Learning Measurement Models

Ricardo Silva, Richard Scheines, Peter Spirtes and Clark Glymour

April 11, 2003
CMU-CALD-03-100

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Observed associations in a database may be due in whole or part to variations in unrecorded ("latent") variables. Identifying such variables and their causal relationships with one another is a principal goal in many scientific and practical domains. Previous work shows that, given a partition of observed variables such that members of a class share only a single latent common cause, standard search algorithms for causal Bayes nets can infer structural relations between latent variables. We introduce an algorithm for discovering such partitions when they exist. Uniquely among available procedures, the algorithm is (asymptotically) correct under standard assumptions in causal Bayes net search algorithms, requires no prior knowledge of the number of latent variables, and does not depend on the mathematical form of the relationships among the latent variables. We evaluate the algorithm on a variety of simulated data sets.

Keywords: Causality discovery, graphical models, structural equation models

1 Introduction

In this work, we study approaches to learn from data which hidden common causes could explain the association of observed variables.

For instance, suppose you are given the following induction problem: discover the causal relationships among variables measuring characteristics of a specie of small mammals in different habitats. There are variables such as increase in height per year (in some scale), increase in length of fur per year (in some other scale), approximate age where each animal achieved sexual maturity, degree of humidity of its habitat, degree of sunlight, amount of a specific kind of leaf that is the staple food of such animals, and so on. A sample of individuals and their respective habitats, measured over those features, is provided.

A further investigation tells you that such variables were originally chosen for a study relating the effects of environment in the growth of such animals. Things become more clear when you see such variables as *indicators* of a couple of unobserved, or *latent*, variables such as "enviromental quality" and "rate of maturity". A candidate model can be as simple as the one depicted in Figure 1.

However, a structure based only in background knowledge may not be satisfactory, and questions about the validity of some relationships will have to be tested. For example, in the model of Figure 1, is our abstract latent variable *Environmental quality* good enough to account for the associations between its indicators, or one has to consider a variable such as *Sunlight measure* as a direct cause of *Food availability*? What about the relation between the latents?

The study of latent variable models is a widely interdisciplinary enterprise affecting different fields of science, such as econometrics and social sciences, natural sciences and psychology (Harman, 1967; Cattel, 1978; Rayment and Jöreskog, 1993; Bartholomew et al., 2002; Malinowski, 2002). A large number of such models adopt, implicitly or explicitly, what we call the *measurement assumption*: observed variables are indicators of latent variables, i.e., observed variables are caused by latent variables, and such latents can be not only physical concepts that were not measured because of some pratical constraint, but also abstract factors hard to quantify with a single measure. Observed variables can be direct causes of other indicators, but not causes of latent variables. While it is easy to come up with examples where this requirement does not hold, many studies are designed in order to fulfill this assumption, such as in the questionnaires used in social sciences research and marketing, and in psychometrical measurements.

The measurement assumption is not merely an artifact for mathematical adequacy, but a standard principle for a variety of data analysis practices. At the end, we have a framework that approachs a problem by dividing it into two parts: a *measurement model*, which describes how latents affect indicators, and a *structural model*, describing how latents affect other latents.

Another motivation for explicitly creating a measurement model are the consequences of measurement error (Bollen, 1989). For instance, suppose we want to quantify the causal effect of exposure to lead in children's cognitive functions. There are direct and indirect mechanisms that might explain this effect, and the issues raised by such problem appear in many research contexts.

Researchers in policy analysis are interested in this type of problem because they need to control the environment in order to achieve the desired effects: should we intervene in how lead is spread in the environment? But what if it does not actually affect cognitive skills of children, but there is some hidden common cause that explains this dependency? How to quantify it? These are typical questions in econometrics and social sciences. But also researchers in artificial intelligence and robotics are attentive to such general problems. How does the world affect my agent/robot? How can it manipulate its environment in order to achieve its goals? If one does not know how to

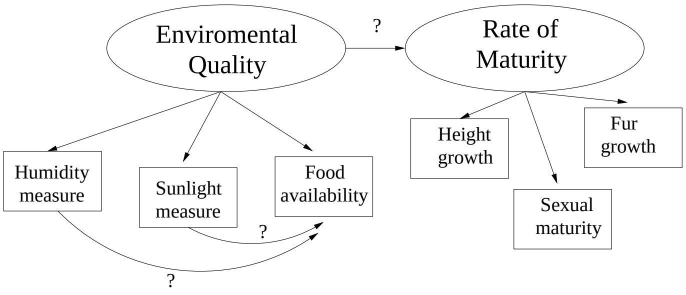


Figure 1: A causal model of enviromental effects on the biological growth of some animal specie. Edges marked with a "?" indicate relations in which the available prior knowledge is less certain.

quantify such effects, it is hard to believe in any decision theoretic machinery used as a criterion for action, since the prediction of the effects of a manipulation will be wrong. In order to perform sound prediction of manipulations, causal models are necessary. Usually such models are not readily available, and algorithms are necessary to learn them from data.

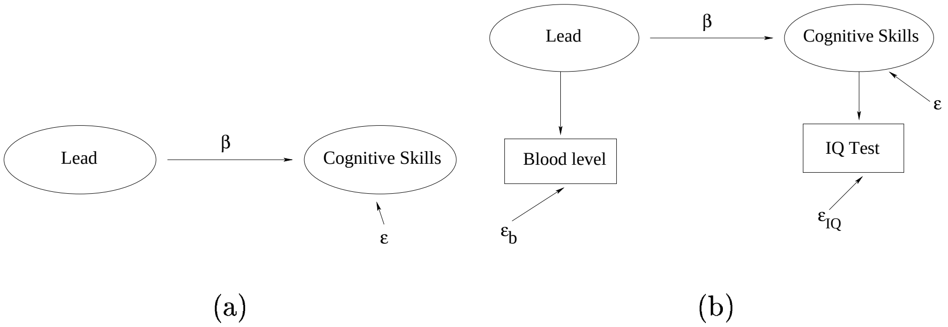


Figure 2: In (a), the underlying hypothetized phenomenon. In (b), how the model assumptions relates the measurements.

A simple causal model for the lead (L) and cognitive skills (C) problem is a linear regression model where L causes C, and they are related by the linear equation[1] $C = \beta L + \epsilon$, where ϵ is the usual zero-mean, normally distributed random variable. Figure 2(a) illustrates this equation as a causal graphical model. There is one important problem: one usually does not have well-defined criteria for quantifying what "lead exposure" and "cognitive skills" should be. The usual approach is relying on indirect measures, such as *Blood level concentration* (BL), which is an indicator of lead exposure. And cognitive functionality is probably one of the most ill-defined concepts in existence. Measures such as *IQ Tests* (IQ) have to be used as indicators of C. Our regression model has to be something along the lines of $IQ = \beta BL + \epsilon_{IQ}$. Figure 2(b) illustrates the new graphical model representing this regression.

However, if the measurement error of L through BL is not zero, i.e., $\epsilon_b \neq 0$, we cannot get a consistent estimator of β under the linear regression model. Without a consistent estimator, one would not be able test if β is zero, for instance. That happens not because regression is a

[1] We assume that both are centered at its mean.

defective method, but because this problem fails to meet its assumptions. By Figure 2(b), we see that there is a common cause between BL and IQ (*Lead*), which is strictly against one of the main assumptions in regression: if one wants consistent estimators of causal effects, there cannot be any hidden common cause between the regressor and the predictor.

The common cause problem does not happen if one is willing to perform randomized experiments, where children are exposed to varying degrees of lead on purpose. However, such experiments are unethical, and only observational data can be used. Techniques such as the ones described in Spirtes et al. (2000) and Pearl (2000) are necessary. Also, even if one has considerable control of its environment, such as an intelligent agent in an idealized world, manipulation may be expensive and cheap observational data should be used as much as possible.

One solution is fully modeling the latent structure. Additional difficulties arise in latent variable models, though. For instance, due to parameter identifiability reasons, we may require multiple indicators per latent (Bollen, 1989; Scheines et al., 1999). Other implications have to be considered. In our example, suppose we take into account a common cause between lead and cognitive abilities: the parent's attentiveness to home environment (P), with multiple indicators P_1, P_2, P_3 (Figure 3). We want to test if L is independent from C given P and, if so, conclude that lead is not a direct cause of alterations in children's cognitive functions. If these variables were observed, well-known methods of testing conditional independencies for certain families of probability distributions could be used. This is not the case. Some (flawed) existing solutions perform a test of conditional independence of *Blood Level* and *IQ* given a function of the indicators P_1, P_2, P_3, a so-called *scale* of P. A look at the graphical model in Figure 3 tells us that the only way such scale would d-separate[2] BL and IQ was if it was an invertible deterministic function of P, which one should not expect to happen in practice.

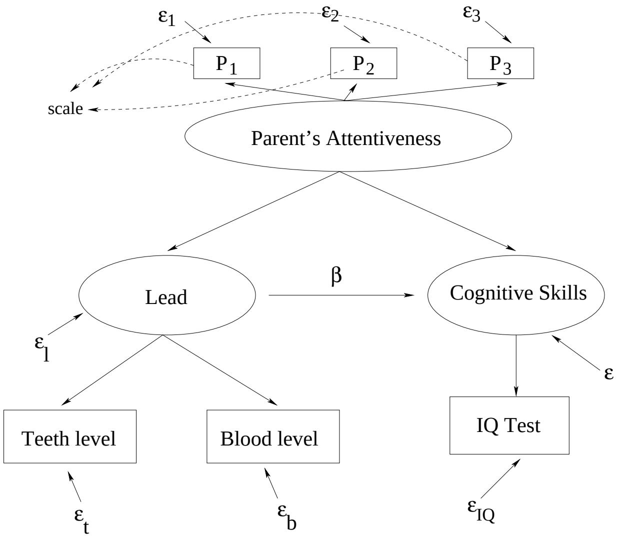


Figure 3: A graphical model with three latents.

In this report we do not discuss how to learn the causal relationships among the latents, as it was done in (Silva, 2002a). Even thought this is one of our main motivations, there are still many applications where knowing the measurement model alone is important. For example, in data mining applications we may want to know how the different measures cluster together and

[2] D-separation is a graphical criterion equivalent to conditional probabilistic independence under our axiomatic system of causality. Details are given in Pearl (1988).

how they can be interpreted as abstract factors with more interesting semantics. Factor analysis, for instance, is a standard latent variable model for this kind of exploratory data analysis, but requiring stronger assumptions and rather arbitrary rotation methods. A lower dimensional space could also be used for creating predictive models based in such abstract factors.

Other fields of applications include cognitive sciences, also interested in learning general factors that explain observed measurements of behavior and brain activity (factor analysis was originally motivated by studies in psychometrics). In artificial intelligence and knowledge engineering, learning underlying unknown factors is related to the concept of *ontology learning*. Such latent variables can be used as new concepts in large human-designed knowledge bases.

2 Problem statement and assumptions

The goal of learning measurement models is identifying abstract or unmeasured concepts ("factors") that causally explain the associations measured over a set of observable random variables. The language of graphical models (Jordan, 1998), a graphical causality calculus and the concept of d-separation will be used as a formal language for our models. If the reader is not familiar with the concept of d-separation and causal models, books such as Pearl (1988, 2000) and Spirtes et al.(2000) present the definitions in full detail. The following definitions introduce the families of measurement models of interest. A random variable and a node in a graph are treated as a single entity when both have the same name.

Definition 1 (Measurement model) *A directed acyclic graph (DAG) containing a set of latent variables $\mathbf{L}$, a set of error variables ϵ, a set of observed variables $\mathbf{O}$, two set of edges $\mathbf{E_O}$ and $\mathbf{E}_\epsilon$, forms a measurement model $M(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_O}, \mathbf{E}_\epsilon)$ if each latent in $\mathbf{L}$ is a parent of at least one variable in $\mathbf{O}$, none of the observed variables is a parent of any variable in $\mathbf{L} \cup \epsilon$, all nodes in $\mathbf{O}$ are children of some node in $\mathbf{L}$, any node in ϵ is a common parent of at least two nodes in $\mathbf{O}$ and is d-separated from every element of $\mathbf{L}$. Also, all edges in $\mathbf{E_O}$ have at least one endpoint in $\mathbf{O}$ and no endpoint in ϵ, and all edges in $\mathbf{E}_\epsilon$ have at least one endpoint in ϵ.*

The definition of a measurement model specifies in which way observed variables are indicators of common latent factors while not considering how such factors are causally connected, since no directed path between two latents is intermediated by an observed variable. Not allowing an observed variable to be a cause of a latent is stronger than not allowing causal paths among latents be intermediated by non-latent variables, but such (widely adopted) assumption will allow us later to derive stronger conclusions about such models. Variables in ϵ are also latent variables, but we refer to them as *error variables*, in a similar role to error terms in regression analysis. They represent uncertainty in the measurement. We do not exclude the possibility of different error terms being linked by paths. It is a standard practice to represent associations due to dependent errors when visually depicting a directed graph by using double-directed edges, which is the usual representation in structural equation models (Bollen, 1989) and causal models (Pearl, 2000; Spirtes et al., 2000). Another variation is to completely ignore error variables when drawing graphs, and any double-directed edges that would exist among them will be drawn among the observed variables that are descendants of such error terms.

Unless it is stated otherwise, we do not include the error variables in the figures depicting graphical models across this document, but we will include any necessary double-directed edges. In Figure 4 we have an example of a graphical model that is a measurement model and one that is not. Relations among latents are not considered.

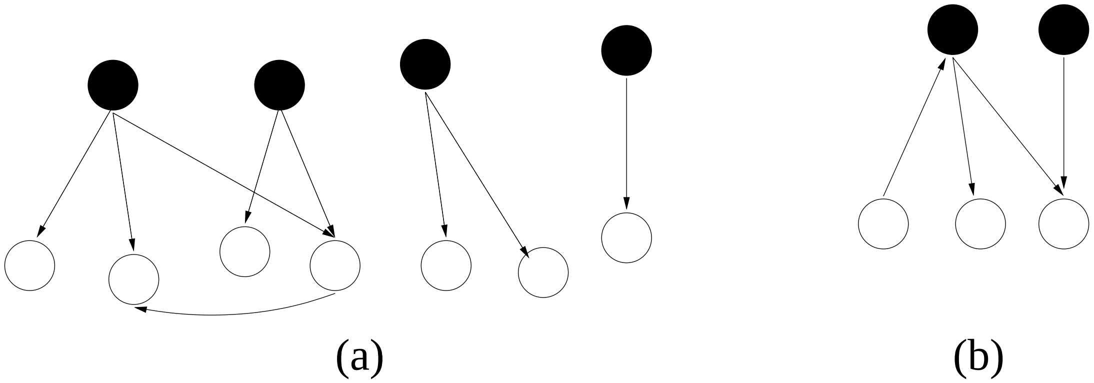


Figure 4: In the graphs above, white nodes represent observed variables and black nodes are latents. Figure (a) is an example of a measurement model. Relations among latents are just ignored. Error nodes are not represented. Figure (b) is an example of a graphical model that is not a measurement model.

The previous definition does not specify which is the parametric relationship between a variable and its direct causes. In this work, we will focus in the following class of measurement models:

Definition 2 (Linear measurement model) *A measurement model is linear if each observed variable is determined by a linear combination of its parents and an additive error term. A linear measurement model is defined by a set of equations* $O_i = \sum_j \lambda_{ij} L_j + \sum_k \eta_{ik} O_k + \sum_e \upsilon_e \epsilon_e + e_i$*,* $\forall O_i \in \mathbf{O}$*, where any* L_j *is in* $\mathbf{L}$*, all* ϵ_e *is in* ϵ*, and* e_i *is an extra error term with non-zero variance independent of every other variable in the measurement model.*

We assume for simplicity that all variables have zero mean. The reason for the extra error term is because we want to exclude deterministic relations among elements in our models. The next definition describes an important subclass of measurement models.

Definition 3 (Pure measurement model) *A measurement model* $M(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_O}, \mathbf{E}_\epsilon)$ *is pure if for every* $O_i \in \mathbf{O}$*,* O_i *is d-separated from every element in* $(\mathbf{L} - L_i) \cup (\mathbf{O} - O_i)$ *conditioned in some* $L_i \in \mathbf{L}$ *such that* L_i *is a parent of* O_i *in* M*.*

Figure 5 depicts a pure and an impure measurement model. In a pure measurement model, $\epsilon = \emptyset$, $\mathbf{E}_\epsilon = \emptyset$. Notice that each observed variable is still a non-deterministic function of its parents in linear measurement models, since a extra error term exists.

We say that a graph G is *faithful* to a joint probability distribution $f(\mathbf{V})$, where $\mathbf{V}$ is a set of random variables and for every $V \in \mathbf{V}$ there is a corresponding node in G, when every conditional (or marginal) independence statement about variables in $\mathbf{V}$ holds in $f(\mathbf{V})$ if and only if it also holds in G according to the graphical criterion of d-separation. The assumption of *faithfulness* is very common in causal models (Spirtes et al., 2000) and is described by Pearl (2000) under the name of *stability*. The following definition makes use of this assumption.

Definition 4 (Latent variable graph) *Given a set of latent variables* $\mathbf{L}$*, a set of error variables* ϵ*, a set of observed variables* $\mathbf{O}$*, three sets of edges* $\mathbf{E_O}$*,* $\mathbf{E_L}$ *and* $\mathbf{E}_\epsilon$*, a latent variable graph* $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E}_\epsilon)$ *is a directed acyclic graph faithful to the joint distribution* $f(\mathbf{L} \cup \mathbf{O} \cup \epsilon)$*. Also, all edges in* $\mathbf{E_L}$ *have both endpoints in* $\mathbf{L}$*, and the directed acyclic graph defined by the tuple* $(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_O}, \mathbf{E}_\epsilon)$ *forms a measurement model.*

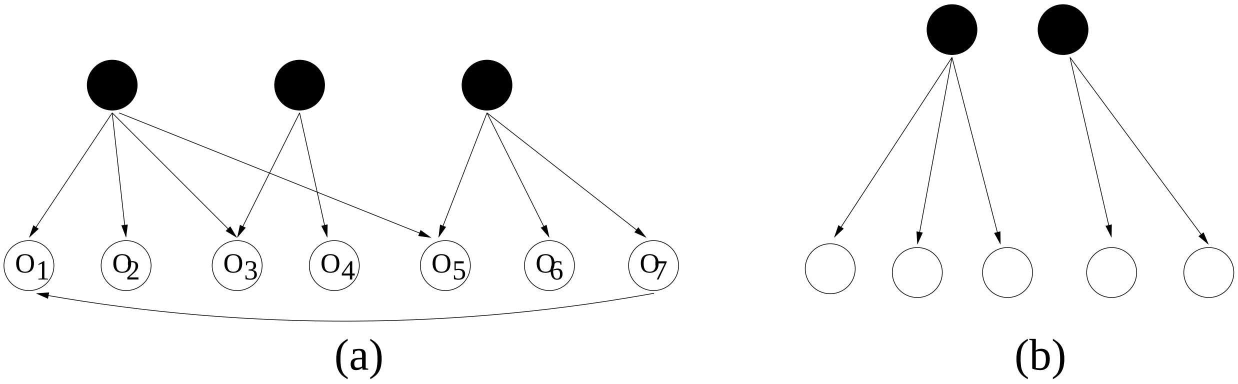


Figure 5: The graph in Figure (a) is not a pure model: O_1 and O_7 are d-connected given their latent parents, O_3 and O_5 have more than one parent. On the other hand, the graph in Figure (b) is a pure measurement model: every observed variable is d-separated from other indicators and latents given its (unique) latent parent.

We will also say that, given a latent variable graph, the tuple $(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_O}, \mathbf{E}_\epsilon)$ is its measurement model. The definition of pure latent variable graph and linear latent variable graph are analogous: they should have pure/linear measurement models. Pure graphs are important for reasons that will be explained later, and it is useful to know how to obtain a pure graph out of an impure one.

Definition 5 (Purification) *Given a latent variable graph $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E}_\epsilon)$, a purification is a pure latent variable graph obtained from G by a sequence of deletions of elements from $\mathbf{O}$ and all elements from ϵ.*

This definition requires that every observed node in the purification has just one latent parent in G. Since the output should be a faithful graph, this deletion cannot be arbitrary. The purpose of this definition is to identify pure measurement models that are a function of G, where each observed variable is d-separated from every other observed node given its (unique) latent parent, conditioning on an observed node can never d-connect indicators unconditionally disconnected, and such d-separations also hold in G. For instance, the graph in Figure 5(a) has a purification containing variables $\{O_2, O_4, O_6, O_7\}$ and any subset of this set. Notice that if O_1 is in a purification, then O_6 cannot be in the same graph even if we remove O_7: in this case, O_1 would be d-separated from O_6 in this graph conditioned on the latent parent of O_1, but the graph would not be faithful since O_1 is still dependent in O_6 given the latent parent of O_1.

In this report we introduce approaches that, given the covariance matrix of a set of random variables, discover the measurement model of some marginal of the distribution. We must formally define the set of assumptions by which we are able to prove the correctness of our method. The following definition specify the special class of latent variable graphs that corresponds to our primary assumptions.

Definition 6 (Purifiable linear latent variable graph) *A purifiable linear latent variable graph $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E}_\epsilon, \mathbf{G_S})$ is a graphical model such that $(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E}_\epsilon)$ is a linear latent variable graph and $\mathbf{G_S}$ is a non-empty set of purifications of G such that, in every graph $G_S \in \mathbf{G_S}$, all latent nodes have at least three observed children in G_S.*

The motivation for requiring at least three observed children per latent in purifications of G comes from identifiability issues. This will be evident in the next sections, where we introduce an algorithm for learning families of measurement models (*equivalence classes*) that fit a given covariance matrix Σ of a set of variables $\mathbf{O}$. The assumptions by which this algorithm work are:

- observed variables are continuous;
- Σ is faithfully generated by an unknown purifiable linear latent variable graph $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E}_\epsilon, \mathbf{G_S})$;
- the distributions of $\mathbf{O}$, $\mathbf{L}$ and ϵ have second moments;

Notice that this assumes faithfulness and linearity of the measurement model, and that by faithfulness no variable has zero variance. However, it does *not* assume which is the specific family of probability distributions for $\mathbf{O}$, $\mathbf{L}$ or ϵ. We will also implicitly assume that each latent is correlated to at least one other latent and there are at least two latents. These last assumptions are not essential. They are made only for the sake of keeping the algorithm slightly simpler and can be removed in later extensions of this report.

We do not expect being able to identify every possible component in a model (i.e., find every edge with every possible orientation, assigning a proper cluster for each observed node, and so on). In Section 3 we define classes of equivalent solutions, or *equivalence classes*, for our problem: a representation that includes the correct measurement model given the assumptions, plus a set of other possible measurement models that could not be distinguished by our algorithm. The output of our procedure is an equivalence class. After the explanation of the algorithm in Section 4 and Section 5 for the case where Σ in known, in Section 6 we explain which statistical tests can be used for the (usual) case where all we have is a covariance matrix obtained from finite samples. We also discuss heuristics for dealing with the fact that we have to perform sequential hypothesis testing that is prone to commit several mistakes. Section 8 shows some empirical results. Appendix C provides a proof of correctness for our approach.

Concerning which kinds of problems for which our methodology is appropriate or not recommendable, the general criterion should be: apply it when one believes that the underlying common causes have some unique indicators, i.e., its own direct effects that are not direct effects of other latents. For instance, many case studies in social sciences and econometrics follow that pattern (Bartholomew et al., 2002). On the other hand, if the problem suggests that most observed variables are measures of a large variety of common latent causes, then the proposed approach should not be able to identify much structure on it. The following list describes some examples of problems that are *not* prone to be solved with our method:

- general blind source separation problems, where measures are usually indicators of most of the latents (i.e., sources) at the same time;
- some chemometrics problems (Malinowski, 2002) for identifying chemical components in samples that contain mixtures of many different elements;
- a common model in analysis of text data: latent semantic analysis (Hofmann, 2001), where documents are considered to be generated by a mixture of different semantical topics.

On the other hand, there are natural scenarios for our algorithms:

- in econometrics, social sciences, psychometrics and natural sciences, it is important to describe the observed measures as indicators of a few, common latent concepts. By describing the relationship of observations through latents, one is able to take in account the consequences of measurement error (Bollen, 1989) and better quantify the effects of theoretical concepts on the observed measures. It also provides building blocks for the creation of new theories and concepts that explain observed phenomena, as well as relating previously known concepts to those obtained through empirical analysis of measurement models. Althought one has to carefully evaluate how to interpret, reify or discard the discovered latent concepts, this is an important step that no science proceeds without [3].

- in general, through data mining one wants to gain insight in a given data generation process in order to implement effective policies. Automatic discovery of measurement models can at least provide a quick and dirty exploratory data analysis tool for coming up with such insights.

- one of the most important problems in artificial intelligence is building large systems of common sense knowledge. Designing knowledge bases by grouping the many concepts of common sense by abstract, latent common causes may reduce this task to a more manageable size.

Also, it should be clear that the learning problem especifically treated in this report is a problem of qualitative nature: the goal is discovering clusters of variables, not quantifying the causal effect of a latent in each of its indicators. For quantification of effects given the structure, standard approaches exist for the parametric case where variables are multivariate normal (Bollen, 1989) and other distributions from the exponential family (Bartholomew and Knott, 1999). Independent component analysis is the semi-parametric method for the special case where latents are independent (and usually the measurement model is error-free, an unlikely case for the problems we are interested in modeling).

