_m)^{-2}\tilde{M}\tilde{L} + \tilde{K}\tilde{M}(\tilde{M}+\epsilon I_m)^{-2}\tilde{M}\tilde{L}\tilde{M}(\tilde{M}+\epsilon I_m)^{-2}\tilde{M})$$

This estimator is quite costly to compute, i.e. $\mathcal{O}(m^3)$ naively, especially if we need to compute it many times to determine a significance threshold as in the permutation approach. Fortunately, we see from (Bach and Jordan, 2002, Appendix C) that the eigenspectra of Gram matrices for Gaussian kernels decay very rapidly, so low rank approximations of these matrices can be obtained even when using a very conservative threshold. We can use the incomplete Cholesky factorization (Fine and Scheinberg, 2001) procedure to obtain such an approximation. For an $m \times m$ Gram matrix Q, this factorization results in an $m \times p$ matrix G, where $p \ll m$, and an $m \times m$ permutation matrix P such that $Q \approx PGG^\top P^\top$. If we simply replace the centered Gram matrices in $\hat{\mathbb{H}}_{XY|Z}$ with their incomplete Cholesky factorizations and use the following equivalence to invert $G^\top G + \epsilon I_p$ for each $m \times p$ matrix G in the resulting factorizations instead of $GG^\top + \epsilon I_m$, we can obtain a $\mathcal{O}(mp^3)$ procedure if implemented carefully.

$$\left(GG^\top + \epsilon I_m\right)^{-1} = \frac{1}{\epsilon} I_m - \frac{1}{\epsilon} G\left(\epsilon I_m + G^\top G\right)^{-1} G^\top$$

Unfortunately, the above procedure is not numerically stable unless a relatively large regularizer ϵ is chosen or only a small number of columns are used in the incomplete Cholesky factorizations.

A more stable (and faster) approach is to obtain incomplete Cholesky factorizations G_X, G_Y, and G_Z with permutation matrices P_X, P_Y, and P_Z, and then obtain the *thin* SVDs for HP_XG_X, HP_YG_Y, and HP_ZG_Z. The thin SVD for a matrix A consists of matrices U, S, and V, such that U is $m \times p$, S is a $p \times p$ diagonal matrix of singular values, V is $p \times p$, and $A = USV$. For $\tilde{K}$, $\tilde{L}$, and $\tilde{M}$, let $U^X S^X V^X$, $U^Y S^Y V^Y$, and $U^Z S^Z V^Z$ be the corresponding thin SVDs, respectively. Now define matrices $\bar{S}^X, \bar{S}^Y$, and $\bar{S}^Z$ and $\bar{G}^X, \bar{G}^Y$, and $\bar{G}^Z$ as follows:

$$\bar{s}^X_{ii} = \left(s^X_{ii}\right)^2 \quad \bar{s}^Y_{ii} = \left(s^Y_{ii}\right)^2 \quad \bar{s}^Z_{ii} = \frac{\left(s^Z_{ii}\right)^2}{\left(s^Z_{ii}\right)^2 + \epsilon}$$

$$\bar{G}^X = U^X \bar{S}^X U^{X\top} \quad \bar{G}^Y = U^Y \bar{S}^Y U^{Y\top} \quad \bar{G}^Z = U^Z \bar{S}^Z U^{Z\top}$$

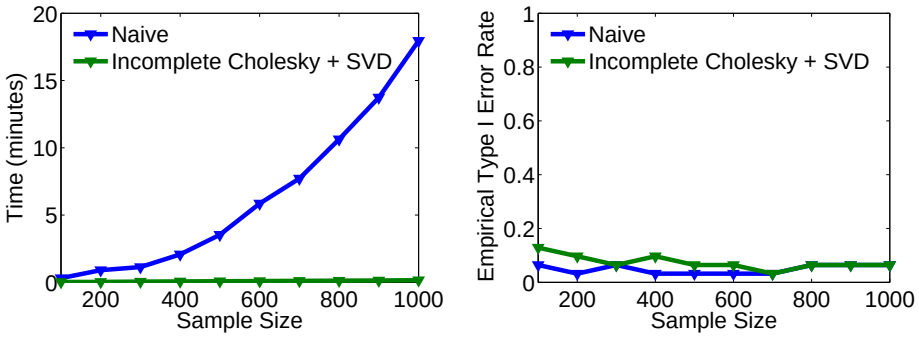


Figure 4: Runtime and Empirical Type I Error Rate. Results are over the generation of 20 3-node DAGs for which $X \perp\!\!\!\perp Y|Z$ and the generating distribution was linear Gaussian.

