Journal X (2005) XX-XX Submitted XX/XX; Published XX/XX

Learning the Structure of Linear Latent Variable Models

Ricardo Silva RBAS@CS.CMU.EDU
Center for Automated Learning and Discovery
School of Computer Science

Richard Scheines SCHEINES@ANDREW.CMU.EDU
Clark Glymour CG09@ANDREW.CMU.EDU
Peter Spirtes PS7Z@ANDREW.CMU.EDU
Department of Philosophy
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Editor: XXX

Abstract

We describe anytime search procedures that (1) find disjoint subsets of recorded variables for which the members of each subset are d-separated by a single common unrecorded cause, if such exists; (2) return information about the causal relations among the latent factors so identified. We prove the procedure is point-wise consistent assuming (a) the causal relations can be represented by a directed acyclic graph (DAG) satisfying the Markov Assumption and the Faithfulness Assumption; (b) unrecorded variables are not caused by recorded variables; and (c) dependencies are linear. We compare the procedure with factor analysis over a variety of simulated structures and sample sizes, and illustrate its practical value with brief studies of social science data sets. Finally, we consider generalizations for non-linear systems.

Keywords: Latent variable models, causality, graphical models, structural equation models

1. What we will show

In many empirical studies that estimate causal relationships, influential variables are unrecorded, or "latent." When unrecorded variables are believed to influence only one recorded variable directly, they are commonly modeled as noise. When, however, they influence two or more measured variables directly, the intent of such studies is to identify them and their influences. In many cases, for example in sociology, social psychology, neuropsychology, epidemiology, climate research, signal source studies, and elsewhere, the chief aim of inquiry is in fact to identify the causal relations of (often unknown) unrecorded variables that influence multiple recorded variables. It is often assumed on good grounds that recorded variables do not influence unrecorded variables, although in some cases recorded variables may influence one another.

When there is uncertainty about the number of latent variables, which measured variables they influence, or which measured variables influence other measured variables, the investigator who aims at a causal explanation is faced with a difficult discovery problem for which currently available methods are at best heuristic. Loehlin (2004) argues that while there are several approaches to automatically learn causal structure, none can be seem as competitors of exploratory factor analysis: the usual focus of automated search procedures for causal Bayes nets is on relations among observed variables. Loehlin's comment overlooks Bayes net search procedures robust to latent variables (Spirtes et al., 2000), but the general sense of his comment is correct. For a kind of model widely used in applied sciences − "multiple indicator models" in which multiple observed measures are assumed to be effects of unrecorded variables and possibly of each other − machine learning has provided no principled alternative to factor analysis, principal components, and regression analysis of proxy scores formed from averages or weighted averages of measured variables, the techniques most commonly used to estimate the existence and influences of variables that are unrecorded. The statistical properties of models produced by these methods are well understood, but there are no proofs, under any general assumptions, of convergence to features of the true causal structure. The few simulation studies of the accuracy of these methods on finite samples with diverse causal structures are not reassuring (Glymour, 1997). The use of proxy scores with regression is demonstrably not consistent, and systematically overestimates dependencies. Better methods are needed.

We describe a two part method for this problem. The method (1) finds clusters of measured variables that are d-separated by a single unrecorded common cause, if such exists; and (2) finds features of the Markov Equivalence class of causal models for the latent variables. Assuming only principles standard in Bayes net search algorithms, and satisfied in almost all social science models, the two procedures converge, probability 1 in the large sample limit, to correct information. The completeness of the information obtained about latent structure depends on how thoroughly confounded the measured variables are, but when, for each unknown latent variable, there in fact exists at least a small number of measured variables that are influenced only by that latent variable, the method returns the complete Markov Equivalence class of the latent structure. We show by simulation studies for three latent structures and for a range of sample sizes that the method identifies the number of latent variables more accurately than does factor analysis. Applying the search procedures for latent structure to the latent variables identified (1) by factor analysis and, alternatively, (2) by our clustering method, we again show by simulation that our clustering procedure is more accurate. We illustrate the procedure with applications to social science cases.

2. Illustrative principles

Consider Figure 1, where X variables are recorded and L variables (in ovals) are unrecorded and unknown to the investigator.

The latent structure, the dependencies of measured variables on individual latent variables, and the linear dependency of the measured variables on their parents and (unrepresented) independent noises in Figure 1 imply a pattern of constraints on the covariance matrix among the X variables. For example, X_1, X_2, X_3 have zero covariances with X_7, X_8, X_9.

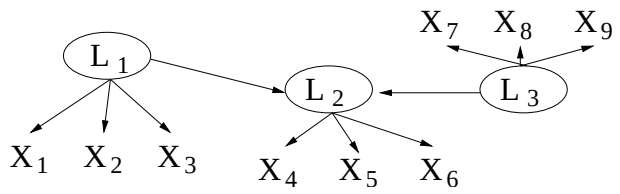


Figure 1: A latent variable model which entails several constraints on the observed covariance matrix. Latent variables are inside ovals.

Less obviously, for X_1, X_2, X_3 and any one of X_4, X_5, X_6, three quadratic constraints (*tetrad* constraints) on the covariance matrix are implied: e.g., for X_4

$$\rho_{12}\rho_{34} = \rho_{14}\rho_{23} = \rho_{13}\rho_{24} \tag{1}$$

where ρ_{12} is the Pearson product moment correlation between X_1, X_2, etc. (Note that any two of the three vanishing tetrad differences above entails the third.) The same is true for X_7, X_8, X_9 and any one of X_4, X_5, X_6; for X_4, X_5, X_6, and any one of X_1, X_2, X_3 or any one of X_7, X_8, X_9. Further, for any two of X_1, X_2, X_3 or of X_7, X_8, X_9 and any two of X_4, X_5, X_6, exactly one such quadratic constraint is implied, e.g., for X_1, X_2 and X_4, X_5, the single constraint

$$\rho_{14}\rho_{25} = \rho_{15}\rho_{24} \tag{2}$$

The constraints hold as well if covariances are substituted for correlations.

Statistical tests for vanishing tetrad differences are available for a wide family of distributions. Linear and non-linear models can imply other constraints on the correlation matrix, but general, feasible computational procedures to determine arbitrary constraints are not available (Geiger and Meek, 1999) nor are there any available statistical tests of good power for higher order constraints.

Given a "pure" set of sets of measured indicators of latent variables, as in Figure 1 − informally, a measurement model specifying, for each latent variable, a set of measured variables influenced only by that latent variable and individual, independent noises − the causal structure among the latent variables can be estimated by any of a variety of methods. Standard chi square tests of latent variable models can be used to compare models with and without a specified edge, providing indirect tests of conditional independence among latent variables. The conditional independence facts can then be input to a constraint based Bayes net search algorithm, such as PC or FCI (Spirtes et al., 2000). Such procedures are asymptotically consistent, but not necessarily optimal on small samples. Alternatively, a correlation matrix among the latent variables can be estimated from the measurement model and the correlations among the measured variables, and a Bayesian search can be used. Score-based approaches for learning the structure of Bayesian networks, such as GES (Chickering, 2002), are usually more accurate with small to medium sized samples than are PC or FCI. Given an identification of the latent variables and a set of "pure" measured effects or indicators of each latent, the correlation matrix among the latent variables can be estimated by expectation maximization, The complete graph on the latent variables is then

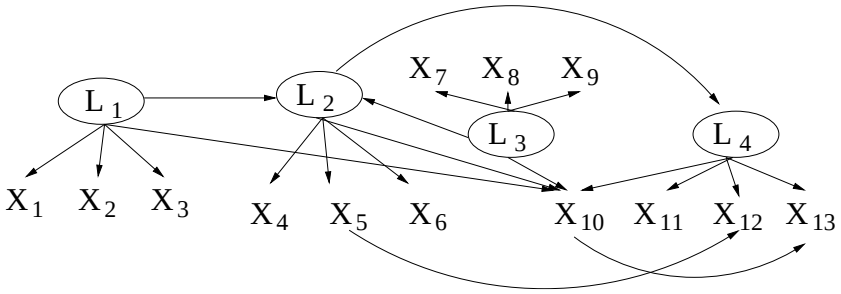


Figure 2: A latent variable model which entails several constraints on the observed covariance matrix.

dispensed with and the latent structure is estimated from the estimated correlations, using GES with the Bayes Information Criterion (BIC) score to estimate posterior probabilities. In Figure 1 the measured variables neatly cluster into disjoint sets of variables and the variables in any one set are influenced only by a single common cause and there are no influences of the measured variables on one another. In many real cases the influences on the measured variables do not separate so simply. Some of the measured variables may influence others (as in signal leakage between channels in spectral measurements), and some or many measured variables may be influenced by two or more latent variables.

For example, the latent structure of a linear, Gaussian system shown in Figure 2 can be recovered by the procedures we propose. Our aim in what follows is to prove and use new results about implied constraints on the covariance matrix of measured variables to form measurement models that enable estimation of features of the Markov Equivalence class of the latent structure in a wide range of cases. We will develop the theory first for linear models with a joint Gaussian distribution on all variables, including latent variables, and then consider possibilities for generalization. In many models of this kind in the applied sciences, some variables are specified with unexplained correlations represented as bidirected edges between the variables. We allow representations of this kind.

The general idea is as follows. We introduce a graphical representation of an equivalence class of models that entail a given set of vanishing partial correlations and vanishing tetrad differences, analogous to the familiar notion of a pattern (Pearl, 1988) used to represent a Markov Equivalence class of directed acyclic graphs (DAGs), We provide an algorithm for discovering features of this Measurement Pattern. Using the Measurement Pattern, further procedures find clusters of measured variables for which the members of each cluster share a latent common cause. A combination of expectation-maximization and the GES algorithm scored by BIC is then used to estimate the causal relations among the latent variables.

3. Related work

The traditional framework for discovering latent variables is factor analysis and its variants (see, e.g., Bartholomew et al., 2002). A number of factors is chosen based on some criterion such as the minimum number of factors that fit the data at a given significance level or

the number that maximizes a score such as BIC. After fitting the data, usually assuming a Gaussian distribution, different transformations (rotations) to the latent covariance matrix are applied in order to satisfy some criteria of simplicity. Latents are interpreted based on the magnitude of the coefficients relating each observed variable to each latent.

In non-Gaussian cases, the usual methods are variations of independent component analysis, such as independent factor analysis (Attias, 1999) and tree-based component analysis (Bach and Jordan, 2003). These methods severely constrain dependency structure among the latent variables. That facilitates joint density estimation or blind source separation, but it is of little use in learning causal structure.

In a similar vein, Zhang (2004) represents latent variable models for discrete variables (both observed and latent) with a multinomial probabilistic model. The model is constrained to be a tree and every observed variable has one and only one (latent) parent and no child. Zhang does not provide a search method to find variables satisfying the assumption, but assumes a priori the variables measured satisfy it.

Elidan et al. (2000) introduces latent variables as common causes of densely connected regions of a DAG learned through standard algorithms for learning Bayesian network structures. Once one latent is introduced as the parent of a set of nodes originally strongly connected, the standard search is executed again. The process can be iterated to introduce multiple latents. Examples are given for which this procedure increases the fit over a latent-free graphical model are provided, but Elidan et al. provide no information about the conditions under which the estimated causal structure is correct. In Silva et al. (2003) we developed an approach to learning measurement models. That procedure requires that the true underlying graph has a "pure" submodel with three measures for each latent variable, which is a strong and generally untestable assumption. That assumption is not needed in the procedures described here.

4. Notation, assumptions and definitions

Our work is in the framework of causal graphical models. Concepts used here without explicit definition, such as d-separation and I-map, can be found in standard sources (Pearl, 1988; Spirtes et al., 2000; Pearl, 2000). We use "variable" and "vertex" interchangeably, and standard kinship terminology ("parent," "child," "descendant," "ancestor") for directed graph relationships. Sets of variables are represented in bold, individual variables and symbols for graphs in italics. The Pearson partial correlation of X, Y controlling for Z is denoted by $\rho_{XY.Z}$. We assume i.i.d. data sampled from a subset $\mathbf{O}$ of the variables of a joint Normal distribution D on variables $\mathbf{V} = \mathbf{O} \cup \mathbf{L}$, subject to the following assumptions:

A1 D factors according to the local Markov assumption for a DAG G with vertex set $\mathbf{V}$. That is, any variable is independent of its non-descendants in G conditional on any values of its parents in G.

A2 No vertex in $\mathbf{O}$ is an ancestor of any vertex in $\mathbf{L}$. We call this property the *measurement assumption*;

A3 Each variable in $\mathbf{V}$ is a linear function of its parents plus an additive error term of positive finite variance

A4 The Faithfulness Assumption: for all $\{X, Y, Z\} \subseteq \mathbf{V}$, X is independent of Y conditional on each assignment of values to variables in Z if and only if the Markov Assumption for G entails such conditional independencies. For models satisfying A1-A3 with Gaussian distributions, Faithfulness is equivalent to assuming that no correlations or partial correlations vanish because of multiple pathways whose influences perfectly cancel one another.

Definition 1 (Linear latent variable model) *A model satisfying A1 − A4 is a linear latent variable model, or for brevity, where the context makes the linearity assumption clear, a latent variable model.*

A single symbol, such as G, will be used to denote both a linear latent variable model and the corresponding latent variable graph. Linear latent variable models are ubiquitous in econometric, psychometric, and social scientific studies (Bollen, 1989), where they are usually known as structural equation models.

Definition 2 (Measurement model) *Given a linear latent variable model G, with vertex set $\mathbf{V}$, the subgraph containing all vertices in $\mathbf{V}$, and all and only those edges directed into vertices in $\mathbf{O}$, is called the measurement model of G.*

Definition 3 (Structural model) *Given a linear latent variable model G, the subgraph containing all and only its latent nodes and respective edges is the structural model of G.*

Definition 4 (Linear entailment) *We say that a DAG G linearly entails a constraint if and only if the constraint holds in every distribution satisfying A1 - A4 for G with covariance matrix parameterized by Θ, the set of linear coefficients and error variances that defines the conditional expectation and variance of a vertex given its parents.*

Definition 5 (Tetrad equivalence class) *Given a set $\mathbf{C}$ of vanishing partial correlations and vanishing tetrad differences, a tetrad equivalence class $\mathcal{T}(\mathbf{C})$ is the set of all latent variable graphs each member of which entails all and only the tetrad constraints and vanishing partial correlations among the measured variables entailed by $\mathbf{C}$.*

Definition 6 (Measurement equivalence class) *An equivalence class of measurement models $\mathcal{M}(\mathbf{C})$ for $\mathbf{C}$ is the union of the measurement models graphs in $\mathcal{T}(\mathbf{C})$. We introduce a graphical representation of common features of all elements of $\mathcal{M}(\mathbf{C})$, analogous to the familiar notion of a pattern representing the Markov Equivalence class of a Bayes net.*

Definition 7 (Measurement pattern) *A measurement pattern, denoted $\mathcal{MP}(\mathbf{C})$, is a graph representing features of the equivalence class $\mathcal{M}(\mathbf{C})$ satisfying the following:*

- *there are latent and observed vertices;*
- *the only edges allowed in an MP are directed edges from latent variables to observed variables, and undirected edges between observed vertices;*
- *every observed variable in a MP has at least one latent parent;*

Algorithm FindPattern
Input: a covariance matrix Σ

1. Start with a complete graph G over the observed variables.
2. Remove edges for pairs that are marginally uncorrelated or uncorrelated conditioned on a third variable.
3. For every pair of nodes linked by an edge in G, test if some rule CS1, CS2 or CS3 applies. Remove an edge between every pair corresponding to a rule that applies.
4. Let H be a graph with no edges and with nodes corresponding to the observed variables.
5. For each maximal clique in G, add a new latent to H and make it a parent to all corresponding nodes in the clique.
6. For each pair (A, B), if there is no other pair (C, D) such that $\sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BC} = \sigma_{AB}\sigma_{CD}$, add an undirected edge $A - B$ to H.
7. Return H.

Table 1: Returns a measurement pattern corresponding to the tetrad and first order vanishing partial correlations of Σ.

- *if two observed variables X and Y in a $\mathcal{MP}(\mathbf{C})$ do not share a common latent parent, then X and Y do not share a common latent parent in any member of $\mathcal{M}(\mathbf{C})$;*
- *if observed variables X and Y are not linked by an undirected edge in $\mathcal{MP}(\mathbf{C})$, then X is not an ancestor of Y in any member of $\mathcal{M}(\mathbf{C})$.*

Definition 8 (Pure measurement model) *A pure measurement model is a measurement model in which each observed variable has only one latent parent, and no observed parent. That is, it is a tree beneath the latents.*

5. Procedures for finding pure measurement models

Our goal is to find pure measurement models whenever possible, and use them to estimate the structural model. To do so, we first use properties relating graphical structure and covariance constraints to identify a measurement pattern, and then turn the measurement pattern into a pure measurement model.

FindPattern, given in Table 1, is an algorithm to learn a measurement pattern from an oracle for vanishing partial correlations and vanishing tetrad differences. The algorithm uses three rules, CS1, CS2, CS3, based on Lemmas that follow, for determining graphical structure from constraints on the correlation matrix of observed variables.

