Automatic Discovery of Latent Variable Models

Ricardo Silva
August 2005
CMU-CALD-05-109

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

Thesis Committee:
Richard Scheines, CMU (Chair)
Clark Glymour, CMU
Tom Mitchell, CMU
Greg Cooper, University of Pittsburgh

This work was partially supported by NASA under Grants No. NCC2-1377, NCC2-1295 and NCC2-1227 to the Institute for Human and Machine Cognition, University of West Florida. This research was also supported by a Siebel Scholarship and a Microsoft Fellowship.

Keywords: graphical models, causality, latent variables

Abstract

Much of our understanding of Nature comes from theories about unobservable entities. Identifying which hidden variables exist given measurements in the observable world is therefore an important step in the process of discovery. Such an enterprise is only possible if the existence of latent factors constrains how the observable world can behave. We do not speak of atoms, genes and antibodies because we see them, but because they indirectly explain observable phenomena in a unique way under generally accepted assumptions.

How to formalize the process of discovering latent variables and models associated with them is the goal of this thesis. More than finding a good probabilistic model that fits the data well, we describe how, in some situations, we can identify causal features common to all models that equally explain the data. Such common features describe causal relations among observed and hidden variables. Although this goal might seem ambitious, it is a natural extension of several years of work in discovering causal models from observational data through the use of graphical models. Learning causal relations without experiments basically amounts to discovering an unobservable fact (does A cause B?) from observable measurements (the joint distribution of a set of variables that include A and B). We take this idea one step further by discovering which hidden variables exist to begin with.

More specifically, we describe algorithms for learning causal latent variable models when observed variables are noisy linear measurements of unobservable entities, without postulating a priori which latents might exist. Most of the thesis concerns how to identify latents by describing which observed variables are their respective measurements. In some situations, we will also assume that latents are linearly dependent, and in this case causal relations among latents can be partially identified. While continuous variables are the main focus of the thesis, we also describe how to adapt this idea to the case where observed variables are ordinal or binary.

Finally, we examine density estimation, where knowing causal relations or the true model behind a data generating process is not necessary. However, we illustrate how ideas developed in causal discovery can help the design of algorithms for multivariate density estimation.

Acknowledgements

Everything passed so fast during my years at CMU, and yet there are so many people to thank. Richard Scheines and Clark Glymour are outstanding tutors. I think I will never again have meetings as challenging and as fun as those that we had. I am also very much in debt to Peter Spirtes, Jiji Zhang and Teddy Seidenfeld for providing a help handing whenever necessary, as well as to my thesis committee members, Tom Mitchell and Greg Cooper. Diane Stidle was also essential to guarantee that everything was on the right track, and CALD would not be the same without her.

It was a great pleasure to be part of CALD on its first years. Deepayan Chakrabarti and Anna Goldenberg have been with me since Day 1, and they know what it means, and how important they were to me in all these years. Many other CALDlings were with us in many occasions: the escapades for food in South Side with Rande Shern and Deepay; the annual Super Bowl parties at Bubba Beasley's and foosball at Daniel Wilson's; the always ready-for-everything CALD KREM: Krishna Kumaraswamy, Elena Eneva, Matteo Matteucci and myself (too bad I broke the pattern of repeated initials. Think of me as the noise term) − these guys could party even during a black-out; Pippin Whitaker, perpetrator of the remarkable feat of convincing me to go to the gym at 5 a.m. (I still don't know how I was able to wake up and find the way to the gym by myself). On top of that, Edoardo Airoldi and Xue Bai were masters of organizing a good CALD weekend, preferably with the company of Leonid Teverovskiy, Jason Ernst and Pradeep Ravikumar; Xue gets additional points for being able to drag me to salsa classes (with the help of Lea Kissner and Chris Colohan); Francisco Pereira is not quite from CALD, but he is not in these acknowledgements just because of his healthy habit of bringing me some fantastic Porto wine straight from the source (yes, I got one for my defense too); and one cannot forget the honorary CALDlings Martin Zinkevich and Shobha Venkataraman.

Josué, Simone and Clara Ramos were fantastic hosts, who made me feel at home when I was just a newcomer. Whenever you show up in my homecity, make sure to knock at my door. It will feel like the days in Pittsburgh, snow not included.

I owe a lot to Einat Minkov, including some of my sweetest memories of Pittsburgh. Will I ever repay for everything? I won't stop trying.

To conclude, it goes without saying that my parents and brother were an essential support on every step of my life. But let me say it anyway: thank you for everything. This thesis is dedicated to you.

Contents

Chapter 1

Introduction

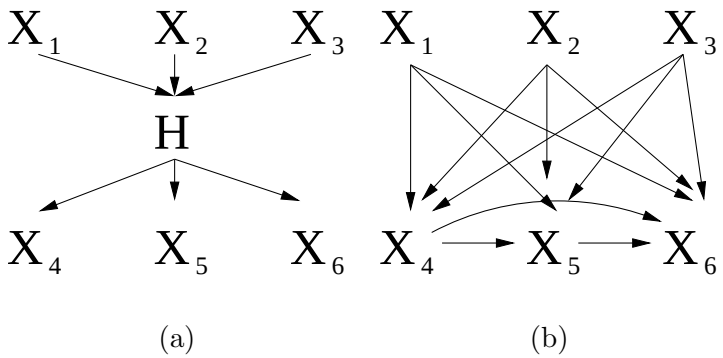
Latent variables, also called *hidden variables*, are variables that are not observed. Concepts such as gravitational fields, subatomic particles, antibodies or "economical stability" are essential building blocks of models of great practical impact, and yet such entities are unobservable (Klee, 1996). Sometimes there is overwhelming evidence that hidden variables are actual physical entities, e.g., quarks, and sometimes they are useful abstractions, e.g., "psychological stress."