Noticing that for a matrix A with thin SVD USV, $A^2 = US^2U^\top$, we can compute $\hat{\mathbb{H}}_{XY|Z}$ as follows:

$$\hat{\mathbb{H}}_{XY|Z} = \frac{1}{m^2} tr\left(\bar{G}^X\bar{G}^Y - 2\bar{G}^X\bar{G}^Z\bar{G}^Y + \bar{G}^X\bar{G}^Z\bar{G}^Y\bar{G}^Z\right)$$

This can be computed stably and efficiently in $\mathcal{O}(mp^3)$ by simply choosing an appropriate associative ordering of matrix multiplications. Figure 4 shows that this method leads to a significant increase in speed when used with the modified permutation approach described below for conditional independence testing without significantly affecting the empirically observed type I error rate for a level-.05 test.

Computing significance thresholds for level-α statistical tests is complicated by the fact that the null distribution of $\hat{\mathbb{H}}_{XY|Z}$ is unknown and difficult to derive, and the permutation approach given in algorithm 3 is not appropriate since permuting Y while leave Z fixed changes the marginal distribution of Y given Z. One approach described in Sun (2008) is a modified permutation procedure where we (making analogy to the discrete case) first cluster Z and then only permute elements which fall within the same cluster. This approach is outlined in algorithm 5. A second possibility is to directly calculate the first and second moments of $\hat{\mathbb{H}}_{XY|Z}$ and use these with the Gamma distribution approach for finding the threshold for a level-α test using HSIC described in Gretton et al. (2008). This approach is complicated by the matrix inversions which appear in $\hat{\mathbb{H}}_{XY|Z}$. One possibility is to rewrite these terms as follows

and then use the Taylor series expansion:

$$\left(\tilde{M} + \epsilon I_m\right)^{-1} \approx \left(\frac{1}{\delta}\tilde{M} + I_m\right)^{-1}$$
$$= I_m + \sum_{i=1}^{\infty} (-1)^i \frac{1}{\delta^i}\tilde{M}^i$$

This approach can be used to derive estimates for the first and second moments of $\hat{\mathbb{H}}_{XY|Z}$, but the computational costs involved in computing these estimators makes them not useful in practice.

Input : Paired i.i.d. samples $(x_1, y_1, z_1), \ldots, (x_m, y_m, z_m)$, $\tilde{K}$, α
Output: Significance threshold t

Cluster $z_1, \ldots, z_m$ into $\mathbf{z}_{(1)}, \ldots, \mathbf{z}_{(k)}$
for $i = 1, \ldots, p$ **do**
  Let $\mathbf{y}_{(1)}, \ldots, \mathbf{y}_{(k)}$ each be random permutations of the elements in $y_1, \ldots, y_n$ matching the indices of the elements in each of $\mathbf{z}_{(1)}, \ldots, \mathbf{z}_{(k)}$
  Let $y'_1, \ldots, y'_m$ be $y_1, \ldots, y_m$ after replacing each y_i for $1 \leq i \leq m$ with the element in $\mathbf{y}_{(1)}, \ldots, \mathbf{y}_{(k)}$ corresponding to the element in $\mathbf{z}_{(1)}, \ldots, \mathbf{z}_{(k)}$ matched to the index of z_i
  Construct the centered Gram matrix $\tilde{L}'$ for $y'_1, \ldots, y'_m$
  Compute $\hat{\mathbb{H}}^{(i)}_{XY|Z}$ using $\tilde{K}, \tilde{L}'$, and $\tilde{M}$
end
$t \leftarrow 1 - \alpha$ quantile of the empirical distribution of $\hat{\mathbb{H}}^{(1)}_{XY|Z}, \ldots \hat{\mathbb{H}}^{(p)}_{XY|Z}$

Algorithm 5: Modified permutation approach for conditional independence threshold

9. Related research

kPC is similar in spirit to the PC-LiNGAM structure learning algorithm (Hoyer et al., 2008a), which assumes dependencies are linear with either Gaussian or non-Gaussian noise. PC-LiNGAM combines the PC algorithm with LiNGAM to learn structures referred to as *ngDAG*s. KCL (Sun et al., 2007) is a heuristic search for a mixed graph that uses a kernel-based dependence measure similar to kPC (while not determining significance thresholds via a hypothesis test), but does not take advantage of additive noise models. Mooij et al. (2009) provides a more efficient algorithm for learning additive noise models, by first finding a causal ordering after doing a series of high