Let $\mathbf{C}$ be a set of linearly entailed constraints that exist in the observed covariance matrix. The first stage of FindPattern searches for subsets of $\mathbf{C}$ that will guarantee

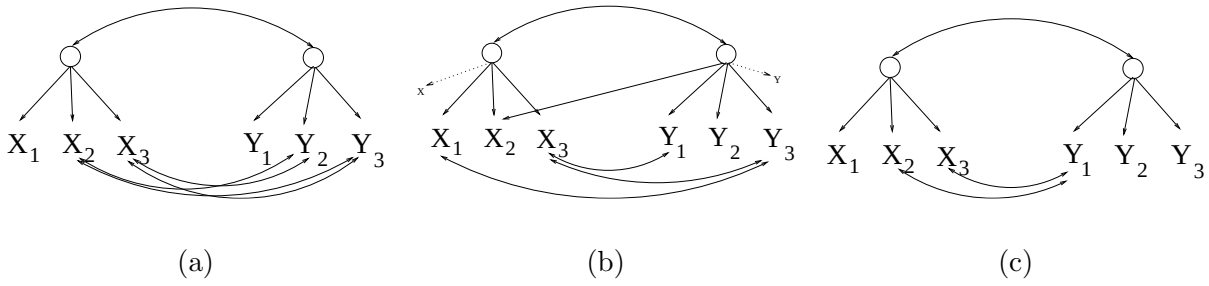


Figure 3: Three examples with two main latents and several independent latent common causes of two indicators (represented by double-directed edges). In (a), CS1 applies, but not CS2 nor CS3 (even when exchanging labels of the variables); In (b), CS2 applies (assuming the conditions for X_1, X_2 and Y_1, Y_2), but not CS1 nor CS3. In (c), CS3 applies, but not CS1 nor CS2.

that two observed variables do not have any latent parent in common. Let G be the latent variable graph for a linear latent variable model with a set of observed variables $\mathbf{O}$. Let $\mathbf{O'} = \{X_1, X_2, X_3, Y_1, Y_2, Y_3\} \subset \mathbf{O}$ such that for all triplets $\{A, B, C\}$, $\{A, B\} \subset \mathbf{O'}$ and $C \in \mathbf{O}$, we have $\rho_{AB} \neq 0, \rho_{AB.C} \neq 0$. Let τ_{IJKL} represent the tetrad constraint $\sigma_{IJ}\sigma_{KL} - \sigma_{IK}\sigma_{JL} = 0$ and $\neg\tau_{IJKL}$ represent the complementary constraint $\sigma_{IJ}\sigma_{KL} - \sigma_{IK}\sigma_{JL} \neq 0$:

Lemma 9 (CS1 Test) *If constraints* $\{\tau_{X_1Y_1X_2X_3}, \tau_{X_1Y_1X_3X_2}, \tau_{Y_1X_1Y_2Y_3}, \tau_{Y_1X_1Y_3Y_2}, \neg\tau_{X_1X_2Y_2Y_1}\}$ *all hold, then* X_1 *and* Y_1 *do not have a common parent in* G.

"CS" here stands for "constraint set," the premises of a rule that can be used to test if two nodes do not share a common parent. Other sets of observable constraints can be used to reach the same conclusion.

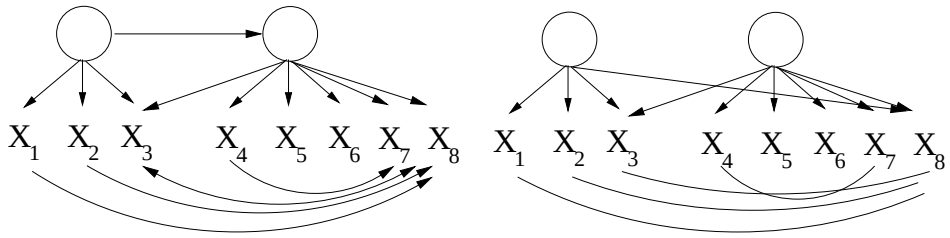
Let the predicate $F_1(X, Y, G)$ be true if and only if there exist two nodes W and Z in latent variable graph G such that τ_{WXYZ} and τ_{WXZY} are both linearly entailed by G, all variables in $\{W, X, Y, Z\}$ are correlated, and there is no observed C in G such that $\rho_{AB.C} = 0$ for $\{A, B\} \subset \{W, X, Y, Z\}$:

Lemma 10 (CS2 Test) *If constraints* $\{\tau_{X_1Y_1Y_2X_2}, \tau_{X_2Y_1Y_3Y_2}, \tau_{X_1X_2Y_2X_3}, \neg\tau_{X_1X_2Y_2Y_1}\}$ *all hold such that* $F_1(X_1, X_2, G) = true$, $F_1(Y_1, Y_2, G) = true$, X_1 *is not an ancestor of* X_3 *and* Y_1 *is not an ancestor of* Y_3, *then* X_1 *and* Y_1 *do not have a common parent in* G.

Lemma 11 (CS3 Test) *If constraints* $\{\tau_{X_1Y_1Y_2Y_3}, \tau_{X_1Y_1Y_3Y_2}, \tau_{X_1Y_2X_2X_3}, \tau_{X_1Y_2X_3X_2}, \tau_{X_1Y_3X_2X_3}, \tau_{X_1Y_3X_3X_2}, \neg\tau_{X_1X_2Y_2Y_3}\}$ *all hold, then* X_1 *and* Y_1 *do not have a common parent in* G.

These rules are illustrated in Figure 3. The rules are not redundant: only one can be applied on each situation. For CS2 (Figure 3(b)), nodes X and Y are depicted as auxiliary nodes that can be used to verify predicates F_1. For instance, $F_1(X_1, X_2, G)$ is true because all three tetrads in the covariance matrix of $\{X_1, X_2, X_3, X\}$ hold.

Sometime it is possible to guarantee that a node is not an ancestor of another, as required, e.g., to apply CS2:



(a) (b)

Figure 4: In (a), a model that generates a covariance matrix Σ. In (b), the output of FindPattern given Σ. Pairs in $\{X_1, X_2\} \times \{X_4, \ldots, X_7\}$ are separated by CS2.

Lemma 12 *If for some set* $\mathbf{O}' = \{X_1, X_2, X_3, X_4\} \subseteq \mathbf{O}$, $\sigma_{X_1X_2}\sigma_{X_3X_4} = \sigma_{X_1X_3}\sigma_{X_2X_4} = \sigma_{X_1X_4}\sigma_{X_2X_3}$ *and for all triplets* $\{A, B, C\}$, $\{A, B\} \subset \mathbf{O}', C \in \mathbf{O}$, *we have* $\rho_{AB.C} \neq 0$ *and* $\rho_{AB} \neq 0$, *then* $A \in \mathbf{O}'$ *is not a descendant in* G *of any element of* $\mathbf{O}' \backslash \{A\}$.

For instance, in Figure 3(b) the existence of the observed node X (linked by a dashed edge to the parent of X_1) allows the inference that X_1 is not an ancestor of X_3, since all three tetrad constraints hold in the covariance matrix of $\{X, X_1, X_2, X_3\}$.

Theorem 13 *The output of* FindPattern *is a measurement pattern* $\mathcal{MP}(\mathbf{C})$ *with respect to the tetrad and zero/first order vanishing partial correlation constraints* $\mathbf{C}$ *of* Σ.

The presence of an undirected edge does not mean that adjacent vertices in the pattern are actually adjacent in the true graph. Figure 4 illustrates this: X_3 and X_8 share a common parent in the true graph, but are not adjacent. Observed variables adjacent in the output pattern always share at least one parent in the pattern, but do not always share a common parent in the true DAG. Vertices sharing a common parent in the pattern might not share a parent in the true graph (e.g., X_1 and X_8 in Figure 4).

The FindPattern algorithm is sound, but not necessarily complete. That is, there might be graphical features shared by all members of the measurement model equivalence class that are not discovered by FindPattern. Using the notion of a pure measurement model, defined above, we can improve the results with respect to a subset of the given variables. A pure measurement model implies a *clustering* of observed variables: each cluster is a set of observed variables that share a common (latent) parent, and the set of latents defines a partition over the observed variables. The output of FindPattern cannot, however, reliably be turned into a pure measurement pattern in the obvious way, by removing from H all nodes that have more than one latent parent and one of every pair of adjacent nodes.

The procedure BuildPureClusters of Table 2 builds a pure measurement model using FindPattern and an oracle for constraints as input. Variables are removed whenever appropriate tetrad constraints are not satisfied. Some extra adjustments concern clusters

with proper subsets that are not consistently correlated to another variable (Steps 6 and 7) and a final merging of clusters (Step 8). We explain the necessity of these steps in Appendix A. As described, BuildPureClusters requires some decisions that are not specified (Steps 2, 4, 5 and 9). We propose an implementation in Appendix C, but various results are indifferent to how these choices are made.

The graphical properties of the output of BuildPureClusters are summarized by the following theorem:

Theorem 14 *Given a covariance matrix Σ assumed to be generated from a linear latent variable model G with observed variables $\mathbf{O}$ and latent variables $\mathbf{L}$, let G_{out} be the output of* BuildPureClusters(Σ) *with observed variables $\mathbf{O_{out}} \subseteq \mathbf{O}$ and latent variables $\mathbf{L_{out}}$. Then G_{out} is a measurement pattern, and there is an unique injective mapping $M : \mathbf{L_{out}} \rightarrow \mathbf{L}$ with the following properties:*

1. *Let $L_{out} \in \mathbf{L_{out}}$. Let X be a child of L_{out} in G_{out}. Then $M(L_{out})$ d-separates X from $\mathbf{O_{out}} \backslash X$ in G;*

2. *$M(L_{out})$ d-separates X from every latent L in G for which $M^{-1}(L)$ is defined;*

3. *Let $\mathbf{O'} \subseteq \mathbf{O_{out}}$ be such that each pair in $\mathbf{O'}$ is correlated. At most one element in $\mathbf{O'}$ has the following property: (i) it is not a descendant of its respective mapped latent parent in G or (ii) it has a hidden common cause with its respective mapped latent parent in G;*

Informally, there is a labeling of latents in G_{out} according to the latents in G, and in this relabeled output graph any d-separation between a measured node and some other node will hold in the true graph, G. For each group of correlated observed variables, we can guaranteee that at most one edge from a latent into an observed variable is incorrectly directed. Notice that we cannot guarantee that an observed node X with latent parent L_{out} in G_{out} will be d-separated from the other nodes in G given $M(L_{out})$: if X has a common cause with $M(L_{out})$, then X will be d-connected to any ancestor of $M(L_{out})$ in G given $M(L_{out})$.

To illustrate BuildPureClusters, suppose the true graph is the one given in Figure 5(a), with two unlabeled latents and 12 observed variables. This graph is unknown to BuildPureClusters, which is given only the covariance matrix of variables $\{X_1, X_2, ..., X_{12}\}$. The task is to learn a measurement pattern, and then a purified measurement model.

In the first stage of BuildPureClusters, the FindPattern algorithm, we start with a fully connected graph among the observed variables (Figure 5(b)), and then proceed to remove edges according to rules CS1, CS2 and CS3, giving the graph shown in Figure 5(c). There are two maximal cliques in this graph: $\{X_1, X_2, X_3, X_7, X_8, X_{11}, X_{12}\}$ and $\{X_4, X_5, X_6, X_8, X_9, X_{10}, X_{12}\}$. They are distinguished in the figure by different edge representations (dashed and solid - with the edge $X_8 - X_{12}$ present in both cliques). The next stage takes these maximal cliques and creates an intermediate graphical representation, as depicted in Figure 5(d). In Figure 5(e), we add the undirected edges $X_7 - X_8$, $X_8 - X_{12}$, $X_9 - X_{10}$ and $X_{11} - X_{12}$, finalizing the measurement pattern returned by FindPattern. Finally, Figure 5(f) represents a possible purified output of BuildPureClusters given this

Algorithm BuildPureClusters
Input: a covariance matrix Σ

1. $G \leftarrow$FindPattern(Σ).
2. *Choose* a set of latents in G. Remove all other latents and all observed nodes that are not children of the remaining latents and all clusters of size 1.
3. Remove all nodes that have more than one latent parent in G.
4. For all pairs of nodes linked by an undirected edge, *choose* one element of each pair to be removed.
5. If for some set of nodes $\{A, B, C\}$, all children of the same latent, there is a fourth node D in G such that $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BC}$ is *not* true, remove one of these four nodes.
6. For every latent L with at least two children, $\{A, B\}$, if there is some node C in G such that $\sigma_{AC} = 0$ and $\sigma_{BC} \neq 0$, split L into two latents L_1 and L_2, where L_1 becomes the only parent of all children of L that are correlated with C, and L_2 becomes the only parent of all children of L that are not correlated with C;
7. Remove any cluster with exactly 3 variables $\{X_1, X_2, X_3\}$ such that there is no X_4 where all three tetrads in the covariance matrix $\mathbf{X} = \{X_1, X_2, X_3, X_4\}$ hold, all variables of $\mathbf{X}$ are correlated and no partial correlation of a pair of elements of $\mathbf{X}$ is zero conditioned on some observed variable;
8. While there is a pair of clusters with latents L_i and L_j, such that for all subsets $\{A, B, C, D\}$ of the union of the children of L_i, L_j we have $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BC}$, and no marginal independence or conditional independence in sets of size 1 are observed in this cluster, set $L_i = L_j$ (i.e., merge the clusters);
9. Again, verify all implied tetrad constraints and remove elements accordingly. Iterate with the previous step till no changes happen;
10. Remove all latents with less than three children, and their respective measures;
11. if G has at least four observed variables, return G. Otherwise, return an empty model.

Table 2: A general strategy to find a pure MP that is also a linear measurement model of a subset of the latents in the true graph. As explained in the body of the text, steps 2, 4, 5 and 9 are not described algorithmically in this Section.

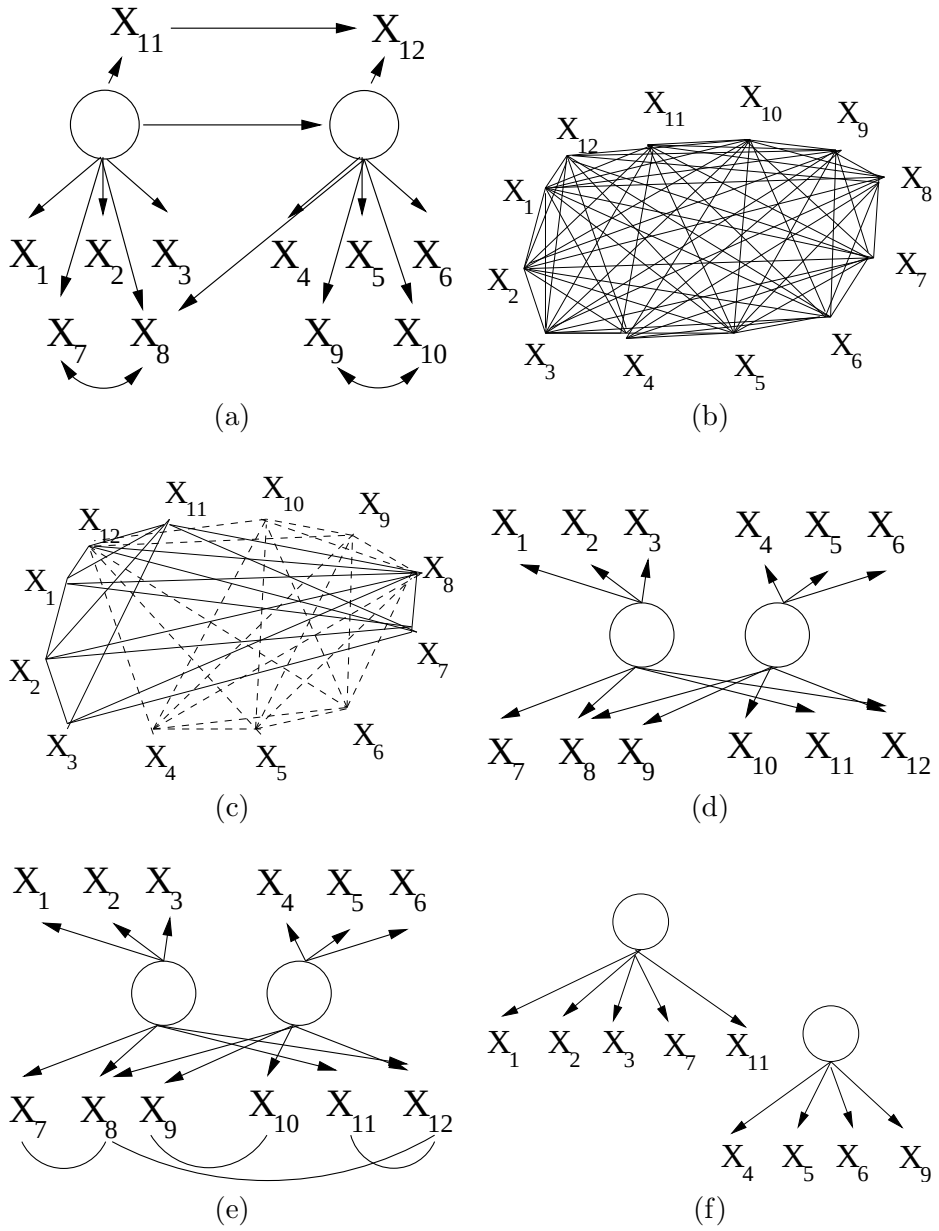


Figure 5: A step-by-step demonstration of how a covariance matrix generated by graph in Figure (a) will induce the pure measurement model in Figure (f).

pattern. Another purification with as many nodes as in the graph in Figure 5(f) substitutes node X_9 for node X_{10}.

The following result is essential to provide an algorithm that is guaranteed to find a Markov equivalence class for the latents in $M(\mathbf{L_{out}})$ using the output of BuildPureClusters as a starting point:

Theorem 15 *Let $M(\mathbf{L_{out}}) \subseteq \mathbf{L}$ be the set of latents in G obtained by the mapping function $M()$. Let $\Sigma_{\mathbf{O_{out}}}$ be the population covariance matrix of $\mathbf{O_{out}}$. Let the DAG G^{aug}_{out} be G_{out} augmented by connecting the elements of $\mathbf{L_{out}}$ such that the structural model of G^{aug}_{out} is an I-map of the distribution of $M(\mathbf{L_{out}})$. Then there exists a linear latent variable model using G^{aug}_{out} as the graphical structure such that the implied covariance matrix of $\mathbf{O_{out}}$ equals $\Sigma_{\mathbf{O_{out}}}$.*

A further reason why we do not provide details of some steps of BuildPureClusters at this point is because there is no unique way of implementing it, and different purifications might be of interest. For instance, one might be interested in the pure model that has the largest possible number of latents. Another one might be interested in the model with the largest number of observed variables. However, some of these criteria might be computationally intractable to achieve. Consider for instance the following criterion, which we denote as $\mathcal{MP}^3$: given a measurement pattern, decide if there is some choice of nodes to be removed such that the resulting graph is a pure measurement model and each latent has at least three children. This problem is intractable:

Theorem 16 *Problem $\mathcal{MP}^3$ is NP-complete.*

There is no need to solve a NP-hard problem in order to have the theoretical guarantees of interpretability of the output given by Theorem 14. For example, there is a stage in FindPattern where it appears necessary to find all maximal cliques, but, in fact, it is not. Identifying more cliques increases the chance of having a larger output (which is good) by the end of the algorithm, but it is not required for the algorithms correctness. Stopping at Step 5 of FindPattern before completion will not affect Theorems 14 or 15.