Often the goal of statistical analysis with latent variables is to reduce the dimensionality of the data. Although in many instances this is a practical necessity, it is a goal that is sometimes in tension with discovering the truth, especially when the truth concerns the causal relations among latent variables. For instance, there are several methods that accomplish effective dimensionality reduction by assuming that the latents under study are independent. Because full independence among random variables is a very strong assumption, models resulting from such methods might not have any correspondence to real causal mechanisms, even if such models fit the data reasonably well.

When there is uncertainty about the number of latent variables, which variables measure them, or which measured variables influence other measured variables, the investigator who aims at a causal explanation is faced with a difficult discovery problem for which currently available methods are at best heuristic. Loehlin (2004) argues that while there are several approaches to automatically learn causal structure (Glymour and Cooper, 1999), none can be seen as competitors of exploratory factor analysis: the usual focus of automated search procedures for causal Bayes nets is on relations among observed variables. Loehlin's comment overlooks Bayes net search procedures robust to the presence latent variables (Spirtes et al., 2000), but the general sense of his comment is correct.

The main goal of this thesis is to fill this gap by formulating algorithms for discovering latent variables that are hidden common causes of a given set of observed variables. Furthermore, we provide strategies for discovering causal relations among the hidden variables themselves. In applications as different as gene expression analysis and marketing, knowing how latents causally interact with the given observed measures and among themselves is essential. This is a question that has been hardly addressed. The common view is that solving this problem is actually impossible, as illustrated by the closing words of a popular textbook on latent variable modeling (Bartholomew and Knott, 1999):

> When we come to models for relationships between latent variables we have reached a point where so much has to be assumed that one might justly conclude that the limits of scientific usefulness have been reached if not exceeded.



(a) (b)

Figure 1.1: An illustration on how the existence of an unrecorded variable can affect a probabilistic model. Figure (b) represents the remaining set of conditional independencies that still exist after removing node H from Figure (a). This figure is adapted from (Binder et al., 1997).

This view is a consequence of formulating the problem of discovering latent variables by using arbitrary methods such as factor analysis, which can generate an infinite number of solutions. Identifiability in this case is treated as a mere case of "interpretation," where all solutions are acceptable, and the preferred ones are just those that are "easier" to interpret. This thesis should be seen as a case against this type of badly formulated approach, and a counter-example to Bartholomew and Knott's statement.

This introduction will explain the general approach for latent variable modeling and causal modeling adopted in this thesis. We first discuss how latent variables are important (Section 1.1), especially in causal models. We then define the scope of the thesis (Section 1.2). Details about causal models are introduced in Section 1.3. In the end of this chapter we provide a thesis outline.

1.1 On the necessity of latent variable models

Consider first the problem of density estimation using the graphical modeling framework (Jordan, 1998). In this framework, one represents joint probability distributions by imposing several conditional independence constraints on the joint, where such constraints are represented by a graph. Assume that we have a distribution that respects the independence constraints represented by Figure 1.1(a). If for some reason variable H is unrecorded in our database and we want to reconstruct the marginal probability distribution of the remaining variables, the simplest graph we can use has at least as many edges as the one depicted in Figure 1.1(b). This graph is relatively dense, which can lead to computationally expensive inferences and statistically inefficient estimation of probabilities. If instead we use the latent variable model of Figure 1.1(a), we can obtain more efficient estimators using standard techniques such as maximum likelihood estimation by gradient descent (Binder et al., 1997). That is, even if we do not have data for particular variables, it is still the case that a latent variable model might provide more reliable information about the observable marginal than a model without latents.

In the given example, the hidden variable was postulated as being part of a true model. Sometimes a probabilistic model contains hidden variables not because such variables represent some physical entity, but because it adds bias to a model in order to reduce the variance of the estimator.

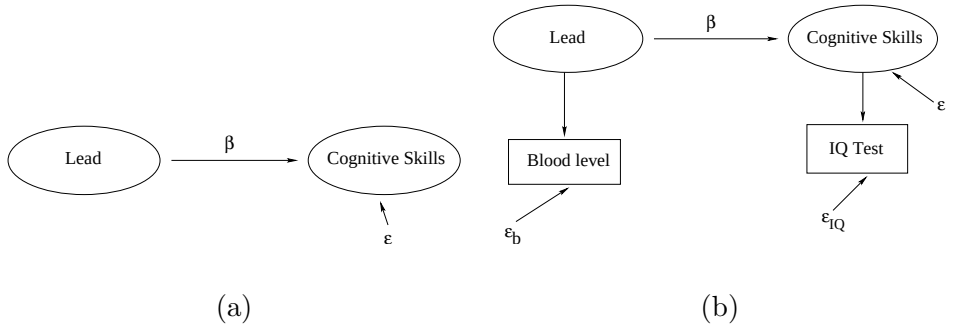


Figure 1.2: In (a), the underlying hypothetized phenomenon. In (b), how the model assumptions relates the measurements.

Even if such model does not correspond to a physical reality, it can aid predictions when data is limited.

However, suppose we are interested not only in good estimates of a joint distribution as the ultimate goal, but on the actual causal structure underlying the joint distribution. Consider first the scenario where we are *given* a set of latent variables. The problem is how to find the correct graphical structure representing the causal connections between latent variables, and between any pair of latent and observed variables.