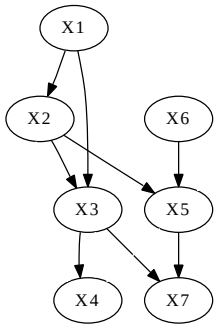


Figure 5: Toy example DAG

dimensional regressions and HSIC independence tests and then pruning the resulting DAG implied by this ordering. Finally, Zhang and Hyvärinen (2009b) proposes a two-stage procedure for learning additive noise models from data that is similar to kPC, but requires the additive noise model assumptions in the first stage where the Markov equivalence class is identified.

10. Experimental results

10.1 Toy data

To evaluate the performance of kPC, we first considered a toy example. We generated 1000 data points by forward sampling from the 7 node DAG in Figure 5. Samples were generated using the following recursive equations and noise distributions:

$$
\begin{aligned}
X_1 &= \epsilon_1 && \epsilon_1 \sim Unif(-1,1) \\
X_2 &= 6\cos(X_1) + \epsilon_2 && \epsilon_2 \sim \mathcal{N}(-1,1) \\
X_3 &= 2\sin(\pi X_1) + X_2 + \epsilon_3 && \epsilon_3 \sim \mathcal{N}(0,1) \\
X_4 &= 3\cos(X_3) + \epsilon_4 && \epsilon_4 \sim \mathcal{N}(0,1) \\
X_5 &= .05(X_2 + X_6)^3 + \epsilon_5 && \epsilon_5 \sim \mathcal{N}(0,1) \\
X_6 &= \epsilon_6 && \epsilon_6 \sim Unif(-1,1) \\
X_7 &= 6\cos(.2[X_3 + \log(6 + X_5) + 2]) + \epsilon_7 && \epsilon_7 \sim Unif(-1,1)
\end{aligned}
$$

We used kPC, PC, GES with the BIC score, and LiNGAM to learn structures from this data with nonlinear dependencies and both Gaussian and non-Gaussian noise. Figure 6 reports the results. We see that kPC learns the correct structure, PC leaves out two edges and reverses one edge, GES leaves out two edges and adds two edges, and LiNGAM leaves out two edges, adds two edges, and reverses two edges.