Another computational concern is the $O(N^5)$ loops in Step 3 of FindPattern, where N is the number of observed variables. Again, it is not necessary to compute this loop entirely. One can stop Step 3 at any time at the price of losing information, but not the theoretical guarantees of BuildPureClusters. This anytime property is summarized by the following corollary:

Corollary 17 *The output of* BuildPureClusters *retains its guarantees even when rules CS1, CS2 and CS3 are applied an arbitrary number of times in* FindPattern *for any arbitrary subset of nodes and an arbitrary number of maximal cliques is found.*

6. Learning the structure of the unobserved

The real motivation for finding a pure measurement model is to obtain reliable statistical access to the relations among the latent variables. Given a pure and correct measurement model, even one involving a fairly small subset of the original measured variables, a variety of algorithms exist for finding a Markov equivalence class of graphs over the set of latents in the given measurement model.

6.1 Constraint-based search

Constraint based search algorithms rely on decisions about independence and conditional independence among a set of variables to find the Markov equivalence class over these

variables. Given a pure and correct measurement model involving at least 2 measures per latent, we can test for independence and conditional independence among the latents, and thus search for equivalence classes of structural models among the latents, by taking advantage of the following theorem Spirtes et al. (2000):

Theorem 18 *Let G be a pure linear latent variable model. Let L_1, L_2 be two latents in G, and $\mathbf{Q}$ a set of latents in G. Let X_1 be a measure of L_1, X_2 be a measure of L_2, and $X_{\mathbf{Q}}$ be a set of measures of $\mathbf{Q}$ containing at least two measures per latent. Then L_1 is d-separated from L_2 given $\mathbf{Q}$ in G if and only if the rank of the correlation matrix of $\{X_1, X_2\} \cup \mathbf{X_Q}$ is less than or equal to $|\mathbf{Q}|$ with probability 1 with respect to the Lebesgue measure over the linear coefficients and error variances of G.*

We can then use this constraint to test[1] for conditional independencies among the latents. Such conditional independence tests can then be used as an oracle for constraint-satisfaction techniques for causality discovery in graphical models, such as the PC algorithm (Spirtes et al., 2000) or the FCI algorithm (Spirtes et al., 2000).

We define the algorithm PC-MIMBuild[2] as the algorithm that takes as input a measurement model satisfying the assumption of purity mentioned above and a covariance matrix, and returns the Markov equivalence class of the structural model among the latents in the measurement model according to the PC algorithm. A FCI-MIMBuild algorithm is defined analogously. In the limit of infinite data, it follows from the preceding and from the consistency of PC and FCI algorithms (Spirtes et al., 2000) that

Corollary 19 *Given a covariance matrix Σ assumed to be generated from a linear latent variable model G, and G_{out} the output of* BuildPureClusters *given Σ, the output of* PC-MIMBuild *or* FCI-MIMBuild *given (Σ, G_{out}) returns the correct Markov equivalence class of the latents in G corresponding to latents in G_{out} according to the mapping implicit in* BuildPureClusters.

6.2 Score-based search

Score-based approaches for learning the structure of Bayesian networks, such as GES (Meek, 1997; Chickering, 2002) are usually more accurate than PC or FCI when there are no omitted common causes, or in other terms, when the set of recorded variables is causally sufficient. We know of no consistent scoring function for linear latent variable models that can be easily computed. As a heuristic, we suggest using the Bayesian Information Criterion (BIC) function. Using BIC with Structural EM (Friedman, 1998) and GES results in a computationally efficient way of learning structural models, where the measurement model is fixed and GES is restricted to modify edges among latents only. Assuming a Gaussian distribution, the first step of our Structural EM implementation uses a fully connected structural model in order to estimate the first expected latent covariance matrix. That is followed by a GES search. We call this algorithm GES-MIMBuild and use it as the

1. One way to test if the rank of a covariance matrix in Gaussian models is at most q is to fit a factor analysis model with q latents and assess its significance.
2. MIM stands for "multiple indicator model", a term in structural equation model literature describing latent variable models with multiple measures per latent.

structural model search component in all of the studies of simulated and empirical data that follow.

7. Simulation studies

In the following simulation studies, we draw samples of three different sizes from 9 different latent variable models. We compare our algorithm against two versions of exploratory factor analysis, and measure the success of each on the following discovery tasks:

DP1. Discover the number of latents in G.

DP2. Discover which observed variables measure each latent G.

DP3. Discover as many features as possible about the causal relationships among the latents in G.

Since factor analysis addresses only tasks DP1 and DP2, we compare it directly to BuildPureClusters on DP1 and DP2. For DP3, we use our procedure and factor analysis to compute measurement models, then discover as much about the features of the structural model among the latents as possible by applying GES-MIMBuild to the measurement models output by BPC and factor analysis.

We hypothesized that three features of the problem would affect the performance of the algorithms compared: sample size; the complexity of the structural model; and, the complexity and level of impurity in the generating measurement model. We use three different sample sizes for each study: 200, 1,000, and 10,000. We constructed nine generating latent variable graphs by using all combinations of the three structural models and three measurement models in Figure 6. For structural model SM3, the respective measurement models are augmented accordingly.

MM1 is a pure measurement model with three indicators per latent. MM2 has five indicators per latent, one of which is impure because its error is correlated with another indicator, and another because it measures two latents directly. MM3 involves six indicators per latent, half of which are impure.

SM1 entails one unconditional independence among the latents: L_1 is independent L_3. SM2 entails one first order conditional independence: $L_1 \perp L_3 | L_2$, and SM3 entails one first order conditional independence: $L_2 \perp L_3 | L_1$, and one second order conditional independence relation: $L_1 \perp L_4 | \{L_2, L_3\}$. Thus the statistical complexity of the structural models increases from SM1 to SM3 and the impurity of measurement models increases from MM1 to MM3.

For each generating latent variable graph, we used the Tetrad IV program[3] with the following procedure to draw 10 multivariate normal samples of size 200, 10 at size 1,000, and 10 at size 10,000.

1. Pick coefficients for each edge in the model randomly from the interval $[-1.5, -0.5] \cup [0.5, 1.5]$.

2. Pick variances for the exogenous nodes (i.e., latents without parents and error nodes) from the interval $[1, 3]$.

3. Available at `http://www.phil.cmu.edu/tetrad`.

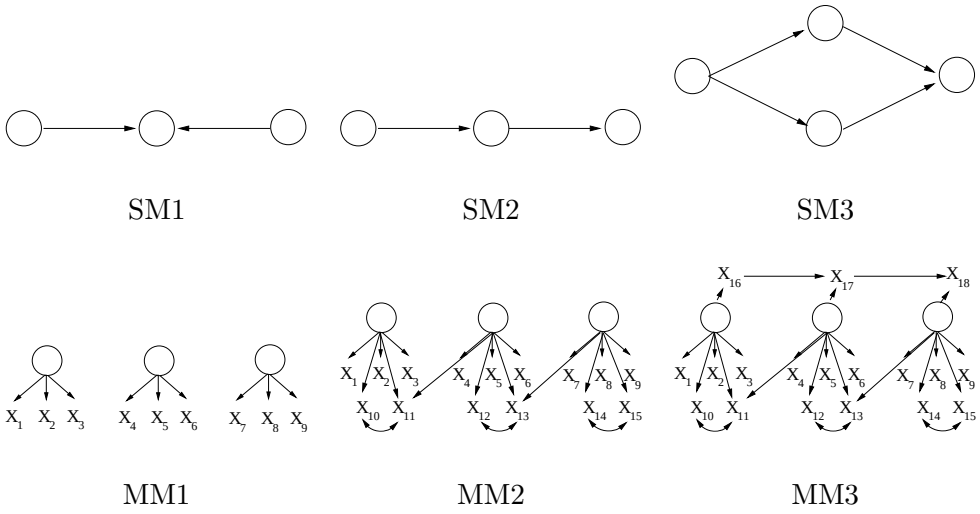


Figure 6: The Structural and Measurement models used in our simulation studies.

3. Draw one pseudo-random sample of size N.

We used three algorithms in our studies:

1. BPC: BuildPureClusters + GES-MIMBuild
2. FA: Factor Analysis + GES-MIMBuild
3. P-FA: Factor Analysis + Purify + GES-MIMBuild

BPC is the implementation of BuildPureClusters and GES-MIMBuild described in C. FA involves combining standard factor analysis to find the measurement model with GES-MIMBuild to find the structural model. For standard factor analysis, we used `factanal` from R 1.9 with the oblique rotation promax. FA and variations are still widely used and are perhaps the most popular approach to latent variable modeling (Bartholomew et al., 2002). We choose the number of latents by iteratively increasing its number till we get a significant fit above 0.05, or till we have to stop due to numerical instabilities.

Factor analysis is not directly comparable to BuildPureClusters since it does not generate pure models only. We extend our comparison of BPC and FA by including a version of factor analysis with a post processing step to purify the output of factor analysis. Purified Factor Analysis, or P-FA, takes the measurement model output by factor analysis and proceeds as follows: 1. for each latent with two children only, remove the child that has the highest number of parents. 2. remove all latents with one child only, unless this latent is the only parent of its child. 3. removes all indicators that load significantly on more than one latent. The measurement model output by P-FA typically contains far fewer latent variables than the measurement model output by FA.

In order to compare the output of BPC, FA, and P-FA on discovery tasks DP1 (finding the correct number of underlying latents) and DP2 (measuring these latents appropriately),

we must map the latents discovered by each algorithm to the latents in the generating model. That is, we must define a mapping of the latents in the G_{out} to those in the true graph G. Although one could do this in many ways, for simplicity we used a majority voting rule in BPC and P-FA. If a majority of the indicators of a latent L^i_{out} in G_{out} are measures of a latent node L^j in G, then we map L^i_{out} to L^j. Ties were in fact rare, but were broken randomly. At most one latent in G_{out} is mapped to a fixed latent L in G, and if a latent in G_{out} had no majority, it was not mapped to any latent in G.

The mapping for FA was done slightly differently. Because the output of FA is typically an extremely impure measurement model with many indicators loading on more than one latent, the simple minded majority method generates too many ties. For FA we do the mapping not by majority voting of indicators according to their true clusters, but by verifying which true latent corresponds to the highest sum of absolute values of factor loadings for a given output latent. For example, let L_{out} be a latent node in G_{out}. Suppose S_1 is the sum of the absolute values of the loadings of L_{out} on measures of the true latent L_1 only, and S_2 is the sum of the absolute values of the loadings of L_{out} on measures of the true latent L_2 only. If $S_2 > S_1$, we rename L_{out} as L_2. If two output latents are mapped to the same true latent, we label only one of them as the true latent by choosing the one that corresponds to the highest sum of absolute loadings.

We compute the following scores for the output model G_{out} from each algorithm, where the true graph is labelled G_I, and where G is a purification of G_I:

- **latent omission**, the number of latents in G that do not appear in G_{out} divided by the total number of true latents in G;
- **latent commission**, the number of latents in G_{out} that could not be mapped to a latent in G divided by the total number of true latents in G;
- **misclustered indicators**, the number of observed variables in G_{out} that end up in the wrong cluster divided by the number of observed variables in G;
- **indicator omission**, the number of observed variables in G that do not appear in the G_{out} divided by the total number of observed variables in G;
- **indicator commission**, the number of observed nodes in G_{out} that are not in G divided by the number of nodes in G that are not in G_I. These are nodes that introduce impurities in the output model;

To be generous to factor analysis we considered only latents with at least three indicators. Even with this help, we still found several cases in which latent commission errors were more than 100%. Again, to be conservative, we calculate the **misclustered indicators** error in the same way as in BuildPureClusters or P-FA. In this calculation, an indicator is not counted as mistakenly clustered if it is a child of the correct latent, even if it is *also* a child of a wrong latent.

Simulation results are given in Tables 3 and 4, where each number is the average error across 10 trials with standard deviations in parentheses for sample sizes of 200, 1000, 10,000. Notice there are at most two maximal pure measurement models for each setup (there are two possible choices of which measures to remove from the last latent in MM2 and MM3)

Evaluation of output measurement models									
	Latent omission			**Latent commission**			**Misclustered indicator**		
Sample	**BPC**	**FA**	**P-FA**	**BPC**	**FA**	**P-FA**	**BPC**	**FA**	**P-FA**
$SM_1 + MM_1$									
200	0.10(.2)	0.00(.0)	0.10(.2)	0.00(.0)	0.00(.0)	0.00(.0)	0.01(.0)	0.00(.0)	0.00(.0)
1000	0.17(.2)	0.00(.0)	0.13(.2)	0.00(.0)	0.00(.0)	0.03(.1)	0.00(.0)	0.01(.0)	0.01(.0)
10000	0.07(.1)	0.00(.0)	0.13(.2)	0.00(.0)	0.00(.0)	0.00(.0)	0.00(.0)	0.00(.0)	0.00(.0)
$SM_1 + MM_2$									
200	0.00(.0)	0.03(.1)	0.60(.3)	0.03(.1)	0.77(.2)	0.10(.2)	0.01(.0)	0.12(.1)	0.02(.0)
1000	0.00(.0)	0.00(.0)	0.17(.2)	0.00(.0)	0.47(.2)	0.27(.3)	0.00(.0)	0.08(.1)	0.10(.1)
10000	0.00(.0)	0.00(.0)	0.23(.2)	0.03(.1)	0.33(.3)	0.17(.2)	0.02(.1)	0.07(.1)	0.03(.1)
$SM_1 + MM_3$									
200	0.00(.0)	0.00(.0)	0.33(.3)	0.07(.1)	1.13(.3)	0.17(.2)	0.03(.1)	0.16(.1)	0.04(.1)
1000	0.00(.0)	0.00(.0)	0.30(.2)	0.07(.1)	0.87(.3)	0.33(.3)	0.03(.1)	0.12(.1)	0.06(.1)
10000	0.03(.1)	0.00(.0)	0.27(.3)	0.00(.0)	0.70(.3)	0.37(.3)	0.00(.0)	0.12(.1)	0.09(.1)
$SM_2 + MM_1$									
200	0.10(.2)	0.00(.0)	0.13(.2)	0.00(.0)	0.00(.0)	0.00(.0)	0.06(.1)	0.01(.0)	0.00(.0)
1000	0.03(.1)	0.00(.0)	0.17(.2)	0.00(.0)	0.00(.0)	0.00(.0)	0.02(.1)	0.00(.0)	0.00(.0)
10000	0.00(.0)	0.00(.0)	0.07(.1)	0.00(.0)	0.00(.0)	0.00(.0)	0.00(.0)	0.00(.0)	0.00(.0)
$SM_2 + MM_2$									
200	0.03(.1)	0.00(.0)	0.33(.2)	0.07(.1)	0.80(.3)	0.17(.2)	0.06(.1)	0.15(.1)	0.04(.1)
1000	0.00(.0)	0.00(.0)	0.27(.2)	0.00(.0)	0.53(.3)	0.23(.3)	0.00(.0)	0.08(.1)	0.06(.1)
10000	0.00(.0)	0.00(.0)	0.10(.2)	0.00(.0)	0.27(.3)	0.23(.3)	0.00(.0)	0.08(.1)	0.06(.1)
$SM_2 + MM_3$									
200	0.00(.0)	0.03(.1)	0.53(.2)	0.00(.0)	1.13(.3)	0.03(.1)	0.01(.0)	0.07(.1)	0.01(.0)
1000	0.00(.0)	0.00(.0)	0.27(.2)	0.00(.0)	0.73(.3)	0.13(.2)	0.00(.0)	0.08(.1)	0.03(.1)
10000	0.00(.0)	0.00(.0)	0.37(.2)	0.00(.0)	0.97(.3)	0.27(.3)	0.00(.0)	0.08(.1)	0.05(.1)
$SM_3 + MM_1$									
200	0.12(.2)	0.02(.1)	0.38(.2)	0.00(.0)	0.05(.1)	0.00(.0)	0.05(.1)	0.02(.1)	0.01(.0)
1000	0.10(.2)	0.02(.1)	0.12(.2)	0.00(.0)	0.02(.1)	0.00(.0)	0.01(.0)	0.02(.1)	0.00(.0)
10000	0.05(.1)	0.00(.0)	0.20(.1)	0.00(.0)	0.00(.0)	0.00(.0)	0.00(.0)	0.00(.0)	0.00(.0)
$SM_3 + MM_2$									
200	0.02(.1)	0.05(.2)	0.60(.2)	0.10(.2)	0.62(.1)	0.08(.2)	0.03(.1)	0.16(.3)	0.01(.0)
1000	0.02(.1)	0.02(.1)	0.30(.3)	0.02(.1)	0.38(.2)	0.10(.1)	0.01(.0)	0.18(.2)	0.07(.1)
10000	0.00(.0)	0.05(.1)	0.45(.2)	0.00(.0)	0.35(.2)	0.10(.2)	0.00(.0)	0.18(.2)	0.04(.1)
$SM_3 + MM_3$									
200	0.02(.1)	0.02(.1)	0.58(.2)	0.05(.1)	0.98(.3)	0.08(.1)	0.04(.1)	0.19(.2)	0.01(.0)
1000	0.02(.1)	0.08(.2)	0.35(.2)	0.00(.0)	0.72(.3)	0.08(.1)	0.00(.0)	0.23(.3)	0.03(.0)
10000	0.00(.0)	0.08(.1)	0.30(.3)	0.00(.0)	0.60(.3)	0.08(.1)	0.00(.0)	0.27(.3)	0.02(.0)

Table 3: Results obtained with BuildPureClusters (BPC), factor analysis (FA) and purified factor analysis (P-FA) for the problem of learning measurement models. Each number is an average over 10 trials, with the standard deviation over these trials in parenthesis.

and for each G_{out} we choose our gold standard G as a maximal pure measurement submodel that contains the most number of nodes found in G_{out}.