3 Equivalence classes

In our context, an equivalence class is a set of solutions that are admissible given the assumptions and the specific instance of a problem. For example, when learning DAGs from observational data, it is common that different DAGs explain the data with exactly the same adequacy. If the true DAG is $A \rightarrow B \rightarrow C$, given only the observed conditional independencies of $\{A, B, C\}$, in general no algorithm will be able to tell which of the graphs in $\{A \rightarrow B \rightarrow C, A \leftarrow B \rightarrow C, A \leftarrow B \leftarrow C\}$ is the one that generated that distribution. An useful output would be the complete set of possible solutions, often expressed in a shortened representation. Pearl (2000) describes an equivalence class of DAGs by *patterns*, a chain graph where some edges are not oriented (the pattern for our example would be $A - B - C$). A *minimal* equivalence class would be a class that includes the minimal possible number of solutions for a problem given the assumptions. The definition of a minimal equivalence class is independent of the algorithm that may be designed for the problem. For some cases, we know which should be the minimal equivalence class and which algorithms can be used to discover the minimal equivalence class of any given instance of our problem (e.g., Meek, 1995).

In our problem, the only input given to us is a set of observed random variables $\mathbf{O}$ and some function of their joint probability distribution. In this report, we will make use only of the co-

[3] For example, there are half a dozen different fundamental models in superstring theory, all of them with the same empirical predictive power, but different in their latent structure. However, it is important for physicists to study this equivalence class of models in order to provide guidance for theories and future experiments that may eventually reduce this class of equivalent models to a smaller one.

variance matrix Σ of the observed variables, and assume such variables have zero mean. There will be situations where some specific properties of a measurement model cannot be identified, and whenever we speak of an equivalence class, it should be implicit this is an equivalence class with respect to the constraints we evaluate. For example, suppose that a given observed indicator O_i of some latent L_j is directly caused by every other indicator in the corresponding latent graph with the exception of some indicator O_k. The algorithm described in the next section uses a specific family of constraints to identify relationships among these variables, but in this case no constraint will hold between O_i and any of the other variables. We will not be able to tell that O_k is a direct cause of O_i or not, and actually O_i will not appear in our measurement pattern.

As another example of feature that we cannot identify, suppose O_i, O_j, O_k are indicators of some latent, and the edges $O_i \rightarrow O_j$, $O_j \rightarrow O_k$ are in the underlying latent variable graph, but there is no edge $O_i \rightarrow O_k$. Our algorithm may be able to identify that there is a path from O_i to O_k that does not go through their common latent parent, but it will not be able to tell if this is a direct cause or not. Notice that, with a few more conditions that could be assumed or verified in some cases, one would be able to detect this chain, but the algorithm here described will not explore this possibility. We do not claim that the algorithm introduced in this report is *complete* with respect to its assumptions, i.e., it will not discover the minimal equivalence class of a measurement model.

The output of the algorithm introduced in the next section is a graph MM_G with directed and undirected edges and the following properties:

- the graph MM_G has a set $\mathbf{T}$ of latent variables and observed variables $\mathbf{O'} \subseteq \mathbf{O}$, where $\mathbf{O}$ is the original set given as input. Notice that we denote latents in the pattern by $\mathbf{T}$ instead of $\mathbf{L}$, because obtaining a one-to-one mapping from one set to the other is not guaranteed;

- every latent has at least two children;

- some pairs of observed variables may be connected by an undirected edge. Some pairs of latents are connected by an undirected edge. No latents have parents;

- there are no error nodes;

Let $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E}_\epsilon, \mathbf{G_S})$ be a purifiable linear latent variable graph such that the covariance matrix of $\mathbf{O}$ is given as input to our algorithm. Then MM_G represents possible measurement models such that the measurement model of every $G_S \in \mathbf{G_S}$ is a subgraph of MM_G. Our problem can be seeing as a *clustering* problem: identifying how variables are clustered together, where a *cluster* in a measurement model is any set of indicators that share a same latent parent. Clusters can overlap in general measurement models. Clusters cannot overlap in pure measurement models. Sometimes we will refer to the elements of $\mathbf{G_S}$ as *solution graphs*, because they can be identified, while this is not usually the case for G. The fact that there are no error nodes in the pattern does not mean that measurement error is not being considered: it is implicity in the linear parameterization of the model.

The information encoded by MM_G explains the tested constraints in the covariance/correlation matrix of the observed variables according to the following properties:

1. if there is no node in MM_G representing a variable O, then there is no possible purifiable linear latent variable graph $G_0(\mathbf{L_0}, \mathbf{O}, \epsilon_\mathbf{0}, \mathbf{E_{L_0}}, \mathbf{E_{O_0}}, \mathbf{E}_{\epsilon_\mathbf{0}}, \mathbf{G_{S_0}})$ where O is included in some $G_S \in \mathbf{G_{S_0}}$;

2. if there is a pair of observed nodes O_1, O_2 connected by an undirected edge in MM_G, then there is no purifiable linear latent variable graph $G_0(\mathbf{L_0}, \mathbf{O}, \epsilon_\mathbf{0}, \mathbf{E_{L_0}}, \mathbf{E_{O_0}}, \mathbf{E}_{\epsilon_\mathbf{0}}, \mathbf{G_{S_0}})$ that can include both O_1 and O_2 in the same $G_S \in \mathbf{G_{S_0}}$;

3. if there is a pair of observed nodes O_1, O_2 that do not appear in any common cluster in MM_G, then there is no purifiable linear latent variable graph $G_0(\mathbf{L_0}, \mathbf{O}, \epsilon_\mathbf{0}, \mathbf{E_{L_0}}, \mathbf{E_{O_0}}, \mathbf{E}_{\epsilon_\mathbf{0}}, \mathbf{G_{S_0}})$ that can include both O_1 and O_2 in the same cluster in some $G_S \in \mathbf{G_{S_0}}$;

4. let $\mathbf{C_T} = \{T_1, T_2, ..., T_{|\mathbf{C_T}|}\}$ be a maximum clique among latent nodes in MM_G such that each $T_i \in \mathbf{C_T}$ has a subset of indicators $\mathbf{O_i}$ with the following joint properties: (i) $\forall i, |\mathbf{O_i}| \geq 3$; (ii) $\forall O_{ip} \in \mathbf{O_i}, O_{jq} \in \mathbf{O_j}, i \neq j$, O_{ip} and O_{jq} do not have any common latent parent in MM_G nor are linked by an undirected edge. Let MM'_G be the measurement model containing each $T_i \in \mathbf{C_T}$ and the respective indicators $\mathbf{O_i}$. Then there is a one-to-one relation $L_G : \mathbf{C_T} \rightarrow \mathbf{L}$ from the set of latents in $\mathbf{C_T}$ onto the set of latents $\mathbf{L}$ in G. We say that $L_G(T) = L$, $T \in \mathbf{C_T}, L \in \mathbf{L}$ if and and only if all children of T in MM'_G are also children of L in G. There is at least one maximum clique satisfying this property, which will then have size $|\mathbf{L}|$.

Any purifiable linear latent variable graph containing the same observed variables and entailing the same constraints we test will lead to the same output in our algorithm. The set of such graphs will be called a *measurement equivalence class*, denoted by $MM(\mathbf{O}, \Sigma)$. As hinted at the beginning of this section, we use this representation as a shortened representation for $MM(\mathbf{O}, Constraints(\Sigma))$, where $Constraints(\Sigma)$ are those constraints that are tested in our algorithm (in the same vein by which standard DAG equivalence classes are defined with respect to conditional independence constraints). The graphical representation of the equivalence class described above is a *measurement pattern*.

4 An algorithm for learning measurement equivalence classes

The algorithm here described builds a measurement pattern of a unknown purifiable linear latent variable graph with a known observed covariance matrix Σ by evaluating the validity of *tetrad constraints* among sets of four variables. Given the covariance matrix of four random variables $\{A, B, C, D\}$, we have that zero, one or three of the following constraints may hold:

$$\begin{aligned}
\sigma_{AB}\sigma_{CD} &= \sigma_{AC}\sigma_{BD} \\
\sigma_{AC}\sigma_{BD} &= \sigma_{AD}\sigma_{BC} \\
\sigma_{AB}\sigma_{CD} &= \sigma_{AD}\sigma_{BC}
\end{aligned}$$

The importance of tetrad constraints is that they can be used to reduce the set of possible relationships among the latent parents and their observed variables if the indicators are a linear combination of its parents. Let $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E}_\epsilon)$ be a linear latent variable graph. Define $L(O) \in \mathbf{L}$ as one specific latent parent of $O \in \mathbf{O}$. The following results are described in Spirtes et al. (2000):

- if all three tetrad constraints hold for a set of random variables $\{A, B, C, D\}$ and no one of these variables is an ancestor of one of the other three, then their respective nodes in the graph are d-separated by the latent parents of these nodes, independently of the relationship among the latents (even if they are the same latents);

- if $\{A, B, C\}$ are three indicators of some given latent $L = L(A) = L(B) = L(C)$, and L d-separates any pair in $\{A, B, C\}$, and there is a fourth variable D in $\mathbf{O}$ such that D is d-separated from $\{A, B, C\}$ given L (or some $L(D)$), then all three tetrad constraints hold in the set $\{A, B, C, D\}$, it does not matter how L interacts with $L(D)$ (or even if $L = L(D)$);
- if $\{A, B\}$ are two indicators of some given latent $L_1 = L(A) = L(B)$, $\{X, Y\}$ are two indicators of some given latent $L_2 = L(X) = L(Y)$, $L_1 \neq L_2$, every element in $\{A, B, X, Y\}$ is d-separated from the others given $\{L_1, L_2\}$, then just one tetrad constraint will hold among elements of the set $\{A, B, X, Y\}$, it does not matter how L_1 interacts with L_2;

The proof of the first assertion requires linearity among latents. The other two do not impose this requirement. An approach for learning a measurement model is finding which variables could be clustered into one-factor models, which are basically conditional naive Bayes models where the observed variables are linear indicators of a given latent, and these observed variables are independent conditioned on the latent. A linear one-factor model has to have at least four indicators to be able to be identified and tested. All three tetrad constraints hold among variables which form a linear one-factor model. One could explore the possibility of clustering together variables that belong to a same one-factor model.

Unfortunately, it is possible to have a set of four variables that satisfies all three tetrad constraints, and yet they belong to different clusters. For example, if $\{X, Y, Z\}$ are pure indicators of L_1 (i.e., each one is d-separated from every other variable in the causal graph given L_1), and K is a pure indicator of L_2, the set $\{X, Y, Z, K\}$ will satisfy all three tetrad constraints. Also, if A is a pure indicator of L_1, $\{X, Y\}$ are pure indicators of L_2 and B is a pure indicator of L_3, the causal relationships among L_1, L_2 and L_3 are linear, and L_2 d-separates L_1 and L_3, then all three tetrad constraints will hold in the set $\{A, B, X, Y\}$. The challenge is knowing when some set of variables should or should not belong to the same cluster and putting together this information, which is not a trivial task.

Yet we present here a feasible approach for this problem. Given a covariance matrix and the assumptions defined in Section 2, tetrad constraints will be used to find an equivalence class as described in the previous section. The two main tasks of this algorithm are clustering and impurity identification. Next subsection describes the general approach for accomplishing these tasks, followed by another subsection that explains how this information is translated to a measurement pattern.

4.1 Clustering and impurity identification

The goal of clustering is grouping the observed variables into sets such that every element in a given set is an indicator of one or more latents in G, but no pair always clustered into different sets can be indicators of a same latent. A detailed account of this procedure is given in Table 1. The function $TetradScore(\mathbf{Set}; \Sigma)$ counts the number of tetrad constraints that hold among elements in $\mathbf{Set}$, which have a covariance matrix as a submatrix of Σ, and where for no triple $\{X, Y, Z\} \subset \mathbf{Set}$ we have $\rho_{XY.Z} = 0$, the partial correlation of X and Y given Z being zero. If for some triplet we have $\rho_{XY.Z} = 0$, the $TetradScore$ is defined to be zero. Denoting the true graph that generated Σ by $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E_\epsilon}, \mathbf{G_S})$, the outline of such procedure is:

- first of all, identify which variables are uncorrelated. By the faithfulness assumption, it turns out that such variables cannot be in a same cluster;

- identify which pairs of variables (X, Y) cannot form a one-factor model with some other pair. If it is not possible to find such one-factor model, X and Y cannot be part of any graph in $\mathbf{G_S}$ at the same time, or otherwise we could construct such a one-factor model (for instance, with two other elements from the cluster of X, if X and Y are not in the same cluster);
- the next step, and the most informative one, is deciding which pairs of variables should not be in a same cluster by evaluating the predicate $Unclustered(\mathbf{Set_1}, \mathbf{Set_2}; \Sigma)$, as defined in Table 2;
- after we have all this pairwise information, the next step is to identify those groups formed by variables where no pair was labeled as incompatible by any of the three criteria above. Finally, a measurement pattern is build out of this clustering and impurity information as discussed in the next subsection.

The algorithm described in Table 1 represents the pairwise information as an undirected graph with colored edges, where each pair of nodes may be linked by at most one edge. Initially, we create a graph where all observed variables are vertices, and every pair is linked by a *Black* edge. This colored edge represents we have no information about the relationship of its endpoints: such variables may or may not belong to the same cluster, and they may or may not appear together in some pure solution of the underlying true latent model G.

The first learning step (Step 4 in Table 1) is getting rid of the edges linking variables that are uncorrelated. Throught the execution of this algorithm, the lack of an edge between a pair will represent that such pair of variables cannot be in a same cluster. For those pairs that cannot appear together in any one-factor model, the edge is changed to *Gray*. This will be typically the case for nodes that are not d-separated by $\mathbf{L}$. Throught the execution of this algorithm, a *Gray* edge between a pair will indicate that such pair of variables cannot be at the same time in any pure model. Those correlated variables that are not impure with respect to each other will be linked by *Blue* edges.

Figure 6(a) illustrates a very simple latent variable graph of three latents. Figure 6(b) is the first auxiliary graphical structure built at Step 3: a complete graph with *Black* edges, NG. At the end of Step 4, we remove edges in NG among uncorrelated pairs. That will get rid of all edges defined by $\{1, 2, 3, 4\} \times \{9, 10, 11\}$. We will also change the color of the edge (in NG) of those pairs that cannot be part of any one-factor model. In our example, this is indicated by an edge of *Gray* color between nodes 3 and 5. The other remaining edges are changed to a different color, *Blue*. Figure 6(c) illustrates the NG graph for this problem at the end of Step 4.

In the following step (Step 5 in Table 1), we want to remove those edges that connect variables that cannot appear in the same cluster. A simple test that guarantees this property is applied: the *Unclustered* test. Given two variables to be tested, O_x and O_y, the motivation behind it is as follows: find other four variables (say, O_1, O_2, O_3, O_4) such that $\{O_x, O_y, O_1, O_2\}$ forms a one-factor model, $\{O_x, O_y, O_3, O_4\}$ forms also a one-factor model, but $\{O_x, O_y, O_1, O_3\}$ entails only one tetrad constraint. If O_x and O_y shared one latent parent, we would have to have all three tetrad constraints entailed. If we do not have all three, then O_x and O_y cannot be indicators of a same latent. More intuition behind this test is discussed in Appendix A.

Also, if O_x and O_y are two variables that could not be proved to belong to different clusters, then they will be tested if they cannot appear together in any solution. If not, the edge between them is changed to *Yellow*. Throught the execution of this algorithm, a *Yellow* edge between a pair will represent that such pair of variables cannot be at the same time in any pure model, but they may still be d-separated given their latent parents. This may happen when two variables are part of some one-factor model with other two variables, but we cannot use *Unclustered* with them because there are no other four variables that satisfy the requirements of this test. The relation between such nodes remains unidentified.

From this point in the execution of the algorithm, a *Blue* edge means that two variables may belong to the same true cluster in a pure measurement model induced by the true underlying graph G (this may still not be true, but we do not have any evidence in contrary). Figure 6(d) illustrates the NG graph for this problem at the end of Step 5. For instance, the edge between nodes 2 and 6 was removed because the sets $\{1, 2, 3\}$ and $\{6, 7, 8\}$ satisfy $Unclustered(\{1, 2, 3\}, \{6, 7, 8\}; \Sigma)$. In order to speed-up the procedure we actually remove all edges in $\{1, 2, 3\} \times \{6, 7, 8\}$ in one step. No $Yellow$ edge appears in this example.

To conclude the clustering procedure, we split the undirected graph NG into components such that no pair of variables that appear in different components can appear together in the same cluster in some purification of a purifiable linear latent variable graph faithfully generating Σ. We accomplish this (Step 6) by generating a set of graphs, where each graph is a component of our current undirected graph NG where all but *Blue* edges are removed: there is no possibility that nodes in different components of such graph will be part of a same cluster. The corresponding *Gray* and $Yellow$ edges are added back when the components are generated.

We are now able to generate a clustering, by obtaining the set of all maximal cliques across all the previously computed components (Step 7) of size two or more. The intuition is that for each latent, each set of its indicators that appear in some $G_S \in \mathbf{G_S}$ will appear together in some of the cliques. Figure 6(e) illustrates the **Clustering** set of graphs (cliques obtained from components of NG_{Blue}, with intra-components *Gray* and $Yellow$ edges added back) for this problem at the end of Step 7. Notice that the edge between 3 and 5 does not show up here, but will be reappear later in the algorithm (it is still in NG, but not in **Clustering** because it is an inter-component edge, not an intra-component one).

Now that we have a clustering, the next task will be recoding it as a measurement pattern.

4.2 Building the measurement pattern

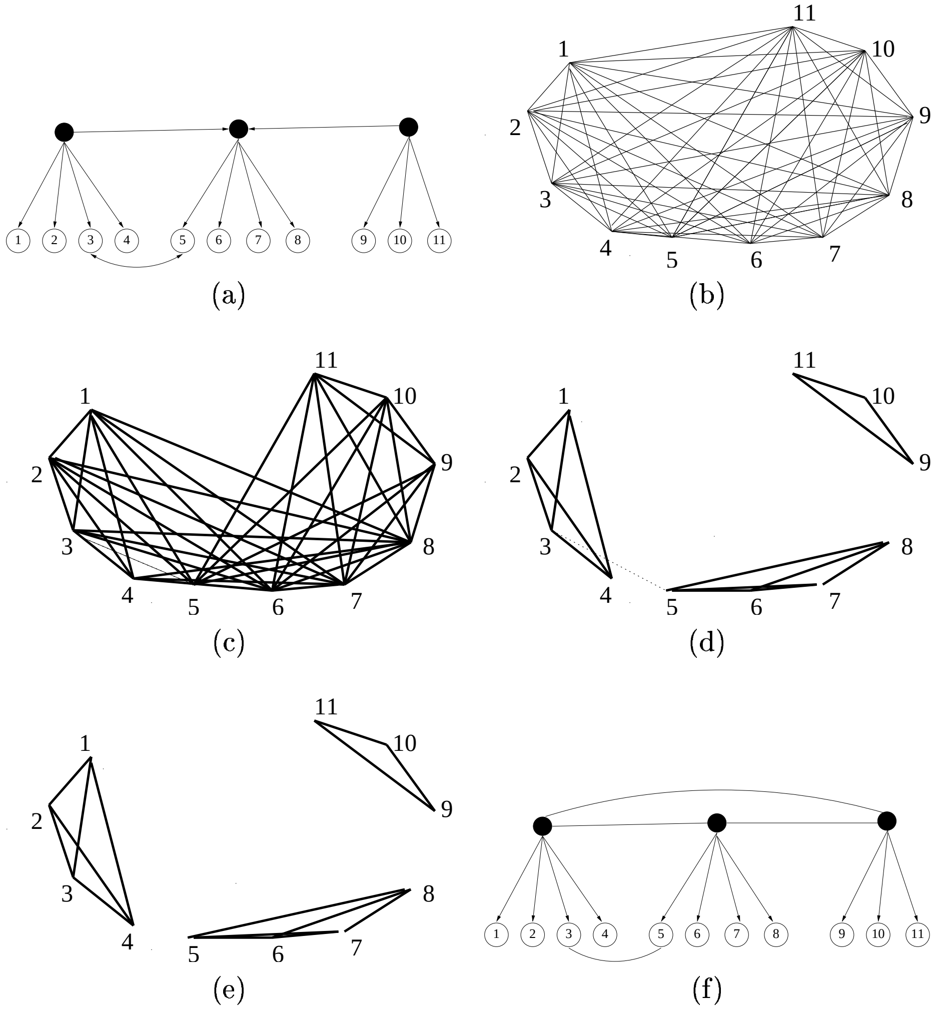
We have a clustering **Clustering** and a graph, NG, with all the necessary pairwise information to build the measurement pattern for $\mathbf{O}, \Sigma$. Table 3 describes this process in full detail.

Each cluster C_i in **Clustering** is transformed into a one-factor model, with an unique latent parent T_i. Directed edges from the latent to each of the observed nodes in the cluster will be added to our output pattern. Nodes that appear in more than one cluster will have multiple latent parents.

A pattern graph $(\mathbf{V_p}, \mathbf{E_p})$, where $\mathbf{E_p}$ may contain directed and undirected edges, is built by grouping together all elements in **Clustering**. Now, the impurities between nodes of different clusters have to be identified. Step 3 of Table 3 simply copies the *Gray* and $Yellow$ edges of NG back to $\mathbf{E_p}$.

The final step is identifying which clusters cannot appear together in a pure measurement model that corresponds to the initial assumptions. An edge between two clusters will be added if and only if there is a subset of six distinct nodes in the combined pair of clusters, three on each, where the *Unclustered* test holds. Figure 6(f) is the final measurement pattern for the true graph of Figure 6(a). More examples are given in Appendix B.

The intuition behind this last step should be made clear: typically, we expect to have more latents in the measurement pattern than in the true (unknown) graph. That can happen because of impurities: there will probably be nodes with more than one parent; also, a node may not



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6: A step-by-step demonstration of how the graph in Figure (a) will generate the measurement pattern in Figure (f). *Blue* edges are represented by bold black edges, and *Gray* edges are represented by dotted ones.

be separable from a cluster that is not one of its in the true graph because of other impurities that makes the predicate *Unclustered* fails whenever we have that node as part of the predicate argument. Moreover, as discussed in Section 6, statistical errors due to using finite samples instead of the true covariance matrix may also induce more latents than those found in the true graph. The criterion of not linking latents that cannot be part of a pure graph with three indicators will be enough for identifying the correct number of latents (assuming the true covariance matrix is known) and creating another kind of output – purified graphs – that is much more robust to statistical errors (in the case of finite samples).

5 Purification

The algorithm described in the previous section is able to find a measurement pattern provided the true covariance matrix of the population. The following theorem states the correctness of the

```
function FindMeasurementPattern
inputs   $\mathbf{O}$, a set of observed random variables
         $\Sigma$, the covariance matrix of $\mathbf{O}$
output   a measurement pattern

1. Let $NG(\mathbf{O}, \mathbf{E})$ be an undirected graph having $\mathbf{O}$ as vertices and $\mathbf{E}$ as edges,
   where each edge is colored and there is at most one edge between each pair
   of vertices
2. Define $NG_{\mathbf{Colors}}$ as the subgraph induced by $NG$ containing all of its
   vertices and only edges in $\mathbf{E}$ that are of some color in $\mathbf{Colors}$
3. Make $NG$ a complete graph with $Black$ edges only
4. while $\exists\{o_i, o_j\} \subset \mathbf{O}$ such that there is a $Black$ edge between then in $NG$
     if $\sigma_{o_io_j} = 0$
        remove the edge between $o_i$ and $o_j$
     else if $\exists\{o_a, o_b\}$ such that $\{o_a, o_b, o_i, o_j\}$ is a clique
     in $NG_{\{Black,Blue\}}$ and $TetradScore(o_a, o_b, o_i, o_j; \Sigma) = 3$
       turn the color of each edge between elements in $\{o_a, o_b, o_i, o_j\}$ to $Blue$
     else turn the color of the edge $(o_i, o_j)$ to $Gray$
5. for all $\{o_i, o_j\} \subset \mathbf{O}$ such that there is a $Blue$ edge between then in $NG$
     if $\exists\{o_a, o_b, o_c, o_d\}$ such that $\{o_a, o_b, o_i\}$, $\{o_c, o_d, o_j\}$ are two cliques in $NG_{Blue}$
     and $Unclustered(\{o_a, o_b, o_i\}, \{o_c, o_d, o_j\}; \Sigma)$
       remove all edges $\{o_a, o_b, o_i\} \times \{o_c, o_d, o_j\}$
     else if $\neg\exists\{o_a, o_b, o_c, o_d\}$ such that $\{o_a, o_b, o_c\}$, $\{o_d, o_i, o_j\}$ are cliques in $NG_{Blue}$
     and $Unclustered(\{o_a, o_b, o_c\}, \{o_d, o_i, o_j\}; \Sigma)$
       turn the color of the edge $(o_i, o_j)$ to $Yellow$
6. Let $\mathbf{Components}$ be the set of disjoint components of the graph $NG_{Blue}$
7. for all $C \in \mathbf{Components}$
     for all $o_i, o_j \in C$
       if $o_i$ and $o_j$ are connected in $NG_{Gray}$, add a $Gray$ edge $(o_i, o_j)$ to $C$
       if $o_i$ and $o_j$ are connected in $NG_{Yellow}$, add a $Yellow$ edge $(o_i, o_j)$ to $C$
8. Let $\mathbf{Clustering}$ be the set of all maximal cliques in $\mathbf{Components}$ of size
   two or more
9. return BuildFinalPattern($\mathbf{Clustering}$, $NG$)
```

Table 1: Returns the measurement pattern of an inducible linear pure latent variable graph.

```
function Unclustered
inputs   $\mathbf{O_1}, \mathbf{O_2}$, two disjoint sets of observed random variables
         $\Sigma$, a covariance matrix including the covariance matrix of $\mathbf{O_1} \cup \mathbf{O_2}$
output   a true/false decision
if all covariances in $\mathbf{O_1} \times \mathbf{O_2}$ are zero
  return true
else return
    $\forall O_x, O_y, O_z \in \mathbf{O_1} \cup \mathbf{O_2}, \sigma_{O_xO_y} \neq 0$ and $\rho_{O_xO_y.O_z} \neq 0$ and
    $\forall O \in \mathbf{O_1}, TetradScore(O, \mathbf{O_2}; \Sigma) = 3$ and
    $\forall O \in \mathbf{O_2}, TetradScore(O, \mathbf{O_1}; \Sigma) = 3$ and
    $\forall \{O_x, O_y\} \subset \mathbf{O_1}, \{O_a, O_b\} \subset \mathbf{O_2}, \sigma_{O_xO_a}\sigma_{O_yO_b} = \sigma_{O_xO_b}\sigma_{O_yO_a} \neq \sigma_{O_xO_y}\sigma_{O_aO_b}$
```

Table 2: Returns true only if no variable in $\mathbf{O_1}$ shares a same parent with any variable in $\mathbf{O_2}$. The symbol $\rho_{O_xO_y.O_z}$ represents the partial correlation of O_x and O_y conditioned on O_z.

procedure `FindMeasurementPattern` (Table 1):

Theorem 1 *Let $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E}_\epsilon, \mathbf{G_S})$ be the purifiable linear latent variable graph that generates the covariance matrix Σ of a set of observed random variables $\mathbf{O}$. Then, G will be in the measurement equivalence class $MM(\mathbf{O}, \Sigma)$, and such class will be given by the measurement pattern obtained throught* `FindMeasurementPattern`$(\mathbf{O}, \Sigma)$.