For example, suppose there is an observed association between exposure to lead and low IQ. Suppose this association is because exposure to lead causes changes in a child's IQ. Policy makers are interested in this type of problem because they need to control the environment in order to achieve a desired effect: 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? These are typical questions in econometrics and social science. But also researchers in artificial intelligence and robotics are attentive to such general problems: how can a robot *intervene* on its environment in order to achieve its goals? If one does not know how to quantify such effects, one cannot build any sound decision theoretic machinery for action, since the prediction of the effects of a manipulation will be wrong to begin with. In order to perform sound prediction of manipulations, causal knowledge is necessary, and algorithms are necessary to learn it from data (Spirtes et al., 2000; Pearl, 2000; Glymour and Cooper, 1999).

A simple causal model for the lead (L) and cognitive skills (C) problem is a linear regression model $C = \beta L + \epsilon$, where ϵ is the usual zero-mean, normally distributed random variable, and the model is interpreted as causal. Figure 1.2(a) illustrates this equation as a graphical model. There is one important problem: how to quantify "lead exposure" and "cognitive skills". The common practice is to rely on indirect measures (*indicators*), such as *Blood level concentration* (of lead) (BL), which is an indicator of lead exposure. In our hypothetical example, BL cannot directly substitute for L in this causal analysis because of *measurement error* (Bollen, 1989), i.e., a significant concentration of lead in someone's blood might not be real, but an artifact of the physical properties of the instruments used in this measurement. Concerning variable C, intelligence itself is probably one of the most ill-defined concepts in existence (Bartholomew, 2004). Measures such as *IQ Tests* (IQ) have to be used as indicators of C. Expressing our regression model directly in terms of observable variables, we obtain $IQ = \beta BL + \epsilon_{IQ}$.

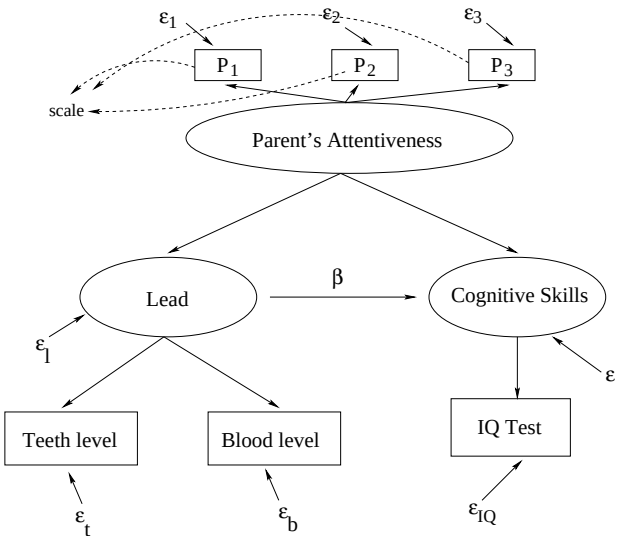


Figure 1.3: A graphical model with three latents. Variable *scale* is a deterministic function of its parents, represented by dashed edges.

However, if the variance of the measurement error of L through BL is not zero, i.e., $E[\epsilon_b^2] \neq 0$, we cannot get a consistent estimator of β by just regressing IQ on BL. This is not because regression is fundamentally flawed, but because this problem fails to meet its assumptions. By Figure 1.2(b), we see that there is a common cause between BL and IQ (*Lead*), which violates an assumptions of regression: if one wants consistent estimators of causal effects, there cannot be any hidden common cause between the regressor and the predictor.

One solution is fully modeling the latent structure. Additional difficulties arise in latent variable models, however. For instance, the model in Figure 1.2(b) is not identified, i.e., the actual parameters that can be used to quantify the causal effect of interested cannot be calculated. This can be solved by using multiple indicators per latent (Bollen, 1989).

Consider the problem of identifying conditional independencies *among latents*. This is an essential pre-requisite in data-driven approaches for causal discovery (Spirtes et al., 2000; Pearl, 2000; Glymour and Cooper, 1999). 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 1.3). 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 could be used.

However, this is not the case. A common practice is to create proxies for such latent variables, and to perform tests with the proxies. For instance, a typical proxy is the average of the respective indicators of the hidden variable of interest. An average of P_1, P_2 and P_3 is a *scale* for P, and scales for L and C can be similarly constructed. In general, however, a scale does not capture all of the variability of the respective hidden variable, and no conditional independence will hold given this scale. Measurement error is responsible for such a difference[1]. Assuming the model is linear,

[1]Using a graphical criterion for independence known as d-separation (Pearl, 1988), one can easily verify that the indicators of L and C cannot be independent of a function of the children on P, unless this function is deterministic

this problem can be solved by fitting the latent variable model and evaluating if the coefficient β parameterizing the edge of interest is zero.

So far, we described a problem where latent variables were given in advance. An even more fundamental problem is discovering *which* latents exist. A solution to this problem can also be indirectly applied to the task of multivariate density estimation. This is one of the most difficult problems in machine learning and statistics, since in general a joint distribution can be generated by an infinite number of different latent variable models. However, under an appropriate set of assumptions, the existence of latents can sometimes be indirectly identified from testable features of the marginal of the observed variables.

The scientific problem is therefore a problem of learning how our observations are causally connected. Since it is often the case that such connections happen through hidden common causes, the scientist has to first infer which relevant latent variables exist. Only then he or she can proceed to identify how such hidden variables are causally connected by examining conditional independencies among latents that can be detected in the observed data. An automatic procedure to aid this accomplishment this is the main contribution of this thesis.