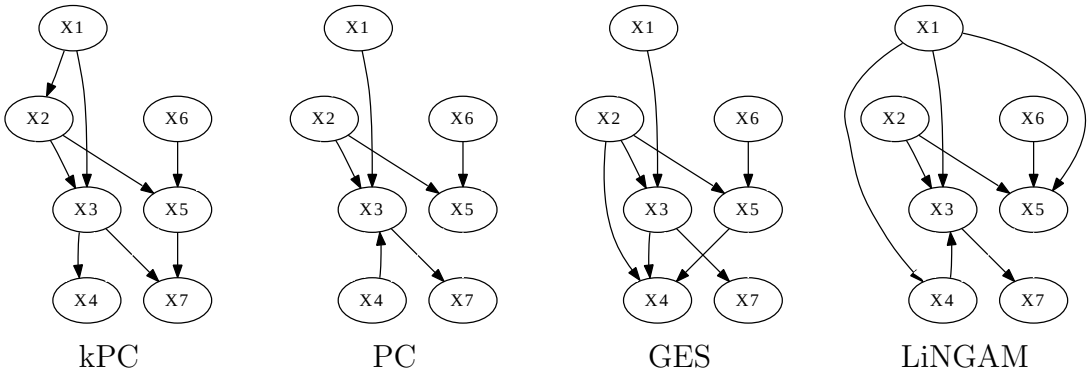


Figure 6: Structures learned by kPC, PC, GES, and LiNGAM using toy data

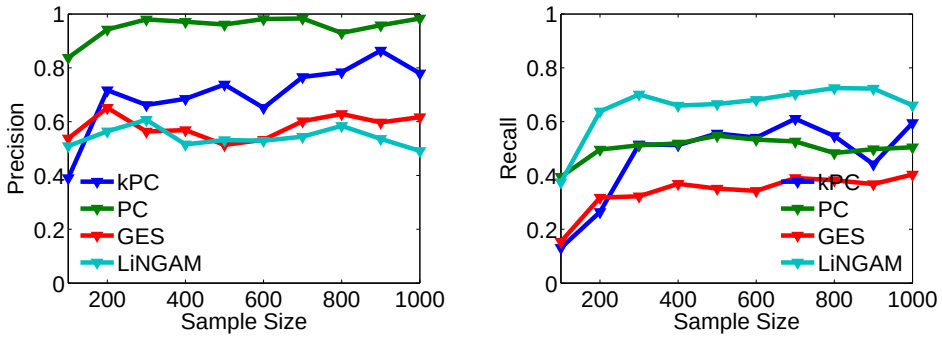
10.2 Simulation data

Next, to evaluate the performance of kPC across a range of possible structures, we generated 20 random 7-nodes DAGs using the MCMC algorithm in Melançon et al. (2000) and forward sampled 1000 data points from each DAG under three conditions: linear dependencies with Gaussian noise, linear dependencies with non-Gaussian noise, and nonlinear dependencies with non-Gaussian noise. We generated non-Gaussian noise using the same procedure as Shimizu et al. (2006) and used polynomial and trigonometric functions for nonlinear dependencies.

We again compared kPC to PC, GES, and LiNGAM. We applied two metrics in measuring performance vs. sample size: precision, i.e. proportion of directed edges in the resulting graph that are in the true DAG, and recall, i.e. proportion of directed edges in the true DAG that are in the resulting graph.

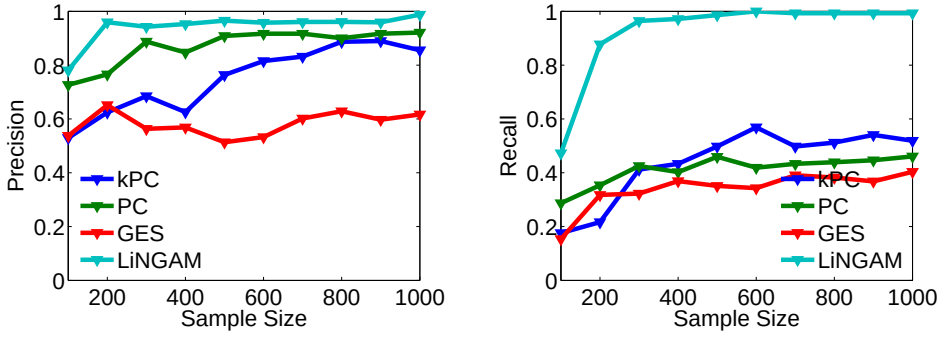
Figure 7 reports the results for the linear Gaussian case. In the linear Gaussian case, we see PC shows slightly better performance than kPC in precision though the difference becomes smaller as the sample size increases, while GES and LiNGAM perform worse than kPC. Recall is about the same for kPC and PC. These results are mostly unsurprising; since PC assumes linear Gaussian distributions whereas kPC uses a more complicated nonparametric conditional independence test, we expect PC to show greater performance for smaller sample sizes when the data are actually linear Gaussian.

Figure 8 shows results for the linear non-Gaussian case. Precision for PC and kPC is about the same, while LiNGAM shows slightly better performance and GES shows worse performance. However, LiNGAM shows significantly better recall than all of the other algorithms, which have about the same recall. These results are also unsurprising since LiNGAM assumes linear relations with non-Gaussian noise and previous simulation results have shown that nonlinearity, but not non-Gaussianity significantly affects the performance of PC (Voortman and Druzdzel, 2008).