This theorem is proved in Appendix C.

Given an equivalence class of measurement models obtained from `FindMeasurementPattern`, we can now perform a purification of this model. This is useful for methods such as the one described in (Silva, 2002a) if one wants to learn causal relationships among the latent variables in the linear case. This is also necessary if one wants to know how many latents are in the underlying latent variable graph that is assumed to have generated the data, since the measurement pattern does not tell you that directly. Unfortunately, by a simple reduction from `3SAT`, finding a pure subgraph given a measurement pattern is NP-Hard: to see this, recast a 3SAT problem as a measurement pattern with a cluster for each clause and each literal as an individual node, where each pair of literals representing different truth assignments for a same variable should be linked by an undirected edge, each latent has two extra and individual dummy indicators not linked to any other indicator, and all latents are pairwise linked[4].

The algorithm `Purify` described in Table 4 takes as an input a measurement pattern. We perform a worst-case exponential search that selects clusters and pure indicators. The output is a linear pure measurement model for the latents in G (without explicit error nodes) using a subset of the indicators given as input. The interesting feature of this algorithm is that is able to select clusters such that each latent in the final solution has a one-to-one correspondence to latents in the original true graph G, as stated by the next theorem and corollary.

Remember the definition of $L_G(\cdot)$ from Section 3. We define the relationship $=_{MM}$ for two

[4] This does not exclude the possible existance of some other approach that can solve that polynomially using, for example, the quantitative information present in the covariance matrix. We do not know if this is possible or not. We still have to prove that finding the measurement pattern is NP-Hard.

```
function BuildFinalPattern
inputs   Clustering, a set of complete undirected graphs of observed random
         variables
         $NG$, a undirected graph of observed random variables
output   a measurement pattern

1. Let $\mathbf{V_p} = \emptyset, \mathbf{E_p} = \emptyset$
2. for each $C_i(\mathbf{O_i}, \mathbf{E_i}) \in \mathbf{Clustering}$
     Let $T_i$ be a new latent node
     $\mathbf{V_p} \leftarrow \mathbf{V_p} \cup \mathbf{O_i} \cup T_i$
     for all $O \in \mathbf{O_i}$
       $\mathbf{E_p} \leftarrow \mathbf{E_p} \cup (T_i, O)$, where $(T_i, O)$ is a directed edge.
3. for all pairs $O_1 \in \mathbf{O_1}, O_2 \in \mathbf{O_2}$, where $\{C_1(\mathbf{O_1}, \mathbf{E_1}), C_2(\mathbf{O_2}, \mathbf{E_2})\} \subseteq \mathbf{Clustering}$
     if there is an edge $(O_1, O_2)$ in $NG_{\{Gray,Yellow\}}$
       $\mathbf{E_p} \leftarrow \mathbf{E_p} \cup (O_1, O_2)$, where $(O_1, O_2)$ is undirected.
4. for all pairs $\{C_i(\mathbf{O_i}, \mathbf{E_i}), C_j(\mathbf{O_j}, \mathbf{E_j})\} \subseteq \mathbf{Clustering}$
     if $\exists\{O_x, O_y, O_z\} \subseteq \mathbf{O_i}, \{O_a, O_b, O_c\} \subseteq \mathbf{O_j}$, where $\{O_x, O_y, O_z\} \cap \{O_a, O_b, O_c\} = \emptyset$
      and $Unclustered(\{O_x, O_y, O_z\}, \{O_a, O_b, O_c\}; \Sigma) = true$,
       $\mathbf{E_p} \leftarrow \mathbf{E_p} \cup (T_1, T_2)$, where $(T_1, T_2)$ is undirected.
5. return $(\mathbf{V_p}, \mathbf{E_p})$
```

Table 3: Given information about possible clusters and impurities, build the corresponding measurement pattern

latent variable graphs graphs $G_1(\mathbf{L_1}, \mathbf{O_1}, \epsilon_1, \mathbf{E_{L_1}}, \mathbf{E_{O_1}}, \mathbf{E_{\epsilon_1}})$ and $G_2(\mathbf{L_2}, \mathbf{O_2}, \epsilon_2, \mathbf{E_{L_2}}, \mathbf{E_{O_2}}, \mathbf{E_{\epsilon_2}})$ as $G_1 =_{MM} G_2$ if and only if $\mathbf{O_1} = \mathbf{O_2}$ and for each $L_1 \in \mathbf{L_1}$ there exists an unique $L_2 \in \mathbf{L_2}$ such that $L_{G_1}(L_1) = L_2$ and $L_{G_2}(L_2) = L_1$. For two sets of latent variable graphs $\mathbf{G_1}$ and $\mathbf{G_2}$, we have $\mathbf{G_1} =_{MM} \mathbf{G_2}$ if for every $G_1 \in \mathbf{G_1}$ there is an unique $G_2 \in \mathbf{G_2}$ such that $G_1 =_{MM} G_2$ and $|\mathbf{G_1}| = |\mathbf{G_2}|$.

We define a purification of a measurement pattern $MM(\mathbf{V_p}, \mathbf{E_p})$ as a directed acyclic graph $MM_{Pure}(\mathbf{V'_p}, \mathbf{E'_p})$ where $\mathbf{V'_p} \subseteq \mathbf{V_p}$, $\mathbf{E'_p} \subseteq \mathbf{E_p}$, latent nodes in $\mathbf{V'_p}$ form a maximum clique in MM, each latent has at least three children in MM_{Pure} and no pair of observed nodes in $\mathbf{V'_p}$ is linked by an undirected edge in MM and MM_{Pure} and no two observed nodes in $\mathbf{V'_p}$ share more than one parent in MM. The following results hold:

Theorem 2 *Let* $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E_\epsilon}, \mathbf{G_S})$ *be the purifiable linear latent variable graph that faithfully generates the covariance matrix* Σ *of a set of observed random variables* $\mathbf{O}$. *Let* MM_G *be the measurement pattern corresponding to the equivalence class* $MM(\mathbf{O}, \Sigma)$. *Let* $\mathbf{MM_{Pure}}$ *be the set of all purifications of* MM_G. *Then* $\mathbf{MM_{Pure}} =_{MM} \mathbf{G_S}$.

Corollary 1 *For every possible pair of purifiable linear latent variable graphs* $G_1(\mathbf{L_1}, \mathbf{O}, \epsilon_1, \mathbf{E_{L_1}}, \mathbf{E_{O_1}}, \mathbf{E_{\epsilon_1}}, \mathbf{G_{S_1}})$ *and* $G_2(\mathbf{L_2}, \mathbf{O}, \epsilon_2, \mathbf{E_{L_2}}, \mathbf{E_{O_2}}, \mathbf{E_{\epsilon_2}}, \mathbf{G_{S_2}})$ *faithfully generating* Σ, *the covariance matrix of* $\mathbf{O}$, *we have* $\mathbf{G_{S_1}} =_{MM} \mathbf{G_{S_2}}$.

6 Statistical tests and practical implementations

Even though the correctness of `FindMeasurementPattern` and `Purify` are guaranteed given the true covariance matrix, in real applications we cannot know this matrix, and a sample covariance matrix has to be used. In order to be able to deal with finite samples, we need statistical tests of tetrad constraints, partial correlations and marginal independencies. Also, one has to consider the computational cost of such algorithms: in the worst case, they are exponential in the number of impurities. In the following section we discuss the importance and reliability of computational and statistical features of our proposed method.

6.1 Statistical robustness

When the distribution of our variables is Gaussian, there are well-known tests of marginal independence and partial correlations. Spirtes et al. (2000) use a normal approximation for each sample tetrad difference $r_{IJ}r_{KL} - r_{IL}r_{JK}$, where r_{XY} is the sample correlation coefficient of X and Y. Mean and variances for such statistics are described in Wishart (1928).

For non-Gaussian distributions, Bollen (1990) describes an asymptotically distribution-free test of vanishing tetrads and a similar method can be used to create distribution-free tests for partial correlations. The computational cost of these tests may slow down the procedure considerably.

In pratice, the measurement pattern can have lots of errors, but still induce a correct purified solution. In our simulated studies reported in Section 8, it was common that the measurement patterns contained many more clusters than the true graph, but the purified solution was very close to the optimal one. Given the difficult nature of our statistical problem, this is actually a good result. Even with measurement patterns widely more complicated than the one that would be obtained given the true covariance matrix, one can interpret this behavior is a way to achieve robust response when learning pure measurement patterns: there is a lot of redundancy in the output of the `FindMeasurementPattern` algorithm, where many of the cliques among the latents

```
function Purify
inputs   $Pattern$, a measurement pattern
         $\Sigma$, a covariance matrix
output   a purified measurement pattern

Let $K$ be the size of the largest clique among latents in $Pattern$
for all sets $\mathbf{S}$ of latents in $Pattern$ that are cliques of size $K$
    Let $InducedG$ be the graph formed by $\mathbf{S}$ and all children of
      each latent in $\mathbf{S}$ that are in $Pattern$, including edges among children
    Remove from $InducedG$ all nodes with more than one latent parent
    For any pair from different clusters in $InducedG$ that share a same
    parent in $Pattern$, add an undirected edge in $InducedG$
    Let Solutions be the set of all subgraphs of $InducedG$ induced by
    removing at least one indicator from all pairs of indicators that are
    connected by an edge and where each latent keeps at least three children
    if Solutions $\neq \emptyset$
      return some random element in Solutions
end for
```

Table 4: The purification algorithm for choosing a subgraph of the measurement pattern where each cluster has at least three elements and no impurities.

have big overlaps. If we find a valid solution among the largests of the many cliques of latents that appear in the noisy measurement pattern, it is very likely it is a good solution, since it was the largest subgraph that endured a battery of many tests.

To summarize, we do recommend trusting the measurement pattern, but *only after a pure solution is found.* After the `Purify` algorithm decides which set of latents can be kept together, one can return to the measurement pattern and add back those indicators that were eliminated during purification and ignore all latents that were not present in the final estimated pure graph. This will give a better idea of the overall picture. There is no guarantee whatsoever that all indicators will appear in the final solution. By the definition of measurement pattern, this may not happen even given the true covariance matrix.

Heuristic statistical tricks are necessary for a better performance. For example, what should be done when the number of tetrad constraints in a given test is 2? This is a logical impossibility. In our implementation (used in the experiments), the solution is parametrically estimating the corresponding one-factor model, and testing for its statistical significance using a simple test such as χ^2. If the model passes the test, we return 3 as the count of valid tetrad constraints. Otherwise, we return 1. For the initial loop for identification of gray edges (Step 4 of `FindMeasurementPattern`), one could decide if all three tetrad constraints hold by also testing one-factor models. For the test of the *Unclustered* predicate (in Step 5), we could test if the one factor model does not hold, and if the pure two-factor model holds. In practice, this may require assuming a parametric form for the joint distribution of the variables. There are also tests of some joint tetrad constraints that could be implemented, as described in Bollen and Ting (1993).

Finally, it is very common that we miss some *Gray* edges among nodes due to the large number of tests (eventually, some one-factor model containing two impure nodes will hold). A suggested heuristic is as follows: for each clique of latents that is evaluated in `Purify`, and for each pair of indicators of a same latent T_1 that are not linked by an estimated impure edge, we look for a third indicator of the same group, plus a fourth indicator of another latent T_2, and test if the one-factor model holds or not. If there is no third indicator in T_1 and fourth indicator in T_2 that can produce such model, we add the *Gray* edge. Our heuristic for choosing T_2 is choosing the latent of the largest cluster that is not T_1. If the computational cost is not too high, one could make this verification for every pair of clusters that is compared in `BuildFinalPattern`.

For the `Purify` procedure, there is always the risk that none of the largest cliques will induce a pure graph. The solution is testing all cliques, starting from the largest to the smallest. In our implementation, we also substitute every clique of size K that fails to produce a result by $K - 1$ cliques where each new clique is a subgraph of the original one (and put them after the other cliques of size K in the queue of cliques to be tested). This can lead to an exponential increase of the number of possible cliques, so some bound should be imposed in the number of expansions a clique can produce in case its subgraphs still fail to produce a valid result.

6.2 Computational cost

Unless we find a polynomial bound of the number of maximal cliques of **Components** (Step 8), the very worst-case performance of `FindMeasurementPattern` is exponential in the number of impurity relations (which will be $O(2^{n^2})$, n being the number of observed variables, in a loose upper bound). Finding a polynomial bound for the number of maximal cliques, or proving it does not exist, is an interesting open problem. Even if we assume such bound does not exist, in practice it is hard to quantify how exactly it will scale in real-world problems, although we conjecture it will not be an important issue in many practical situations. Worst-case exponential algorithms are the rule, not the exception, in Bayesian network literature. Algorithms such as the PC algorithm (Spirtes et al., 2000) and GES (Chickering, 2002) are correct solutions for finding DAGs that in the worst case will take an exponential number of steps, but reported experiments with real data usually converged to solutions in reasonable time. In our case, for instance, if the true latent variable graph does not have any impurities, the actual execution time `FindMeasurementPattern` will be polynomial in the number of observed variables. Also, it is still not totally clear which kind of impurities will lead to extra computational burden: if all impurities are within each true cluster, our algorithm for learning measurement patterns will also take polynomial time (since the number of maximal cliques will be equal to the number of latents); if the true graph is identifiable enough to allow us to verify, for every pair of variables in different true clusters, if they actually belong to different estimated clusters, and no indicator is a child of more than one latent, the algorithm will also take polynomial time.

Of course, for these later remarks we assumed that no statistical errors will be commited while learning the pattern. Such idealized sceneario will never happen. Learning from small samples, even if generated by true graphs that are pure from the beginning, may generate largely complicated measurement patterns, full of latent cliques that will not have any pure solution. We actually observed that happening a couple of times in our experiments with true pure graphs reported in Section 8. Currently, our solution is halting the purification after 100,000 combinations were tried without success. An expansion of our work is coming up with useful approximation algorithms, but we still believe that in practice even the "exact" solution of trying all combinations till a solution is found (or halting after a number of steps and moving for the next latent clique found in the measurement pattern) might work well in a variety of domains.

One has to consider also the cost of regular steps before finding cliques. Step 5 can take up to $O(n^6)$ steps, where n is the number of observed variables. However, in practice it may be much faster and not come close to require a full exploration of all pairs of triplets. The loop at Step 5 can be reasonably fast when the graph does not have many impurities due to agressive elimination of edges in NG. Also, many problems of interest usually do not require more than a few hundred variables and many others just a few dozens, which is quite feasible for this algorithm. We are still studying different ways to achieve the same functionality of the *Unclustered* test, i.e., approaches for detecting when two variables cannot share a same latent parent.

In our implementation, we also save steps by testing *Unclustered* throught nested loops where we do not proceed past the fourth loop for a given variable if the current four elements do not form a one-factor model. Another practical step that can be implemented is storing the information that O_1 and O_2 were in a valid *Unclustered* test, when we were focusing in O_x, O_y. It may be the case that O_1 and O_2 are from the same cluster, and so we will have to enter the second part of Step 5, which will also require $O(n^6)$ steps. But if this information was obtained for free from another test, we will not need to repeat it here. A surpringly large number of steps can be saved with this implementation trick.

The reason why in the formal definition of our algorithm we used a *Unclustered* test of two triplets is because of problems introduced by indicators of multiple latents. Suppose $\{O_1, O_2, O_3\}$ are indicators of a single latent L_1 in the true graph, O_4 is an indicator of L_1 and L_2 and O_5 is an indicator of L_2, and there are no edges between indicators in the true graph. Then the predicate $Unclustered(\{O_1, O_2, O_3\}, \{O_4, O_5\}; \Sigma)$ will hold (the same *Unclustered* as defined in Table 2, but without the tests that are now undefined, eg., choosing one element from the first set against three from the second). Notice that, while it is true that O_4 does not have the same parents as O_1, for instance, ideally we would like not to remove the edge between O_1 and O_4 in the NG graph (because they do share *some* parent. It is just that O_4 has another one). For purification purposes, we have to add an indication that O_4 cannot appear in a final solution, but leaving the algorithm "as is" will not guarantee that (since it may be possible that, according to the order of tests, there will be no gray edge between O_1 and O_4). We may compensate for that by checking again, for every pair (O_x, O_y), O_x indicator of T_i, O_y indicator of T_j, if there is some set $\{O_a, O_b, O_c, O_d\}$, $\{O_a, O_b\}$ indicators of T_i, $\{O_c, O_d\}$ indicators of T_j such that $Unclustered(\{O_x, O_a, O_b\}, \{O_y, O_c, O_d\})$ holds. Some special new steps may be required when trying to get the reconstruct the measurement pattern after we get the purified graph, if one wants to determine which indicators are children of multiple latents.

Finally, the initialization of the algorithm (Step 4) requires the evaluation of a considerably large number of subsets of size four. In the worst case, this first loop will take $O(n^4)$ steps. On average, we expect it to take much less than that, but certainly not less than $O(n^2)$.

With all such issues of worst-case complexity and steps that can get up to $O(n^6)$ in complexity, one can wonder why not use the following trivial algorithm to get pure models:

1. Let **Triplets** be the set of all subsets of size three from $\mathbf{O}$;
2. Let NG be the graph where each node N_{xyz} represents a triplet with variables $\{X, Y, Z\}$. Initialize NG such that it contains no edges;
3. For each pair of nodes (N_{xyz}, N_{abc}), $\{X, Y, Z\} \cap \{A, B, C\} = \emptyset$ such that $Unclustered(\{X, Y, Z\}, \{A, B, C\}; \Sigma)$ holds, add the edge (N_{xyz}, N_{abc}) to NG;
4. Return the largest clique in NG

Depending on how large is the number of pure graphs induced by our latent graph with respect to the number of variables, the real computational cost of our algorithm for a given problem can be much less than the trivial solution. With the trivial algorithm, we will need to evaluate all non-overlapping triplet comparisons, which only happens as a (presumably unlikely) worst-case scenario in `FindMeasurementPattern`. Also, the graph where we look for cliques is expected to contain many more edges than the one defined in our algorithm. These are some of the largest computational advantages of using `FindMeasurementPattern`, besides the fact it can provide information about the true model without requiring purification, or by putting back (impure) indicators into a pure model according to the information in the pattern.

6.3 Metaclustering

A side effect of statistical mistakes is not only introducing errors in the final output, but the fact that the order by which such tests are applied may affect the outcome. For instance, at Step 5 of `FindMeasurementPattern`, we only consider triplets that are cliques in NG_{Blue}, because we know that given the true covariance matrix, one does not need to spend time considering elements not connected by *Blue* edges: they would fail the *Unclustered* test anyway.

However, it may be the case that one *Blue* edge connecting a pair of vertices $\{A, B\}$ was removed due to a statistical mistake at some point of the execution, and the vertices of that edge were the only ones that could be able to make a *Unclustered* test pass for another pair of vertices $\{C, D\}$. This is a case where errors propagate. If we had tested the *Unclustered* predicate for C and D before eliminating the edge between A and B, this second mistake would not happen. Trying to avoid such a problem by considering all pairs of triplets, instead only those that are cliques in NG_{Blue}, can cause an enormous increase in the number of tests: actually, the main reason why our algorithm may be computationally feasible even if one has many observed variables is due mostly to the fact that we discard many tests based on the results of other tests. Also, it is not clear why doing all tests would be any better, since we can error now on the other side: by coincidence, we may remove more edges erroneously by chance because we made many more tests.

Besides statistical mistakes, another source of sensibility to the order of tests in our algorithm is failure of assumptions. While the measurement assumption and linearity of measurement are quite reasonable for many studies, the assumption of having a purification that includes all latents with at least three children per latent is much stronger. It may be the case that you can find purifications with three indicators per latent for only a subset of the latents at a time, but losing some of the guarantees of measurement patterns as described in Section 3.

Figure 7 illustrates a case where we can have pure graphs with three children per latent created from the latent variable graph $G(\{L_1, L_2, L_3\}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E}_\epsilon)$, but where such pure graphs would include only two latents at a time. A common mistake is thinking that we can actually redefine the problem as learning the pattern of, for instance, $G'(\{L_1, L_2\}, \mathbf{O}, \epsilon', \mathbf{E_L}', \mathbf{E_O}', \mathbf{E}_\epsilon')$ and use the same algorithm while expecting the same guarantees. But G' is not a latent variable graph: there are elements in $\mathbf{O}$ that are not children of L_1 or L_2. Still, we can learn valuable information if the true graph is the one depicted in Figure 7(a): Figures 7 (b)–(f) show the progress of our algorithm given the covariance matrix over variables $\{1, 2, ..., 9\}$. Notice that we are still able to find a pure measurement model that is a subgraph of the original one. For example, we are able to separate nodes $\{1, 2, 3\}$ from $\{7, 8, 9\}$, even if the *Unclustered* test does not hold for any combination including at least one element from each set[5]: the corresponding $Yellow$ edges that appear in Figure 7(c) are not carried to **Clustering**, as described by the algorithm in Table 1.

It remains an open question which is the full characterization of measurement equivalence classes under a weaker assumption that there are two indicators O_1 and O_2 such that no parents of O_1 can appear in a pure measurement pattern with any parents of O_2. We conjecture that each latent appearing in a puried outcome will have at least three children of some true latent and probably all of its children have a common cause in the true graph: they might not share a same parent, but a same ancestor, if linearity holds in every path from the common ancestor to each of such indicators. We also have to better define what a "true" latent is: any (possibly indirect) hidden common cause of at least three indicators? Should it be part of any pure measurement model with at least another latent?

In order to minimize the effect of such sources of errors, we recommend the following heuristic, which we denominate *metaclustering*: run the clustering/purification algorithm N times, using a random order of tests each time, and save the purified graph for each run. After that, merge all graphs as a single one (if two latents in a different pair of graphs has exactly the same children, treat them as a single node in the merged graph).

Initially, in the merged graph, latents that came from different purified patterns will not have an edge between them. For each pair of latents L_x and L_y from different sources and without an edge, verify if they satisfy the following properties: there is at least one subset $\mathbf{O_x}$ of the children of L_x of size three; there is at least one subset $\mathbf{O_y}$ of the children of L_y of size three; $\mathbf{O_x} \cap \mathbf{O_y} = \emptyset$; $Unclustered(\mathbf{O_x}, \mathbf{O_y}; \Sigma)$ holds.

If such properties hold, add an edge between such latents in the merged graph. Unsurprisingly, these are properties analogous to Property 4 of measurement patterns.

Finally, find the largest clique of latents in the merged graph that satisfy this property. If there is more than one and computational resources allow for it, one may want also find a clique that

[5] We are assuming that the structural relationships among the latents are non-linear. If they were all linear, the predicate $Unclustered(\{1, 4, 5\}, \{7, 8, 9\}; \Sigma)$ would hold, for example. In this case, we would not have any $Yellow$ edge, but the final measurement pattern would be the same.