1.2 Thesis scope

Given the large number of reasons for the importance of latent variable models, we describe here a simplified categorization of tasks and which ones are relevant to this thesis:

- causal inference. This is the main motivation of this thesis, and it is described in more detail in the next sections;
- density estimation. This is a secondary goal, achieved as a by-product of the thesis's main results. We evaluate empirically how variations of our methods perform in this task;
- latent prediction. Sometimes predicting the values of the latents themselves is the goal of the analysis. For instance, in independent component analysis (ICA) (Hyvarinen, 1999) the latents are signals that have to be recovered. In educational testing, latents represent the abilities of an individual. Mathematical and verbal abilities in an exam such as GRE, for instance, can be treated as latent variables, and individuals are ranked according to their predicted values (Junker and Sijtsma, 2001). Similarly, in model-based clustering the latent space can be used to group individuals: the modes of the latent posterior distribution can be used to represent different market groups, for instance. We do not evaluate our methods in the latent prediction task, but our results might be useful in some domains;
- dimensionality reduction. Sometimes a latent space can be used to perform lossy compression of the observable data. For instance, Bishop (1998) describes an application in image compressing using latent variable models. This is an example of an application where the main theoretical results of this thesis are unlikely to be useful;

Within these tasks, there are different classes of problems. In some, for example, the observed variables are basically measurements of some physical or social process. If, for example, we take dozens of measures of the incident light hitting the surface of the earth, some at ultra-violet wavelengths, some at infra-red, etc., then it is reasonable to assume that such observed variables are

on P and invertible.

measurements of a set of *unrecorded* physical variables, such as atmospherical and solar processes. The pixels that compose fMRI images are indirect measurements of the chemical and electrical processes in human brains. Educational tests intend to measure abstract latent abilities of students, such as verbal and mathematical skills. Questionnaires used in social studies are intended to analyse latent features of the population of interest, such as the "attitude" of single mothers with respect to their children. In all these problems, it is also reasonable to assume that observed variables are *indicators* of latents of interest, and therefore they are effects, not causes, of latents. This type of data generating process is the focus of this thesis.

Moreover, because measures are massively connected by hidden common causes, it is unlikely that conditional independencies hold among such measures unless such independencies are loosely approximated, e.g., in cases where measures are nearly perfectly correlated with the latents. It would be extremely useful to have a machine learning procedure that might discover which latent common causes of such measures were operative, and do so in a way that allowed for discovering something about how they were related, especially causally. But for that one cannot rely only on observed conditional independencies. New techniques for causality discovery that do not directly rely on observed independence constraints is the focus of this thesis.

1.3 Causal models, observational studies and graphical models

In this section we make more precise what we mean by causal modeling and how it is different from non-causal modeling. There are two basic types of prediction problems: *prediction under observation* and *prediction under manipulation*. In the first type, given an observation of the current state of the world, an agent infers the probability distribution of a set of variables conditioned on this observation. For instance, predicting the probability of rain given the measure of a barometer is such a prediction problem.

The second type consists in predicting the effect of a *manipulation* on a set of variables. A manipulation consists on a modification of the probability distribution of a set of variables in a given system by an agent outside the system. For instance, it is known that some specific range of atmospherical pressure is a good indication of rain. A barometer measures atmospherical pressure. If one wants to make rain, why not intervene on a barometer by modifying its sensors? If the probability of rain is high for a given measure, then providing such a measure might appear as a good idea.

The important difference between the two types of prediction is intuitive. If the intervention on our barometer consists on attaching a random number generator in place of the actual physical sensors, we do not expect the barometer to affect the probability of rain, even if the resulting measure is a strong indication of rain under proper physical conditions. We know this because we know that rain causes changes in the barometer reading, not the opposite. A causal model is therefore essential to predict the effects of an intervention.

The standard method of estimating a causal model is by performing experiments. Different manipulations are assigned to different samples following a distribution that is independent of the possible causal mechanisms of interest (uniformly random assignments are a common practice). The different effects are then estimated using standard statistical techniques. Double-blinded treatments in the medical literature are a classical example of experimental design (Rosebaum, 2002).

However, experiments might not be possible for several reasons: they can be unethical (as in estimating the effects of smoking in lung cancer), too expensive (as in manipulating a large

number of sets of genes, one set at a time), or simply technologically impossible (as in several subatomic physics problems). Instead, one must rely on *observational studies*, which attempt to obtain estimates of causal effects from *observational data*, i.e., data representative of the population of interest, but obtained with no manipulations. This can only be accomplished by adopting extra assumptions that link the population joint distribution to causal mechanisms.

An account of classical techniques of observational studies is given by Rosebaum (2002). In most cases, the direction of causality is given a priori. The goal is estimating the causal effect of a variable X on a variable Y, i.e., how Y varies given different manipulated values of X. One tries to measure as many possible common causes between the X and Y in order to estimate the desired effect, since the presence of hidden common causes will result in biased estimates.

Much background knowledge is required in these methods and, if incorrect, can severely affect one's conclusions. For instance, if Z is actually a *common effect* of X and Y, conditioning on Z adds bias to the estimate of the desired effect, instead of removing it.

Instead, this thesis advocates the framework of data-driven *causal graphical models*, or *causal Bayesian networks*, as described by Spirtes et al. (2000) and Pearl (2000). Such models not only encompass a wide variety of models used ubiquitously in social sciences, statistics, and economics, but they are uniquely well suited for computing the effects of interventions.

We still need to adopt assumptions relating causality and joint probabilities. However, such assumptions rely on a fairly general axiomatic calculus of causality instead of being strongly domain dependent. The fundamental property of this calculus is assuming that qualitative features of a true causal model can be represented by a graph. We will focus mostly on directed acyclic graphs (DAGs), so any reference to a graph in this thesis is an implicit reference to a DAG, unless otherwise specified. There are, however, extensions of this calculus to cyclic graphs and other types of graphs (Spirtes et al., 2000).