Evaluation of output measurement models				
	Indicator omission		**Indicator commission**	
Sample	**BPC**	**P-FA**	**BPC**	**P-FA**
$SM_1 + MM_1$				
200	0.12(.2)	0.41(.3)	– – –	– – –
1000	0.18(.2)	0.19(.2)	– – –	– – –
10000	0.09(.2)	0.14(.2)	– – –	– – –
$SM_1 + MM_2$				
200	0.08(.0)	0.87(.1)	0.07(.1)	0.07(.1)
1000	0.07(.1)	0.46(.2)	0.00(.0)	0.13(.2)
10000	0.06(.1)	0.38(.2)	0.03(.1)	0.10(.2)
$SM_1 + MM_3$				
200	0.17(.1)	0.78(.2)	0.04(.1)	0.08(.1)
1000	0.12(.1)	0.58(.2)	0.06(.1)	0.10(.2)
10000	0.13(.1)	0.42(.3)	0.00(.0)	0.06(.1)
$SM_2 + MM_1$				
200	0.10(.1)	0.43(.2)	– – –	– – –
1000	0.03(.1)	0.23(.2)	– – –	– – –
10000	0.03(.1)	0.11(.1)	– – –	– – –
$SM_2 + MM_2$				
200	0.16(.1)	0.77(.1)	0.30(.3)	0.03(.1)
1000	0.06(.1)	0.57(.1)	0.00(.0)	0.07(.2)
10000	0.06(.1)	0.31(.2)	0.00(.0)	0.10(.2)
$SM_2 + MM_3$				
200	0.16(.1)	0.85(.1)	0.18(.2)	0.04(.1)
1000	0.08(.1)	0.56(.2)	0.02(.1)	0.10(.1)
10000	0.05(.1)	0.72(.1)	0.00(.0)	0.16(.1)
$SM_3 + MM_1$				
200	0.14(.1)	0.65(.2)	– – –	– – –
1000	0.12(.2)	0.28(.2)	– – –	– – –
10000	0.08(.1)	0.21(.1)	– – –	– – –
$SM_3 + MM_2$				
200	0.14(.1)	0.84(.1)	0.10(.2)	0.02(.1)
1000	0.11(.1)	0.51(.2)	0.00(.0)	0.02(.1)
10000	0.05(.0)	0.56(.2)	0.00(.0)	0.02(.1)
$SM_3 + MM_3$				
200	0.14(.1)	0.87(.1)	0.17(.1)	0.02(.1)
1000	0.13(.1)	0.66(.1)	0.03(.1)	0.07(.1)
10000	0.13(.1)	0.52(.2)	0.00(.0)	0.08(.1)

Table 4: Results obtained with BuildPureClusters (BPC) and purified factor analysis (P-FA) for the problem of learning measurement models. Each number is an average over 10 trials, with standard deviations in parens.

Table 3 evaluates all three procedures on the first two discovery tasks: DP1 and DP2. As expected, all three procedures had very low error rates in rows involving MM1 and sample sizes of 10,000. Over all conditions, FA has very low rates of latent omission, but very high rates of latent commission, and P-FA, not surprisingly, does the opposite: very

high rates of latent omission but very low rates of commission. In particular, FA is very sensitive to the purity of the generating measurement model. With MM2, the rate of latent commission for FA was moderate; with MM3 it was horrendous. BPC does reasonably well on all measures in Tables 3 at all sample sizes and for all generating models.

Table 4 gives results regarding indicator omissions and commission, which, because FA keeps the original set of indicators it is given, only make sense for BPC and P-FA. P-FA omits far too many indicators, a behavior that we hypothesize will make it difficult for GES-MIMBuild on the measurement model output by P-FA.

In the final piece of the simulation study, we applied the best causal model search algorithm we know of, GES, modified for this purpose as GES-MIMbuild, to the measurement models output by BPC, FA, and P-FA.

If the output measurement model has no errors of latent omission or commission, then scoring the result of the structural model search is fairly easy. The GES-MIMbuild search outputs an equivalence class, with certain adjacencies unoriented and certain adjacencies oriented. If there is an adjacency of any sort between two latents in the output, but no such adjacency in the true graph, then we have an error of edge commission. If there is no adjacency of any sort between two latents in the output, but there is an edge in the true graph, then we have an error of edge omission. For orientation, if there is an oriented edge in the output that is not oriented in the equivalence class for the true structural model, then we have an error of orientation commission. If there is an unoriented edge in the output which is oriented in the equivalence class for the true model, we have an error of orientation omission.

If the output measurement model has any errors of latent commission, then we simply leave out the excess latents in the measurement model given to GES-MIMbuild. This helps FA primarily, as it was the only procedure of the three that had high errors of latent commission.

If the output measurement model has errors of latent omission, then we compare the marginal involving the latents in the output model for the true structural model graph to the output structural model equivalence class. For each of the structural models we selected, SM1, SM2, and SM3, all marginals can be represented faithfully as DAGs. Our measure of successful causal discovery, therefore, for a measurement model involving a small subset of the latents in the true graph is very lenient. For example, if the generating model was SM3, which involves four latents, but the output measurement model involved only two of these latents, then a perfect search result in this case would amount to finding that the two latents are associated. This feature of our evaluation procedure favors P-FA, which tends to omit latents.

In summary then, our measures for assessing the ability of these algorithms to correctly discover at least features of the causal relationships among the latents are as follows:

- **edge omission (EO)**, the number of edges in the structural model of G that do not appear in G_{out} divided by the possible number of edge omissions (2 in SM_1 and SM_2, and 4 in SM_3, i.e., the number of edges in the respective structural models);
- **edge commission (EC)**, the number of edges in the structural model of G_{out} that do not exist in G divided by the possible number of edge commissions (only 1 in SM_1 and SM_2, and 2 in SM_3);

- **orientation omission (OO)**, the number of arrows in the structural model of G that do not appear in G_{out} divided by the possible number of orientation omissions in G (2 in SM_1 and SM_3, 0 in SM_2);
- **orientation commission (OC)**, the number of arrows in the structural model of G_{out} that do not exist in G divided by the number of edges in the structural model of G_{out};

We have bent over, not quite backwards, to favor variations of factor analysis. Tables 5 and 5 summarize the results. Along with each average we provide the number of trials where no errors of a specific type were made.

Although P-FA seems more reliable than FA, this is in reality an artifact of P-FA outputting fewer latents on average than FA and BPC. Although it is clear from Tables 5 and 6 that factor analysis works well when the true models are pure, in cases in which the generating measurement model is impure factor analysis commits high proportions of edge commission. This in turn is because FA commits so many latents, which leads to spurious dependence paths among the latents we scored, which leads to orientation omissions follow.

Figure 7 illustrates the trade-off each of the three procedures make between latent omission/commission and accuracy. Each picture contains a plot of the latent omission against the average edge error of each algorithm (i.e., the average of all four error statistics from Tables 5 and 6), with three points plotted for each algorithm representing different sample sizes (grouped graphically). The optimal performance is the bottom left, where latent omission and edge error are low. The top plot shows the performance of the algorithms on data from a pure measurement model, the middle plot on data from a slightly impure measurement model, and the bottom plot on data from a highly impure measurement model. It is clear that P-FA achieves relatively high accuracy solely because of high percentage of latent omission. FAachieves high accuracy only when the true measurement model is pure. Otherwise it makes very high errors in edge accuracy. BuildPureClusters finds the right balance between latent omission and edge accuracy, and its performance relative to the other procedures improves as the generating measurement model becomes more and more complex. We take this to be the most important feature of our algorithm, as real data is rarely generated from a pure measurement model. Although 7shows the performance of the algorithms for SM2, the pattern of performance was similar across all structural models.

In summary, factor analysis provides little useful information out of the given datasets. In contrast, the combination of BuildPureClusters and GES-MIMBuild largely succeeds in such a difficult task, even at small sample sizes.

8. Real data applications

We now briefly present the results for two real data sets. Data collected from such domains may pose significant problems for exploratory data analysis since sample sizes are usually small and noisy, nevertheless they have a very useful property for our empirical evaluation. In particular, data obtained by questionnaires are designed to target specific latent factors (such as "stress", "job satisfaction", and so on) and a theoretical measurement model is developed by experts in the area to measure the desired latent variables. Very generally, experts are more confident about their choice of measures than about the structural model.

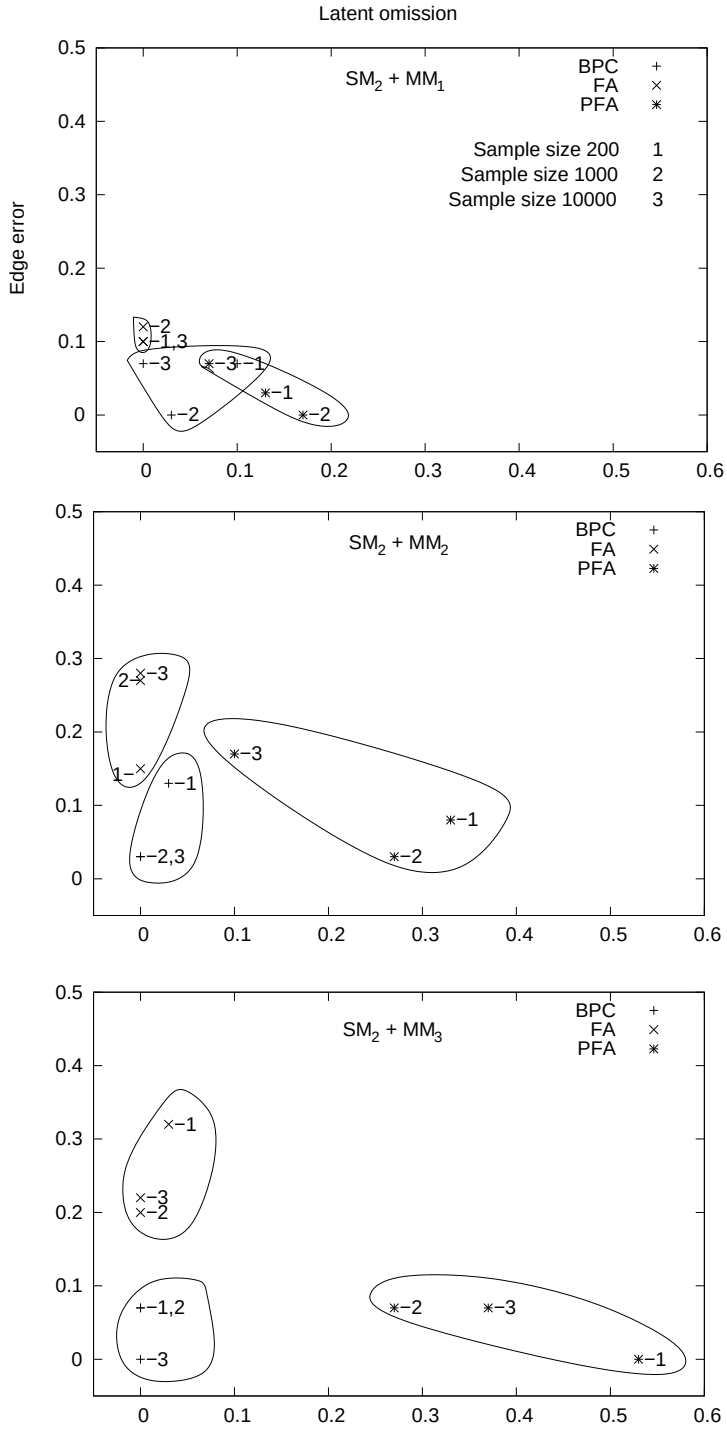


Figure 7: Comparisons of methods on measurement models of increasing complexity (from MM_1 to MM_3). While BPC tends to have low error on both dimensions (latent omission and edge error), the other two methods fail on either one.

Evaluation of output structural models						
	Edge omission			Edge commission		
Sample	**BPC**	**FA**	**P-FA**	**BPC**	**FA**	**P-FA**
$SM_1 + MM_1$						
200	$0.05 - 09$	$0.05 - 09$	$0.10 - 08$	$0.10 - 09$	$0.30 - 07$	$0.20 - 08$
1000	$0.05 - 09$	$0.10 - 08$	$0.05 - 09$	$0.20 - 08$	$0.30 - 07$	$0.10 - 09$
10000	$0.00 - 10$	$0.05 - 09$	$0.15 - 07$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$
$SM_1 + MM_2$						
200	$0.00 - 10$	$0.15 - 07$	$0.00 - 10$	$0.00 - 10$	$0.40 - 06$	$0.00 - 10$
1000	$0.00 - 10$	$0.00 - 10$	$0.15 - 07$	$0.10 - 09$	$0.40 - 06$	$0.20 - 08$
10000	$0.00 - 10$	$0.05 - 09$	$0.25 - 05$	$0.20 - 08$	$0.50 - 05$	$0.20 - 08$
$SM_1 + MM_3$						
200	$0.00 - 10$	$0.25 - 05$	$0.05 - 09$	$0.20 - 08$	$0.70 - 03$	$0.10 - 09$
1000	$0.00 - 10$	$0.15 - 07$	$0.10 - 08$	$0.10 - 09$	$0.70 - 03$	$0.10 - 09$
10000	$0.00 - 10$	$0.05 - 09$	$0.05 - 09$	$0.00 - 10$	$0.40 - 06$	$0.20 - 08$
$SM_2 + MM_1$						
200	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$	$0.20 - 08$	$0.30 - 07$	$0.10 - 09$
1000	$0.00 - 10$	$0.05 - 09$	$0.00 - 10$	$0.00 - 10$	$0.30 - 07$	$0.00 - 10$
10000	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$	$0.20 - 08$	$0.30 - 07$	$0.20 - 08$
$SM_2 + MM_2$						
200	$0.00 - 10$	$0.15 - 07$	$0.05 - 09$	$0.40 - 06$	$0.30 - 07$	$0.20 - 08$
1000	$0.00 - 10$	$0.10 - 09$	$0.00 - 10$	$0.10 - 09$	$0.60 - 04$	$0.10 - 09$
10000	$0.00 - 10$	$0.05 - 09$	$0.00 - 10$	$0.10 - 09$	$0.70 - 03$	$0.50 - 05$
$SM_2 + MM_3$						
200	$0.00 - 10$	$0.15 - 07$	$0.00 - 10$	$0.20 - 08$	$0.70 - 03$	$0.00 - 10$
1000	$0.00 - 10$	$0.15 - 07$	$0.00 - 10$	$0.20 - 08$	$0.40 - 06$	$0.20 - 08$
10000	$0.00 - 10$	$0.10 - 08$	$0.00 - 10$	$0.00 - 10$	$0.50 - 05$	$0.20 - 08$
$SM_3 + MM_1$						
200	$0.12 - 05$	$0.12 - 06$	$0.08 - 08$	$0.20 - 06$	$0.20 - 06$	$0.10 - 09$
1000	$0.05 - 08$	$0.08 - 08$	$0.08 - 07$	$0.15 - 08$	$0.10 - 08$	$0.15 - 07$
10000	$0.05 - 08$	$0.15 - 04$	$0.15 - 04$	$0.15 - 08$	$0.15 - 08$	$0.05 - 09$
$SM_3 + MM_2$						
200	$0.02 - 09$	$0.28 - 03$	$0.05 - 08$	$0.55 - 03$	$0.55 - 02$	$0.00 - 10$
1000	$0.00 - 10$	$0.12 - 07$	$0.05 - 08$	$0.25 - 07$	$0.75 - 02$	$0.20 - 07$
10000	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$	$0.10 - 08$	$0.80 - 02$	$0.00 - 10$
$SM_3 + MM_3$						
200	$0.02 - 09$	$0.32 - 02$	$0.08 - 07$	$0.40 - 05$	$0.50 - 02$	$0.00 - 10$
1000	$0.08 - 07$	$0.02 - 09$	$0.10 - 06$	$0.30 - 06$	$0.65 - 02$	$0.00 - 10$
10000	$0.00 - 10$	$0.05 - 08$	$0.02 - 09$	$0.15 - 07$	$0.65 - 03$	$0.25 - 07$

Table 5: Results obtained with the application of GES-MIMBuild to the output of BPC, FA, and P-FA, with the number of perfect solutions over ten trials on the right of each result.