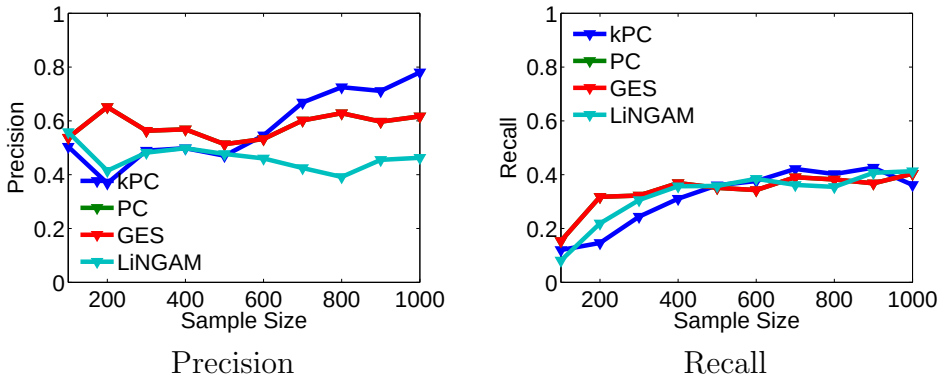
Precision Recall

Figure 7: Simulations with linear Gaussian data



Precision Recall

Figure 8: Simulations with linear non-Gaussian data



Precision Recall

Figure 9: Simulations with nonlinear non-Gaussian data

In the nonlinear non-Gaussian case, kPC performs slightly better than PC and GES (which show almost exactly the same performance) in precision, while LiNGAM performs worse. All algorithms perform about the same in terms of recall. We note, however, that in some of these cases the performance of kPC was significantly better than the other algorithms so these simulations may not provide the best comparison of kPC to the other algorithms in cases where dependencies are nonlinear. When simulating nonlinear data, we must be careful to ensure that variances do not blow up and result in data for which no finite sample method can show adequate performance. This has the unfortunate side effect that the nonlinear data generated may be well approximated using linear methods. Future research will consider more sophisticated methods for simulating data that is more appropriate when comparing kPC to linear methods.

10.3 Climate teleconnection data

Climate scientists use a number of indices, measuring atmospheric pressures, ocean surface temperatures, etc., in prediction models for future climate patterns. Associations between these indices (*teleconnections*) are well documented and various physical mechanisms have been proposed to explain these associations. Since many of these associations have been shown to be nonlinear (Chu and Glymour, 2008), we used data consisting of measurements of six of these indices recorded monthly between 1953 and 2000 (576 total measurements) to evaluate the performance of kPC on real data. We used the following indices:

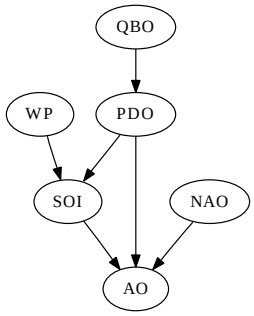


Figure 10: Structure learned by kPC

QBO **Quasi-Biennial Oscillation** - wind oscillation (easterly to westerly) in equatorial stratosphere
SOI **Southern Oscillation Index** - air pressure between Tahiti and Darwin
WP **West Pacific** - wave structure at surface of Pacific Ocean
PDO **Pacific Decadal Oscillation** - warm or cool surface temperatures in the Pacific Ocean
AO **Arctic Oscillation** - atmospheric pressure in Arctic region
NAO **North Atlantic Oscillation** - atmospheric pressure in North Atlantic region

The most support has been given to a physical mechanism between NAO and AO. NAO is thought to be an early regional predictor of future AO indices. A second teleconnection is believed to exist between QBO and PDO, where QBO winter indices are thought to predict spring PDO indices. Figure 10 shows the structure learned by kPC. We note that both of these teleconnections are consistent with the learned structure. We do not know at this time whether there are other physical mechanisms which may explain the remainder of the relationships indicated by kPC.

10.4 fMRI data

We also ran kPC on data from an fMRI experiment that is analyzed in Ramsey et al. (2009) where nonlinear dependencies can be observed. In the experiment, subjects were repeatedly asked to decide whether pairs of words rhymed over the course of a few minutes while fMRI images were obtained. Figure 11 shows the structure that kPC learned, where each of the nodes corresponds to a particular brain region where strong neuronal activity was observed during the experiment (data points represent level of activation among clusters of voxels in these areas). This structure is the same as the one learned by the score-based IMaGES algorithm, which uses additional penalties to effectively deal with observed nonlinearities, that was used in Ramsey et al. (2009) except for the absence of one edge. However, IMaGES

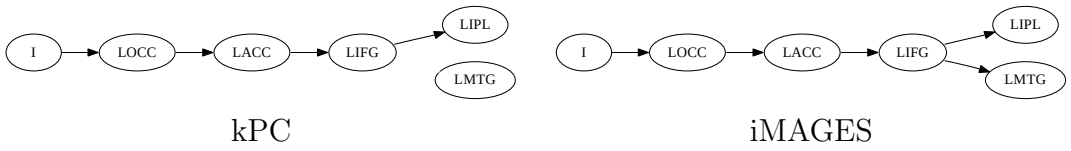


Figure 11: Structures learned by kPC and iMAGES

required background knowledge to direct the edges (IMaGES learns an undirected graph without background knowledge). kPC successfully found the same directed edges without using any background knowledge. Theses edges are consistent with neuroscience domain knowledge.