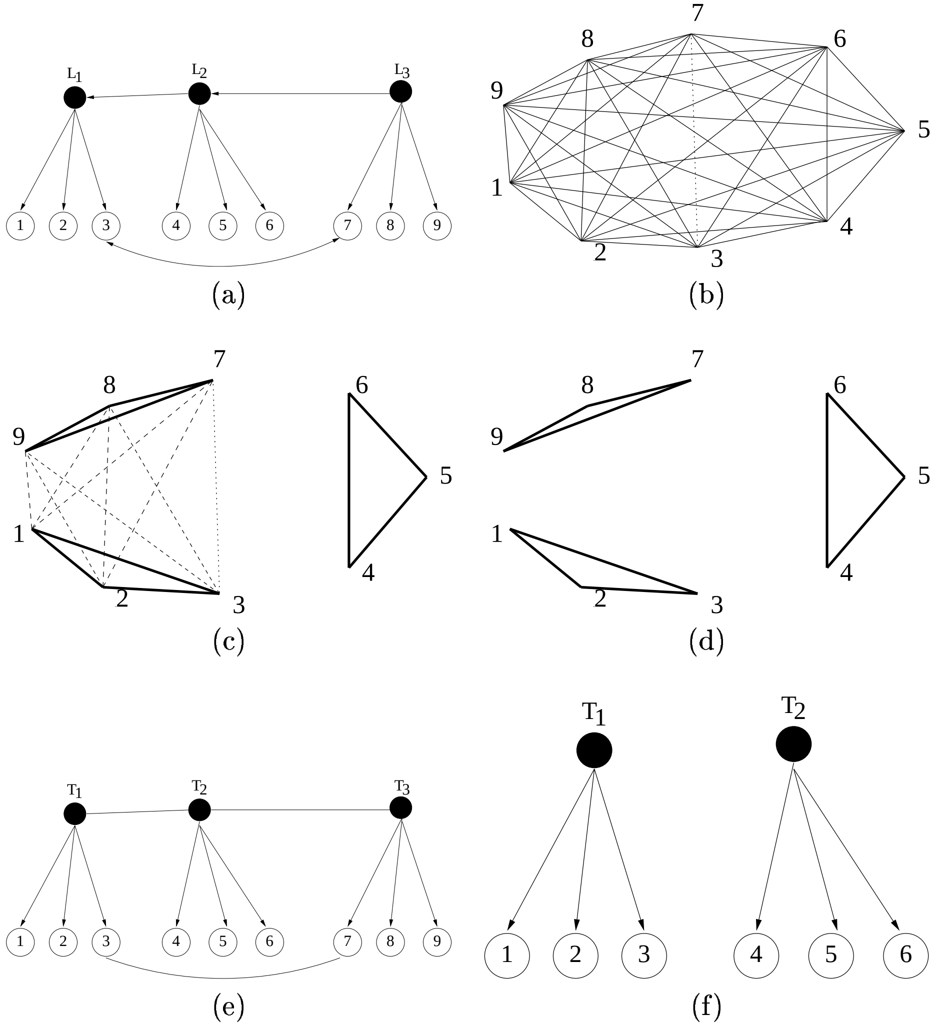


Figure 7: (a) A true graph that fails to meet the assumptions of three pure children per latent. (b) The NG graph at the end of Step 4. Notice the dotted edge (representing a *Gray* color) between nodes 3 and 7. (c) The NG graph at the end of Step 5 of `FindMeasurementPattern`. (d) The *Components* graph. (e) The final measurement pattern. Notice the lack of edge between T_1 and T_3. This will have an implication in the purification. (f) An example of measurement model obtained throught `Purify`. Notice it could have been the combination of T_2 and T_3, depending on the order of the tests.

maximizes the number of indicators in the final puried measurement model. We believe that this heuristic will be reasonably robust and useful, but we cannot provide a principled way to choose the number N of iterations to be applied. One heuristic is verifying how many "new latents" are discovered at every iteration, where a latent counts as new if it has a set of children that is not contained in the set of children of any other latent discovered in previous iterations, and stop when this number approaches zero.

7 Related work

In this section, we briefly describe standard techniques for representing measurement models, and which shortcomings they may have.

7.1 Factor analysis

Factor analysis (FA) embodies the standard set of tools used for modeling and testing measurement models. The usual structural assumption is that the each observed variable is a linear combination of hidden variables (factors), plus an additive error term. Error variables are mutually independent and independent of latent factors. Latent variables are mutually independent in principle.

When estimating such parameters by maximum likelihood, one usually assumes that latents and error variables are multivariate normal, which implies a multivariate normal distribution among the observed variables. Since then, a variety of methodologies were created in order to generalize standard FA to the case where latents are not necessarily assumed to be Gaussian. For instance, independent component analysis (ICA) is a family of tools motivated by blind source separation problems where estimation requires assuming that latents are *not* Gaussian, and instead some measure of independence is maximized without making use of strong assumptions about the marginal distribution of each latent. For instance, Attias (1999) assumes that each latent is distributed accordingly to a semiparametric family of mixture of Gaussians. Still, at its heart ICA relies heavily in factor analysis fundamental idea of interpreting observed variables as joint measurements of a set of independent latents. For problems such as blind source separation such assumptions may be reasonable in some cases, but for large families of different applications this is certainly not the case. Other variations of factor analysis, while useful for dimensionality reduction (Minka, 2000; Bishop, 1998) and data interpretation by visualization, are still limited for more complete insights and again make the assumption of independence among latents.

For instance, Bartholomew et al. (2002) describe a series of applications of factor analysis in social sciences. For confirmatory factor analysis, one is usually interested in verifying some a priori hypothesis about the common causes of the observed variables and quantifying the strength of the causal effect of each latent on each indicator obtained my methods such as maximum likelihood estimation (Bollen, 1989). In real world applications, it is common practice to ignore loadings with absolute values smaller than some threshold.

In exploratory factor analysis, the main goal is discovering the existence of abstract concepts such as political attitude and socio-economic status as common causes of the indicators. One has to choose the number of factors and observe which ones correlate more with which indicators.

That also leads to the question of how to choose the number of latents. One standard approach is testing models with an increasing number of latents till one fits the data at a given significance level. Bartholomew et al. (2002) claim this is not exactly a good practice because it overstimates the number of latents, and advocate using criteria from principal component analysis for choosing the number of factors. These authors also explicitly claim they are not too much concerned if the model gives a bad fit as long as it provides an intuitive insight of the relationships among observed variables.

In general, any domain to which one is interested in applying data mining techniques may benefit from such exploratory procedure, but a more serious shortcoming of standard factor analysis is the non-identifiability of such models: there is an infinite number of loading matrices that give exactly the same fit. On the other hand, unlike ICA and variants, there are straightforward heuristics able to relax the assumption of independence in standard FA.

In order to introduce an more flexible interpretation criterion, there is a large set of approaches to rotate the factor matrix. Actually, a common complain about the (lack of) objectivity of factor analysis is that there are too many of such rotation methods. The general idea is it should maximize the variance of the loadings (usually the variance of the squared loadings to avoid taking signal differences into account). The idea is unveiling something close to a so-called "simple structure" (Harman, 1967): a structure where each indicator loads only in one of the latents, i.e., each indicator has only one latent cause. That is similar to trying to discover pure measurement models. It seems intuitive that if the true model is a pure one, such rotation criteria should work very well given the number of latents is chosen correctly. In an oblique rotation, factors cease to be independent. Bartholomew et al. (2002) point out that oblique rotation usually gives simpler models, and this can be a good argument against the assumption of independence. It seems there is no reason at all to make the assumption of independence among the latents, unless there is very strong belief on it.

It is counterintuitive that one starts fitting a model where latents are independent and then in the same analysis an oblique rotation is introduced. The usual argument is that both models give exactly the same fit. But if one takes the point of view of defining learning as searching through a space of models (Mitchell, 1997), then starting from the assumption of independence and later throwing away this very idea is not the same of starting with the possibility of dependence from the very beginning. No surprise that rotation itself is ill-defined, allowing the existance of different criteria.

7.2 Feature construction

If the goal is to cluster indicators as measurements of latent variables and use this information to find causal relationships among the hidden variables, then one approach that sounds appealing in principle is finding such clustering and creating a new feature for each group of indicators. Each feature is basically a function, eg., the average value of the corresponding indicators. In order to carry on this approach, one should find a principled way to choose among the available indicators those that are considered to be indicators of a single latent. In many cases, the choice of indicators is done before data collection, eg., when designing questionnaries for social studies or marketing research. Each group should form a one-factor model, or *construct*. The term *scale* is commonly used for a feature build upon the observed variables of a single latent variable (Carmines and Zeller, 1979).

In principle, factor analysis can be used in the design of constructs. Actually, one the the fundamental ideas used to motivate factor analysis is that a group of random variables can be clustered accordingly to the strength of their correlations. As put by a traditional textbook in multivariate analysis (Johnson and Wichern, 1998, p. 514):

> Basically, the factor model is motivated by the following argument: suppose variables can be grouped by their correlations. That is, suppose all variables within a particular group are highly correlated among themselves, but have relatively small correlations with variables in a different group. Then it is conceivable that each group of variables represents a single underlying construct, or factor, that is responsible for the observed correlations.

Also, Harman (1967) suggests this criterion as an heuristic for clustering variables, achieving a model closer to a "simple structure". We argue that the assumption that the simple structure can be obtained by such criterion is unnecessary. Actually, there is no reason why it should hold even in a linear model. As an alternative, one could try to find a pure (sub)model, which usually requires throwing away some indicators to get only pure ones. The groups can then be treated as individual constructs.

Many construct design techniques start with background theories for selecting the initial set of indicators to be tested as valid measures of an abstract concept chosen a priori. This is mainly a confirmatory analysis process, where statistical and theoretical tools here aim at achieving validity and reliability assessment. A construct is valid if it actually measures the desired concept, and it is reliable if, for any given value of the latent variable, the conditional variance of the elements in the construct is not too high. Since these criteria rely on unobservable quantities, they are not easy to evaluate.

The methods for evaluating constructs are not approaches for clustering individual indicators in a set of a unknown number of clusters but basically score functions for quantifying the fitness of a one-factor model. The procedure itself does make any use of constraints in the observed joint distribution to decide if one indicator should be grouped with a specific construct or another.

Against the less theoretically-driven exploratory factor analysis approach, Carmines and Zeller (1979) argue that in general it is difficult for factor analysis to distinguish a model with few factors against an one-factor model. The argument is that factor analysis may identify a systematic error variance component as an extra factor. On an example about indicators of self-steem, they write (p. 67):

> In summary, the factor analysis summary of scale data does not provide unambiguous, and even less unimpeachable, evidence of the theoretical dimensionality underlying these self-steem items. On the contrary, since the bifactorial structure can be a function of a single theoretical dimension which is contaminated by a method artifact as well as being indicative of two separate, substantive dimensions, the factor analysis leaves the theoretical structure of self-steem indeterminate.

Clearly, the criticism is on determining the number of factors based merely in a criterion of statistical fitness. Again, as already discussed in the previous section, statistical fitness does not seem to be a strong enough criterion in such applications (Bartholomew et al., 2002). In the self-steem problem, the proposed solution was relying on an extra set of "theoretically relevant external variables", other observed variables that are, by domain-knowledge assumptions, related to the concept of self-steem. First, a scale was formed for each of the two latents in the factor analysis solution. Then, for each external variable, the correlation with both scales was computed. Since the pattern of correlations for the two scales was very similar, and there was no statistically significant difference between the correlations for any external variable comparison, the final conclusion was that the indicators were actually measuring a single abstract factor.

The problem with this approach is relying on strong background knowledge and the lack of a more theoretical, domain-independent, justification for the procedure. Also, it is not clear what should be done if the pattern of correlations does not match. Still, it uses the idea of determining clustering by contrasting sets of indicators (summarized by a scale) with indicators from another factor. Our proposed method of discoverying measurement models tries to overcome such issues by general, weaker assumptions about the structure of the unknown true model, and then obtaining clustering by entailment.

7.3 Graphical models

Graphical models became a representation of choice for computer science and artificial intelligence applications for systems operating under conditions of uncertainty, such as in probabilistic expert systems (Pearl, 1988). Bayesian networks and belief networks are the common denominations under such contexts. They have been used also for decades in econometrics and social sciences (Bollen, 1989), usually to represent linear relations with additive errors. Such models are then denominated structural equation models (SEMs).

The very idea of using graphical models is to be able to express qualitative information that is difficult or impossible to express with probability distributions only. For instance, the consequences of conditional independence conditions can be carried on with much less effort under the language of graphs than under the probability calculus. It becomes easier to add prior knowledge, as well as using the machinery of graph theory to develop exact and approximate inference algorithms. However, perhaps the greatest gain in expressive power is allowing the expression of causal relations, which seems impossible to achieve (at least in a more general sense) by means of probability calculus only (Spirtes et al., 2000; Pearl, 2000).

Many standard models can be recast in graphical representations (e.g., factor analysis as a graph where latents are not connected by any path). Under the graphical modeling literature, there are several approaches for dealing with latent variables. Many of them are techniques for fitting parameters giving the structure (Binder et al., 1997; Bollen, 1989) or choosing the number of latents for a factor analysis model (eg., Minka, 2000).

Elidan et al. (2000) introduce some heuristics for discovering latent variables. But such heuristics have as their sole goal reducing the number of parameters in a Bayesian network for more robust learning, achieving a better estimation of the observed joint probability distribution. They do not provide any formal interpretation of what the resulting structure actually is. For causality discovery and data mining one has to adopt an approach oriented to understanding the structure.

For instance, by assuming a discrete distribution of latent variables and observed measurements in a hidden Markov model (HMM), Beal et al. (2002) present algorithms for learning the transition and emission probabilities with very good empirical results. The only assumptions about the structure of the true graph is that it is a hidden Markov model, but no a priori information on the number of latents or which observed variables are indicators of which latents is necessary. No tests of significance for the parameters are discussed, since that was not the point of the paper. However, if one wants to have qualitative information of independence (as necessary in our axiomatic causality calculus), such analysis has to be carried on. We are also interested in continuous distributions, since in such work HMMs are discrete.

While factor analysis and feature construction techniques are not concerned about discovering relations among latents, there is some work done in graphical models research. A recurring debate is structural equation modeling literature is whether one should learn models from data by first finding the measurement model and then proceeding to the structural model, or if both should be analyzed at the same time (Fornell and Yi, 1992; Hayduk and Glaser, 2000; Bollen, 2000). It should be clear that our methodology strongly supports a two-step procedure. A good deal of criticism on two-or-more-steps approaches concerns in choosing a initial number of factors using methods that suffer from non-identifiability, such as factor analysis. One way to overcome this problem is by explicitly defining classes of models that empirically undistinguishable, a methodology that is a key component in Spirtes et al. (2000).

Concerning estimation methods, Bayesian approaches for learning graphical models are well established (Heckerman et al., 1999), but currently there is no known consistent score function for such methods when considering general latent variable models (Geiger et al., 2001), unless one is willing to compute the full posterior by numerical methods. On the practical side, such methods can be slow, since there no known method of decomposing general latent models into local scores, as it is can be done for the latent-free case (Chickering, 2002). If one wants to learn the causal relationships among latents, another approach, described and evaluated in (Silva, 2002a), consists in comparing nested models, one with a direct edge between two latent variables of interest, where each latent has at least two pure measures. The problem is that in many times we do not have this pure measurement model: it would be interesting to learn it automatically from data given only a sample of the observed variables and no assumptions about the number of latents, or how they are causally related.

In a previous report (Silva, 2002a), we introduced the `Washdown` algorithm as an approach to find a correct pure measurement model for some unknown linear measurement/structural model, and a proof of correctness was provided in (Silva, 2002b). This report discusses a variation of the problem with weaker assumptions and a totally different approach. While in the original method we required that each latent had at least three indicators that were pure *with respect to every other indicator* in the model, here we require that exists a subgraph of the unknown complete graph where each latent has at least three indicators that are pure with respect only to those indicators in that subgraph, a much weaker assumption. A theoretical advantage is not requiring knowing the family of the joint probability distribution of the latents, or even if they are linearly related. In practice, one may still need to use parametric tests if the sample sizes are not large.

8 Empirical results

Evaluating automated knowledge discovery algorithms is often a difficult task because of the lack of a readily available gold standard by which comparisons could be made. This is especially true for unsupervised learning techniques such as clustering and causality discovery. In this section, we will take two approaches: a comparison of our output to random clustering and factor analysis in a real-world example where domain knowledge is not a crucial requirement for interpreting results; comparisons with simulated data from models where we know the true underlying structure, and therefore we can come up with objective measures of success.

8.1 Test anxiety data

We will take a real-world example from Bartholomew et al. (2002). A survey of test anxiety indicators were collected among 335 grade school male students in British Columbia. The survey consisted in 20 measures on how frequent were determined symptoms of anxiety under test conditions. A brief description of the 20 indicators is shown in Table 5. The covariance matrix of such variables is given in Bartholomew et al. (2002), p. 163.

One type of analysis that can be done with such data is inferring which are the common causes that explain the correlation of the given variables. It could be used, for example, to design policies aiming at reducing anxiety (although this example is too simple to surprise the analysist with unexpected facts). As it can be noticed from reading the description of the variables, such indicators are highly correlated[6]. It is not immediately obvious which, and even how many, latent factors are there. Many different ways of clustering may sound plausible at first sight.

In order to evaluate our approach, we will perform a simple psychological test. Five candidate models are given below:

1. $\{x_6, x_9, x_{13}, x_{20}\}, \{x_2, x_5, x_7, x_{18}\}, \{x_4, x_{14}, x_{15}, x_{17}\}$

2. $\{x_5, x_6, x_7, x_{14}, x_{17}\}, \{x_2, x_9, x_{13}, x_{18}\}, \{x_4, x_{15}, x_{20}\}$

[6] The sample correlation matrix of such items has only positive entries, most of them between 0.30 and 0.50.

1. Lack of confidence during tests
2. Uneasy, upset feeling
3. Thinking about grades
4. Freeze up
5. Thinking about getting through school
6. The harder I work, the more confused I get
7. Thought interfere with concentration
8. Jittery when taking tests
9. Even when prepared, get nervous
10. Uneasy before getting the test back
11. Tense during test
12. Exams bother me
13. Tense/stomach upset
14. Defeat myself during tests
15. Panicky during tests
16. Worry before important tests
17. Think about failing
18. Heart beating fast during tests
19. Can't stop worrying
20. Nervous during test, forget facts

Table 5: Indicators of test anxiety described in Bartholomew et al. (2002).

3. $\{x_2, x_4, x_5, x_{13}\}, \{x_9, x_{14}, x_{15}, x_{18}\}, \{x_6, x_{17}, x_{20}\}$
4. $\{x_{13}, x_{17}, x_{18}, x_{20}\}, \{x_5, x_6, x_9, x_{15}\}, \{x_2, x_4, x_{17}, x_{14}\}$
5. $\{x_2, x_{15}, x_{17}, x_{20}\}, \{x_4, x_5, x_6, x_7, x_9\}, \{x_{13}, x_{14}, x_{18}\}$

where symbol x_i represents the ith entry in Table 5 and each set between brackets is a different cluster. Four of such clusters where generated randomly. One of them is the output of our algorithm. To make things less arbitrary, the random clusters are not completely random: they are partial rearrangements of our algorithm output, keeping the same number of clusters. The reader is invited to pick the one he/she thinks it is the most insightful before moving to Section 8.3, where our analysis is discussed along with a comparison with the factor analysis solution proposed by Bartholomew et al.

8.2 Simulated data

The data sets we used in this section are synthetic data sets. The importance of synthetic data is the fact that we know which is each true model that generated the given samples, and therefore we can calculate precisely some measures of distance from our induced models to the true structure. To simplify our task, we will evaluate the following features for each *pure* model we get with respect to a purified true graph:

- **proportion of missing latents**, the number of latents in the true graph that do not appear in the estimated pure graph, divided by the number of latents in the true graph;
- **proportion of missing measurements**, the number of indicators in the true purified graph that do not appear in the estimated pure graph, divided by the number of indicators in the true purified graph;
- **proportion of misplaced measurements**, the number of indicators in the estimated pure graph that end up in the the wrong cluster, divided by the number of indicators in the estimated pure graph;
- **proportion of impurities**, the number of impurities in the estimated pure graph divided by the number of impurities in the true (non-purified) graph. Notice that a node that is impure in the measurement pattern may not be impure with respect to the other nodes in the purified estimated graph. In this case, we do not count them. For each pair of nodes that forms a localized impurity (e.g., indicators with correlated errors, or an indicator that is a direct cause of another, while both are children of a same and single latent), we count this pair as one impurity, since removing one of them will eliminate that impurity. Each indicator that has more than one immediate latent ancestor (i.e., a latent ancestor with a directed path to that indicator that does not include any other element in the latent set) is counted as one impurity, since it has to be removed from all purified graphs.

To perform the comparison, we should indicate which latent found in the estimation corresponds to which of the original latents. The straightforward way is making the match according to the original parent of the majority of the indicators in a given estimated cluster: for example, suppose we have an estimated latent L_E. If, for instance, 70% of the measures in L_E are measures of the true latent L_2, we label L_E as L_2 in the estimated graph and calculate the statistics of comparison

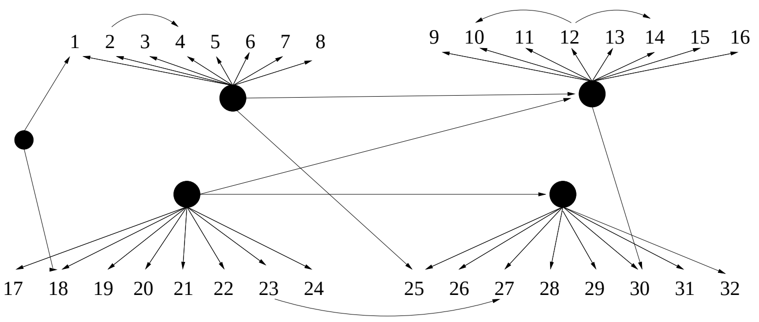


Figure 8: Example of an impure latent variable graph used in one of the experiments. Black nodes represent latent variables. The smaller latent node represents one of the error nodes, a common cause that should not appear in the purified measurement model. Notice that nodes 10 and 14 are also impure with respect to each other, since 12 is another common cause of them.

as described above. Some few ties happened in our experiments, but labeling the latent in one way or another did not change the final statistics.

In order to better compare with factor analysis, and also to provide an upper-bound of how good our results can be, for this subsection we generated only multivariate normal indicators, with a linear latent structure. We used the Wishart test of tetrad constraints (Spirtes et al., 2000; Wishart, 1928). Samples were generated using the Tetrad IV program [7]. Values for the cofficients are then uniformly sampled from the interval $[-1.5, -0.5] \cup [0.5, 1.5]$. Variances for the exogenous nodes (i.e., latents without parents and error nodes) are uniformly sampled from the interval $[1, 3]$. The motivation for choosing such intervals is generating artificial models where the causal effects are not too big or too small. After the full parameterized model is set, independent samples are pseudoramdonly generated. The pseudorandom number generator used in the following experiments was the one used in the Java 1.4 virtual machine. The α-value used in all tests in this section was 0.05.

The first batch of experiments concerns true models that are pure: for a given number m of latents, we add n pure indicators to each latent, where $m = 5, 10$ and $n = 3, 4, 5$. We used two different sample sizes: 1000 and 5000 observations. For a graph with m latents, we added edges among latents aiming at an average degree of h neighbors by latent in the following way: we iterate through all pairs of latent nodes and add a directed edge [8] if a random number generated uniformly in the interval $[0, 1]$ was less than $\frac{h}{m-1}$. For graphs with 5 latents, we had $h = 2$. For graphs with 10 latents, we had $h = 4$. Results are summarized in Table 6.

From the results on Table 6 it is clear that the number of indicators contributes more to the sucess of the algorithm than the sample size. With exact three indicators per latent, there is little margin for redundancy and any statistical mistake when evaluating a constraint may be enough to eliminate a whole cluster. There is a huge leap of quality when latents have four indicators: in this case, results are extremely good and adding more samples do not change it much. A similar pattern follows for the case with 5 and 10 latents, althought the case for 10 latents, 3 indicators per latent and 5000 examples deserves further study.

[7] Available at `http://www.phil.cmu.edu/tetrad`.

[8] We define an arbitrary order among the latents, such that an edge is directed only from the node in the lowest position in this order to the node if the highest position. This avoids circularity.