Such data thus provide a basis for comparison with the output of our algorithm. The chance that various observed variables are not pure measures of their theoretical latents is high. Measures are usually discrete, but often ordinal with a Likert-scale that can be treated as

Evaluation of output structural models						
	Orientation omission			Orientation commission		
Sample	**BPC**	**FA**	**P-FA**	**BPC**	**FA**	**P-FA**
$SM_1 + MM_1$						
200	$0.10 - 09$	$0.15 - 08$	$0.05 - 09$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$
1000	$0.20 - 08$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$	$0.05 - 09$	$0.00 - 10$
10000	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$
$SM_1 + MM_2$						
200	$0.00 - 10$	$0.20 - 07$	$0.00 - 10$	$0.00 - 10$	$0.05 - 09$	$0.00 - 10$
1000	$0.10 - 09$	$0.20 - 07$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$
10000	$0.20 - 08$	$0.25 - 05$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$
$SM_1 + MM_3$						
200	$0.20 - 08$	$0.40 - 04$	$0.10 - 09$	$0.00 - 10$	$0.05 - 09$	$0.00 - 10$
1000	$0.10 - 09$	$0.10 - 09$	$0.10 - 09$	$0.00 - 10$	$0.10 - 08$	$0.00 - 10$
10000	$0.00 - 10$	$0.30 - 06$	$0.10 - 09$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$
$SM_2 + MM_1$						
200	$- - -$	$- - -$	$- - -$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$
1000	$- - -$	$- - -$	$- - -$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$
10000	$- - -$	$- - -$	$- - -$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$
$SM_2 + MM_2$						
200	$- - -$	$- - -$	$- - -$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$
1000	$- - -$	$- - -$	$- - -$	$0.00 - 10$	$0.10 - 09$	$0.00 - 10$
10000	$- - -$	$- - -$	$- - -$	$0.00 - 10$	$0.10 - 09$	$0.00 - 10$
$SM_2 + MM_3$						
200	$- - -$	$- - -$	$- - -$	$0.00 - 10$	$0.10 - 08$	$0.00 - 10$
1000	$- - -$	$- - -$	$- - -$	$0.00 - 10$	$0.05 - 09$	$0.00 - 10$
10000	$- - -$	$- - -$	$- - -$	$0.00 - 10$	$0.05 - 09$	$0.00 - 10$
$SM_3 + MM_1$						
200	$0.15 - 08$	$0.00 - 10$	$0.00 - 10$	$0.22 - 07$	$0.35 - 06$	$0.15 - 08$
1000	$0.10 - 09$	$0.00 - 10$	$0.05 - 09$	$0.10 - 09$	$0.00 - 10$	$0.04 - 09$
10000	$0.05 - 09$	$0.00 - 10$	$0.00 - 10$	$0.04 - 09$	$0.00 - 10$	$0.00 - 10$
$SM_3 + MM_2$						
200	$0.50 - 05$	$0.30 - 06$	$0.00 - 10$	$0.08 - 09$	$0.16 - 07$	$0.00 - 10$
1000	$0.30 - 07$	$0.45 - 04$	$0.10 - 09$	$0.00 - 10$	$0.05 - 09$	$0.04 - 09$
10000	$0.20 - 08$	$0.40 - 06$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$
$SM_3 + MM_3$						
200	$0.50 - 04$	$0.15 - 08$	$0.00 - 10$	$0.19 - 06$	$0.14 - 08$	$0.10 - 09$
1000	$0.20 - 07$	$0.35 - 05$	$0.00 - 10$	$0.15 - 07$	$0.02 - 09$	$0.10 - 09$
10000	$0.00 - 10$	$0.35 - 05$	$0.20 - 07$	$0.00 - 10$	$0.00 - 10$	$0.00 - 10$

Table 6: Results obtained with the application of GES-MIMBuild to the output of BPC, FA, and P-FA, with the number of perfect solutions over ten trials on the right of each result.

normally distributed measures with little loss (Bollen, 1989). In the examples, we compare our procedures with models produced by domain researchers.

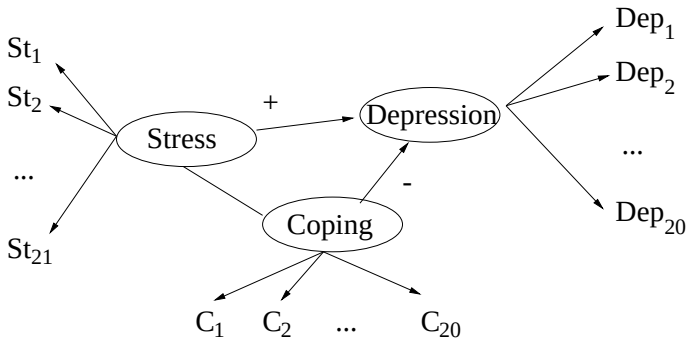


Figure 8: A theoretical model for the interaction of religious coping, stress and depression. The signs on the edges depicts the theoretical signs of the corresponding effects.

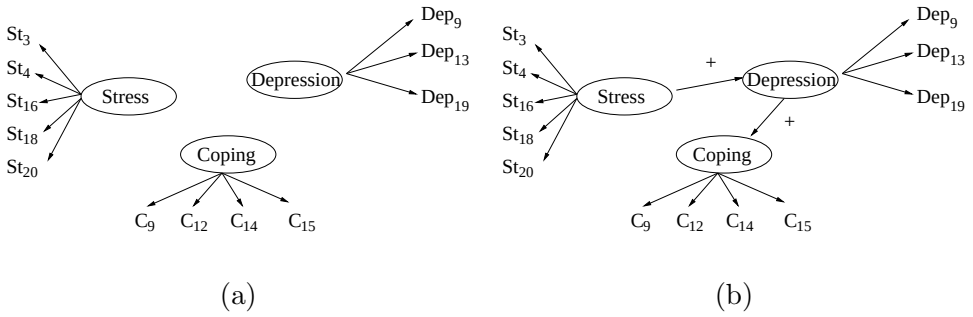


Figure 9: The output of BPC and GES-MIMBuild for the coping study.

8.1 Stress Religious Coping and Depression

Bongjae Lee from the University of Pittsburgh conducted a study of religious/spiritual coping and stress in graduate students. In December of 2003, 127 students answered a questionnaire intended to measure three main factors: stress (measured with 21 items), depression (measured with 20 items) and religious/spiritual coping (measured with 20 items). The full questionnaire is given by Silva and Scheines (2004). Lee's model is shown in Figure 8.

This model fails a chi square test: p = 0. The measurement model produced by BuildPureClusters is shown in Figure 9(a). Note that the variables selected automatically are proper subsets of Lee's substantive clustering. The full model automatically produced with GEM-MIMBUILD with the prior knowledge that STRESS is not an effect of other latent variables is given in Figure 9(b). This model passes a chi square test: p= 0.28.

8.2 Test Anxiety

In the now standard text on exploratory factor analysis, Bartholomew illustrates the procedure with data from a 20 item survey of test anxiety from 12th grade males in British

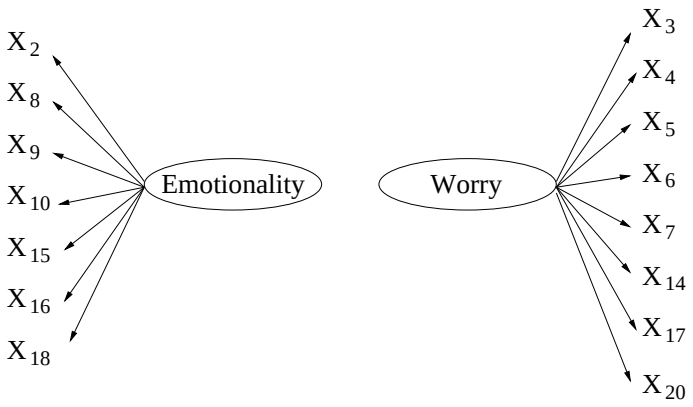


Figure 10: A theoretical model for psychological factors of test anxiety.

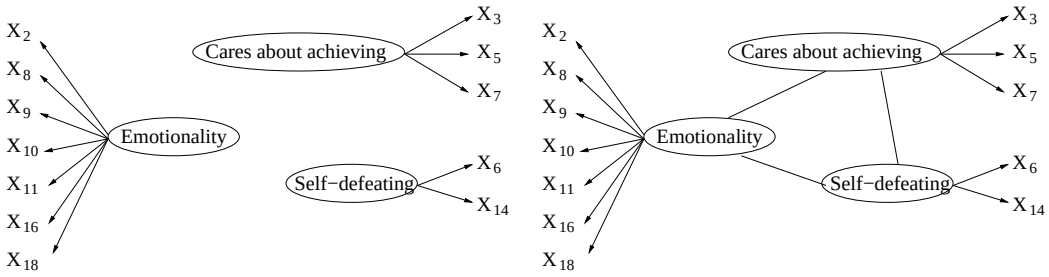


Figure 11: The output of BPC and GES-MIMBuild for the test anxiety study.

Columbia (sample size $N = 335$)[4]. Bartholomew's exploratory factor analysis model for a subset of the variables is shown in Figure 10. The original model is not pure. We show a purified version by keeping only the strongest edge for each possible latent parent. The BuildPureClusters measurement model, with our interpretation of the latents based on the questionnaire contents, is shown in Figure 11(a).

If we correlate the latent variables of the purified factor analysis model, the result is a model that fails a chi-square test, p = 0. Applying GEM-MIMBUILD to the BPC measurement model of Figure 11(a) we obtain Figure 11(b). The model passes a chi square test handily, p = 0.47.

9. Generalizations

In many social science studies, latent structure is represented by so called "non-recursive" structure. In graphical terms, the dependency graph is cyclic. Richardson (1996) has developed a consistent constraint based search for cyclic graphical models of linear systems,

4. The data are available online at http://multilevel.ioe.ac.uk/team/aimdss.html.

and our procedures for identifying measurement models can be combined with it to search for such structure.

The procedure we have described here can, however, straightforwardly be generalized to cases with measured variables taking a small finite range of values by treating the discrete variables as projections from a Gaussian distribution. Much larger sample sizes are required than for linear, Gaussian measured variables.

In a previous work (Silva et al., 2003), we developed an approach to learn measurement models even when the functional relationships among latents are non-linear. In practice, that generality is of limited use because there are at present no consistent search methods available for structures with continuous, non-linear variables. Theorem 14 proved here makes slightly weaker claims than its counterpart given by Silva et al. (2003). Theorem 15 becomes a necessary addition, since the output might not be a subgraph of the true graph. What is needed to extend our results here to systems in which the latent structure is non-linear are results analogous to these two theorems.

10. Conclusion

Our experiments provide evidence that our procedures can be useful in practice, but there are certainly classes of problems where BuildPureClusters will not be of practical value. For instance, learning the causal structure of general blind source separation problems, where measures are usually indicators of most of the latents (i.e., sources) at the same time.

A number of open problems invite further research, including these:

- completeness of the tetrad equivalence class of measurement models: can we identify all the common features of measurement models in the same tetrad equivalence class?
- Bayesian learning of measurement models. The given identification rules (i.e., CS1, CS2, and CS3) already provide principled search operators that can be used to create a GES-like algorithm for learning this type of models, although the difficulty in defining a consistent score function might limit the theoretical guarantees of this approach;
- using the more generic rank constraints of covariance matrices to learn measurement models, possibly identifying the nature of some impure relationships;
- better treatment of discrete variables. Bartholomew and Knott (1999) survey different ways of integrating factor analysis and discrete variables that can be readily adapted, but the computational cost of this procedure is high;
- finding non-linear causal relationships among latent variables given a fixed linear measurement model, and in other families of multivariate continuous distributions besides the Gaussian;

The fundamental point is that common and appealing heuristics (e.g., factor rotation methods) fail when the goal is structure learning with a causal interpretation. In many cases it is preferable to model the relationships of a subset of the given variables than trying to force a bad model over all of them (Kano and Harada, 2000). Better methods are available now, and further improvements will surely come from machine learning research.

Acknowledgments

Research for this paper was supported by NASA NCC 2-1377 to the University of West Florida, NASA NRA A2-37143 to CMU and ONR contract N00014-03-01-0516 to the University of West Florida.

References

H. Attias. Independent factor analysis. *Graphical Models: foundations of neural computation*, pages 207–257, 1999.

F. Bach and M. Jordan. Beyond independent components: trees and clusters. *Journal of Machine Learning Research*, 4:1205–1233, 2003.

D. Bartholomew and M. Knott. *Latent Variable Models and Factor Analysis*. Arnold Publishers, 1999.

D. Bartholomew, F. Steele, I. Moustaki, and J. Galbraith. *The Analysis and Interpretation of Multivariate Data for Social Scientists*. Arnold Publishers, 2002.

K. Bollen. *Structural Equation Models with Latent Variables*. John Wiley & Sons, 1989.

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

G. Elidan, N. Lotner, N. Friedman, and D. Koller. Discovering hidden variables: a structure-based approach. *Neural Information Processing Systems*, 13:479–485, 2000.

N. Friedman. The bayesian structural em algorithm. *Proceedings of 14th Conference on Uncertainty in Artificial Intelligence*, 1998.

D. Geiger and C. Meek. Quantifier elimination for statistical problems. *Proceedings of 15th Conference on Uncertainty in Artificial Intelligence*, 1999.

C. Glymour. Social statistics and genuine inquiry: reflections on *the bell curve. Intelligence, Genes and Sucess: Scientists Respond to The Bell Curve*, 1997.

Y. Kano and A. Harada. Stepwise variable selection in factor analysis. *Psychometrika*, 65: 7–22, 2000.

J. Loehlin. *Latent Variable Models: An Introduction to Factor, Path and Structural Equation Analysis*. Lawrence Erlbaum, 2004.

C. Meek. *Graphical Models: Selecting Causal and Statistical Models*. PhD Thesis, Carnegie Mellon University, 1997.

J. Pearl. *Probabilistic Reasoning in Expert Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

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

T. Richardson. A discovery algorithm for directed cyclic graphs. *Proceedings of 12th Conference on Uncertainty in Artificial Intelligence*, 1996.

G. Shafer, A. Kogan, and P.Spirtes. Generalization of the tetrad representation theorem. *DIMACS Technical Report*, 1993.

R. Silva, R. Scheines, C. Glymour, and P. Spirtes. Learning measurement models for unobserved variables. *Proceedings of 19th Conference on Uncertainty in Artificial Intelligence*, pages 543–550, 2003.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and Search*. Cambridge University Press, 2000.

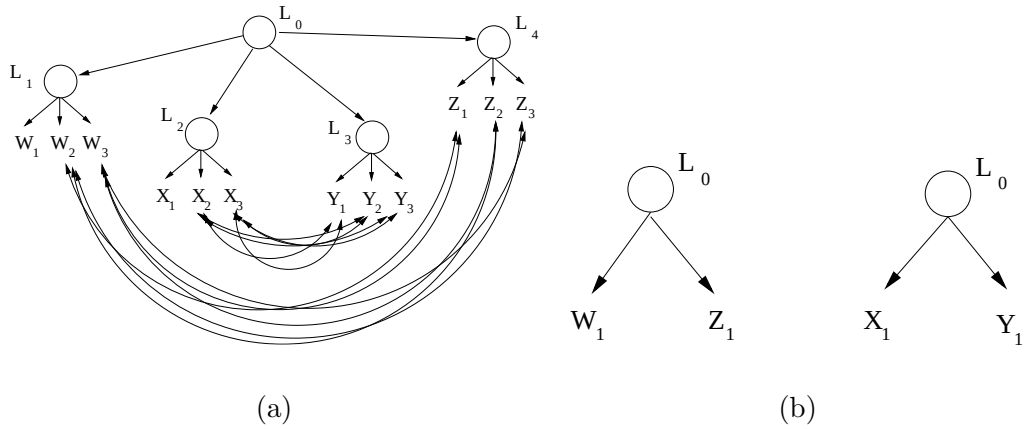
N. Zhang. Hierarchical latent class models for cluster analysis. *Journal of Machine Learning Research*, 5:697–723, 2004.

Appendix A. BuildPureClusters: refinement steps

Concerning the final steps of Table 2, it might be surprising that we merge clusters of variables that we know cannot share a common latent parent in the true graph. However, we are not guaranteed to find a large enough number of pure indicators for each of the original latent parents, and as a consequence only a subset of the true latents will be represented in the measurement pattern. It might be the case that, with respect to the variables present in the output, the observed variables in two different clusters might be directly measuring some ancestor common to all variables in these two clusters. As an illustration, consider the graph in Figure 12(a), where double-directed edges represent independent hidden common causes. Assume any sensible purification procedure will choose to eliminate all elements in $\{W_2, W_3, X_2, X_3, Y_2, Y_3, Z_2, Z_3\}$ because they are directly correlated with a large number of other observed variables (extra edges and nodes not depicted).

Meanwhile, one can verify that all three tetrad constraints hold in the covariance matrix of $\{W_1, X_1, Y_1, Z_1\}$, and therefore there will be no undirected edges connecting pairs of elements in this set in the corresponding measurement pattern. Rule CS1 is able to separate W_1 and X_1 into two different clusters by using $\{W_2, W_3, X_2, X_3\}$ as the support nodes, and analogously the same happens to Y_1 and Z_1, W_1 and Y_1, X_1 and Z_1. However, no test can separate W_1 and Z_1, nor X_1 and Y_1. If we do not merge clusters, we will end up with the graph seen in Figure 12(b) as part of our output pattern. Although this is a valid measurement pattern, and in some situations we might want to output such a model, it is also true that W_1 and Z_1 measure a same latent L_0 (as well as X_1 and Y_1). It would be problematic to learn a structural model with such a measurement model. There is a deterministic relation between the latent measured by W_1 and Z_1, and the latent measured by X_1 and Y_1: they are the same latent! Probability distributions with deterministic relations are not faithful, and that causes problems for learning algorithms.

Finally, we show examples where Steps 6 and 7 of BuildPureClusters are necessary. In Figure 13(a) we have a partial view of a latent variable graph, where two of the latents are marginally independent. Suppose that nodes X_4, X_5 and X_6 are correlated to many other measured nodes not in this figure, and therefore are removed by our purification



(a) (b)

Figure 12: The true graph in (a) will generate at some point a purified measurement pattern as in (b). It is desirable to merge both clusters.

procedure. If we ignore Step 6, the resulting pure submodel over $\{X_1, X_2, X_3, X_7, X_8, X_9\}$ will be the one depicted in Figure 13(b) ($\{X_1, X_2\}$ are clustered apart from $\{X_7, X_8, X_9\}$ because of marginal zero correlation, and X_3 is clustered apart from $\{X_7, X_8, X_9\}$ because of CS1 applied to $\{X_3, X_4, X_5\} \times \{X_7, X_8, X_9\}$). However, no linear latent variable model can be parameterized by this graph: if we let the two latents to be correlated, this will imply X_1 and X_7 being correlated. If we make the two latents uncorrelated, X_3 and X_7 will be uncorrelated.

Step 7 exists to avoid rare situations where three observed variables are clustered together and are *pairwise* part of some foursome entailing all three tetrad constraints with no vanishing marginal and partial correlation, but still should be removed because they are not *simultaneously* in such a foursome. They might not be detected by Step 4 if, e.g., all three of them are uncorrelated with all other remaining observed variables.