11. Conclusion

We considered the current approaches to learning directed graphical models from continuous data which relax the assumptions that (i) dependencies are linear and (ii) noise is Gaussian, while highlighting their limitations. We then introduced the weakly additive noise model framework, which extends the additive noise model framework to cases such as the linear Gaussian, where the additive noise model is invertible and thus unidentifiable, as well as cases where noise may not be additive. The weakly additive noise model framework allows us to identify a unique DAG when the additive noise model assumptions hold, and a structure that is at least as specific as a PDAG (possibly still a unique DAG) when some additive noise model assumptions fail. We defined equivalence classes for such models and introduced the kPC algorithm for learning these equivalence classes from data. We also considered efficient methods for conditional independence testing which do not assume dependencies are linear with Gaussian noise. Finally, we found that the kPC algorithm showed good performance on both synthetic and real data. Areas for future research include extending the weakly additive noise model framework to cases where latent confounding variables and directed cycles may be present as well as speeding up the conditional independence test described in section 8, which is currently the bottleneck for kPC.

Acknowledgments

This work is largely the result of numerous helpful discussions and direct collaboration with Peter Spirtes. Most of the novel results originally appeared in a joint publication with Arthur Gretton and Peter Spirtes (Tillman et al., 2009). Others who have contributed directed to this work or indirectly through helpful discussions and comments include (alphabetically) Tianjiao Chu, Kenji Fukumizu, Clark Glymour, Patrik Hoyer, Dominik Janzing, Joseph Ramsey, and Bernhard Schölkopf. Robert E. Tillman was funded by a grant from the James S. McDonnel Foundation throughout the course of this work.

References

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337404, 1950.

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

T. Chu and C. Glymour. Search for additive nonlinear time series causal models. *Journal of Machine Learning Research*, 9, 2008.

S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2001.

K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems 20*, 2008.

A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems 19*, 2007.

A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. J. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems 20*, 2008.

P. O. Hoyer, A. Hyvärinen, R. Scheines, P. Spirtes, J. Ramsey, G. Lacerda, and S. Shimizu. Causal discovery of linear acyclic models with arbitrary distributions. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, 2008a.

P. O. Hoyer, S. Shimizu, A. J. Kerminen, and M. Palviainen. Estimation of causal effects using linear non-gaussian causal models with hidden variables. *International Journal of Approximate Reasoning*, 49:362–378, 2008b.

P. O. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in Neural Information Processing Systems 21*, 2009.

A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

M. Kalisch and P. Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, 8, 2007.

G. Lacerda, P. Spirtes, J. Ramsey, and P. O. Hoyer. Discovering cyclic causal models by independent components analysis. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, 2008.

C. Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, 1995.

G. Melançon, I. Dutour, and M. Bousquet-Mélou. Random generation of dags for graph drawing. Technical Report INS-R0005, Centre for Mathematics and Computer Sciences, 2000.

J. M. Mooij, D. Janzing, J. Peters, and B. Scholköpf. Regression by dependence minimization and its application to causal inference in additive noise models. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

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

J. Pellet and A. Elisseeff. Finding latent causes in causal networks: an efficient approach based on markov blankets. In *Advances in Neural Information Processing Systems 21*, 2009.

J. D. Ramsey, S. J. Hanson, C. Hanson, Y. O. Halchenko, R. A. Poldrack, and C. Glymour. Six problems for causal inference from fMRI. *NeuroImage*, 2009. In press.

T. Richardson. A discovery algorithm for directed cyclic graphs. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence*, 1996.

S. Shimizu, P. Hoyer, A. Hyvärinen, and A. Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7:1003–2030, 2006.

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

B. Sriperumbudur, A. Gretton, K. Fukumizu, G. Lanckriet, and B. Schölkopf. Injective hilbert space embeddings of probability measures. In *Proceedings of the 21st Annual Conference on Learning Theory*, 2008.

X. Sun. *Causal inference from statistical data*. PhD thesis, Max Plank Institute for Biological Cybernetics, 2008.

X. Sun, D. Janzing, B. Scholköpf, and K. Fukumizu. A kernel-based causal learning algorithm. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

R. E. Tillman, A. Gretton, and P. Spirtes. Nonlinear directed acyclic structure learning with weakly additive noise models. In *Advances in Neural Information Processing Systems 22*, 2009.

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

M. Voortman and M. J. Druzdzel. Insensitivity of constraint-based causal discovery algorithms to violations of the assumption of multivariate normality. In *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference*, 2008.

K. Zhang and A. Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 2009a.

K. Zhang and A. Hyvärinen. Acyclic causality discovery with additive noise: An information-theoretical perspective. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases 2009*, 2009b.