Each random variable is a node in the corresponding graph, and there is an edge $X \rightarrow Y$ in the graph if and only if X is a *direct cause* of Y, i.e., the effect of X on Y when X is manipulated is non-zero when conditioning on all other causes of Y. Notice that causality itself is not defined. Instead we rely on the concepts of manipulation and effect, which are causal concepts themselves, to provide a calculus to solve the practical problems of causal prediction.

Two essential definitions are at the core of the graphical causal framework:

Definition 1.1 (Causal Markov Condition) *Any given variable is independent of its non-effects given its direct causes.*

Definition 1.2 (Faithfulness Condition) *A conditional independence holds in the joint distribution if and only if it is entailed by the Causal Markov condition in the corresponding graph.*

The only difference between the causal and the "non-causal" Markov conditions is that in the former a parent is assumed to be a direct cause. The non-causal Markov condition is widely used in graphical probabilistic modeling (Jordan, 1998). For DAGs, *d-separation* is a sound and complete system to deduce the conditional independencies entailed by the Markov condition (Pearl, 1988), which in principle can be used to verify if a probability distribution is faithful to a given DAG. We will use the concept of d-separation in several points of this thesis as as synonym for conditional independence. The faithfulness condition is also called "stability" by Pearl (2000). Spirtes et al. (2000) and Pearl (2000) discuss the implications and suitability of such assumptions.

Why does the faithfulness condition help us to learn causal models from observational data? The Markov condition applied to different DAGs entails different sets of conditional independence constraints. Such constraints can in principle be detected from data. If constraints in the distribution are allowed to be arbitrarily disconnected from the underlying causal graph, then any probability distribution can be generated from a fully connected graph, and nothing can be learned. This, however, requires that independencies are generated by cancellation of causal paths. For instance, if variables X and Y are probabilistically independent, but X and Y are causally connected, then all causal paths between X and Y cancel each other, amounting to zero association. Our axioms deem such an event impossible, and in fact this assumption seems to be very common in observational studies (Spirtes et al., 2000), even though in many cases it is not explicit. We make it explicit.

Therefore, a set of independencies observed in a joint probability distribution can highly constrain the possible set of graphs that generated such independencies. It might be the case that all compatible graphs agree on specific edges, allowing one to create algorithms to identify such edges. Section 1.4 gives more details about discovery algorithms in the context of this thesis.

It is important to stress that a causal graph is not a full causal model. A graph only indicates which conditional independencies exist, i.e., it is an *independence model*, not a probabilistic model as required to compute causal effects. A full causal model should also describe the joint probability distribution of its variables. Most graphical models used in practice are parametric, and defined by local functions: the conditional density of a variable given its parents. In this thesis we will adopt parametric formulations, mostly multivariate Gaussians or finite mixtures of multivariate Gaussians.

Once parametric formulations are introduced, *other types of constraints are entailed by parameterized causal graphs*. That is, given a causal graph with a respective parameterization, some constraints on the joint distribution will hold for *any* choice of parameter values. One can adopt a different form of faithfulness on which such non-independence constraints observed in the joint distribution are a result of the underlying causal graph, reducing the set of possible graphs compatible with the data. This will be essential in the automatic discovery of latent variable models, as explained in Section 1.5.

1.4 Learning causal structure

Suppose one is given a joint distribution of two variables, X and Y, which are known to be dependent. Both graphs $X \leftarrow Y$ and $X \rightarrow Y$ are compatible with this observation (plus an infinite number of graphs where an arbitrary number of hidden common causes of X and Y exist). In this case, the causal relationship of X and Y is not identifiable from conditional independencies. However, with three or more variables, several sets of conditional independence constraints uniquely identify the directionality of some edges.

Consider Figure 1.4(a), where variables H_i are possible hidden variables. If hidden variables are assumed to not exist, then the directed edges $X \rightarrow Z$ and $Y \rightarrow Z$ can be identified from data generated by this model. If hidden variables are not discarded a priori, one can still learn that Z is not a cause of either X or Y. If the true model is the one shown in Figure 1.4(b), in the large sample limit it is possible to determine that Z is a cause of W under the faithfulness assumption, even if one allows the possibility of hidden common causes.

In general, we do not have enough information to identify all features of the true causal graph without experiments. The problem of causal discovery without experimental data should be for-

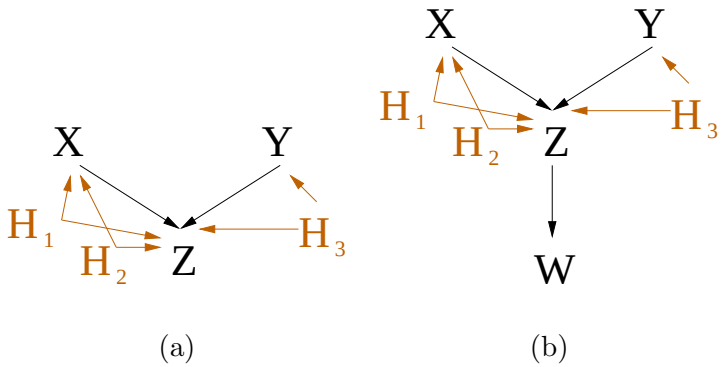


Figure 1.4: Two examples of graphs where the directionality of the edges can be inferred.

mulated as a problem of finding *equivalence classes* of graphs. That is, instead of learning a causal graph, we learn a set of graphs that cannot be distinguished given the observations. This set forms the equivalence class of the given set of observed constraints. The most common equivalence class of causal models is defined by conditional independencies:

Definition 1.3 (Markov equivalence class) *The set of graphs that entail exactly the same conditional independencies by the Markov condition.*

Enumerating all members of an equivalence class might be unfeasible, because in the worst case this number is exponential in the number of nodes in the graph. Fortunately, there are compact representations for Markov equivalence classes. For instance, a *pattern* (Pearl, 2000; Spirtes et al., 2000) is a representation for Markov equivalence classes of DAGs when no pair of nodes has a hidden common cause. A pattern has either directed or undirected edges with the following interpretation:

- two nodes are adjacent in a pattern if and only if they are adjacent in all members of the corresponding Markov equivalence class;
- there is an *unshielded collider* $A \rightarrow B \leftarrow C$ (i.e., a substructure where A and C are parents of B, and A, C are not adjacent) if and only if the same unshielded collider appears in all members of the Markov equivalence class;
- there is a directed edge $A \rightarrow B$ in the pattern only if the same edge appears in all members of the Markov equivalence class;

As hinted by the "only if" condition in the last item, patterns can differ with respect to the *completeness* of their orientations. All members of an Markov equivalence class might agree on the same directed edge that is not part of an unshielded collider (for example, edge $Z \rightarrow W$ in Figure 1.4(b)), and yet it might not be represented in a valid pattern. The original PC algorithm described by Spirtes et al. (2000) is not guaranteed to provide a fully informative pattern, but there are known extensions that provide such a guarantee (Meek, 1997). Some issues of completeness of causal learning algorithms are discussed in this thesis.

Therefore, a key aspect of causal discovery is providing not only a model that fits the data, but *all models that fit the data equally well according to a family of constraints*, i.e., equivalence classes.

There are basically two families of algorithms for learning causal structure from data (Cooper, 1999).

Constraint-satisfaction algorithms check if specific constraints are judged to hold in the population by some decision procedure such as hypothesis testing. Each constraint is tested individually. The choice of which constraints to test is usually based on the outcomes of the previous tests, which increase the computational efficiency of this strategy. Moreover, each test tends to be computationally unexpensive, since only a handful of variables are included in the hypothesis to be tested.

For example, the PC algorithm of Spirtes et al. (2000) learns Markov equivalence classes under the assumption of no hidden common causes by starting with a fully connected undirected graph. An undirected edge is removed if the variables at the endpoints are judged to be independent conditioned on some set of variables. The order by which these tests are performed is in such a way that the algorithm is exponential only in the maximum number of parents among all variables in the true model. If this number is small, then the algorithm is tractable even for problems with a large number of variables. After removing all possible undirected edges, directionality of edges is determined according to which constraints were used in the first stage. Details are given by Spirtes et al. (2000).

A second family of algorithms is the *score-based* family. Instead of testing specific constraints, a score-based algorithm uses a score function to rank how well different graphs explain the data. Since scoring all possible models in unfeasible in all but very small problems, most score-based algorithms are greedy hill-climbing search algorithms. Starting from some candidate model, a greedy algorithm applies operators that create a new set of candidates based on modifications of the current graph. New candidates represent different sets of independence (or other) constraints. The best scoring model among this set of candidates will become the new current graph, unless the current graph itself has a higher score. In this case we reached a local maximum and the search is halted. For instance, the K2 algorithm of Cooper and Herskovits (1992) was one of the first algorithms of this kind. The usual machinery of combinatorial optimization, such as simulated annealing and tabu search, can be adapted to this problem in order to reach better local maxima.

In Figure 1.5, we show an example of the PC algorithm in action. Figure 1.5(a) shows the true model, which is unknown to the algorithm. However, this model entails several conditional independence constraints. For instance, X_1 and X_2 are marginally independent. X_1 and X_4 are independent given X_3, and so on. Starting from a fully connected undirected graph, as shown in Figure 1.5(b), the PC algorithm will remove edges between any pair that is independent conditioned on some other set of variables. This will result in the graph shown in Figure 1.5(c). Conditional independencies allow us to identify which unshielded colliders exist, and the graph in Figure 1.5(d) illustrates a valid pattern for the true graph. However, in this particular case it is also possible to direct the edge $X_3 \rightarrow X_4$. In this case, the most complete pattern represents an unique graph. An example of a score-based algorithm is given in Chapter 2.

Constraint-satisfaction and score-based algorithms are closely related. For instance, several score-based algorithms generate new candidate models that are either nested within the current candidate or vice-versa. Common score functions that compare the current and new candidates are asymptotically equivalent to likelihood-ratio tests, and therefore choosing a new graph amounts to "accepting"[2] or rejecting the null hypothesis corresponding to the more constrained model.

[2]We use a non-orthodox application of hypothesis testing on which failing to reject the null hypothesis is interpreted as accepting the null hypothesis.