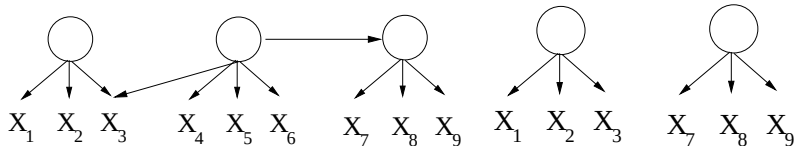
Appendix B. Proofs

Before we present the proofs of our results, we need a few more definitions:

- a *path* in a graph G is a sequence of nodes $\{X_1, \ldots, X_n\}$ such that X_i and X_{i+1} are adjacent in G, $1 \leq i < n$. Paths are assumed to be *simple* by definition, i.e., no node appears more than once. Notice there is an unique set of edges associated with each given path. A path is *into* X_1 (or X_n) if the arrow of the edge $\{X_1, X_2\}$ is into X_1 ($\{X_{n-1}, X_n\}$ into X_n);

- a *collider* on a path $\{X_1, \ldots, X_n\}$ is a node X_i, $1 < i < n$, such that X_{i-1} and X_{i+1} are parents of X_i;

- a *trek* is a path that does not contain any collider;



(a) (b)

Figure 13: Suppose (a) is our true model. If for some reason we need to remove nodes X_4, X_5 and X_6 from our final pure graph, the result will be as shown in Figure (b), unless we apply Step 6 of BuildPureClusters. There are several problems with (b), as explained in the text.

- the *source* of a trek is the unique node in a trek to which no arrows are directed;
- the *I side* of a trek between nodes I and J with source X is the subpath directed from X to I. It is possible that $X = I$, and the I side is just node I;
- a *choke point* CP between two sets of nodes $\mathbf{I}$ and $\mathbf{J}$ is a node that lies on every trek between any element of $\mathbf{I}$ and any element of $\mathbf{J}$ such that CP is either (i) on the $\mathbf{I}$ side of every such trek [5] or (ii) on the $\mathbf{J}$ side or every such trek.

Given these definitions, we state the Tetrad Representation Theorem as follows:

Theorem 20 (The Tetrad Representation Theorem) *Let G be a linear latent variable model, and let I_1, I_2, J_1, J_2 be four variables in G. Then $\sigma_{I_1J_1}\sigma_{I_2J_2} = \sigma_{I_1J_2}\sigma_{I_2J_1}$ if and only if there is a choke point between $\{I_1, I_2\}$ and $\{J_1, J_2\}$.*

Proof: The original proof was given by Spirtes et al. (2000). Shafer et al. (1993) provide an alternative and simplied proof. □

We will use the Tetrad Representation Theorem to prove most of our results. Shafer et al. (1993) also provide more details on the definitions and several examples.

In the following proofs, we will frequently use the symbol $G(\mathbf{O})$ to represent a linear latent variable model with a set of observed nodes $\mathbf{O}$. A choke point between sets $\mathbf{I}$ and $\mathbf{J}$ will be denoted as $\mathbf{I} \times \mathbf{J}$. We will first introduce a lemma that is going to be useful to prove several other results:

Lemma 21 *Let $G(\mathbf{O})$ be a linear latent variable model, and let $\{X_1, X_2, X_3, X_4\} \subset \mathbf{O}$ be such that $\sigma_{X_1X_2}\sigma_{X_3X_4} = \sigma_{X_1X_3}\sigma_{X_2X_4} = \sigma_{X_1X_4}\sigma_{X_2X_3}$. If $\rho_{AB} \neq 0$ for all $\{A, B\} \subset \{X_1, X_2, X_3, X_4\}$, then an unique choke point P entails all the given tetrad constraints, and P d-separates all elements in $\{X_1, X_2, X_3, X_4\}$.*

5. That is, for every $\{I, J\} \in \mathbf{I} \times \mathbf{J}$, CP is on the I side of every trek $T = \{I, \ldots, X, \ldots, J\}$, X being the source of T.

Proof: Let P be a choke point for pairs $\{X_1, X_2\} \times \{X_3, X_4\}$. Let Q be a choke point for pairs $\{X_1, X_3\} \times \{X_2, X_4\}$. We will show that $P = Q$ by contradiction.

Assume $P \neq Q$. Because there is a trek that links X_1 and X_4 throught P (since $\rho_{X_1X_4} \neq 0$), we have that Q should also be on that trek. Suppose T is a trek connecting X_1 to X_4 through P and Q, and without loss of generality assume this trek follows an order that defines three subtreks: T_0, from X_1 to P; T_1, from P to Q; and T_2, from Q to X_4, as illustrated by Figure 14(a). In principle, T_0 and T_2 might be empty, i.e., we are not excluding the possibility that $X_1 = P$ or $X_4 = Q$.

There must be at least one trek T_{Q2} connecting X_2 and Q, since Q is on every trek between X_1 and X_2 and there is at least one such trek (since $\rho_{X_1X_2} \neq 0$). We have the following cases:

Case 1: T_{Q2} includes P. T_{Q2} has to be into P, and $P \neq X_1$, or otherwise there will be a trek connecting X_2 to X_1 through a (possibly empty) trek T_0 that does not include Q, contrary to our hypothesis. For the same reason, T_0 has to be into P. This will imply that T_1 is a directed path from P to Q, and T_2 is a directed path from Q to X_4 (Figure 14(b)).

Because there is at least one trek connecting X_1 and X_2 (since $\rho_{X_1X_2} \neq 0$), and because Q is on every such trek, Q has to be an ancestor of at least one member of $\{X_1, X_2\}$. Without loss of generality, assume Q is an ancestor of X_1. No directed path from Q to X_1 can include P, since P is an ancestor of Q and the graph is acyclic. Therefore, there is a trek connecting X_1 and X_4 with Q as the source that does not include P, contrary to our hypothesis.

Case 2: T_{Q2} does not include P. This is case is similar to Case 1. T_{Q2} has to be into Q, and $Q \neq X_4$, or otherwise there will be a trek connecting X_2 to X_4 through a (possible empty) trek T_2 that does not include P, contrary to our hypothesis. For the same reason, T_2 has to be into P. This will imply that T_1 is a directed path from Q to P, and T_0 is a directed path from P to X_1. An argument analogous to Case 1 will follow.

We will now show by contradiction that P d-separates all nodes in $\{X_1, X_2, X_3, X_4\}$. From the $P = Q$ result, we know that P lies on every trek between any pair of elements in $\{X_1, X_2, X_3, X_4\}$. First consider the case where at most one element of $\{X_1, X_2, X_3, X_4\}$ is linked to P through a trek that is into P. By the Tetrad Representation Theorem, any trek connecting two elements of $\{X_1, X_2, X_3, X_4\}$ goes through P. Since P cannot be a collider on any trek, then P d-separates these two elements.

Without loss of generality, assume there is a trek connecting X_1 and P that is into P, and a trek connecting X_2 and P that is into P. If there is no trek connecting X_1 and P that is out of P neither any trek connecting X_2 and P that is out of P, then there is no trek connecting X_1 and X_2, since P is on every trek connecting these two elements according to the Tetrad Representation Theorem. But this implies $\rho_{X_1X_2} = 0$, a contradiction, as illustrated by Figure 14(c).

Consider the case where there is also a trek out of P and into X_2. Then there is a trek connecting X_1 to X_2 through P that is not on the $\{X_1, X_3\}$ side of pair $\{X_1, X_3\} \times \{X_2, X_4\}$ to which P is a choke point. Therefore, P should be on the $\{X_2, X_4\}$ of every trek connecting elements pairs in $\{X_1, X_3\} \times \{X_2, X_4\}$. Without loss of generality, assume there is a trek out of P and into X_3 (because if there is no such trek for either X_3 and X_4, we fall in

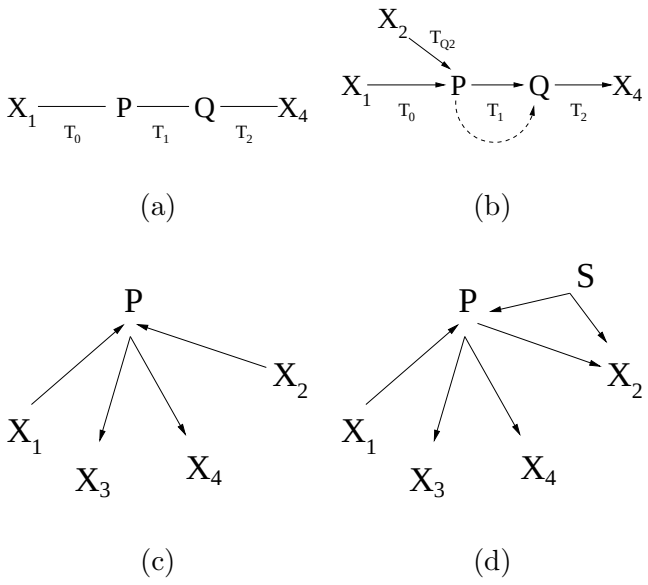


Figure 14: In (a), a depiction of a trek T linking X_1 and X_4 through P and Q, creating three subtreks labeled as T_0, T_1 and T_2. Directions in such treks are left unspecified. In (b), the existence of a trek T_{Q2} linking X_2 and Q through P will compel the directions depicted as a consequence of the given tetrad and correlation constraints (the dotted path represents any possible continuation of T_{Q2} that does not coincide with T). The configuration in (c) cannot happen if P is a choke point entailing all three tetrads among marginally dependent nodes $\{X_1, X_2, X_3, X_4\}$. The configuration in (d) cannot happen if P is a choke point for $\{X_1, X_3\} \times \{X_2, X_4\}$, since there is a trek $X_1 - P - X_2$ such that P is not on the $\{X_1, X_3\}$ side of it, and another trek $X_2 - S - P - X_3$ such that P is not on the $\{X_2, X_4\}$ side of it.

the previous case by symmetry). Let S be the source of a trek into P and X_2, which should exist since X_2 is not an ancestor of P. Then there is a trek of source S connecting X_3 and X_2 such that P is not on the $\{X_2, X_4\}$ side of it as shown in Figure 14(d). Therefore P cannot be a choke point for $\{X_1, X_3\} \times \{X_2, X_4\}$. Contradiction. □

Lemma 12 *Let $G(\mathbf{O})$ be a linear latent variable model. If for some set $\mathbf{O}' = \{X_1, X_2, X_3, X_4\} \subseteq \mathbf{O}$, $\sigma_{X_1X_2}\sigma_{X_3X_4} = \sigma_{X_1X_3}\sigma_{X_2X_4} = \sigma_{X_1X_4}\sigma_{X_2X_3}$ and for all triplets $\{A, B, C\}$, $\{A, B\} \subset \mathbf{O}', C \in \mathbf{O}$, we have $\rho_{AB.C} \neq 0$ and $\rho_{AB} \neq 0$, then no element $A \in \mathbf{O}'$ is a descendant of an element of $\mathbf{O}'\backslash\{A\}$ in G.*

Proof: Without loss of generality, assume for the sake of contradiction that X_1 is an ancestor of X_2. From the given tetrad and correlation constraints and Lemma 21, there is a node P that lies on every trek between X_1 and X_2 and d-separates these two nodes. Since P lies

on the directed path from X_1 to X_2, P is a descendant of X_1, and therefore an observed node. However, this implies $\rho_{X_1X_2.P} = 0$, contrary to our hypothesis. □

Lemma 9 *Let $G(\mathbf{O})$ be a linear latent variable model. Assume $\mathbf{O'} = \{X_1, X_2, X_3, Y_1, Y_2, Y_3\} \subseteq \mathbf{O}$. If constraints $\{\tau_{X_1Y_1X_2X_3}, \tau_{X_1Y_1X_3X_2}, \tau_{Y_1X_1Y_2Y_3}, \tau_{Y_1X_1Y_3Y_2}, \neg\tau_{X_1X_2Y_2Y_1}\}$ all hold, and that for all triplets $\{A, B, C\}, \{A, B\} \subset \mathbf{O'}$, $C \in \mathbf{O}$, we have $\rho_{AB} \neq 0, \rho_{AB.C} \neq 0$, then X_1 and Y_1 do not have a common parent in G.*

Proof: We will prove this result by contradiction. Suppose that X_1 and Y_1 have a common parent L in G. Suppose L is not a choke point for $\{X_1, X_2\} \times \{Y_1, X_3\}$ corresponding to one of the tetrad constraints given by hypothesis. Because of the trek $X_1 \leftarrow L \rightarrow Y_1$, then either X_1 or Y_1 is a choke point. Without loss of generality, assume X_1 is a choke point in this case. By Lemma 12 and the given constraints, X_1 cannot be an ancestor of either X_2 or X_3, and by Lemma 21 it is also the choke point for $\{X_1, Y_1\} \times \{X_2, X_3\}$. That means that all treks connecting X_1 and X_2, and X_1 and X_3 should be into X_1. Since there are no treks between X_2 and X_3 that do not include X_1, and all paths between X_2 and X_3 that include X_1 collide at X_1, that implies $\rho_{X_2X_3} = 0$, contrary to our hypothesis. By symmetry, Y_1 cannot be a choke point. Therefore, L is a choke point for $\{X_1, Y_1\} \times \{X_2, X_3\}$ and by Lemma 21, it also lies on every trek for any pair in $\mathbf{S_1} = \{X_1, X_2, X_3, Y_1\}$.

Analogously, L is on every trek connecting any pair from the set $\mathbf{S_2} = \{X_1, Y_1, Y_2, Y_3\}$. It follows that L is on every trek connecting any pair from the set $\mathbf{S_3} = \{X_1, X_2, Y_1, Y_2\}$, and it is on the $\{X_1, Y_1\}$ side of $\{X_1, Y_1\} \times \{X_2, Y_2\}$, i.e., L is a choke point that implies $\tau_{X_1X_2Y_2Y_1}$. Contradiction. □

Remember that predicate $F_1(X, Y, G)$ is true if and only if there exist two nodes W and Z in G such that τ_{WXYZ} and τ_{WXZY} are both entailed, all nodes in $\{W, X, Y, Z\}$ are correlated, and there is no observed C in G such that $\rho_{AB.C} = 0$ for $\{A, B\} \subset \{W, X, Y, Z\}$.

Lemma 10 *Let $G(\mathbf{O})$ be a linear latent variable model. Assume $\mathbf{O'} = \{X_1, X_2, X_3, Y_1, Y_2, Y_3\} \subseteq \mathbf{O}$, such that $F_1(X_1, X_2, G)$ and $F_1(Y_1, Y_2, G)$ hold, Y_1 is not an ancestor of Y_3 and X_1 is not an ancestor of X_3. If constraints $\{\tau_{X_1Y_1Y_2X_2}, \tau_{X_2Y_1Y_3Y_2}, \tau_{X_1X_2Y_2X_3}, \neg\tau_{X_1X_2Y_2Y_1}\}$ all hold, and that for all triplets $\{A, B, C\}, \{A, B\} \subset \mathbf{O'}, C \in \mathbf{O}$, we have $\rho_{AB} \neq 0, \rho_{AB.C} \neq 0$, then X_1 and Y_1 do not have a common parent in G.*

Proof: We will prove this result by contradiction. Assume X_1 and Y_1 have a common parent L. Because of the tetrad constraints given by hypothesis and the existence of the trek $X_1 \leftarrow L \rightarrow Y_1$, one node in $\{X_1, L, Y_1\}$ should be a choke point for the pair $\{X_1, X_2\} \times \{Y_1, Y_2\}$. We will first show that L has to be such a choke point, and therefore lies on every trek connecting X_1 and Y_2, as well as X_2 and Y_1. We then show that L lies on every trek connecting Y_1 and Y_2, as well as X_1 and X_2. Finally, we show that L is a choke point for $\{X_1, Y_1\} \times \{X_2, Y_2\}$, contrary to our hypothesis.

Step 1: If there is a common parent L to X_1 and Y_1, then L is a $\{X_1, X_2\} \times \{Y_1, Y_2\}$ choke point. For the sake of contradiction, assume X_1 is a choke point in this case. By Lemma 12 and assumption $F_1(X_1, X_2, G)$, we have that X_1 is not an ancestor of X_2, and therefore

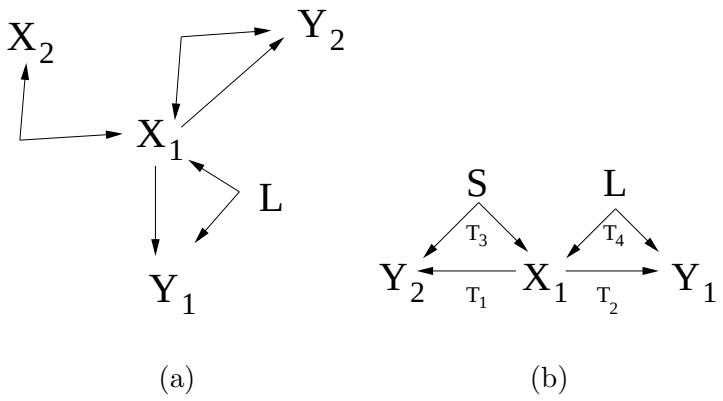


Figure 15: Figure (a) illustrates necessary treks among elements of $\{X_1, X_2, Y_1, Y_2, L\}$ according to the assumptions of Lemma 10 if we further assume that X_1 is a choke point for pairs $\{X_1, X_2\} \times \{Y_1, Y_2\}$ (other treks might exist). Figure (b) rearranges (a) by emphasizing that Y_1 and Y_2 cannot be d-separated by a single node.

all treks connecting X_1 and X_2 should be into X_1. Since $\rho_{X_2Y_2} \neq 0$ by assumption and X_1 is on all treks connecting X_2 and Y_2, there must be a directed path out of X_1 and into Y_2. Since $\rho_{X_2Y_2.X_1} \neq 0$ by assumption and X_1 is on all treks connecting X_2 and Y_2, there must be a trek into X_1 and Y_2. Because $\rho_{X_2Y_1} \neq 0$, there must be a trek out of X_1 and into Y_1. Figure 15(a) illustrates the configuration.

Since $F_1(Y_1, Y_2, G)$ is true, by Lemma 21 there must be a node d-separating Y_1 and Y_2 (neither Y_1 nor Y_2 can be the choke point in $F_1(Y_1, Y_2, G)$ because this choke point has to be latent, according to the partial correlation conditions of F_1). However, by Figure 15(b), treks $T_2 - T_3$ and $T_1 - T_4$ cannot both be blocked by a single node. Contradiction. Therefore X_1 cannot be a choke point for $\{X_1, X_2\} \times \{Y_1, Y_2\}$ and, by symmetry, neither can Y_1.