Evaluation of estimated purified models				
	5L/1000E	**5L/5000E**	**10L/1000E**	**10L/5000E**
3 indicators, pure				
missing latents	0.42 ± 0.15	0.28 ± 0.10	0.40 ± 0.08	0.45 ± 0.08
missing indicators	0.36 ± 0.16	0.26 ± 0.10	0.37 ± 0.09	0.43 ± 0.11
misplaced indicators	0.11 ± 0.12	0.03 ± 0.08	0.05 ± 0.08	0.03 ± 0.06
4 indicators, pure				
missing latents	0.0 ± 0.0	0.02 ± 0.06	0.07 ± 0.08	0.05 ± 0.07
missing indicators	0.08 ± 0.05	0.06 ± 0.07	0.11 ± 0.09	0.10 ± 0.06
misplaced indicators	0.0 ± 0.0	0.0 ± 0.0	0.02 ± 0.04	0.0 ± 0.0
5 indicators, pure				
missing latents	0.0 ± 0.0	0.02 ± 0.06	0.02 ± 0.04	0.0 ± 0.00
missing indicators	0.03 ± 0.03	0.06 ± 0.08	0.09 ± 0.07	0.06 ± 0.05
misplaced indicators	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
3 indicators + impurities				
missing latents	0.40 ± 0.13	0.34 ± 0.16	−−	−−
missing indicators	0.40 ± 0.15	0.37 ± 0.20	−−	−−
misplaced indicators	0.0 ± 0.0	0.01 ± 0.03	−−	−−
impurities	0.06 ± 0.08	0.03 ± 0.07	−−	−−
4 indicators + impurities				
missing latents	0.0 ± 0.0	0.04 ± 0.08	−−	−−
missing indicators	0.05 ± 0.08	0.14 ± 0.13	−−	−−
misplaced indicators	0.01 ± 0.01	0.0 ± 0.0	−−	−−
impurities	0.03 ± 0.09	0.0 ± 0.0	−−	−−
5 indicators + impurities				
missing latents	0.0 ± 0.0	0.0 ± 0.0	−−	−−
missing indicators	0.05 ± 0.04	0.03 ± 0.03	−−	−−
misplaced indicators	0.0 ± 0.0	0.0 ± 0.0	−−	−−
impurities	0.03 ± 0.09	0.0 ± 0.0	−−	−−

Table 6: Results obtained for estimated purified graphs. Each number is an average over 10 trials, with an indication of the standard deviation over these trials. The four columns represent the cases with 5 latents/1000 observations, 5 latents/5000 observations, 10 latents/1000 observations and 10 latents/5000 observations, respectively.

The second batch of experiments concerns impure models with embedded pure models taken from the previous case. In other words, from a pure graph with m latents and n indicators per latent, we create an impure graph by adding $2m$ more indicators, and making each latent a parent of these extra indicators. It barely affects the case for 4 and 5 latents, but it was somewhat worse for the case with three pure indicators per latent graphs.

The third experiment uses the graph in Figure 8. In this case, we have 7 impure nodes that have to be removed. The result for sample size of 1000, averaged over 10 trials, was: 0 missing latents, 0.056 (±0.047) missing indicators, 0 misplaced indicators, 0.24 (±0.17) impurities. The result for sample size of 5000, averaged over 10 trials, was: 0 missing latents, 0.032 (±0.065) missing indicators, 0 misplaced indicators, 0.2 (±0.1) impurities.

We used metaclustering in all experiments, merging clusters out of a cycle of 10 repetitions with a randomized order of tests. Metaclustering did not help in most of the situations, but in a few cases it was able to add one or two extra clusters. None of the solutions actually required merging the output of any of the last 5 runs for each metaclustering. For the metaclustering procedure, we adopted a slightly different method of verifying if two latents can appear together in the final model: by fitting a one factor model with six indicators, three from each latent, and verifying it is not significant. If we passed the first test (i.e., the one-factor model is not significant), we fit a two-factor model and verify if it is significant. If true, then these two latents can be kept together. The test used was a chi-square test of significance. In practice, we also verified if two indicators were impure with respect to each other by running the same tetrad tests again, but now using only indicators from each respective cluster (instead of all indicators). This double-check was useful to detect some extra impurities in the last experiment.

We also generated factor analysis models for each of the data sets used in these experiments. We used the PROC FACTOR procedure from SAS v.8e, and two criteria for choosing the number of latents: the default SAS criterion that basically chooses the number of latents by some thresholding on the amount of variance explained, and an iterative procedure that chooses the number of latents by the first statistically significant model when we start from 1 latent and increase the number by 1 at each iteration. A chi-square test was used[9]. In order to evaluate the final outcome, we first perform an oblique rotation (we used the oblimin rotation). We then heuristically cluster the indicators by associating each one with the latent with the respective highest loading (in absolute value). Finally, we just examine how the original pure indicators were clustered, and how it matches the purified true graphs.

The default criterion of choosing the number of latents widely underestimated the true number, in many times by just keeping half of it. The chi-square criterion worked extremely well. The combination of the chi-square criterion and the heuristic clustering criterion was close to perfection, achieving nearly zero error by all our evaluation measures (except impurity detection, which was not considered. Notice also that there is no problem with missing indicators, since nothing is discarded). For the last experiment, using samples from Figure 8, SAS failed to find a statistically significant model before having convergence problems with maximum likelihood estimation. We then used the default SAS criterion, which in this case did not understimate the minimum number of latents (4) necessary for perfect clustering.

The very good performance of factor analysis in these data sets was somewhat surprising, but again the heuristic methods of rotation and clustering do not provide any theoretical guarantee. It is not clear also how to distinguish what is a pure measure from those that are not, since it is not uncommon that pure indicators have large loadings (> 0.20) in more than one latent.

[9] We compared this criterion against searching with the BIC score and stopping when the score decreases, and the results were virtually the same.

1.0	0.707	0.701	0.988	0.888	0.808	0.807	0.888	0.868	0.85	0.848	0.868
0.707	1.0	0.487	0.705	0.64	0.58	0.585	0.639	0.626	0.616	0.61	0.626
0.701	0.487	1.0	0.7	0.625	0.571	0.56	0.624	0.608	0.595	0.592	0.607
0.988	0.705	0.7	1.0	0.887	0.808	0.806	0.888	0.868	0.85	0.848	0.868
0.888	0.64	0.625	0.887	1.0	0.907	0.909	0.997	0.974	0.952	0.952	0.974
0.808	0.58	0.571	0.808	0.907	1.0	0.824	0.907	0.887	0.866	0.866	0.887
0.807	0.585	0.56	0.806	0.909	0.824	1.0	0.909	0.888	0.869	0.867	0.888
0.888	0.639	0.624	0.888	0.997	0.907	0.909	1.0	0.974	0.952	0.952	0.974
0.868	0.626	0.608	0.868	0.974	0.887	0.888	0.974	1.0	0.977	0.976	0.999
0.85	0.616	0.595	0.85	0.952	0.866	0.869	0.952	0.977	1.0	0.955	0.977
0.848	0.61	0.592	0.848	0.952	0.866	0.867	0.952	0.976	0.955	1.0	0.976
0.868	0.626	0.607	0.868	0.974	0.887	0.888	0.974	0.999	0.977	0.976	1.0

Table 7: Correlation matrix for a case where factor analysis clustering heuristics fail.

8.2.1 Tricking factor analysis

In some sense, the observed agreement with factor analysis should be interpreted as an indication of soundness of our approach, althought we would be more satisfied by showing examples where factor analysis heuristics for clustering fail.

One of the motivations for using such heuristics is suggesting that elements in a same cluster are more strongly correlated than those in different clusters, as quoted in Section 7.2. In this section, we will artificially generate a graph where this assumption does not hold and evaluate the behavior of factor analysis clustering under these conditions.

The graph used in this experiment has three latents forming a directed chain (i.e., $L_1 \rightarrow L_2 \rightarrow L_3$). The structural equations for L_2 and L_3 are $L_2 = 2L_1 + \epsilon_{L_2}$, $L_3 = 2L_2 + \epsilon_{L_3}$, where L_1, ϵ_{L_2} and ϵ_{L_3} are independent standard normal variables. Each latent has four pure indicators. The first and fourth indicators of each latent have a loading of 9. The second and third have a loading of 1. This means, for example, that the first indicator of L_1 is more strongly correlated to the first indicator of L_2 than to some other indicators of L_1. Each indicator has an additive standard normal error term. A typical covariance matrix is shown in Table 7. Correlations tend to be rather strong.

By performing 10 simulations with samples of 5000 observations, we got very good results with our algorithm as expected. In three times the algorithm failed to include one indicator overall, and in one time it did not include one indicator from each cluster. No misplaced indicators, no lost latents.

A scree plot of each simulated data set would strongly suggest just one factor (or at most 2). Applying maximum likelihood estimation and principal component analysis with three factors would cluster the indicators of L_1 together, and indicators of L_2 and L_3 together, where no variable would be clustered with the third factor. It was also interesting to observe that we needed at least four factors to get a significant fit with maximum likelihood. Even with four latents, our suggested heuristic clustering would make use of only two of the four factors.

We do not know yet how realistic this example can be, but a similar outcome happened in the analysis of non-simulated data discussed in Section 8.3.

8.2.2 Nonlinear latent structure

In this section we perform a first experiment with a nonlinear latent structure and non-normally distributed data. The graph in Figure 9 is parameterized by the following nonlinear structural

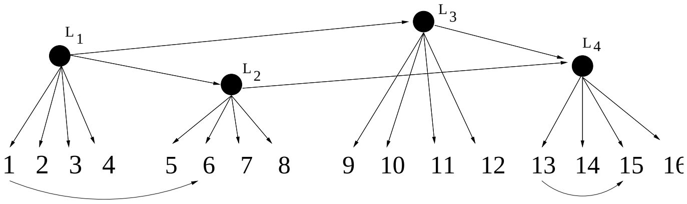


Figure 9: An impure model with a diamond-like latent structure. Notice there are two ways to purify this graph: by removing 6 and 13 or removing 6 and 15.

1.0	-0.683	-0.693	-0.559	-0.414	-0.78	-0.369	-0.396	-0.306	0.328	-0.309	-0.3	-0.231	0.227	0.276	-0.278
-0.683	1.0	0.735	0.603	0.442	0.64	0.389	0.425	0.347	-0.363	0.338	0.339	0.243	-0.238	-0.282	0.282
-0.693	0.735	1.0	0.603	0.426	0.637	0.378	0.408	0.348	-0.365	0.341	0.337	0.236	-0.239	-0.279	0.284
-0.559	0.603	0.603	1.0	0.357	0.524	0.316	0.334	0.282	-0.298	0.279	0.287	0.18	-0.196	-0.222	0.227
-0.414	0.442	0.426	0.357	1.0	0.789	0.761	0.811	0.19	-0.203	0.197	0.194	0.356	-0.371	-0.429	0.439
-0.78	0.64	0.637	0.524	0.789	1.0	0.713	0.757	0.284	-0.304	0.289	0.284	0.354	-0.364	-0.429	0.438
-0.369	0.389	0.378	0.316	0.761	0.713	1.0	0.734	0.171	-0.183	0.174	0.174	0.321	-0.333	-0.387	0.401
-0.396	0.425	0.408	0.334	0.811	0.757	0.734	1.0	0.175	-0.188	0.184	0.183	0.326	-0.34	-0.402	0.41
-0.306	0.347	0.348	0.282	0.19	0.284	0.171	0.175	1.0	-0.858	0.821	0.818	0.199	-0.191	-0.239	0.239
0.328	-0.363	-0.365	-0.298	-0.203	-0.304	-0.183	-0.188	-0.858	1.0	-0.848	-0.843	-0.212	0.204	0.256	-0.25
-0.309	0.338	0.341	0.279	0.197	0.289	0.174	0.184	0.821	-0.848	1.0	0.805	0.201	-0.19	-0.238	0.237
-0.3	0.339	0.337	0.287	0.194	0.284	0.174	0.183	0.818	-0.843	0.805	1.0	0.211	-0.2	-0.246	0.244
-0.231	0.243	0.236	0.18	0.356	0.354	0.321	0.326	0.199	-0.212	0.201	0.211	1.0	-0.654	-0.898	0.777
0.227	-0.238	-0.239	-0.196	-0.371	-0.364	-0.333	-0.34	-0.191	0.204	-0.19	-0.2	-0.654	1.0	0.78	-0.787
0.276	-0.282	-0.279	-0.222	-0.429	-0.429	-0.387	-0.402	-0.239	0.256	-0.238	-0.246	-0.898	0.78	1.0	-0.92
-0.278	0.282	0.284	0.227	0.439	0.438	0.401	0.41	0.239	-0.25	0.237	0.244	0.777	-0.787	-0.92	1.0

Table 8: An example of a sample correlation matrix of a sample of size 5000.

equations:

$$\begin{aligned} L_2 &= L_1^2 + \epsilon_{L2} \\ L_3 &= \sqrt{L_1} + \epsilon_{L3} \\ L_4 &= \sin{(L_2/L_3)} + \epsilon_{L4} \end{aligned}$$

where L_1 is distributed as a mixture of two beta distributions, $Beta(2,4)$ and $Beta(4,2)$, where each one has prior probability of 0.5. Each error term ϵ_{L_v} is distributed as a mixture of a $Beta(4,2)$ and the symmetric of a $Beta(2,4)$, where each component in the mixture has a prior probability that is uniformly distributed in $[0,1]$, and the mixture priors are drawn individually for each latent in $\{L_2, L_3, L_4\}$. The error terms for the indicators also follow a mixture of betas $(2,4)$ and $(4,2)$, each one with a mixing proportion individually chosen according to a uniform distribution in $[0,1]$. The linear coefficients relating latents to indicators and indicators to indicators were chosen uniformly in the interval $[-1.5, -0.5] \cup [0.5, 1.5]$.

To give an idea of how nonnormal the observed distribution can be, we submitted a sample of size 5000 for a Shapiro-Wilk normality test in R 1.6.2 for each variable, and the hypothesis of normality in all 16 variables was strongly rejected, where the highest p-value was at the order of 10^{-11}. Figure 14 depicts histograms for each variable in a specific sample. We show a randomly selected correlation matrix from a sample of size 5000 in Table 8.

In principle, the asymptotic distribution free test of tetrad constraints from (Bollen, 1990) should be the method of choice if the data does not pass a normality test. However, such test uses the fourth moments of the empirical distribution, which can take a long time to be computed if the number of variables is large (since it takes $O(mn^4)$ steps, where m is the number of data points and n is the number of variables). Caching a large matrix of fourth moments may require secondary memory storage, unless one is willing to pay for multiple passes through the data set every

Evaluation of estimated purified models			
	1000	**5000**	**50000**
Wishart test			
missing latents	0.20 ± 0.11	0.20 ± 0.11	0.18 ± 0.12
missing indicators	0.21 ± 0.11	0.22 ± 0.08	0.10 ± 0.13
misplaced indicators	0.01 ± 0.02	0.0 ± 0.0	0.0 ± 0.0
impurities	0.0 ± 0.0	0.0 ± 0.0	0.1 ± 0.21
Bollen test			
missing latents	0.18 ± 0.12	0.13 ± 0.13	0.10 ± 0.13
missing indicators	0.15 ± 0.09	0.16 ± 0.14	0.14 ± 0.11
misplaced indicators	0.02 ± 0.05	0.0 ± 0.0	0.1 ± 0.03
impurities	0.15 ± 0.24	0.10 ± 0.21	0.0 ± 0.0

Table 9: Results obtained for estimated purified graphs with the nonlinear graph. Each number is an average over 10 trials, with an indication of the standard deviation over these trials.

time a test is demanded or if a large amount of RAM is available. Therefore, we also evaluate the behavior of the algorithm using the Wishart test (see Spirtes et al., 2000 for details), which assumes multivariate normality[10]. Samples of size 1000, 5000 and 50000 were used. Results are given in Table 9. Such test might be useful as an approximation, even though it is not the theoretically correct way of approaching such kind of data.

The results are quite close to each other, although the Bollen test at least seems to get better with more data. Results for the proportion of impurities vary more, since we have only two impurities in the true graph. The major difficulty in this example is again the fact that we have two clusters with only three pure latents each. It was quite common that we could not keep the cluster with variables $\{5, 7, 8\}$ and some other cluster in the same final solution because the test (which requires the evaluation of many tetrad constraints) that contrasts two clusters would fail (Step 4 of `BuildFinalPattern`). To give an idea of how having more than three indicators per latent can affect the result, running this same example with 5 indicators per latent (which means at least four pure indicators for each latent) produce better results than anything reported in Table 9 with samples smaller than 1000. That happens because Step 4 of `BuildFinalPattern` only needs *one* triplet from each cluster, and the chances of having at least one triplet from each group that satisfies its criterion increases with a higher number of pure indicators per latent.

Again, factor analysis with oblique rotation and heuristic clustering performed surprisingly well here (ignoring how to interpret the loadings of the known impure indicators), with only an occasional indicator ending up in a wrong cluster. The major difference was the instability of the maximum likelihood estimator, which assumes multivariate normality: in many cases it would require as many as ten random restarts to converge, or not even converge in this given number of trials. Whenever it was possible, significant fits would happen with 6 latents, which makes sense since there are six nodes with children in the true graph (excluding error terms). Using principal component analysis was never a problem, although the SAS default criterion would never choose more than three components. That would usually result in two groups being clustered together, while the other two would remain separated. Other experiments with a larger variety of impurities will be performed in a future work.

[10] We did not implement yet distribution-free tests of vanishing partial correlations. In these experiments we will be using the tests for jointly normal variables, which did not seem to affect the results.

8.3 Test anxiety revisited

One of the reasons why we chose the test anxiety data to illustrate our methodology was due to the similarity among items. A quick examination may even suggest that clustering can be arbitrary without much loss of insight, adding difficulty to this otherwise simple example.

And yet we expect to propose a meaningful model, reinforcing the usefulness of exploratory model-building algorithms. We do not expect that everyone will agree with us[11] when we say that the output of the algorithm, which is the second clustering among those proposed in Section 8.1, is the most meaningful one, but we do believe it gives quite detailed aspects of the unknown data generating process. In this case, there is evidence that variables are approximately multivariate normal.

For instance, cluster $\{x_5, x_6, x_7, x_{14}, x_{17}\}$ seems to reflect how the student's own thoughts interfere with his performance. Cluster $\{x_2, x_9, x_{13}, x_{18}\}$ shows a more distinct grouping of psychological/physiological conditions induced by stress. Cluster $\{x_4, x_{15}, x_{20}\}$ is interesting because it contains the more extreme psychological reactions to an exam. The results may not surprise anyone, but it must be stressed that they were obtained with no prior knowledge (except, of course, that such variables were related under the more general aspect of "test anxiety").

Notice this is a purified model. We know, for instance, that the original latent that had $\{x_5, x_6, x_7, x_{14}, x_{17}\}$ as its children also had x_3 (an item which, interestingly, is also about student's thoughts about his success). As discussed in Section 6.1, if we just ignore those latents that did not appear in the final pure measurement model and if we add back the deleted indicators from the latents in the pure model, we will have the following set of indicators:

- $x_3, x_5, x_6, x_7, x_{14}, x_{17}$
- $x_1, x_2, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{16}, x_{18}$
- $x_4, x_{12}, x_{15}, x_{20}$

The second latent variable as defined by the list above has a much less understandable meaning, which may be an artifact of statistical variability or unidentifiability of the pattern. In a future work we will explore different ways of doing metaclustering in order to take into account what happens when we group together not only the purified latents from each run of the algorithm, but the full set of indicators per latent. Our clustering was generated using 10 cycles of metaclustering. Unlike the simulated data of previous section, metaclustering seems to have made a practical difference, increasing the number of clusters from 2 to 3 by combining clusters from two different runs[12], but in this case that can also be explained by the rather small number of latents we have here.

Conclusions derived from factor analysis, as described in Bartholomew et al. (2002), reveal some subtle differences. Using oblique rotation and two factors, all loadings were quite far from zero. Assigning a cluster to each indicator by choosing the one with the highest loading will give a cluster with $x_3, x_5, x_6, x_7, x_{14}$ and x_{17}, and a second cluster with the remaining ones, $x_1, x_2, x_4, x_8, ..., x_{13}, x_{15}, x_{16}, x_{18}, x_{19}, x_{20}$. The authors interpret the largest cluster as a factor of "emotionality", i.e., "reactions evoked by the nervous system", while the smaller one would be indications of a type of anxiety categorized as "worry" (but notice how variable x_{16} ended up in the large cluster, where it was just thrown away in our approach as an impurity). The smaller cluster is

[11] In part, also, to the fact that such clusters were not completely random, by derived from the variables selected by our algorithm.

[12] We also ran metaclustering independently three times (which means 30 runs of the algorithm, but grouping results into groups of 10) achieving the same result.

virtually one of those we got from our analysis. We divided the larger cluster into two subcategories, one containing the more extreme "emotionality" factors. The authors could have used three factors (or more!) as well, but two were enough according to their criteria of fitness, which was done by basically looking at a scree plot. In general, both analysis agree in the most essential points of this simple example. However, we should stress that our approach requires less subjective decisions: no need for picking the number of latents in advance, no need of choosing some rather arbitrary rotation method, no need of choosing some theoretically unjustified method of clustering indicators by using the strenght of the loadings. For instance, if we perform a chi-square test of statistical fitness using the given covariance matrix, the factor analysis implementation in SAS reveals that just one factor is enough with a p-value of 0.09. This is also the result that minimizes BIC. The default criterion used by SAS also chooses 1 factor only. This is not the kind of insight one would want from such analysis.

As a sidenote, one important fact pointed out by Bartholomew et al. is that orthogonal rotation is hopeless in this data set if the goal is unveiling any kind of "simple structure". That reinforces the notion that assumptions about the independence of latents are not desirable except in very especialized domains.

9 Discussion and future work

The results described in the previous section should be considered as very promising. With sample sizes that are not very large given the number of variables (e.g., 1000 points for 32 variables, and could possibly be less than that for similar results), fewer than usual assumptions, theoretically rigorous algorithms, the algorithm exceeded expectations, considering the previous results presented in (Silva, 2002a). Constraint-satisfaction techniques for learning graphical models performed better than score-based approaches described in this previous work. It does not mean that new and efficient score-based algorithm using approximate Bayesian scores cannot be designed for this problem, but there are some theoretical issues that have to solved, such as how to define a consistent score of a latent graph without evaluating the full posterior. There are computational issues, such as how to efficiently compute the score of a new candidate since latent variable graphs are not decomposable. Another fundamental problem is how to structure the search space in a way a solution can be found by greedy search. One still has to consider how to make this search as distribution-free as possible.

Considering the generality of the problem, the solution here presented should be of practical applicability for many scenarios. However, there is still a lot of improvements that can be done, and the following topics can be considered as immediate starting points for more research:

- since this report introduced our first approach for this problem, we did not worry about optimization in order to make the algorithm as simple as possible, but in future versions we may be concerned to improve its scalability at the expense of possibly missing exact solutions. We currently researching alternative algorithms that can reduce its computational complexity. For instance, the work by Bansal et al. (2002) describes approximation algorithms for problems closely related to some of the clustering tasks we need to accomplish;

- there are a large number of alternatives when one intends to increase the statistical robustness of this approach. For example, using false discovery rate procedures instead of Bonferroni adjustments within each set of three tetrad constraints that are tested, or scoring components of the final model (for example, using a χ^2 score assuming some parametric form for the joint distribution) in order to detect wrong decisions, or bootstrapping the statistical tests. Important attention should be paid to the case where the computational cost may be prohibitive;

- one specific way of improving robustness is exploring redundancy in tetrad constraints. One of the reasons why experiments with models with four pure indicators worked much better than the ones with three was due to the extra amount of redundancy. A starting point is using the ideas from Bollen and Ting (1993) in order to specify how to use redundancy to perform more reliable decisions;
- a more extensive experimental evaluation, including more tests with non-Gaussian data and real-world data, as well as simulations where assumptions do not hold;
- designing alternatives that use weaker assumptions. For instance, determining which identifiability guarantees we have when the pure models do not include all latents, or when we have three pure indicators per pair of latents, instead with respect to the whole set of latents. This seems to be one of the most important practical extensions of this procedure and a way to provide formal justification for the `Metaclustering` procedure;
- creating similar approaches for discrete data. The factor analysis framework discussed in Bartholomew and Knott (1999) describes how to use factor analysis for any exponential family distribution, and that can be a starting point;
- complementary approaches for maximizing the number of measures we keep in a pattern (for better ontology learning and data mining, for instance), as well as trying to identify the nature of the impurities (direct causes, error correlations, etc.);
- pure models are identifiable. Which reliable estimation techniques can be designed for estimating the effects of latents into their indicators in pure model without assuming a parametric distribution for the observed variables?
- since this method uses only the second moments of the distribution, one could also explore how kernel methods could be integrated into this problem;
- we will also work in the problem of learning the structure among the latents. Current methods (e.g., Silva (2002a)) makes use of the assumption that latents are linearly related. Can we use the techniques here describe to generalize this method?

10 Acknowledgements

Thanks for Joseph Ramsey for the help with the Tetrad 4.2 code and Martin Zinkevich for the suggestion of how to reduce 3SAT to the pattern purification problem. This work was supported by a NASA grant #NCC2-1227.

References

Attias, H. (1999). “Independent factor analysis”. *Neural Computation* 11, 803-851.

Bansal, B., Blum, A. and Chawla, S. (2002). “Correlation clustering”. *Proceedings of the 43th Symposium on Foundations of Computer Science*, p. 238-247.

Bartholomew, D. and Knott, M. (1999). *Latent Variable Models and Factor Analysis.* 2^{nd} edition, Arnold Publishers.

Bartholomew, D.; Steele, F.; Moustaki, I. and Galbraith, J. (2002). *The Analysis and Interpretation of Multivariate Data for Social Scientists.* Chapman & Hall.

Beal, M.; Ghahramani, Z. and Rasmussen C. (2002). "The infinite hidden Markov model". *Advances in Neural Information Processing Systems* 14.

Bishop, C. (1998). "Latent variable models". In *Learning in Graphical Models*, p. 371-403. MIT Press.