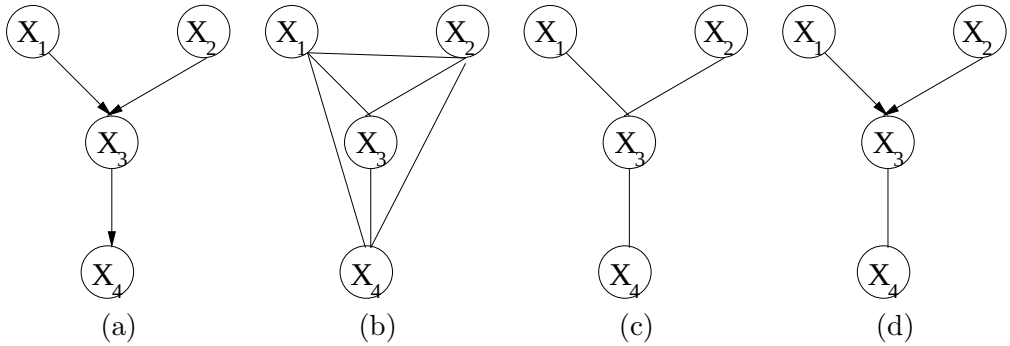


Figure 1.5: A step-by-step demonstration of the PC algorithm. The true model is given in (a). We start with a full undirected graph among latents (b) and remove edges according the independence constraints that hold among the given variables. For example, X_1 and X_2 are marginally independent. Therefore, the edge $X_1 - X_2$ is removed. However, X_1 and X_3 are not independent conditioned on any subset of $\{X_2, X_4\}$. The edge $X_1 - X_3$ remains. At the end of this stage, we obtain graph (c). By orienting unshielded colliders, we get graph (d). Extra steps of orientation detailed by Spirtes et al. (2000) will recreate the true graph.

However, with finite samples, algorithms in different families can get different results. Usually score-based search will give better results, but the computational cost might be much higher. Constraint-satisfaction algorithms tend to be "greedier," in the sense that they might remove more candidates at each step.

Score-based algorithms are especially problematic when latent variables are introduced. While scoring DAGs without hidden variables can be done as efficiently as performing hypothesis tests in a typical constraint-satisfaction algorithm, this is not true when hidden variables are present. In practice, strategies such as Structural EM (Friedman, 1998), as explained in Chapter 2, have to be used. However, Structural EM might increase the chances of an algorithm getting trapped in a local maxima. Another problem is the consistency of the score function, which we discuss in Chapter 6.

1.5 Using parametric constraints

We emphasized Markov equivalence classes in the previous section, but there are other important constraints, besides independence constraints, which can be used for learning causal graphs. They are crucial when several important variables are hidden.

When latent variables are included in a graph, different graphs might represent the same marginal over the observed variables, even if these graphs represent different independencies in the original graph. A classical example is factor analysis (Johnson and Wichern, 2002). Consider the graphs in Figure 1.6, where circles represent latent variables. Assume this is a linear model with additive noise where variables are distributed as multivariate Gaussian. A simple linear transformation of the parameters of a model corresponding to Figure 1.6(a) will generate a model as in Figure 1.6(b) such that two models represent different sets of conditional independencies, but the observed marginal distribution is identical.

Moreover, observed conditional independencies are of no use here. There are no observed

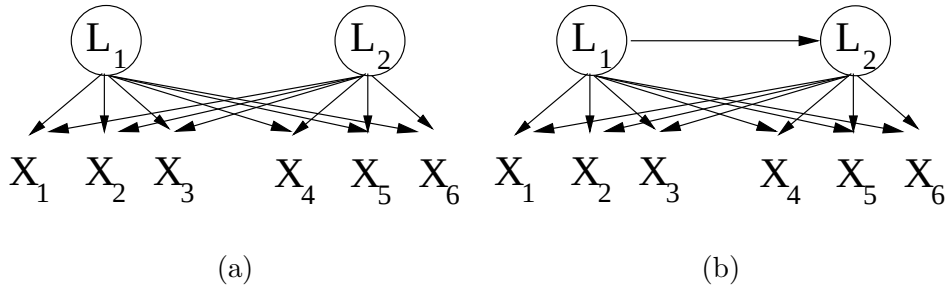


Figure 1.6: These two graphs with two latent variables are indistinguishable for an infinite number of normal distributions.

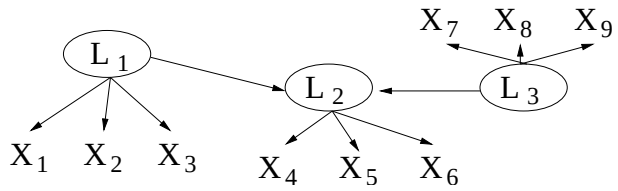


Figure 1.7: A latent variable model which entails several constraints on the observed covariance matrix. Latent variables are inside ovals.

conditional independencies. One has to appeal to other types of constraints. Consider Figure 1.7, where X variables are recorded and L variables (in ovals) are unrecorded and unknown to the investigator. Assume this model is linear.

The latent structure, the dependencies of measured variables on individual latent variables, and the linear dependency of the measured variables on their parents and (unrepresented) independent noises in Figure 1.7 imply a pattern of constraints on the covariance matrix among the X variables. For example, X_1, X_2, X_3 have zero covariances with X_7, X_8, X_9. Less obviously, for X_1, X_2, X_3 and any one of X_4, X_5, X_6, three quadratic constraints (*tetrad* constraints) on the covariance matrix are implied: e.g., for X_4

$$\rho_{12}\rho_{34} = \rho_{14}\rho_{23} = \rho_{13}\rho_{24} \tag{1.1}$$

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

$$\rho_{14}\rho_{25} = \rho_{15}\rho_{24} \tag{1.2}$$

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

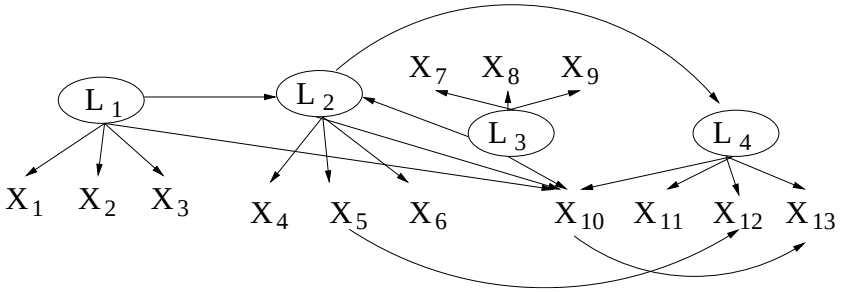


Figure 1.8: A more complicated latent variable model which still entails several observable constraints.