Step 2: L is on every trek connecting Y_1 and Y_2 and on every trek connecting X_1 and X_2. Let L be the choke point for pairs $\{X_1, X_2\} \times \{Y_1, Y_2\}$. As a consequence, all treks between Y_2 and X_1 go through L. All treks between X_2 and Y_1 go through L. All treks between X_2 and Y_2 go through L. Such treks exist, since no respective correlation vanishes.

Consider the given hypothesis $\sigma_{X_2Y_1}\sigma_{Y_2Y_3} = \sigma_{X_2Y_3}\sigma_{Y_2Y_1}$, corresponding to a choke point $\{X_2, Y_2\} \times \{Y_1, Y_3\}$. From the previous paragraph, we know there is a trek linking Y_2 and L. L is a parent of Y_1 by construction. That means Y_2 and Y_1 are connected by a trek through L.

We will show by contradiction that L is on every trek connecting Y_1 and Y_2. Assume there is a trek T_Y connecting Y_2 and Y_1 that does not contain L. Let P be the first point of intersection of T_Y and a trek T_X connecting X_2 to Y_1, starting from X_2. If T_Y exists, such point should exist, since T_Y should contain a choke point $\{X_2, Y_2\} \times \{Y_1, Y_3\}$, and all treks connecting X_2 and Y_1 (including T_X) contain the same choke point.

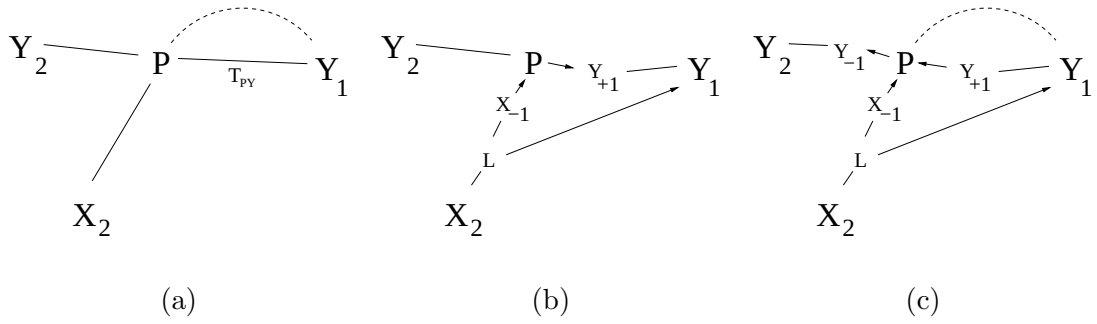


Figure 16: In (a), a depiction of T_Y and T_X, where edges represent treks (T_X can be seen more generally as the combination of the solid edge between X_2 and P concatenated with a dashed edge between P and Y_1 representing the possibility that T_Y and T_X might intersect multiple times in T_{PY}, but in principle do not need to coincide in T_{PY} if P is not a choke point.) In (b), a possible configurations of edges $< X_{-1}, P >$ and $< P, Y_{+1} >$ that do not collide in P, and P is a choke point (and $Y_{+1} \neq Y$). In (c), the edge $< Y_{-1}, P >$ is compelled to be directed away from P because of the collider with the other two neighbors of P.

Let T_{PY} be the subtrek of T_Y starting on P and ending one node before Y_1. Any choke point $\{X_2, Y_2\} \times \{Y_1, Y_3\}$ should lie on T_{PY} (Figure 16(a)). (Y_1 cannot be such a choke point, since all treks connecting Y_1 and Y_2 are into Y_1, and by hypothesis all treks connecting Y_1 and Y_3 are into Y_1. Since all treks connecting Y_2 and Y_3 would need to go through Y_1 by definition, then there would be no such trek, implying $\rho_{Y_2 Y_3} = 0$, contrary to our hypothesis.)

Assume first that $X_2 \neq P$ and $Y_2 \neq P$. Let X_{-1} be the node before P in T_X starting from X_2. Let Y_{-1} be the node before P in T_Y starting from Y_2. Let Y_{+1} be the node after P in T_Y starting from Y_2 (notice that it is possible that $Y_{+1} = Y_1$). If X_{-1} and Y_{+1} do not collide on P (i.e., there is no structure $X_{-1} \rightarrow P \leftarrow Y_{+1}$), then there will be a trek connecting X_2 to Y_1 through T_{PY} after P. Since L is not in T_{PY}, L should be before P in T_X. But then there will be a trek connecting X_2 and Y_1 that does not intersect T_{PY}, which is a contradiction (Figure 16(b)). If the collider does exist, we have the edge $P \leftarrow Y_{+1}$. Since no collider $Y_{-1} \rightarrow P \leftarrow Y_{+1}$ can exist because T_Y is a trek, the edge between Y_{-1} and P is out of P. But that forms a trek connecting X_2 and Y_2 (Figure 16(c)), and since L is in every trek between X_2 and Y_2 and T_Y does not contain L, then T_X should contain L before P, which again creates a trek between X_2 and Y_1 that does not intersect T_{PY}.

If $X_2 = P$, then T_{PY} has to contain L, because every trek between X_2 and Y_1 contains L. Therefore, $X_2 \neq P$. If $Y_2 = P$, then because every trek between X_2 and Y_2 should contain L, we again have that L lies in T_X before P, which creates a trek between X_2 and Y_1 that does not intersect T_{PY}. Therefore, we showed by contradiction that L lies on every trek between Y_2 and Y_1.

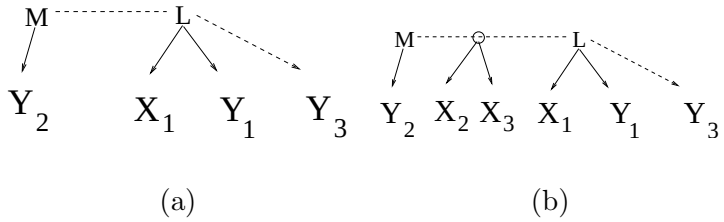


Figure 17: In (a), Y_2 and X_1 cannot share a parent, and because of the given tetrad constraints, L should d-separate M and Y_3. Y_3 is not a child of L either, but there will be a trek linking L and Y_3. In (b), an (invalid) configuration for X_2 and X_3, where they share an ancestor between M and L.

Consider now the given hypothesis $\sigma_{X_1X_2}\sigma_{X_3Y_2} = \sigma_{X_1Y_2}\sigma_{X_3X_2}$, corresponding to a choke point $\{X_2, Y_2\} \times \{X_1, X_3\}$. By symmetry with the previous case, all treks between X_1 and X_2 go through L.

Step 3: If L exists, so does a choke point $\{X_1, Y_1\} \times \{X_2, Y_2\}$. By the previous steps, L intermediates all treks between elements of the pair $\{X_1, Y_1\} \times \{X_2, Y_2\}$. Because L is a common parent of $\{X_1, Y_1\}$, it lies on the $\{X_1, Y_1\}$ side of every trek connecting pairs of elements in $\{X_1, Y_1\} \times \{X_2, Y_2\}$. L is a choke point for this pair. This implies $\tau_{X_1X_2Y_2Y_1}$. Contradiction. □

Lemma 11 *Let $G(\mathbf{O})$ be a linear latent variable model. Let $\mathbf{O}' = \{X_1, X_2, X_3, Y_1, Y_2, Y_3\} \subseteq \mathbf{O}$. If constraints $\{\tau_{X_1Y_1Y_2Y_3}, \tau_{X_1Y_1Y_3Y_2}, \tau_{X_1Y_2X_2X_3}, \tau_{X_1Y_2X_3X_2}, \tau_{X_1Y_3X_2X_3}, \tau_{X_1Y_3X_3X_2}, \neg\tau_{X_1X_2Y_2Y_3}\}$ all hold, and that for all triplets $\{A, B, C\}, \{A, B\} \subset \mathbf{O}', C \in \mathbf{O}$, we have $\rho_{AB} \neq 0, \rho_{AB.C} \neq 0$, then X_1 and Y_1 do not have a common parent in G.*

Proof: We will prove this result by contradiction. Suppose X_1 and Y_1 have a common parent L in G. Since all three tetrads hold in the covariance matrix of $\{X_1, Y_1, Y_2, Y_3\}$, by Lemma 21 the choke point that entails these constraints d-separates the elements of $\{X_1, Y_1, Y_2, Y_3\}$. The choke point should be in the trek $X_1 \leftarrow L \rightarrow Y_1$, and since it cannot be an observed node because by hypothesis no d-separation conditioned on a single node holds among elements of $\{X_1, Y_1, Y_2, Y_3\}$, L has to be a latent choke point for all pairs of pairs in $\{X_1, Y_1, Y_2, Y_3\}$.

It is also given that $\{\tau_{X_1Y_2X_2X_3}, \tau_{X_1Y_2X_3X_2}, \tau_{X_1Y_1Y_2Y_3}, \tau_{X_1Y_1Y_3Y_2}\}$ holds. Since it is the case that $\neg\tau_{X_1X_2Y_2Y_3}$, by Lemma 9 X_1 and Y_2 cannot share a parent. Let T_{ML} be a trek connecting some parent M of Y_2 and L. Such a trek exists because $\rho_{X_1Y_2} \neq 0$.

We will show by contradiction that there is no node in $T_{ML} \backslash L$ that is connected to Y_3 by a trek that does not go through L. Suppose there is such a node, and call it V. If the trek connecting V and Y_3 is into V, and since V is not a collider in T_{ML}, then V is either an ancestor of M or an ancestor of L. If V is an ancestor of M, then there will be a trek connecting Y_2 and Y_3 that is not through L, which is a contradiction. If V is an

ancestor of L but not M, then both Y_2 and Y_3 are d-connected to a node V is a collider at the intersection of such d-connecting treks. However, V is an ancestor of L, which means L cannot d-separate Y_2 and Y_3, a contradiction. Finally, if the trek connecting V and Y_3 is out of V, then Y_2 and Y_3 will be connected by a trek that does not include L, which again is not allowed. We therefore showed there is no node with the properties of V. This configuration is illustrated by Figure 17(a).

Since all three tetrads hold among elements of $\{X_1, X_2, X_3, Y_2\}$, then by Lemma 21, there is a single choke point P that entails such tetrads and d-separates elements of this set. Since T_{ML} is a trek connecting Y_2 to X_1 through L, then there are three possible locations for P in G:

Case 1: $P = M$. We have all treks between X_3 and X_2 go through M but not through L, and some trek from X_1 to Y_3 goes through L but not through M. No choke point can exist for pairs $\{X_1, X_3\} \times \{X_2, Y_3\}$, which by the Tetrad Representation Theorem means that the tetrad $\sigma_{X_1Y_3}\sigma_{X_2X_3} = \sigma_{X_1X_2}\sigma_{Y_3X_3}$ cannot hold, contrary to our hypothesis.

Case 2: P lies between M and L in T_{ML}. This configuration is illustrated by Figure 17(b). As before, no choke point exists for pairs $\{X_1, X_3\} \times \{X_2, Y_3\}$, contrary to our hypothesis.

Case 3: $P = L$. Because all three tetrads hold in $\{X_1, X_2, X_3, Y_3\}$ and L d-separates all pairs in $\{X_1, X_2, X_3\}$, one can verify that L d-separates all pairs in $\{X_1, X_2, X_3, Y_3\}$. This will imply a $\{X_1, Y_3\} \times \{X_2, Y_2\}$ choke point, contrary to our hypothesis. □

Theorem 13 *The output of* FindPattern *is a measurement pattern with respect to the tetrad and vanishing partial correlation constraints of* Σ

Proof: Two nodes will not share a common latent parent in a measurement pattern if and only if they are not linked by an edge in graph C constructed by algorithm FindPattern and that happens if and only if some partial correlation vanishes or if any of rules CS1, CS2 or CS3 applies. But then by Lemmas 9, 10, 11 and the equivalence of vanishing partial correlations and conditional independence in linearly faithful distributions (Spirtes et al., 2000) the claim is proved. The claim about undirected edges follows from Lemma 12. □

Theorem 14 *Given a covariance matrix* Σ *assumed to be generated from a linear latent variable model* $G(\mathbf{O})$ *with latent variables* $\mathbf{L}$*, let* G_{out} *be the output of* BuildPureClusters(Σ) *with observed variables* $\mathbf{O_{out}} \subseteq \mathbf{O}$ *and latent variables* $\mathbf{L_{out}}$*. Then* G_{out} *is a measurement pattern, and there is an injective mapping* $M : \mathbf{L_{out}} \rightarrow \mathbf{L}$ *with the following properties:*

1. *Let* $L_{out} \in \mathbf{L_{out}}$*. Let* $\mathbf{X}$ *be the children of* L_{out} *in* G_{out}*. Then* $M(L_{out})$ *d-separates any element* $X \in \mathbf{X}$ *from* $\mathbf{O_{out}} \backslash X$ *in* G*;*

2. $M(L_{out})$ *d-separates* X *from every latent in* G *for which* $M^{-1}(.)$ *exists;*

3. *Let* $\mathbf{O'} \subseteq \mathbf{O_{out}}$ *be such that each pair in* $\mathbf{O'}$ *is correlated. At most one element in* $\mathbf{O'}$ *with latent parent* L_{out} *in* G_{out} *is not a descendant of* $M(L_{out})$ *in* G*, or has a hidden common cause with it;*

Proof: We will start by showing that for each cluster Cl_i in G_{out}, there exists an unique latent L_i in G that d-separates all elements of Cl_i. This shows the existance of an unique function from latents in G_{out} to latents in G. We then proceed to prove the three claims given in the theorem, and finish by proving that the given function is injective.

Let Cl_i be a cluster in a non-empty G_{out}. Cl_i has three elements X, Y and Z, and there is at least some W in G_{out} such that all three tetrad constraints hold in the covariance matrix of $\{W, X, Y, Z\}$, where no pair of elements in $\{X, Y, Z\}$ is marginally d-separated or d-separated by an observable variable. By Lemma 21, it follows that there is an unique latent L_i d-separating X, Y and Z. If Cl_i has more than three elements, it follows that since no node other than L_i can d-separate all three elements in $\{X, Y, Z\}$, and any choke point for $\{W', X, Y, Z\}$, $W' \in Cl_i$, will d-separate all elements in $\{W', X, Y, Z\}$, then there is an unique latent L_i d-separating all elements in C_i. An analogous argument concerns the d-separation of any element of Cl_i and observed nodes in other clusters.

Now we will show that each L_i d-separates each X in Cl_i from all other mapped latents. As a byproduct, we will also show the validity of the third claim of the theorem. Consider $\{Y, Z\}$, two other elements of Cl_i besides X, and $\{A, B, C\}$, three elements of Cl_j. Since L_i and L_j each d-separate all pairs in $\{X, Y\} \times \{A, B\}$, and no pair in $\{X, Y\} \times \{A, B\}$ has both of its elements connected to L_i (L_j) through a trek that is into L_i (L_j) (since L_i, or L_j, d-separates then), then both L_i and L_j are choke points for $\{X, Y\} \times \{A, B\}$. According to Lemma 2.5 given by Shafer et al. (1993), any trek connecting an element from $\{X, Y\}$ to an element in $\{A, B\}$ passes through both choke points in the same order. Without loss of generality, assume the order is first L_i, then L_j.

If there is no trek connecting X to L_i that is into L_i, then L_i d-separates X and L_j. The same holds for L_j and A with respect to L_i. If there is a trek T connecting X and L_i that is into L_i, and since all three tetrad constraints hold in the covariance matrix of $\{X, Y, Z, A\}$ by construction, then there is no trek connecting A and L_i that is into L_i (Lemma 21). Since there are treks connecting L_i and L_j, they should be all out of L_i and into L_j. This means that L_i d-separates X and L_j. But this also creates a trek connecting X and L_j that is into L_j. Since all three tetrad constraints hold in the covariance matrix of $\{X, A, B, C\}$ by construction, then there is no trek connecting A and L_j that is into L_j (by the d-separation implied by Lemma 21). This means that L_j d-separates A from L_i. This also means that the existance of such a trek T out of X and into L_i forbids the existance of any trek connecting a variable correlated to X that is into L_i (since all treks connecting L_i and some L_j are out of L_i), which proves the third claim of the theorem.

We will conclude by showing that given two clusters Cl_i and Cl_j with respective latents L_i and L_j, where each cluster is of size at least three, if they are not merged, then $L_i \neq L_j$. That is, the mapping from latents in G_{out} to latents in G, as defined at the beginning of the proof, is injective.

Assume $L_i = L_j$. We will show that these clusters will be merged by the algorithm, proving the counterpositive argument. Let X and Y be elements of Cl_i and W, Z elements of Cl_j. It immediately follows that L_i is a choke point for all pairs in $\{W, X, Y, Z\}$, since L_i d-separates any pair of elements of $\{W, X, Y, Z\}$, which means all three tetrads will hold in the covariance matrix of any subset of size four from $Cl_i \cup Cl_j$. These two clusters will then be merged by BuildPureClusters. □

Theorem 15 *Given a covariance matrix Σ assumed to be generated from a linear latent variable model $G(\mathbf{O})$ with latent variables $\mathbf{L}$, let G_{out} be the output of* BuildPureClusters(Σ) *with observed variables $\mathbf{O_{out}} \subseteq \mathbf{O}$ and latent variables $\mathbf{L_{out}}$. Let $M(\mathbf{L_{out}}) \subseteq \mathbf{L}$ be the set of latents in G obtained by the mapping function $M()$. Let $\Sigma_{\mathbf{O_{out}}}$ be the population covariance matrix of $\mathbf{O_{out}}$, i.e., the corresponding marginal of Σ. Let the DAG G^{aug}_{out} be G_{out} augmented by connecting the elements of $\mathbf{L_{out}}$ such that the structural model of G^{aug}_{out} is an I-map of the distribution of $M(\mathbf{L_{out}})$. Then there exists a linear latent variable model using G^{aug}_{out} as the graphical structure such that the implied covariance matrix of $\mathbf{O_{out}}$ equals $\Sigma_{\mathbf{O_{out}}}$.*

Proof: If a linear model is an I-map DAG of the true distribution of its variables, then there is a well-known natural instantiation of the parameters of this model that will represent the true covariance matrix (Spirtes et al., 2000). We will assume such parametrization for the structural model, and denote as $\Sigma_L(\Theta)$ the parameterized latent covariance matrix. Instead of showing that G^{aug}_{out} is an I-map of the respective set of latents and observed variables and using the same argument, we will show a valid instantion of its parameters directly.