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

Bollen, K. (1990). "Outlier screening and a distribution-free test for vanishing tetrads". *Sociological Methods and Research* 19: 80-92.

Bollen, K. and Ting. K. (1993). "Confirmatory Tetrad analysis". In *Sociological Methodology*, p. 147-176. Blackwell Publishers.

Carmines, E. and Zeller, R. (1979). *Reliability and Validity Assessment.* Quantitative Applications in the Social Sciences 17. Sage Publications.

Cattel, R. (1978). *The Scientific Use of Factor Analysis in Behavioral and Life Sciences.* Plenum Press, NY.

Chickering, D. (2002). "Learning equivalence classes of Bayesian networks". *Journal of Machine Learning Research* 2, 445-498.

Elidan, G.; Lotner, N.; Friedman, N. and Koller, D. (2000). "Discovering hidden variables: a structure-based approach". Neural Information Processing Systems 13.

Fornell, C. and Yi, Y. (1992) "Assumptions of the two-step approach to latent variable modeling". *Sociological Methods & Research* 20 (3), 291-320.

Geiger, D., Heckerman, D., King, H. and Meek, C. (2001) "Stratified exponential families: graphical models and model selection". *Annals of Statistics* 29, 505-529.

Harman, H. (1967). *Modern Factor Analysis.* University of Chicago Press, 2nd edition.

Hayduk, L. and Glaser, D. (2000). "Jiving the four-step, waltzing around factor analysis, and other serious fun". *Structural Equation Modeling* 7(1), 1-35.

Heckerman, D.; Meek, C. and Cooper, G. (1999). "A Bayesian approach to causal discovery". In *Computation, Causation and Discovery*, 141-166. AAAI Press.

Hofmann, T. (2001). "Unsupervised learning by probabilistic semantic analysis". *Machine Learning* 42, p. 177-196. Kluwer Academic Publishers.

Johnson, R. and Wichern, D. (1998). *Applied Multivariate Statistical Analysis.* Prentice-Hall.

Jordan, M. (ed.) (1998). *Learning in Graphical Models.* MIT Press.

Malinowski, E. (2002). *Factor Analysis in Chemistry.* John Wiley & Sons, 3rd edition.

Meek, C. (1995) "Causal inference and causal explanation with background knowledge". *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, 403 - 418. Morgan Kaufmann.

Minka, T. (2000). "Automatic choice of dimensionality for PCA". Advances in Neural Information Processing Systems 13, 598-604.

Mitchell, T. (1997). *Machine Learning.* McGraw Hill.

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

Pearl, J. (2000) *Causality.* Cambridge University Press.

Rayment, R. and Jöreskog, K. (1993). *Applied Factor Analysis in the Natural Sciences.* Cambridge University Press.

Scheines, R.; Hoijtink, H. and Boomsa, A. (1999). "Bayesian estimation and testing of structural equation models". *Psychometrika* 64, 37–52.

Shafer, G.; Kogan, A. and Spirtes, P. (1993). "Generalization of the Tetrad Representation Theorem". DIMACS Technical Report 93-68.

Silva, R. (2002a) "The structure of the unobserved". Technical report CMU-CALD-02-102, Carnegie Mellon University, Pittsburgh, PA.

Silva, R. (2002b) "The new Washdown algorithm". Unpublished research note.

Spirtes, P; Glymour, C. and Scheines, R. (2000). *Causation, Prediction and Search.* MIT Press.

Wishart, J. (1928). "Sampling errors in the theory of two factors". *British Journal of Psychology* 19, 180-187.

Appendix

A More about the *Unclustered* test

The core of our methodology is the procedure to detect when two indicators cannot share a same parent, which we called the *Unclustered* test. In Appendix C we prove its correctness for the

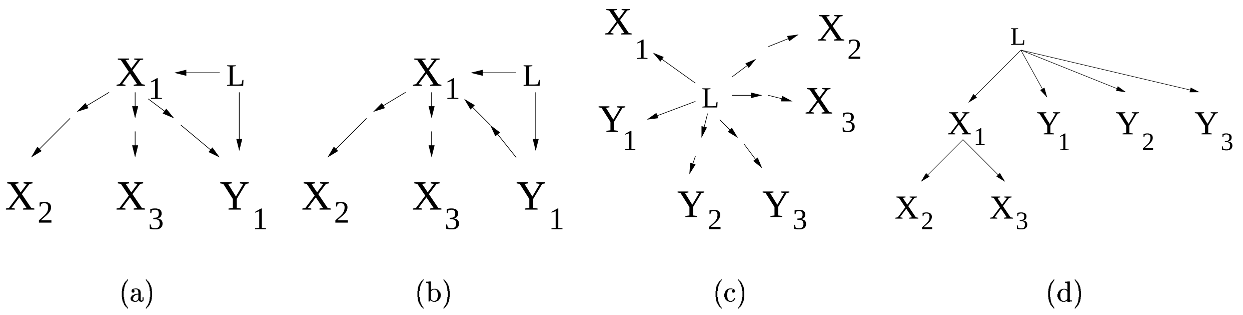


Figure 10: (a) X_1 as a choke point for $\{X_2, X_3, Y_1\}$. We represent the fact that X_1 may be only an ancestor instead of a parent by a sequence of two edges $\rightarrow\rightarrow$. (b) Another example of how X_1 chould be a choke point among the same elements. In (c), the global picture when L is a choke point for $\{X_1, X_2, X_3, Y_1\}$ and $\{X_1, Y_1, Y_2, Y_3\}$. (d) A case where all conditions for *Unclustered* holds, with the exception of #1.

general case where a latent can be a non-linear function of its parents. In this section, we will give an intuition of why it works in the special case where the relationships among latents are linear.

Under the assumption we have a linear causal graph faithful to a distribution, the Tetrad Representation Theorem described in Spirtes et al. (2000) and Shafer et al. (1993) gives a graphical condition that holds if and only if a tetrad constraint holds. Given four random variables, A, B, C and D, the constraint $\sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BC}$ is entailed by a *linear* causal graph faithful to their distribution if and only if there is a *choke point* between the pair $\{A, B\}$ and $\{C, D\}$.

Before defining choke points, we will first introduce the concept of treks in a directed graph. Let a *trek* in a directed acyclic graph be any path where no two consecutive edges in the path point to the same node. A directed path $A \rightarrow B \rightarrow C \rightarrow D$ is an example of a trek. A path with a single *source* such as $A \leftarrow B \leftarrow C \rightarrow D$ is an example of a trek, where C is the source[13]. A path with a *collider* such as $A \rightarrow B \leftarrow C \leftarrow D$ is not a trek, where B is the collider.

If CP is a choke point of the pair of pairs $\{\{A, B\}, \{C, D\}\}$, then every trek between an element of $\{A, B\}$ and an element of $\{C, D\}$ has to include CP[14]. Given two sets of variables $\mathbf{X} = \{X_1, X_2, X_3\}, \mathbf{Y} = \{Y_1, Y_2, Y_3\}$, a covariance matrix Σ that includes the covariance matrix of $\mathbf{X} \cup \mathbf{Y}$, we define $Unclustered(\mathbf{X}, \mathbf{Y}; \Sigma)$ as true if and only if every element in $\mathbf{X}$ is uncorrelated with every element in $\mathbf{Y}$ or the following conditions hold:

1. for all $\{A, B, C\} \subset \mathbf{X} \cup \mathbf{Y}, \rho_{AB.C} \neq 0$ and $\rho_{AB} \neq 0$;
2. for all $Y \in \mathbf{Y}, TetradScore(\mathbf{X} \cup Y; \Sigma) = 3$;
3. for all $X \in \mathbf{X}, TetradScore(\mathbf{Y} \cup X; \Sigma) = 3$;
4. for all $\{X_a, X_b\} \subset \mathbf{X}, \{Y_a, Y_b\} \subset \mathbf{Y}, \sigma_{X_aY_a}\sigma_{X_bY_b} = \sigma_{X_aY_b}\sigma_{X_bY_a} \neq \sigma_{X_aX_b}\sigma_{Y_aY_b}$

[13] A was the source in the previous example. Essentially, a source node should have a directed path from it to each of the other nodes in the path.

[14] The definition of choke point also requires it being on a given "side" of each trek, but to keep the exposition simple, we will not comment further on it. Shafer et al. (1993) give a rather complete coverage of the topic.

The claim is: if *Unclustered*$(\mathbf{X}, \mathbf{Y}; \Sigma) = true$, then no element in $\mathbf{X}$ can share a parent with any element in $\mathbf{Y}$. We will show here an intuitive demonstration of this argument using the Tetrad Representation Theorem.

Suppose X_1 and Y_1 share a same latent parent, L. Since all three tetrads hold in $\{X_1, X_2, X_3, Y_1\}$, there must be a choke point between any two elements of this set. But the trek $X_1 \leftarrow L \rightarrow Y_1$ exists, so the choke point must be X_1, L or Y_1. Suppose it is X_1. So, given any pair in $\{X_2, X_3, Y_1\}$, all treks must be intermediated by X_1. Figures 10a and 10b illustrate two typical cases. In all this cases, every pair in $\{X_2, X_3, Y_1\}$ is d-separated by X_1, which contradicts Condition #1 in the definition of *Unclustered*. Therefore, X_1 (nor Y_1) can be the choke point.

Since L is a choke point in $\{X_1, X_2, X_3, Y_1\}$, L is in every trek among elements of such set[15]. Since L is a choke point in $\{X_1, Y_1, Y_2, Y_3\}$, then L is in every trek among elements of such set. Therefore, as Figure 10c suggests, L is in every trek among elements in $\{X_1, X_2, Y_1, Y_2\}$, which by the Tetrad Representation Theorem will imply $\sigma_{X_1Y_1}\sigma_{X_2Y_2} = \sigma_{X_1Y_2}\sigma_{X_2Y_1} = \sigma_{X_1X_2}\sigma_{Y_1Y_2}$. Contradiction.

Could we simplify the definition of *Unclustered*? Figure 10d illustrates a case where Conditions #2, #3 and #4 hold while #1 does not, yet X_1 and Y_1 share a same latent parent. If we are interested only in verifying if one given pair of nodes share a same parent (as suggested by the way loops are organized in the `FindMeasurementPattern` algorithm), then it will not be necessary to check all those tetrad constraints: for X_1 and Y_1, we can check only if $\sigma_{X_1X_2}\sigma_{X_3Y_1} = \sigma_{X_1X_3}\sigma_{X_2Y_1} = \sigma_{X_1Y_1}\sigma_{X_2Y_3}$, $\sigma_{Y_1Y_2}\sigma_{Y_3X_1} = \sigma_{Y_1Y_3}\sigma_{Y_2X_1} = \sigma_{Y_1X_1}\sigma_{Y_2Y_3}$ and $\sigma_{X_1Y_1}\sigma_{X_2Y_2} = \sigma_{X_1Y_2}\sigma_{X_2Y_1} \neq \sigma_{X_1X_2}\sigma_{Y_1Y_2}$. It is not clear if this approach can actually save computational time (since we still need to deal with six nodes at a time anyway) or if it increases statistical robustness. In preliminary experiments, it was actually harmful in both ways: not only it was prone to make more statistical mistakes of separating nodes that were actually children of the same latent, but it took considerably more computational time, since we only eliminate edges among two nodes at a time while we still keep looking at six nodes in a single test. Intermediate approaches, such as evaluating only the tetrads that allow us to decide when a subset of $\mathbf{X}$ is separated from a subset of $\mathbf{Y}$, could be evaluated in the future.

It remains as an open question if we can come out with a considerably different way of deciding when two nodes do not share a same latent parent, or even proving that is not possible and all such approaches should test essentially the same constraints as *Unclustered* does. Also, it would be interesting to come up with a test of deciding when two nodes *do* share a common latent parent. The way by which we are able to accomplish this in the present work requires having the global picture, i.e., knowing the measurement pattern and then using the assumption of existance of a pure measurement model including *all* latents with at least three children. It is unclear right now how to relax this assumption, if possible, while guaranteeing the same identifiability results of pure measurement models.

An interesting variation would be deciding when two nodes do not share some common parent. While the *Unclustered* test is general enough to the point it does not need to consider if there are other common parents, we may try to get a different test that separates nodes that cannot have at least one parent in common, but that may affect other properties of the algorithm.

[15] Also, since L cannot be a descendant on an indicator, then L is an ancestor of every one of such nodes, which is the reason why we should not worry about which "side" of each trek L should be, as suggested in the previous footnote.

B More examples

In this subsection, we illustrate how the algorithm works by describing step-by-step solutions for two more simple cases.

Our first example is depicted in Figure 11. The true model (Figure 11a) consists of two latents with 6 indicators each, where many pairs of indicators have correlated errors. Adopting the representation $(\{cluster_1\}, \{cluster_2\})$ to indicate graphs in $\mathbf{G_S}$ for the given problem, the possible induced pure measurement models are: $(\{1, 2, 3\}, \{10, 11, 12\})$; any subgraph of $(\{1, 2, 4, 5, 6\}, \{10, 11, 12\})$ with at least three members from each cluster; any subgraph of $(\{4, 5, 6\}, \{7, 8, 10, 11, 12\})$ with at least three members from each cluster; and, finally, $(\{4, 5, 6\}, \{7, 8, 9\})$.

The initial step is depicted in Figure 11b, which shows the initial configuration of our clustering after Step 3 of `FindMeasurementPattern`. Here, solid edges correspond to the *Black* edges described in Table 1. After the first pass, Step 4 will change the color of some edges to *Blue* and *Gray*. Figure 11c shows the NG_{Blue} graph at the end of this step (*Blue* edges are represent by bold solid edges), while Figure 11d shows NG_{Gray} (dotted edges). At Step 5, edges such as the ones between elements in $\{4, 5, 6\}$ and elements in $\{10, 11, 12\}$ are removed. Notice the presence of *Yellow* edges $(3, 7)$, $(3, 8)$ and $(3, 9)$ (represented as dashed edges in Figure 11e, the outcome of Step 5). This happens because, for instance, nodes 3 and 7 never appear together in any solution graph, yet they are not impure with respect to each other. Situations like that are the main responsible for the existence of many maximal cliques that overlap other maximal cliques in **Clustering**. It will not be the case here, thought: **Components**, the set of disjoint graphs shown in Figure 11f actually finds the right partition with a one-to-one correspondence to each latent in the true graph. Figure 11g corresponds to **Clustering**, the graph obtained when we put back the *Gray* and *Yellow* edges withing each disjoint component. Figure 11h shows the first sketch of a measurement pattern as computed at the end of Step 2 of `BuildFinalPattern`. Step 3 of `BuildFinalPattern` simply adds the cross-construct impurity edges (Figure 11i), while the final step evaluates which pairs of latents could be part of an induced pure measurement model. Figure 11j shows the final measurement pattern as computed by our algorithm.

Our next example is depicted in Figure 12a. Only a handful pairs of impurities exists, as illustrated by Figure 12b, where the only *Gray* edges are $(2, 5), (2, 6), (3, 5)$. Let's assume the relationship among the latents is linear. With linear latents, the predicate $Unclustered(\{2, 7, 8\}, \{9, 10, 11\})$ will hold because L_2 d-separates L_1 from L_3 (Spirtes et al., 2000). This will cluster together indicator 2 with measures $\{5, 6, 7, 8\}$. Figure 12c shows that we have two disjoint components, but three maximal cliques. **Clustering** will be equal to $\{1, 2, 3, 4, 5\}, \{2, 5, 6, 7, 8\}$ and $\{9, 10, 11\}$. Figure 12d depicts the final measurement pattern. The presence of 2 and 5 in different clusters indicates that a variety of latent variable graphs may have generated that pattern. For example, we cannot tell if indicator 5 is a measure of T_1, T_2 or both (it is an indicator of both in the true graph, but if the edge between 3 and 5 was double-edged, the pattern would still be the same). Fortunately, when we purify the measurement pattern we are guaranteed to have a correct model, in the sense of having proper clustering assignments, as explained in Section 5.

C Proofs

Before presenting proofs for the lemmas and theorems stated in the body of this text, we will introduce the following notation. Let σ_{XY} denote the covariance of any two random variable X and Y and $\rho_{XY.Z}$ denotes the partial correlation of X and Y given Z.

Also, let $X = \lambda_{x0}L + \sum_i^k \lambda_{xi}\eta_i$ and Y be random variables with zero mean, L being another random variable with zero mean. We define σ_{XYL}, the "covariance of X and Y through L", as $\sigma_{XYL} \equiv \lambda_{x0}E[LY]$. This concept is going to be used in Lemmas 3 and 4. The definition of trek is also used in the proofs and can be found in Appendix A.

We will consistently make use of polynomial identities defined by the entailed tetrad constraints in such graphs. The variables in such polynomials are the free linear parameters linking each observed node to its parents. The results hold almost surely with respect to a Lebesgue measure over the free linear coefficients. Therefore, these results may fail in sets with a Lebesgue measure of zero, but so does the faithfulness assumption in linear parameterization of graphs (Spirtes et al, 2000). In a similar way, we will assume that such cases of probability zero are of practical irrelevance.

Also, most of the times we will not explicitly represent the extra independent error term that is included in the definition of linear measurement models: without loss of generality, we will treat them as regular parents with a dummy linear coefficient that is always set to 1, but that will be represented by some arbitrary symbol. The only exception will be in Lemma 5.

Lemma 1 *Let $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E}_\epsilon)$ be a linear latent variable graph. Let $\{A, B\} \subset \mathbf{O}$ be nodes of G. If $\sigma_{AB} = 0$, then A and B do not share a same parent in G.*

Proof of Lemma 1: Assume $\sigma_{AB} = 0$. Suppose A and B have a common parent L in G. Let the structural equations for A and B be $A = aL + \sum_i a_i A_i$, $B = bL + \sum_j b_j B_j$, where $\{a, a_1, a_2, ..., a_{\#A}, b, b_1, b_2, ..., b_{\#B}\}$ are real coefficients, $\{L, A_1, A_2, ..., A_{\#A}\}$ are all parents of A in G and $\{L, B_1, B_2, ..., B_{\#B}\}$ are all parents of B in G. Then the following holds:

$$\sigma_{AB} = E[AB] = ab\sigma_L^2 + f(G, A, B) = 0 \qquad (1)$$

The polynomial $f(G, A, B)$ cannot possibly contain any term with the product ab. Since this polynomial is identically zero, we have to have $ab\sigma_L^2 = 0$. Since $a \neq 0, b \neq 0$ and by faithfulness, $\sigma_L^2 \neq 0$, this is a contradiction. □

Lemma 2 *Let $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E}_\epsilon)$ be a linear latent variable graph. For any set $\mathbf{O'} = \{A, B, C, D\} \subseteq \mathbf{O}$, if G entails $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BD}$ and for no set $\{X, Y, Z\} \subset \mathbf{O'}$ we have $\rho_{XY.Z} = 0$ and $\rho_{XY} \neq 0$, then no element in $\mathbf{O'}$ is a descendant of another element of $\mathbf{O'}$ in G.*

Proof of Lemma 2: Since G is acyclic, then at least one element in $\mathbf{O'}$ is not an ancestor in G of any other element in this set. By symmetry, we can assume without loss of generality that D is such node. Since the measurement model is linear, we can write A, B, C, D as linear functions of their parents:

$$\begin{array}{rcl} A & = & \sum_p a_p A_p \\ B & = & \sum_i b_i B_i \\ C & = & \sum_j c_j C_j \\ D & = & \sum_k d_k D_k \end{array}$$

where on the right-hand side of each equation we have the respective parents of A, B, C and D. Such parents can be latents or another indicators, but each indicator has at least one latent parent. Since each indicator is always a linear function of its parents, by composition of linear functions we have that each $X \in \mathbf{O'}$ will be a linear function of its *immediate latent ancestors*, i.e., latent ancestors[16] L_{X_v} of X such that there is a directed path from L_{X_v} to X in G that does not contain any other element of $\mathbf{L}$. The equations above can then be rewritten as:

$$\begin{array}{rcl} A & = & \sum_p \lambda_{A_p} L_{A_p} \\ B & = & \sum_i \lambda_{B_i} L_{B_i} \\ C & = & \sum_j \lambda_{C_j} L_{C_j} \\ D & = & \sum_k \lambda_{D_k} L_{D_k} \end{array}$$

where on the right-hand side of each equation we have the respective immediate latent ancestors of A, B, C and D and the λ parameters are functions of the original coefficients of the measurement model.

Assume C is an ancestor of D. Let L be a latent parent of C, where the edge from L into C is labeled with c, corresponding to its linear coefficient. We can rewrite the equation for C as

$$C = cL + \sum_j \lambda_{C_j} L_{C_j} \tag{2}$$

where by an abuse of notation we are keeping the same index j to represent the other latent ancestors of C. Moreover, L can appear again in the summation if there is more than one directed path from L to C. In this case, the corresponding coefficient λ is modified by subtracting c. What is important here is that the symbol c does not appear in $\sum_j \lambda_{C_j} L_{C_j}$.

By a final abuse of notation, rewrite A, B and D as

$$\begin{array}{l} A = c\omega_a L + \sum_p \lambda_{A_p} L_{A_p} \\ B = c\omega_b L + \sum_i \lambda_{B_i} L_{B_i} \\ D = c\omega_d L + \sum_k \lambda_{D_k} L_{D_k} \end{array}$$

Each ω_v symbol is a polynomial function of all (possible) directed paths from C to $X_v \in \{A, B, D\}$, as illustrated in Figure 13. The corresponding $\lambda_{X_{v_t}}$ coefficient for L is adjusted in the summation. Again, L may appear in the summation if there are directed paths from L to X_v that do not go through C. If C has more than one parent, then the expression for ω_v will appear again into some $\lambda_{X_{vt}}$. However, the symbol c *cannot* appear again into any $\lambda_{X_{vt}}$, since ω_v summarizes all possible directed paths from C to X_v. This remark will be very important later when we will factorize the expression corresponding to the tetrad constraints. Notice that while is possible to have $\omega_a = 0$ or $\omega_b = 0$, by assumption $\omega_d \neq 0$.

Another important point to be emphasized is that *no term inside ω_d can appear in the expression for A and B*. That happens because D is not an ancestor of A, B or C, and at least the edges from the parents of D to D cannot appear in any trek between any pair of elements in $\{A, B, C\}$ and every term inside ω_d contains the label of one edge between a parent of D and D. This remark will also be very important later when we will factorize the expression corresponding to the tetrad constraints.

By the definitions above, we have:

$$\begin{array}{rcl} \sigma_{AB} & = & c^2\omega_a\omega_b\sigma_L^2 + c\omega_a \sum \lambda_{B_i}\sigma_{L_{B_i}L} + c\omega_b \sum \lambda_{A_p}\sigma_{L_{A_p}L} + \sum\sum \lambda_{A_p}\lambda_{B_i}\sigma_{L_{A_p}L_{B_i}} \\ \sigma_{CD} & = & c^2\omega_d\sigma_L^2 + c\sum \lambda_{D_k}\sigma_{L_{D_k}L} + c\omega_d \sum \lambda_{C_j}\sigma_{L_{C_j}L} + \sum\sum \lambda_{C_j}\lambda_{D_k}\sigma_{L_{C_j}L_{D_k}} \\ \sigma_{AC} & = & c^2\omega_a\sigma_L^2 + c\omega_a \sum \lambda_{C_j}\sigma_{L_{C_j}L} + c\sum \lambda_{A_p}\sigma_{L_{A_p}L} + \sum\sum \lambda_{A_p}\lambda_{C_j}\sigma_{L_{A_p}L_{C_j}} \\ \sigma_{BD} & = & c^2\omega_b\omega_d\sigma_L^2 + c\omega_b \sum \lambda_{D_k}\sigma_{L_{D_k}L} + c\omega_d \sum \lambda_{B_i}\sigma_{L_{B_i}L} + \sum\sum \lambda_{B_i}\lambda_{D_k}\sigma_{L_{B_i}L_{D_k}} \end{array}$$

[16] We will also treat error nodes as latent parents in this lemma.

Since the polynomial identity $\sigma_{AB}\sigma_{CD} - \sigma_{AC}\sigma_{BD} = 0$ should hold for every set of parameters in the measurement model, then the sum of every term including the product $c^2\omega_{dt}$ should vanish to zero, where ω_{dt} is some term inside the polynomial ω_d.

Before using this result, we need to identify precisely which elements of the polynomial $\sigma_{AB}\sigma_{CD} - \sigma_{AC}\sigma_{BD}$ can be factored by $c^2\omega_{dt}$, for some arbitrary ω_{dt}. This will clearly include elements from any term that will explicitly include $c^2\omega_d$ when multiplying the covariance equations above. Notice that some λ_{d_k} will be functions of ω_{dt}: every immediate latent ancestor of C is an immediate latent ancestor of D. Therefore, for each common latent ancestor parent L_q of C and D, we have that $\lambda_{d_q} = \omega_d\lambda_{c_q} + t(L_q, D) = \omega_{dt}\lambda_{c_q} + (\omega_d - \omega_{dt})\lambda_{c_q} + t(L_q, D)$, where $t(L_q, D)$ is a polynomial representing other directed paths from L_q to D that do not go through C.