Assume without loss of generality that all variables have zero mean. To each observed node X with latent ancestor L_X in G such that $M^{-1}(L_X)$ is a parent of X in G_{out}, the linear model representation is:

$$X = \lambda_X L_X + \epsilon_X$$

For this equation, we have two associated parameters, λ_X and $\sigma^2_{\epsilon_X}$, where $\sigma^2_{\epsilon_X}$ is the variance of ϵ_X. We instantiate them by the linear regression values, i.e., $\lambda_X = \sigma_{XL_X}/\sigma^2_{L_X}$, and $\sigma^2_{\epsilon_X}$ is the respective residual variance. The set $\{\lambda_X\} \cup \{\sigma^2_{\epsilon_X}\}$ of all λ_X and $\sigma^2_{\epsilon_X}$, along with the parameters used in $\Sigma_L(\Theta)$, is our full set of parameters Θ.

Our definition of linear latent variable model requires $\sigma_{\epsilon_X \epsilon_Y} = 0$, $\sigma_{\epsilon_X L_X} = 0$ and $\sigma_{\epsilon_X L_Y} = 0$, for all $X \neq Y$. This corresponds to a covariance matrix $\Sigma(\Theta)$ of the observed variables with entries defined as:

$$E[X^2](\Theta) = \sigma^2_X(\Theta) = \lambda^2_X \sigma^2_{L_X} + \sigma^2_{\epsilon_X}$$

$$E[XY](\Theta) = \sigma_{XY}(\Theta) = \lambda_X \lambda_T \sigma_{L_X L_Y}$$

To prove the theorem, we have to show that $\Sigma_{\mathbf{O_{out}}} = \Sigma(\Theta)$ by showing that correlations between different residuals, and residuals and latent variables, are actually zero.

The relation $\sigma_{\epsilon_X L_X} = 0$ follows directly from the fact that λ_X is defined by the regression coefficient of X on L_X. Notice that if X and L_X do not have a common ancestor, λ_X is the direct effect of L_X in X with respect to G_{out}. As we know, by Theorem 14, at most one variable in any set of correlated variables will not fulfill this condition.

We have to show also that $\sigma_{XY} = \sigma_{XY}(\Theta)$ for any pair X, Y in G_{out}. Residuals ϵ_X and ϵ_Y are uncorrelated due to the fact that X and Y are independent given their latent ancestors in G_{out}, and therefore $\sigma_{\epsilon_X \epsilon_Y} = 0$. To verify that $\sigma_{\epsilon_X L_Y} = 0$ is less straightforward, but one can appeal to the graphical formulation of the problem. In a linear model, the residual ϵ_X is a function only of the variables that are not independent of X given L_X. None of this variables can be nodes in G_{out}, since L_X d-separates X from all such variables. Therefore, given L_X none of the variables that define ϵ_X can be dependent on L_Y, implying $\sigma_{\epsilon_X L_Y} = 0$.

□

Theorem 16 *Problem $\mathcal{MP}^3$ is NP-complete.*

Proof: Direct reduction from the 3-SAT problem: let S be a 3-CNF formula from which we want to decide if there is an assignment for its variables that makes the expression true. Define G as a latent variable graph with a latent node L_i for each clause C_i in M, with an arbitrary fully connected structural model. For each latent in G, add five pure children. Choose three arbitrary children of each latent L_i, naming them $\{C_i^1, C_i^2, C_i^3\}$. Add a bi-directed edge $C_i^p \leftrightarrow C_j^q$ for each pair $C_i^p, C_j^q, i \neq j$, if and only that they represent literals over the same variable but of opposite values. As in the maximum clique problem, one can verify that there is a pure submodel of G with at least three indicators per latent if and only if S is satisfiable. □

The next corollay suggests that even an invalid measurement pattern could be used in BuildPureClusters instead of the output of FindPattern. However, an arbitrary (invalid) measurement pattern is unlikely to be informative at all after being purified. In constrast, FindPattern can be highly informative.

Corollary 17 *The output of* BuildPureClusters *retains its guarantees even when rules CS1, CS2 and CS3 are applied an arbitrary number of times in* FindPattern *for any arbitrary subset of nodes and an arbitrary number of maximal cliques is found.*

Proof: Independently of the choice made on Step 2 of BuildPureClusters and which nodes are not separated into different cliques in FindPattern, the exhaustive verification of tetrad constraints by BuildPureClusters provides all the necessary conditions for the proof of Theorem 14. □

Corollary 19 *Given a covariance matrix Σ assumed to be generated from a linear latent variable model G, and G_{out} the output of* BuildPureClusters *given Σ, the output of* PC-MIMBuild *or* FCI-MIMBuild *given (Σ, G_{out}) returns the correct Markov equivalence class of the latents in G corresponding to latents in G_{out} according to the mapping implicit in* BuildPureClusters

Proof: By Theorem 14, each observed variable is d-separated from all other variables in G_{out} given its latent parent. By Theorem 15, one can parameterize G_{out} as a linear model such that the observed covariance matrix as a function of the parameterized G_{out} equals its corresponding marginal of Σ. By Theorem 18, the rank test using the measurement model of G_{out} is therefore a consistent independence test of latent variables. The rest follows immediately from the consistency property of PC and FCI given a valid oracle for conditional independencies. □

Appendix C. Implementation

Statistical tests for tetrad constraints are described by Spirtes et al. (2000). Although it is known that in practice constraint-based approaches for learning graphical model structure are outperformed on accuracy by score-based algorithms such as GES (Chickering, 2002), we favor a constraint-based approach due mostly to computational efficiency. Moreover, a smart implementation of can avoid many statistical shortcomings.

C.1 Robust purification

We do avoid a constraint-satisfaction approach for purification. At least for a fixed p-value and using false discovery rates to control for multiplicity of tests, purification by testing tetrad constraints often throws away many more nodes than necessary when the number of variables is relative small, and does not eliminate many impurities when the number of variables is too large. We suggest a robust purification approach as follows.

Suppose we are given a clustering of variables (not necessarily disjoint clusters) and a undirect graph indicating which variables might be ancestors of each other, analogous to the undirect edges generated in FindPattern. We purify this clustering not by testing multiple tetrad constraints, but through a greedy search that eliminates nodes from a linear measurement model that entails tetrad constraints. This is iterated till the current model fits the data according to a chi-square test of significance (Bollen, 1989) and a given acceptance level. Details are given in Table 7.

This implementation is used as a subroutine for a more robust implementation of BuildPureClusters described in the next section. However, it can be considerably slow. An alternative is using the approximation derived by Kano and Harada (2000) to rapidly calculate the fitness of a factor analysis model when a variable is removed. Another alternative is a greedy search over the initial measurement model, freeing correlations of pairs of measured variables. Once we found which variables are directly connected, we eliminate some of them till no pair is impure. In our experiments with synthetic data, it did not work as well as the iterative removal of variables described in Table 7. However, we do apply this variation in the last experiment described in Section 6, because it is computationally cheaper. If the model search in RobustPurify does not fit the data after we eliminate too many variables (i.e., when we cannot statistically test the model) we just return an empty model.

C.2 Finding a robust initial clustering

The main problem of applying FindPattern directly by using statistical tests of tetrad constraints is the number of false positives: accepting a rule (CS1, CS2, or CS3) as true when it does not hold in the population. One can see that might happen relatively often when there are large groups of observed variables that are pure indicators of some latent: for instance, assume there is a latent L_0 with 10 pure indicators. Consider applying CS1 to a group of six pure indicators of L_0. The first two constraints of CS1 hold in the population, and so assume they are correctly identified by the statistical test. The last constraint, $\sigma_{X_1X_2}\sigma_{Y_1Y_2} \neq \sigma_{X_1Y_2}\sigma_{X_2Y_1}$, should not hold in the population, but will not be rejected by

the test with some probability. Since there are $10!/(6!4!) = 210$ ways of CS1 being wrongly applied due to a statistical mistake, we *will* get many false positives in all certainty.

We can highly minimize this problem by separating *groups* of variables instead of pairs. Consider the test DisjointGroup$(X_i, X_j, X_k, Y_a, Y_b, Y_c; \Sigma)$:

- DisjointGroup$(X_i, X_j, X_k, Y_a, Y_b, Y_c; \Sigma) = true$ if and only if CS1 returns true for all sets $\{X_1, X_2, X_3, Y_1, Y_2, Y_3\}$, where $\{X_1, X_2, X_3\}$ is a permutation of $\{X_i, X_j, X_k\}$ and $\{Y_1, Y_2, Y_3\}$ is a permutation of $\{Y_a, Y_b, Y_c\}$. Also, we test an extra redundant constraint: for every pair $\{X_1, X_2\} \subset \{X_i, X_j, X_k\}$ and every pair $\{Y_1, Y_2\} \subset \{Y_a, Y_b, Y_c\}$ we also require that $\sigma_{X_1Y_1}\sigma_{X_2Y_2} = \sigma_{X_1Y_2}\sigma_{X_2Y_1}$.

Notice it is much harder to obtain a false positive with DisjointGroup than, say, with CS1 applied to a single pair. This test can be implemented in steps: for instance, if for no four foursome including X_i and Y_a we have that all tetrad constraints hold, then we do not consider X_i and Y_a in DisjoingGroup.

Based on DisjointGroup, we propose here a modification to increase the robustness of BuildPureClusters, the RobustBuildPureClusters algorithm, as given in Table 8. It starts with a first step called FindInitialSelection (Table 9). The goal of FindInitialSelection is to find a pure model using only DisjointGroup instead of CS1, CS2 or CS3. This pure model is then used as an starting point for learning a more complete model in the remaining stages of RobustBuildPureClusters.

In FindInitialSelection, if a pair $\{X, Y\}$ cannot be separated into different clusters, but also does not participate in any successful application of DisjointGroup, then this pair will be connected by a GRAY or YELLOW edge: this indicates that these two nodes cannot be in a pure submodel with three indicators per latent. Otherwise, these nodes are "compatible", meaning that they *might* be in such a pure model. This is indicated by a BLUE edge.

In FindInitialSelection we then find cliques of compatible nodes (Step 8)[6]. Each clique is a candidate for a one-factor model (a latent model with one latent only). We purify every clique found to create pure one-factor models (Step 9). This avoids using clusters that are large not because they are all unique children of the same latent, but because there was no way of separating its elements. This adds considerably more computational cost to the whole procedure.

After we find pure one-factor models M_i, we search for a combination of compatible groups. Step 10 first indicates which pairs of one-factor models cannot be part of a pure model with three indicators each: if M_i and M_j are not pairwise a two-factor model with three pure indicators (as tested by DisjointGroup), they cannot be both part of a valid solution.

ChooseClusteringClique is a heuristic designed to find a large set of one-factor models (nodes of H) that can be grouped into a pure model with three indicators per latent (we need a heuristic since finding a maximum clique in H is NP-hard). First, we define the *size* of a clustering $H_{candidate}$ (a set of nodes from H) as the number of variables that remain according to the following elimination criteria: 1. eliminate all variables that appear

6. Any algorithm can be used to find maximal cliques. Notice that, by the anytime properties of our approach, one does not need to find all maximal cliques

Algorithm RobustPurify
Inputs: *Clusters*, a set of subsets of some set **O**;
C, an undirect graph over **O**;
Σ, a sample covariance matrix of **O**.

1. Remove all nodes that have appear in more than one set in *Clusters*.
2. For all pairs of nodes that belong to two different sets in *Clusters* and are adjacent in C, remove the one from the largest cluster or the one from the smallest cluster if this has less than three elements.
3. Let G be a graph. For each set $S \in Clusters$, add all nodes in S to G and a new latent as the only common parent of all nodes in S. Create an arbitrary full DAG among latents.
4. For each variable V in G, fit a graph $G'(V)$ obtained from G by removing V. Update G by choosing the graph $G'(V)$ with the smallest chi-square score. If some latent ends up with less than two children, remove it. Iterate till a significance level is achieved.
5. Do mergings if that increases the fitness. Iterate 4 and 5 till no improvement can be done.
6. Eliminate all clusters with less than three variables and return G.

Table 7: A score-based purification.

Algorithm RobustBuildPureClusters
Input: Σ, a sample covariance matrix of a set of variables **O**

1. $(Selection, C, C_0) \leftarrow$FindInitialSelection($\Sigma$).
2. For every pair of nonadjacent nodes $\{N_1, N_2\}$ in C where at least one of them is not in *Selection* and an edge $N_1 - N_2$ exists in C_0, add a RED edge $N_1 - N_2$ to C.
3. For every pair of nodes linked by a RED edge in C, apply successively rules CS1, CS2 and CS3. Remove an edge between every pair corresponding to a rule that applies.
4. Let H be a complete graph where each node corresponds to a maximal clique in C.
5. $FinalClustering \leftarrow$ ChooseClusteringClique(H).
6. Return RobustPurify($FinalClustering, C, \Sigma$).

Table 8: A modified BuildPureClusters algorithm.

in more than one one-factor model inside $H_{candidate}$; 2. for each pair of variables $\{X_1, X_2\}$ such that X_1 and X_2 belong to different one-factor models in $H_{candidate}$, if there is an edge $X_1 - X_2$ in C, then we remove one element $\{X_1, X_2\}$ from $H_{candidate}$ (i.e., guarantee that no pair of variables from different clusters which were not shown to have any common latent parent will exist in $H_{candidate}$). We eliminate the one that belongs to the largest cluster, unless the smallest cluster has less than three elements to avoid extra fragmentation; 3. eliminate clusters that have less than three variables.

The heuristic motivation is that we expected that a model with a large size will have a large number of variables after purification. Our suggested heuristic to be implemented as ChooseClusteringClique is trying to find a good model using a very simple hill-climbing algorithm that starts from an arbitrary node in H and add new clusters to the current candidate according to the one that will increase its size mostly while still forming a maximal clique in H. We stop when we cannot increase the size of the candidate. This is calculated using each node in H as a starting point, and the largest candidate is returned by ChooseClusteringClique.

C.3 Clustering refinement

The next steps in RobustBuildPureClusters are basically the FindPattern algorithm of Table 1 with a final purification. The main difference is that we do not check anymore if pairs of nodes in the initial clustering given by *Selection* should be separated. The intuition explaining the usefulness of this implementation is as follows: if there is a group of latents forming a pure subgraph of the true graph with a large number of pure indicators for each latent, then the initial step should identify such group. The consecutive steps will refine this solution without the risk of splitting the large clusters of variables, which are exactly the ones most likely to produce false positive decisions. RobustBuildPureClusters has the power of identifying the latents with large sets of pure indicators and refining this solution with more flexible rules, covering also cases where DisjointGroup fails.

Notice that the order by which tests are applied might influence the outcome of the algorithms, since if we remove an edge $X - Y$ in C at some point, then we are excluding the possibility of using some tests where X and Y are required. Imposing such restriction reduces the overall computational cost and statistical mistakes. To minimize the ordering effect, an option is to run the algorithm multiple times and select the output with the highest number of nodes.

Algorithm FindInitialSelection
Input: Σ, a sample covariance matrix of a set of variables $\mathbf{O}$

1. Start with a complete graph C over $\mathbf{O}$.
2. Remove edges of pairs that are marginally uncorrelated or uncorrelated conditioned on a third variable.
3. $C_0 \leftarrow C$.
4. Color every edge of C as BLUE.
5. For all edges $N_1 - N_2$ in C, if there is no other pair $\{N_3, N_4\}$ such that all three tetrads constraints hold in the covariance matrix of $\{N_1, N_2, N_3, N_4\}$, change the color of the edge $N_1 - N_2$ to GRAY.
6. For all pairs of variables $\{N_1, N_2\}$ linked by a BLUE edge in C

   If there exists a pair $\{N_3, N_4\}$ that forms a BLUE clique with N_1 in C, and a pair $\{N_5, N_6\}$ that forms a BLUE clique with N_2 in C, all six nodes form a clique in C_0 and DisjointGroup$(N_1, N_3, N_4, N_2, N_5, N_6; \Sigma) = true$, then remove all edges linking elements in $\{N_1, N_3, N_4\}$ to $\{N_2, N_5, N_6\}$.

   Otherwise, if there is no node N_3 that forms a BLUE clique with $\{N_1, N_2\}$ in C, and no BLUE clique in $\{N_4, N_5, N_6\}$ such that all six nodes form a clique in C_0 and DisjointGroup$(N_1, N_2, N_3, N_4, N_5, N_6; \Sigma) = true$, then change the color of the edge $N_1 - N_2$ to YELLOW.
7. Remove all GRAY and YELLOW edges from C.
8. $List_C \leftarrow$FindMaximalCliques(C).
9. Let H be a graph where each node corresponds to an element of $List_C$ and with no edges. Let M_i denote both a node in H and the respective set of nodes in $List_C$. Let $M_i \leftarrow$ RobustPurify(M_i, C, Σ);
10. Add an edge $M_1 - M_2$ to H only if there exists $\{N_1, N_2, N_3\} \subseteq M_1$ and $\{N_4, N_5, N_6\} \subseteq M_2$ such that DisjointGroup$(N_1, N_2, N_3, N_4, N_5, N_6; \Sigma) = true$.
11. $H_{choice} \leftarrow$ChooseClusteringClique(H).
12. Let $H_{clusters}$ be the corresponding set of clusters, i.e., the set of sets of observed variables, where each set in $H_{clusters}$ correspond to some M_i in H_{choice}.
13. $Selection \leftarrow$RobustPurify$(H_{clusters}, C, \Sigma)$.
14. Return $(Selection, C, C_0)$.

Table 9: Selects an initial pure model.