For example, consider the expression $c^2\omega_a\left(\sum \lambda_{B_i}\sigma_{L_{B_i}L}\right)\left(\sum \lambda_{D_k}\sigma_{L_{D_k}L}\right)$, which is an additive term inside the product $\sigma_{AB}\sigma_{CD}$. If we group only those terms inside this expression that contain ω_{dt}, we will get $c^2\omega_a\omega_{dt}\left(\sum \lambda_{B_i}\sigma_{L_{B_i}L}\right)\left(\sum \lambda_{C_j}\sigma_{L_{C_j}L}\right)$ where the index j runs over the same latent ancestors as in (2). As discussed before, no term in ω_{dt} can be a factor inside any λ_{B_i}. For the same reason, it cannot appear inside ω_a.

When one writes down the algebraic expression for $\sigma_{AB}\sigma_{CD} - \sigma_{AC}\sigma_{BD}$ as functions of λs, c, $\omega_a, \omega_b, \omega_{dt}$, the terms

$$\begin{aligned}
&c^2\omega_{dt}[\sigma_L^2 \sum\sum \lambda_{A_p}\lambda_{B_i}\sigma_{L_{A_p}L_{B_i}} + \omega_a\omega_b\sigma_L^2 \sum\sum \lambda_{C_j}\lambda_{C_{j'}}\sigma_{L_{C_j}L_{C_{j'}}} + \omega_a \sum \lambda_{B_i}\sigma_{L_{B_i}L} \sum \lambda_{C_j}\sigma_{L_{C_j}L} + \\
&\omega_b \sum \lambda_{A_p}\sigma_{L_{A_p}L} \sum \lambda_{C_j}\sigma_{L_{C_j}L}] - \\
&c^2\omega_{dt}[\omega_b\sigma_L^2 \sum\sum \lambda_{A_p}\lambda_{C_j}\sigma_{L_{A_p}L_{C_j}} + \omega_a\sigma_L^2 \sum\sum \lambda_{B_j}\lambda_{C_j}\sigma_{L_{B_i}L_{C_j}} + \omega_a\omega_b \sum \lambda_{C_j}\sigma_{L_{C_j}L} \sum \lambda_{C_j}\sigma_{L_{C_j}L} + \\
&\sum \lambda_{A_p}\sigma_{L_{A_p}L} \sum \lambda_{B_i}\sigma_{L_{B_i}L}]
\end{aligned}$$

will be the *only* ones being multiplied by $c^2\omega_{dt}$. Since this has to be identically zero and $\omega_{dt} \neq 0$, we have the following relation:

$$f_1(G) = f_2(G) \tag{3}$$

where

$$\begin{aligned}
f_1(G) = c^2[&\sigma_L^2 \sum\sum \lambda_{A_p}\lambda_{B_i}\sigma_{L_{A_p}L_{B_i}} + \omega_a\omega_b\sigma_L^2 \sum\sum \lambda_{C_j}\lambda_{C_{j'}}\sigma_{L_{C_j}L_{C_{j'}}} + \omega_a \sum \lambda_{B_i}\sigma_{L_{B_i}L} \sum \lambda_{C_j}\sigma_{L_{C_j}L} + \\
&\omega_b \sum \lambda_{A_p}\sigma_{L_{A_p}L} \sum \lambda_{C_j}\sigma_{L_{C_j}L}]
\end{aligned}$$

$$\begin{aligned}
f_2(G) = c^2[&\omega_b\sigma_L^2 \sum\sum \lambda_{A_p}\lambda_{C_j}\sigma_{L_{A_p}L_{C_j}} + \omega_a\sigma_L^2 \sum\sum \lambda_{B_j}\lambda_{C_j}\sigma_{L_{B_i}L_{C_j}} + \omega_a\omega_b \sum \lambda_{C_j}\sigma_{L_{C_j}L} \sum \lambda_{C_j}\sigma_{L_{C_j}L} + \\
&\sum \lambda_{A_p}\sigma_{L_{A_p}L} \sum \lambda_{B_i}\sigma_{L_{B_i}L}]
\end{aligned}$$

Similarly, when we factor terms multiplying $c\omega_{dt}$ (i.e., the power of c in the term has to be 1), we get the following expression as an additive term of $\sigma_{AB}\sigma_{CD} - \sigma_{AC}\sigma_{BD}$:

$$\begin{aligned}
&c\omega_{dt}[\omega_a \sum \lambda_{B_i}\sigma_{L_{B_i}L} \sum\sum \lambda_{C_j}\lambda_{C_{j'}}\sigma_{L_{C_j}L_{C_{j'}}} + \omega_b \sum \lambda_{A_p}\sigma_{L_{A_p}L} \sum\sum \lambda_{C_j}\lambda_{C_{j'}}\sigma_{L_{C_j}L_{C_{j'}}} + \\
&2 \sum \lambda_{C_j}\sigma_{L_{C_j}L} \sum\sum \lambda_{A_p}\lambda_{B_i}\sigma_{L_{A_p}L_{B_i}}] - \\
&c\omega_{dt}[\omega_a \sum \lambda_{C_j}\sigma_{L_{C_j}L} \sum\sum \lambda_{B_i}\lambda_{C_j}\sigma_{L_{B_i}L_{C_j}} + \sum \lambda_{A_p}\sigma_{L_{A_p}L} \sum\sum \lambda_{B_i}\lambda_{C_j}\sigma_{L_{B_i}L_{C_j}} + \\
&\omega_b \sum \lambda_{C_j}\sigma_{L_{C_j}L} \sum\sum \lambda_{A_p}\lambda_{C_j}\sigma_{L_{A_p}L_{C_j}} + \sum \lambda_{B_i}\sigma_{L_{B_i}L} \sum\sum \lambda_{A_p}\lambda_{C_j}\sigma_{L_{A_p}L_{C_j}}]
\end{aligned}$$

for which we must have:

$$g_1(G) = g_2(G) \tag{4}$$

where

$g_1(G) = c[\omega_a \sum \lambda_{B_i}\sigma_{L_{B_i}L} \sum\sum \lambda_{C_j}\lambda_{C_{j'}}\sigma_{L_{C_j}L_{C_{j'}}} + \omega_b \sum \lambda_{A_p}\sigma_{L_{A_p}L} \sum\sum \lambda_{C_j}\lambda_{C_{j'}}\sigma_{L_{C_j}L_{C_{j'}}} +$
$2\sum \lambda_{C_j}\sigma_{L_{C_j}L} \sum\sum \lambda_{A_p}\lambda_{B_i}\sigma_{L_{A_p}L_{B_i}}]$

$g_2(G) = c[\omega_a \sum \lambda_{C_j}\sigma_{L_{C_j}L} \sum\sum \lambda_{B_i}\lambda_{C_j}\sigma_{L_{B_i}L_{C_j}} + \sum \lambda_{A_p}\sigma_{L_{A_p}L} \sum\sum \lambda_{B_i}\lambda_{C_j}\sigma_{L_{B_i}L_{C_j}} +$
$\omega_b \sum \lambda_{C_j}\sigma_{L_{C_j}L} \sum\sum \lambda_{A_p}\lambda_{C_j}\sigma_{L_{A_p}L_{C_j}} + \sum \lambda_{B_i}\sigma_{L_{B_i}L} \sum\sum \lambda_{A_p}\lambda_{C_j}\sigma_{L_{A_p}L_{C_j}}]$

Finally, we look at terms multiplying ω_{dt} without c, which will result in:

$$h_1(G) = h_2(G) \tag{5}$$

where

$$h_1(G) = \sum\sum \lambda_{A_p}\lambda_{B_i}\sigma_{L_{A_p}L_{B_i}} \sum\sum \lambda_{C_j}\lambda_{C_{j'}}\sigma_{L_{C_j}L_{C_{j'}}}$$

$$h_2(G) = \sum\sum \lambda_{A_p}\lambda_{C_j}\sigma_{L_{A_p}L_{C_j}} \sum\sum \lambda_{B_i}\lambda_{C_j}\sigma_{L_{B_i}L_{C_j}}$$

Writing down the full expression for $\sigma_{AC}\sigma_{BC}$ and $\sigma^2_C\sigma_{AB}$ will result in:

$$\sigma_{AC}\sigma_{BC} = P(G) + \omega_d[f_2(G) + g_2(G) + h_2(G)] \tag{6}$$

$$\sigma^2_C\sigma_{AB} = P(G) + \omega_d[f_1(G) + g_1(G) + h_1(G)] \tag{7}$$

where

$$\begin{aligned} P(G) \quad = \quad & c^4\omega_a\omega_b(\sigma^2_L)^2 + c^3\omega_a\omega_b\sigma^2_L \sum \lambda_{C_j}\sigma_{L_{C_j}L} + c^3\omega_a\sigma^2_L \sum \lambda_{B_i}\sigma_{L_{B_i}L} + \\ & c^3\omega_a\omega_b\sigma^2_L \sum \lambda_{C_j}\sigma_{L_{C_j}L} + c^2\omega_a \sum \lambda_{C_j}\sigma_{L_{C_j}L} \sum \lambda_{B_i}\sigma_{L_{B_i}L} + \\ & c^3\omega_b\sigma^2_L \sum \lambda_{A_p}\sigma_{L_{A_p}L} + c^2\omega_b \sum \lambda_{C_j}\sigma_{L_{C_j}L} \sum \lambda_{A_p}\sigma_{L_{A_p}L} \end{aligned}$$

By (3), (4), (5), (6) and (7), we have:

$$\sigma_{AC}\sigma_{BC} = \sigma^2_C\sigma_{AB} \Rightarrow \sigma_{AB} - \sigma_{AC}\sigma_{BC}(\sigma^2_C)^{-1} = 0 \Rightarrow \rho_{AB.C} = 0$$

Contradiction. Therefore, C cannot be an ancestor of D. By symmetry, neither are A and B.

Among A, B and C, at least one element is not an ancestor of the others since the graph is acyclic. By symmetry, assume without loss of generality that C is not an ancestor of A and B. Therefore, C is not an ancestor of any node in $\mathbf{O'}$, and by symmetry with D, we have that A and B cannot be ancestors of C.

Between A and B, one element cannot be an ancestor of the other since the graph is acyclic. By symmetry, assume without loss of generality that B is not an ancestor of A. Therefore, B is not an ancestor of any other one in $\mathbf{O'}$, and by symmetry with D, we have that A cannot be an ancestor of B. □

Lemma 3 *Let $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E}_\epsilon)$ be a linear latent variable graph. Let A, B, C and D be four elements in $\mathbf{O}$ such that no element in $\{A, B, C, D\}$ is an ancestor of any other element of this set in G, and A has a parent L in G, and no element of the covariance matrix over A, B, C and D is zero. If $\sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BC}$ is faithfully entailed by G, then $\sigma_{ACL} = \sigma_{ADL} = 0$ or $\sigma_{ACL}/\sigma_{ADL} = \sigma_{AC}/\sigma_{AD} = \sigma_{BC}/\sigma_{BD}$.*

Proof of Lemma 3: Since G is a linear latent variable graph, we can express A, B, C and D as linear functions of their parents as follows:

$$\begin{array}{rcl} A & = & aL + \sum_p a_p A_p \\ B & = & \sum_i b_i B_i \\ C & = & \sum_j c_j C_j \\ D & = & \sum_k d_k D_k \end{array}$$

where on the right-hand side of each equation the uppercase symbols denote the respective parents of each variable on the left side.

Given the assumptions, we have:

$$\begin{array}{lr} \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BC} & \Rightarrow \\ E[a\sum_j c_j L C_j + \sum_p \sum_j a_p c_j A_p C_j]\sigma_{BD} = E[a \sum_k d_k L D_k + \sum_p \sum_k a_p d_k A_p D_k]\sigma_{BC} & \Rightarrow \\ a(\sum_j c_j \sigma_{LC_j})\sigma_{BD} + \sum_p \sum_j a_p c_j \sigma_{A_pC_j}\sigma_{BD} = a(\sum_k d_k \sigma_{LD_k})\sigma_{BC} + \sum_p \sum_k a_p d_k \sigma_{A_pD_k}\sigma_{BC} & \Rightarrow \\ a[(\sum_j c_j \sigma_{LC_j})\sigma_{BD} - (\sum_k d_k \sigma_{LD_k})\sigma_{BC})] + [\sum_p \sum_j a_p c_j \sigma_{A_pC_j}\sigma_{BD} - \sum_p \sum_k a_p d_k \sigma_{A_pD_k}\sigma_{BC}] = 0 & \end{array}$$

Since no element in $\{A, B, C, D\}$ is an ancestor of another element of this set, then there is no trek among elements of $\{B, C, D\}$ containing both L and A, and therefore the symbol a cannot appear in $\sum_p \sum_j a_p c_j \sigma_{A_pC_j}\sigma_{BD} - \sum_p \sum_k a_p d_k \sigma_{A_pD_k}\sigma_{BC}$ when we expand each covariance as a function of the parameters of G. Therefore, since this polynomial is identically zero, we have to have the coefficient for a equal to zero, which implies:

$$a(\sum_j c_j \sigma_{LC_j})\sigma_{BD} = a(\sum_k d_k \sigma_{LD_k})\sigma_{BC} \Rightarrow \sigma_{ACL}\sigma_{BD} = \sigma_{ADL}\sigma_{BC}$$

Since no element in Σ_{ABCD} is zero, then $\sigma_{ACL} = 0 \Leftrightarrow \sigma_{ADL} = 0$. If $\sigma_{ACL} \neq 0$, then $\sigma_{ACL}/\sigma_{ADL} = \sigma_{AC}/\sigma_{AD} = \sigma_{BC}/\sigma_{BD}$. □

Lemma 4 *Let $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{G_S}, \mathbf{E}_\epsilon)$ be a linear latent variable graph and Σ the covariance matrix of $\mathbf{O}$ with two triplets $\mathbf{X} = \{X_1, X_2, X_3\} \subset \mathbf{O}, \mathbf{Y} = \{Y_1, Y_2, Y_3\} \subset \mathbf{O}, \mathbf{X} \cap \mathbf{Y} = \emptyset$, such that: (i) for every triplet $\{A, B, C\} \subset \mathbf{X} \cup \mathbf{Y}$, $\rho_{AB.C} \neq 0, \rho_{AB} \neq 0$ (ii) $\forall Y \in \mathbf{Y}, TetradScore(X_1, X_2, X_3, Y; \Sigma) = 3$; (iii) $\forall X \in \mathbf{X}, TetradScore(Y_1, Y_2, Y_3, X; \Sigma) = 3$. Then, if $\forall \{X_i, X_j\} \subset \mathbf{X}, \{Y_p, Y_q\} \subset \mathbf{Y}, \sigma_{X_iY_p}\sigma_{X_jY_q} = \sigma_{X_iY_q}\sigma_{X_jY_p} \neq \sigma_{X_iX_j}\sigma_{Y_pY_q}$, we have that $\forall X \in \mathbf{X}, Y \in \mathbf{Y}$, X and Y do not have a common parent in G.*

Proof of Lemma 4: Suppose X_1 and Y_1 have a common parent L in G. Let $X_1 = aL + \sum_p a_p A_p$ and $Y_1 = bL + \sum_i b_i B_i$, where each A_p, B_i are parents in G of X_1 and Y_1, respectively.

By Lemma 2 and the given tetrad constraints, for any pair of elements in $\mathbf{X} \cup \mathbf{Y}$, no element in this pair can be an ancestor of the other. By definition, $\sigma_{X_1VL} = (a/b)\sigma_{Y_1VL}$ for some element V, and therefore $\sigma_{X_1VL} = 0 \Leftrightarrow \sigma_{Y_1VL} = 0$. Since by assumption $\sigma_{X_1X_2}\sigma_{Y_1X_3} = \sigma_{X_1X_3}\sigma_{Y_1X_2}$ and $\sigma_{X_1Y_2}\sigma_{Y_1Y_3} = \sigma_{X_1Y_3}\sigma_{Y_1Y_2}$, then by Lemma 3 we have $\sigma_{X_1X_2L} = 0 \Leftrightarrow \sigma_{X_1X_3L} = 0 \Leftrightarrow \sigma_{Y_1X_2L} = 0 \Leftrightarrow \sigma_{Y_1X_3L} = 0$ and also $\sigma_{X_1Y_2L} = 0 \Leftrightarrow \sigma_{X_1Y_3L} = 0 \Leftrightarrow \sigma_{Y_1Y_2L} = 0 \Leftrightarrow \sigma_{Y_1Y_3L} = 0$.

By $\sigma_{X_1Y_1}\sigma_{X_2Y_2} = \sigma_{X_1Y_2}\sigma_{X_2Y_1}$, we have $\sigma_{X_1Y_1L} = 0 \Leftrightarrow \sigma_{X_1Y_2L} = 0$ and $\sigma_{X_1Y_1L} = 0 \Leftrightarrow \sigma_{X_2Y_1L} = 0$, which will imply $\sigma_{X_1Y_2L} = 0 \Leftrightarrow \sigma_{X_2Y_1L} = 0$.

Assume $\sigma_{X_1Y_2L} = \sigma_{Y_1X_2L} = 0$. Let $X_2 = \sum_j b_jC_j$ and $Y_2 = \sum_k d_kD_k$. Since by assumption $\sigma_{X_iY_q}\sigma_{X_jY_p}$ for all $p \in \{1,2,3\}, q \in \{1,2,3\}$, we have:

$$\begin{array}{rcl} \sigma_{X_1Y_1}\sigma_{X_2Y_2} & = & \sigma_{X_2Y_1}\sigma_{X_1Y_2} \\ (ab\sigma_L^2 + \sum_p\sum_i a_pb_i\sigma_{A_pB_i})\sigma_{X_2Y_2} & = & \sigma_{X_2Y_1}\sigma_{X_1Y_2} \end{array}$$

Again, this identity should hold for all values of a. Since a does not appear in any other term than $ab\sigma_L^2\sigma_{X_2Y_2}$ (because $\sigma_{X_1Y_2L} = \sigma_{Y_1X_2L} = 0$ and no element is an ancestor of the other), then $ab\sigma_L^2\sigma_{X_2Y_2} = 0$. But $a \neq 0, b \neq 0, \sigma_L^2 \neq 0, \sigma_{X_2Y_2} \neq 0$, which is a contradiction. A similar result will follow if we assume $\sigma_{X_1X_3L} = 0$. Therefore, assume no element in $\{\sigma_{X_1X_2L}, \sigma_{X_1X_3L}, \sigma_{Y_1X_2L}, \sigma_{Y_1X_3L}, \sigma_{X_1Y_2L}, \sigma_{X_1Y_3L}, \sigma_{Y_1Y_2L}, \sigma_{Y_1Y_3L}\}$ is zero.

From the assumed entailed constraint $\sigma_{Y_1X_2}\sigma_{Y_2Y_3} = \sigma_{Y_1Y_2}\sigma_{X_2Y_3}$ and Lemma 3, we have

$$\frac{\sigma_{Y_1X_2}}{\sigma_{Y_1Y_2}} = \frac{\sigma_{Y_1X_2L}}{\sigma_{Y_1Y_2L}}$$

From the assumed entailed constraint $\sigma_{X_1Y_2}\sigma_{X_2X_3} = \sigma_{X_1X_2}\sigma_{X_3Y_2}$ and Lemma 3, we have

$$\sigma_{X_1X_2} = \sigma_{X_1Y_2}\frac{\sigma_{X_1X_2L}}{\sigma_{X_1Y_2L}} = \sigma_{X_1Y_2}\frac{(b/a)\sigma_{Y_1X_2L}}{(b/a)\sigma_{Y_1Y_2L}} = \sigma_{X_1Y_2}\frac{\sigma_{Y_1X_2}}{\sigma_{Y_1Y_2}}$$

which implies

$$\sigma_{X_1X_2}\sigma_{Y_1Y_2} = \sigma_{X_1Y_2}\sigma_{Y_1X_2}$$

Contradiction. The result follows for all pairs in $\mathbf{X} \times \mathbf{Y}$ by symmetry. □

Lemma 5 *Let $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E}_\epsilon)$ be a linear latent variable graph. Let $\{A, B, C\} \subset \mathbf{O}$ be some triplet such that A and B have parents L_1 and L_2, respectively (where it is possible that $L_1 = L_2$), and C is not an ancestor of A or B. Then, if $\sigma_{L_1L_2} \neq 0$, it follows that $\rho_{XY.Z} \neq 0$.*

Proof of Lemma 5: Let the structural equations for A, B and C be $A = aL_1 + \sum_i a_iA_i + e_a$, $B = bL_2 + \sum_j b_jB_j + e_b$ and $C = \sum_k c_kC_k + e_c$, where e_a, e_b and e_c are independent random variables, and independent of every other random variable in G besides A, B and C, respectively.

We have that $\rho_{AB.C} \neq 0 \Leftrightarrow \sigma_{AB}\sigma_C^2 - \sigma_{AC}\sigma_{BC} \neq 0$. We will prove that $\sigma_{AB}\sigma_C^2 - \sigma_{AC}\sigma_{BC} \neq 0$. From the above equations, we have that $\sigma_{AB}\sigma_C^2 - \sigma_{AC}\sigma_{BC} = [ab\sigma_{L_1L_2} + F_1(A, B)](F_2(C) + \psi_c) - \sigma_{AC}\sigma_{BC}$, where no term in $F_1(A, B)$ can contain the product ab, every term in $F_2(C)$ contains some variable c_k as well as every term in $\sigma_{AC}\sigma_{BC}$, and ψ_c is the variance of e_c. The term $\sigma_{L_1L_2}$ cannot contain any variable c_k, since C is not an ancestor of A or B. Therefore, no term in this polynomial can cancel the term $ab\sigma_{L_1L_2}\psi_c$, and since $ab\sigma_{L_1L_2}\psi_c \neq 0$, it follows that $\rho_{AB.C} \neq 0$. □

Theorem 1 *Let $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E}_\epsilon, \mathbf{G_S})$ be the purifiable linear latent variable graph that generates the covariance matrix Σ of a set of observed random variables $\mathbf{O}$. Then, G will be in the measurement equivalence class $MM(\mathbf{O}, \Sigma)$, and such class will be given by the measurement pattern obtained throught* `FindMeasurementPattern`$(\mathbf{O}, \Sigma)$.

Proof of Theorem 1: Step 4 of the algorithm `FindMeasurementPattern` cannot remove edges among elements in the same cluster in G by Lemma 1. If two nodes O_i, O_j belong to some solution

$G_S \in \mathbf{G_S}$, there will be some set of four variables where all three tetrad constraints hold and by Lemma 5 there will be no vanishing partial correlations among such elements, so the edge between these variables cannot be turned into *Gray*, but *Blue* instead. Also, during our search for two extra nodes O_a, O_b, we do not need to consider those that are linked to each other or to O_i, O_j with a *Gray* edge, since we know by construction that not all three tetrad constraints can hold if one of these nodes is included in the set of four that is tested.

On Step 5, if for two nodes O_i, O_j there are four others O_a, O_b, O_c, O_d such that $Unclustered(\{O_a, O_b, O_i\}, \{O_c, O_d, O_j\}; \Sigma)$ holds, we know by Lemma 4 that O_i and O_j cannot be in the same cluster. Also, during our seach for four extra nodes O_a, O_b, O_c, O_d, we do not need to consider those linked by *Gray* or *Yellow* edges, since by construction they would not satisfy the conditions for the *Unclustered* predicate. Analogously, if two nodes O_i, O_j are pure indicators of a same cluster in some solution graph $G_S \in \mathbf{G_S}$, then there is a third node O_a in the same cluster, and three others O_b, O_c, O_d that are pure indicators of a different latent, and $Unclustered(\{O_i, O_j, O_a\}, \{O_b, O_c, O_d\}; \Sigma)$ holds by entailment, where all necessary conditions are guaranteed by Lemma 5 if those latents are correlated. Therefore, if two nodes are linked by a *Yellow* edge, they never appear together in some solution graph. Since only *Gray* and *Yellow* edges will be transformed to undirected edges among indicators in the output of the algorithm, therefore property 2 of measurement patterns will be satisfied.

Now, if we consider only the *Blue* edges of NG, we know from the previous paragraphs that any set of indicators of a same latent that appear in some purified solution graph $G_S \in \mathbf{G_S}$ will appear together in some connected component, since they will form a clique in NG_{Blue}. Only components of NG_{Blue} of size 1 will be removed, which implies that if some variable Y does not appear in any graph in **Clustering**, then Y cannot appear in any graph in $\mathbf{G_S}$. Therefore, property 1 of measurement patterns will be satisfied by our algorithm.

Given two latents T_1 and T_2, if both latents contain three distinct indicators and all six indicators appear in a same solution graph, then such latents will be linked at Step 4 of `BuildFinalPattern`. Now we have to prove that the last two properties of a measurement pattern are satisfied by the output of our algorithm. By the previous paragraphs, we know that for each triplet of variables that are indicators of a single latent in a pure graph $G_S \in \mathbf{G_S}$ there will be at least one cluster in the pattern graph containing them and there will be at least one cluster for each latent in G. Since indicators in different clusters under some $G_S \in \mathbf{G_S}$ cannot be linked by any edge at the end of the algorithm, they will not be in any common cluster in the pattern, and so there is at least one clique of size $|\mathbf{L}|$ among latent variables in the pattern $(\mathbf{O_p}, \mathbf{E_p})$ that satisfies property 4. Property 3 follows from the argument above.

Now suppose there is another clique $\mathbf{C_T^0}$ of size $|\mathbf{L}|$ among latents in the pattern $MM_G(\mathbf{O_p}, \mathbf{E_p})$ such that each $T_i \in \mathbf{C_T}$ has a subset of indicators $\mathbf{O_i}$ with the following properties: (i) $\forall i, |\mathbf{O_i}| \geq 3$; (ii) $\forall O_{ip} \in \mathbf{O_i}, O_{jq} \in \mathbf{O_j}, i \neq j$, O_{ip} and O_{jq} do not have any common latent parent in MM_G nor are linked by an undirected edge. Suppose the graph composed by $\mathbf{C_T^0}$ and their respective indicators in $\mathbf{O_1} \cup \mathbf{O_2} \cup ... \cup \mathbf{O}_{|\mathbf{L}|}$ does not satisfy property 4 because one of the latents in this clique, T_i, cannot be mapped to a latent in G, as defined by the function $L_G(\cdot)$ (see Section 3): $\mathbf{O_i}$ contains indicators of at least two different latents in $\mathbf{L}$. Because elements in different $\mathbf{O_i}$ sets do not share any latent parent and are not linked by an edge, that means they were not linked by any edge at the end of the clustering procedure, which can only happen if the *Unclustered* condition separates them. But then, by Lemma 4, no other of the $|\mathbf{L}| - 1$ sets $\mathbf{O_j}$, $j \in \{1, 2, ..., |\mathbf{L}|\} - i$ can contain any indicator of the same latents represented by $\mathbf{O_i}$, which means that we have indicators of at most $|\mathbf{L}| - 2$ different latents to distribute in the remaining $|\mathbf{L}| - 1$ sets. This means that at least one latent of the remaining $|\mathbf{L}| - 2$ ones will have to be a parent of indicators in at least two triplets

$\mathbf{O_p}$, $\mathbf{O_q}$, $p \neq q$, which is a contradiction.

Now suppose the same scenario from the previous paragraph holds, but property 4 fails because the mapping from $\mathbf{C_T}$ to $\mathbf{L}$ is not one-to-one, i.e., two elements T_1, T_2 of $\mathbf{C_T}$ are mapped to the same latent in $\mathbf{L}$. But then they could not possibly be separated, because they share a common latent parent in G.

Similarly, now suppose the maximum cliques among the latents in $(\mathbf{O_p}, \mathbf{E_p})$ that satisfies property 4 of the measurement pattern definition are of size $|\mathbf{L}| + k, k > 0$. Again, that leaves indicators of at most $|\mathbf{L}|$ latents to be distributed among $|\mathbf{L}| + k > |\mathbf{L}|$ clusters which is not possible unless a same latent is a parent of indicators in $\mathbf{O_p}$, $\mathbf{O_q}$, $p \neq q$, which is a contradiction. □

As a sidenote, a situation where the measurement pattern will have more than one maximum clique among estimated latents is when, for instance, a latent has two disjoint set of indicators, $\mathbf{O_1}$ and $\mathbf{O_2}$, both pure with respect to indicators of other latents and within each $\mathbf{O_i}$ set, but with the property that every element in $\mathbf{O_1}$ is impure with respect to every element in $\mathbf{O_2}$ (e.g., having correlated error terms). Such latent will be represented by two estimated latents in the pattern, but they will not be linked together.

Lemma 6 *Let $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E_\epsilon})$ be a linear latent variable graph. For any set $\mathbf{O'} = \{A, B, C, D\} \subseteq \mathbf{O}$, if G entails $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BD}$ and for every set $\{X, Y, Z\} \subset \mathbf{O'}$ we have $\rho_{XY.Z} \neq 0$ and $\rho_{XY} \neq 0$, and A and B have a common parent in G, then A and B cannot have any other common parent in G.*

Proof of Lemma 6: Assume L_1 and L_2 are two common parents of A and B in G. Let the graph G' have the same structure as G, but without all edges from other possible parents of A and B not in $\{L_1, L_2\}$. Since G' is more constrained than G, if a tetrad constraint holds in G, then it holds in G'. By Lemma 2, no element in $\mathbf{O'}$ is an ancestor of any other element in this set. Let the structural equations for A, B, C and D in G' be:

$$\begin{array}{rcl} A & = & \alpha_1 L_1 + \alpha_2 L_2 \\ B & = & \beta_1 L_2 + \beta_2 L_2 \\ C & = & \sum_j c_j C_j \\ D & = & \sum_k d_k D_k \end{array}$$

Since the tetrad constraint $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD}$ holds in G', we have $\sigma_{AB}\sigma_{CD} - \sigma_{AC}\sigma_{BD} = 0 \Rightarrow (\alpha_1\beta_1\sigma^2_{L_1} + \alpha_1\beta_2\sigma_{L_1L_2} + \alpha_2\beta_1\sigma_{L_1L_2} + \alpha_2\beta_2\sigma^2_{L_2})\sigma_{CD} - (\alpha_1 \sum_j c_j\sigma_{C_jL_1} + \alpha_2 \sum_j c_j\sigma_{C_jL_2})(\beta_1 \sum_k d_k\sigma_{D_kL_1} + \beta_2 \sum_k d_k\sigma_{D_kL_2}) = 0 \Rightarrow \alpha_1\beta_1(\sigma^2_{L_1}\sigma_{CD} - (\sum_j c_j\sigma_{C_jL_1})(\sum_k d_k\sigma_{D_kL_1})) + f(G) = 0$, where

$$f(G) = (\alpha_1\beta_2\sigma_{L_1L_2} + \alpha_2\beta_1\sigma_{L_1L_2} + \alpha_2\beta_2\sigma^2_{L_2})\sigma_{CD} - \alpha_2 \sum_j c_j\sigma_{C_jL_2}(\beta_1 \sum_k d_k\sigma_{D_kL_1} + \beta_2 \sum_k d_k\sigma_{D_kL_2})$$

When fully expanding $f(G)$ as a function of the linear parameters of G, the product $\alpha_1\beta_1$ cannot possibly appear, since no element in $\mathbf{O'}$ is an ancestor of any other element in this set, Therefore, since the polynomial constraint is identically zero and nothing in $f(G)$ can cancel the term $\alpha_1\beta_1$, we have:

$$\sigma^2_{L_1}\sigma_{CD} = \sum_j c_j\sigma_{C_jL_1} \sum_k d_k\sigma_{D_kL_1} \qquad (8)$$

Using a similar argument for the coefficients of $\alpha_1\beta_2$, $\alpha_2\beta_1$ and $\alpha_2\beta_2$, we get:

$$\sigma_{L_1L_2}\sigma_{CD} = \sum_j c_j\sigma_{C_jL_1}\sum_k d_k\sigma_{D_kL_2} \tag{9}$$

$$\sigma_{L_1L_2}\sigma_{CD} = \sum_j c_j\sigma_{C_jL_2}\sum_k d_k\sigma_{D_kL_1} \tag{10}$$

$$\sigma^2_{L_2}\sigma_{CD} = \sum_j c_j\sigma_{C_jL_2}\sum_k d_k\sigma_{D_kL_2} \tag{11}$$

From (8),(9), (10), (11), it follows:

$$\begin{array}{rcl}
\sigma_{AC}\sigma_{AD} & = & [\alpha_1\sum_j c_j\sigma_{C_jL_1} + \alpha_2\sum_j c_j\sigma_{C_jL_2}][\alpha_1\sum_k d_k\sigma_{D_kL_1} + \alpha_2\sum_k d_k\sigma_{D_kL_2}] \\
& = & \alpha_1^2\sum_j c_g\sigma_{C_jL_1}\sum_k d_k\sigma_{D_kL_1} + \alpha_1\alpha_2\sum_j c_j\sigma_{C_jL_1}\sum_k d_k\sigma_{D_kL_2} + \\
& & \alpha_1\alpha_2\sum_j c_j\sigma_{C_jL_2}\sum_k d_k\sigma_{D_kL_1} + \alpha_2^2\sum_j c_j\sigma_{C_jL_2}\sum_k d_k\sigma_{D_kL_2} \\
& = & [\alpha_1^2\sigma^2_{L_1} + 2\alpha_1\alpha_2\sigma_{L_1L_2} + \alpha_2^2\sigma^2_{L_2}]\sigma_{CD} \\
& = & \sigma^2_A\sigma_{CD}
\end{array}$$

which implies $\sigma_{CD} - \sigma_{AC}\sigma_{AD}(\sigma^2_A)^{-1} = 0 \Rightarrow \rho_{CD.A} = 0$. By Lemma 5, C and D have no correlated parents, which entails $\sigma_{CD} = 0$ in G'. Since all treks between C and D in G are preserved in G', that implies $\sigma_{CD} = 0$ is entailed by G. Contradiction. □

Theorem 2 *Let $G(\mathbf{L}, \mathbf{O}, \epsilon, \mathbf{E_L}, \mathbf{E_O}, \mathbf{E_\epsilon}, \mathbf{G_S})$ be the purifiable linear latent variable graph that faithfully generates the covariance matrix Σ of a set of observed random variables $\mathbf{O}$. Let MM_G be the measurement pattern corresponding to the equivalence class $MM(\mathbf{O}, \Sigma)$. Let $\mathbf{MM_{Pure}}$ be the set of all purifications of MM_G. Then $\mathbf{MM_{Pure}} =_{MM} \mathbf{G_S}$.*

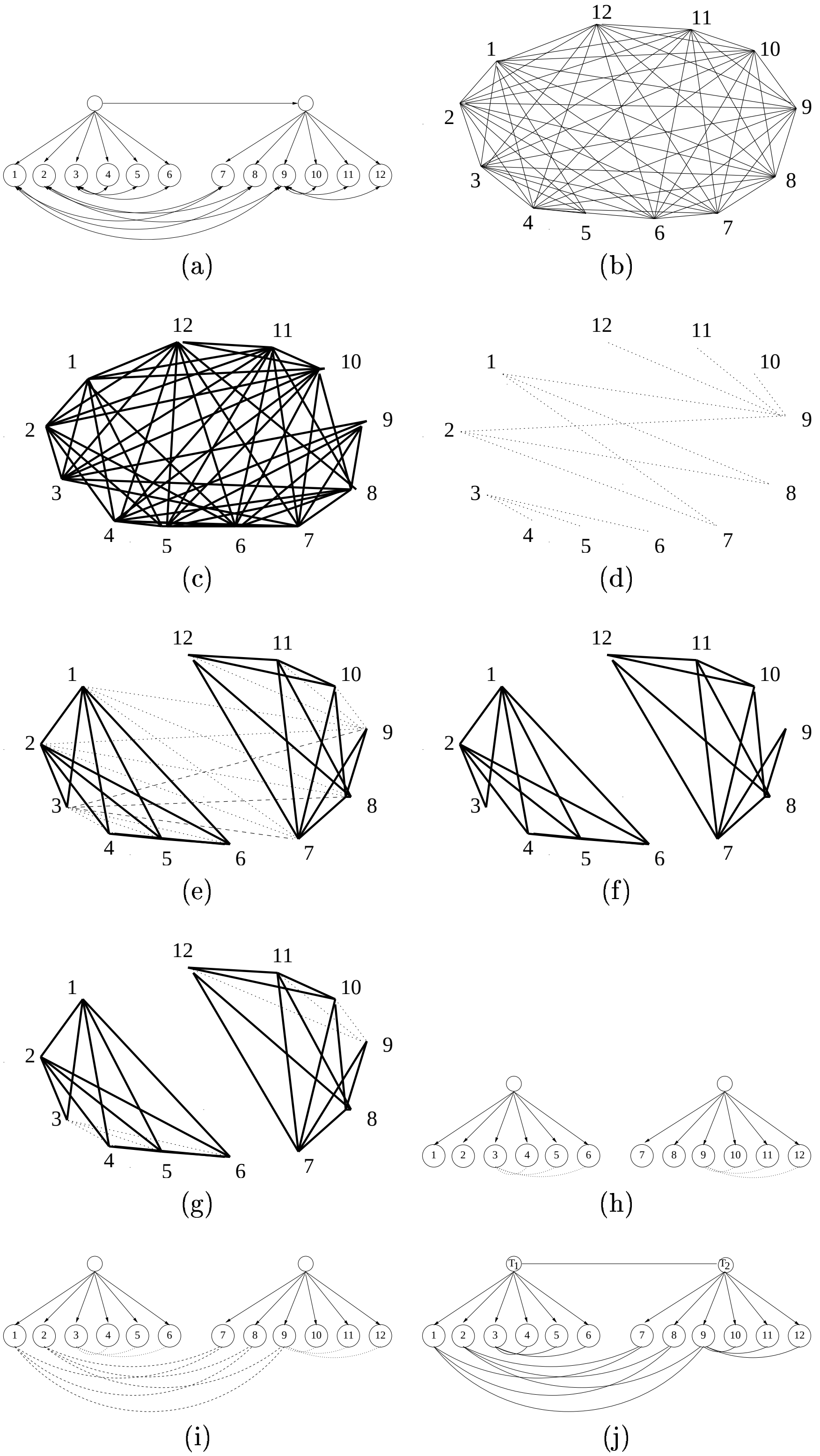
Proof of Theorem 2: $\mathbf{MM_{Pure}}$ is not empty, because by definition of measurement pattern there are at least three indicators per latent that can satisfy the requirements of purification. Let MM_{Pure} be some element of $\mathbf{MM_{Pure}}$. By the definition of measurement pattern and the definition of purification of a measurement pattern, MM_{Pure} has as many latents as any element in $\mathbf{G_S}$ and all indicators of a latent in MM_{Pure} are children of a same latent in G, and there is an one-to-one mapping from latents in MM_{Pure} to latents in $\mathbf{L}$.

No pair of indicators $\{O_x, O_y\}$ in MM_{Pure} are linked by an undirected edge in MM_G. That means that for each pair there exists another pair $\{O_a, O_b\} \subset \mathbf{O}$ such that $TetradScore(\{O_x, O_y, O_a, O_b\}; \Sigma) = 3$. By Lemma 2, no element in $\{O_x, O_y, O_a, O_b\}$ is an ancestor of another element in the same set. Therefore, no indicator in MM_{Pure} can be an ancestor of another indicator of MM_{Pure} in G. By Lemma 6, no two indicators in a same cluster in MM_{Pure} can have another common parent in G besides their common latent parent. By construction and Lemma 3, no two indicators in different clusters can have a common parent in G. Therefore there are no impurities in MM_{Pure}, and by mapping the estimated latents of this graph to the true latents in G, we have $MM_{Pure} =_{MM} G_S$ for some $G_S \in \mathbf{G_S}$.

Now, let G_S be some element of $\mathbf{G_S}$. By property 1 of measurement patterns, all indicators in $\mathbf{G_S}$ have to be in the measurement pattern MM_G. Let I_1 and I_2 be two indicators in G_S. By Lemma 4, if I_1 and I_2 are indicators of a same latent in G, then they have to be indicators of a same latent in MM_G. Suppose now they are indicators of different latents. Since they are in G_S, there are four other indicators $\{I_3, I_4, I_5, I_6\}$ such that $Unclustered(\{I_1, I_3, I_5\}, \{I_2, I_4, I_6\}; \Sigma)$ holds, and by the algorithm they cannot be under a same latent parent in MM_G. Also, there cannot be an undirected edge between I_1 and I_2 in MM_G because there exists another pair of indicators $\{I_3, I_4\}$ in G_S such that $TetradScore(\{I_1, I_2, I_3, I_4\}; \Sigma) = 3$. It follows that for each latent L in G_S, there is some latent T in MM_G such that $G_T(L) = T$. By construction of the algorithm, these latents form a clique in MM_G. So G_S is a subgraph of MM_G with $|\mathbf{L}|$ latents inducing subgraphs that satisfy Property 4 of measurement patterns, i.e., there is some $MM_{Pure} \in \mathbf{MM_{Pure}}$ such that $G_S =_{MM} MM_{Pure}$. □

Corollary 1 *For every possible pair of purifiable linear latent variable graphs* $G_1(\mathbf{L_1}, \mathbf{O}, \epsilon_\mathbf{1}, \mathbf{E_{L_1}}, \mathbf{E_{O_1}}, \mathbf{E_{\epsilon_1}}, \mathbf{G_{S_1}})$ *and* $G_2(\mathbf{L_2}, \mathbf{O}, \epsilon_\mathbf{2}, \mathbf{E_{L_2}}, \mathbf{E_{O_2}}, \mathbf{E_{\epsilon_2}}, \mathbf{G_{S_2}})$ *faithfully generating* Σ*, the covariance matrix of* $\mathbf{O}$*, we have* $\mathbf{G_{S_1}} =_{MM} \mathbf{G_{S_2}}$.

Proof of Corollary 1: By Theorem 1, both graphs fall under the same equivalence class $MM(\mathbf{O}, \Sigma)$, since they provide the same input for the algorithm `FindPattern`. The result follows immediately from Theorem 2. □



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 11: A step-by-step demonstration of how the graph in Figure (a) will give rise to the measurement pattern in Figure (j).

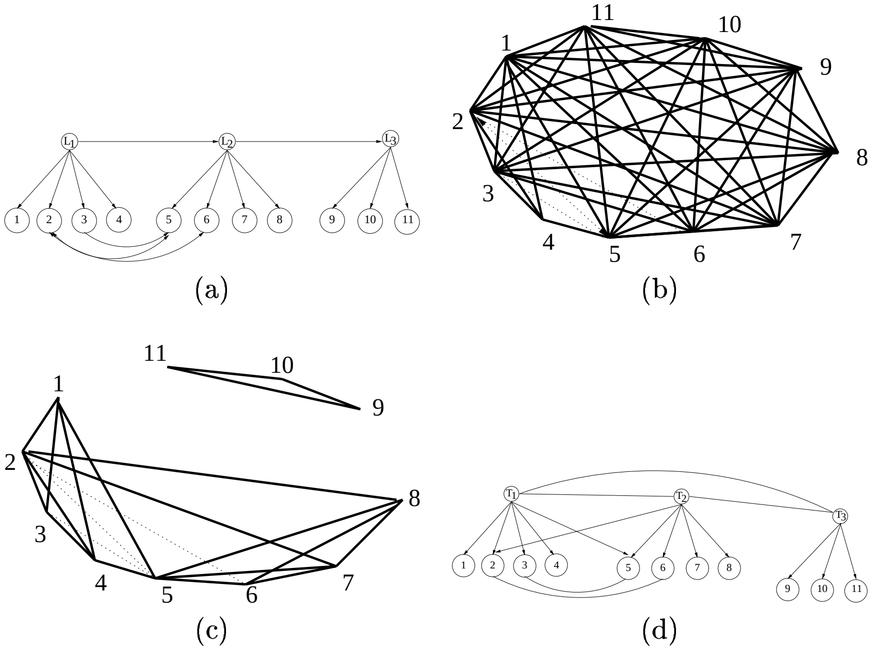


Figure 12: Another example where two of the measures are positioned in two clusters at the same time.

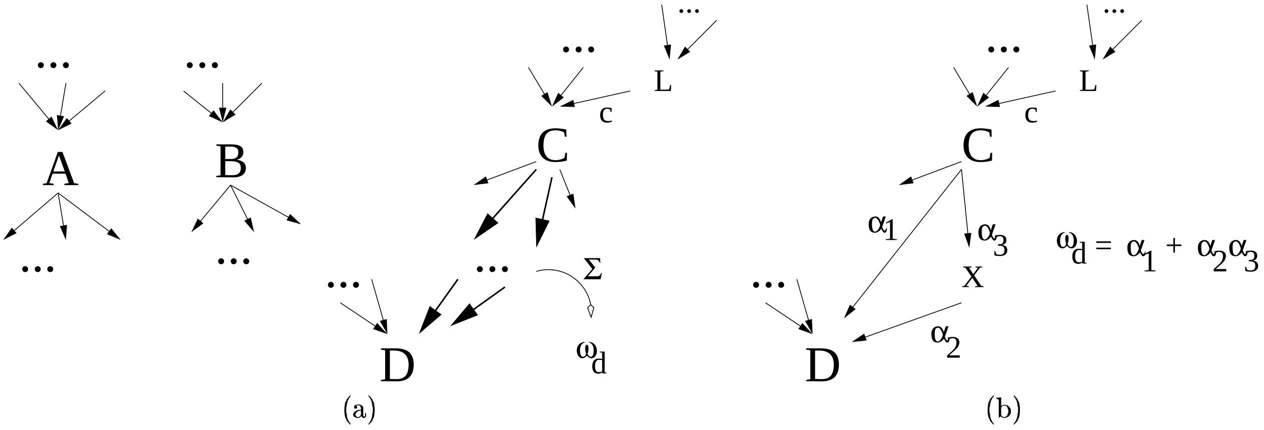


Figure 13: (a) The symbol ω_d is defined as the sum of the product of the labels of each edge that appears in a directed path from C to D. Here the bold edges represent edges in such directed paths. (b) An example: we have two directed paths from C to D. The symbol ω_d then stands for $\alpha_1 + \alpha_2\alpha_3$, where each term in this polynomial corresponds to one directed path. Notice that it is not possible to obtain any additive term that forms ω_d out of the product of some $\lambda_{A_p}, \lambda_{B_i}, \lambda_{C_j}$, since D is not an ancestor of any of them: in our example, α_1 and α_2 cannot appear in any $\lambda_{A_p}\lambda_{B_i}\lambda_{C_j}$ product (α_3 may appear if X is an ancestor of A or B).

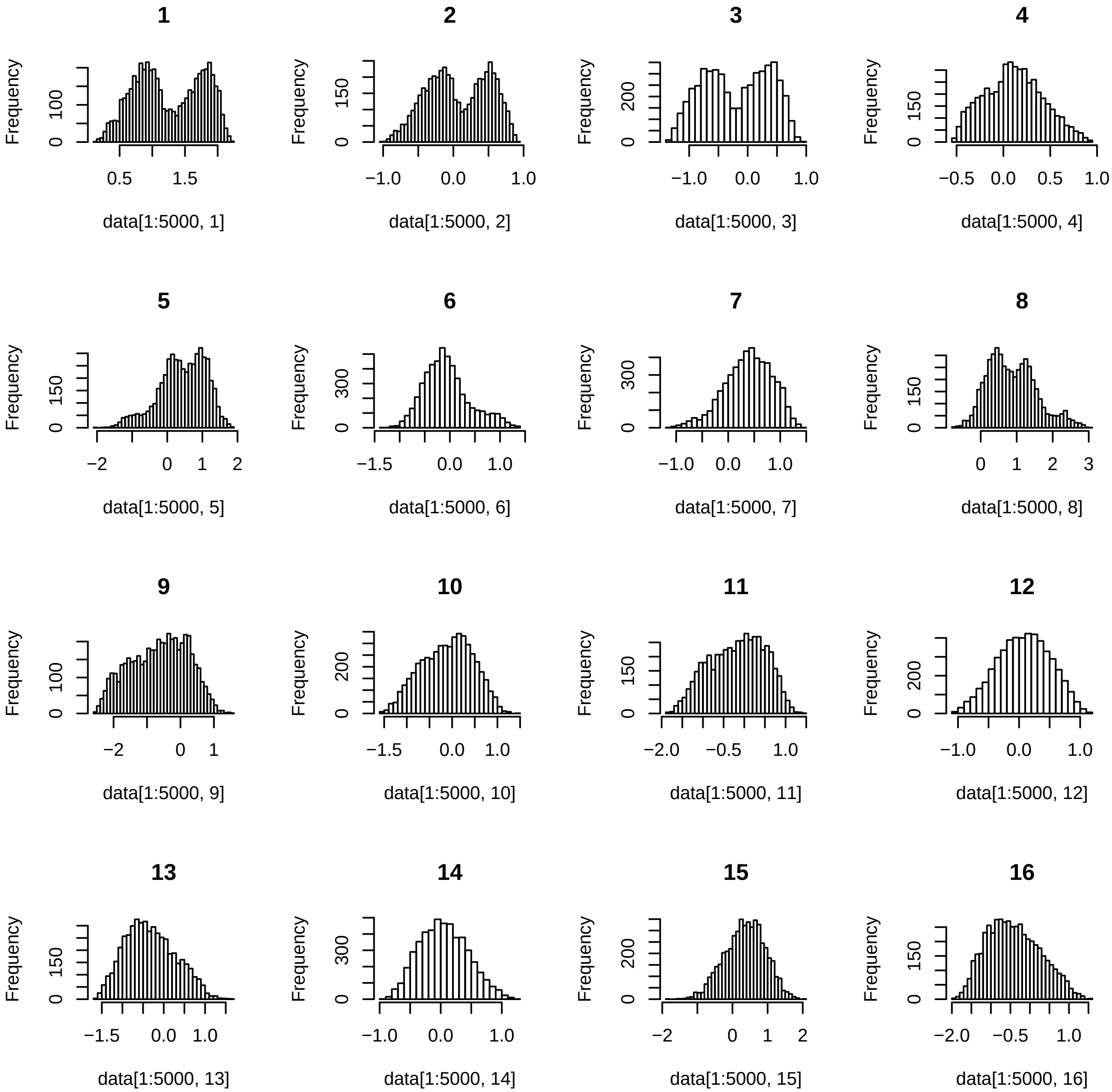


Figure 14: Univariate histograms for each of the 16 variables (organized by row) from a data set of 5000 observations sampled from the graph in Figure 9. 30 bins were used.