Given a "pure" set of sets of measured indicators of latent variables, as in Figure 1.7 − informally, a measurement model specifying, for each latent variable, a set of measured variables influenced only by that latent variable and individual, independent noises − the causal structure among the latent variables can be estimated by any of a variety of methods. Standard tests of latent variable models (e.g., χ^2) can be used to compare models with and without a specified edge, providing indirect tests of conditional independence among latent variables. The conditional independence facts can then be input to standard Bayes net search algorithms.

In Figure 1.7, the measured variables neatly cluster into disjoint sets of variables and the variables in any one set are influenced only by a single common cause and there are no influences of the measured variables on one another. In many real cases the influences on the measured variables do not separate so simply. Some of the measured variables may influence others (as in signal leakage between channels in spectral measurements), and some or many measured variables may be influenced by two or more latent variables.

For example, the structure among the latents of a linear, Gaussian system shown in Figure 1.8 can be recovered by the procedures we propose. Our aim in what follows is to prove and use new results about implied constraints on the covariance matrix of measured variables to form measurement models that enable estimation of features of the Markov equivalence class of the latent structure in a wide range of cases. We will develop the theory first for linear models with a joint Gaussian distribution on all variables, including latent variables, and then consider possibilities for generalization.

These examples illustrate that, where appropriate parameterizations can be used, new types of constraints on the observed marginal will correspond to different independencies in the latent variable graph, even though these conditional independencies themselves cannot be directly tested. This thesis is entirely built upon this observation. Extra parametric assumptions will be necessary, but at the benefit of broader identifiability guarantees. Considering the large number of applications that adopt such parametric assumptions, our final results should benefit researchers across many fields, such as as econometrics, social sciences, psychology, etc. (Bollen, 1989; Bartholomew et al., 2002). From Chapter 3 to 6 we discuss our approach along possible applications.

1.6 Thesis outline

This thesis concerns algorithms for learning causal and probabilistic graphs with latent variables. The ultimate goal is learning causal relations among latent variables, but most of the thesis will focus on discovering which latents exist and how they are related to the observed variables. We provide theoretical results that our algorithms asymptotically generate outputs with a sound interpretation. Sound algorithms for learning causal structures indirectly provide a suitable approach for density estimation, which we show through experiments. The outline of the thesis is as follows:

- our first goal is to learn the structure of linear latent variable models under the assumption that latents are not children of observed variables. This is the common assumption of factor analysis and its variants, which are applied to several domains where observed variables are measures, and not causes, of a large set of hidden common causes. We provide an algorithm that can learn a specific parametric type of equivalence class according to tetrad constraints in order to identify which latents exist and which observed variables are their respective measures. Given this *measurement model*, we then proceed to find the Markov equivalence class among the hidden variables. We prove the pointwise consistency of this procedure. This is the subject of Chapter 3;

- in Chapter 4, we relax the assumption of linearity among latents. That is, hidden variables can be non-linear functions of their parents, while observed variables are still linear functions of their respective parents. We show that several theoretical results from Chapter 3 still hold under this case. We also show that some of the results do not hold for non-linear models;

- discrete models are considered in Chapter 5. There is a straightforward adaptation of our approach in linear models for the case where measurement are discrete ordinal variables. Because of the extra computational cost of estimating discrete models, we will develop this case under a different framework for learning a set of models for single latent variables. This has a correspondence with the goal of mining databases for association and causal rules;

- finally, in Chapter 6 we develop a heuristic Bayesian learning algorithm for learning latent variable models in more flexible families of probabilistic models and graphs. We emphasize results in density estimation, since the causal theory for such more general graphical models is not as developed as the ones studied in the previous chapters.

Chapter 2

Related work

Latent variable modeling is a century-old enterprise. In this chapter, we provide a brief overview of existing approaches.

2.1 Factor analysis and its variants

The classical technique of factor analysis (FA) is the foundation for many latent variable modeling techniques. In factor analysis, 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. Principal component analysis (PCA) can be seen as a special case of FA, where the variances of the error terms are constrained to be equal (Bishop, 1998).

Let $\mathbf{X}$ represent a vector of observed variables, $\mathbf{L}$ represent a vector of latent variables, and ϵ a vector of error terms. A factor analysis model can then be described as

$$\mathbf{X} = \Lambda\mathbf{L} + \epsilon$$

where Λ is a matrix of parameters, with entry λ_{ij} corresponding to the linear coefficient of L_j in the linear equation defining X_i. In this parameterization, we are setting the mean of each variable to zero to simplify the presentation.

When estimating parameters, one usually assumes that latents and error variables are multivariate normal, which implies a multivariate normal distribution among the observed variables. The covariance of $\mathbf{X}$ is given by

$$\Sigma_{\mathbf{X}} = E[\mathbf{X}\mathbf{X}^{\mathbf{T}}] = \Lambda E[\mathbf{L}\mathbf{L}^{\mathbf{T}}]\Lambda^T + E[\epsilon\epsilon^{\mathbf{T}}] = \Lambda\Sigma_{\mathbf{L}}\Lambda^T + \mathbf{\Psi}$$

where M^T is the transpose of matrix M, $E[X]$ is the expected value of random variable X, $\Sigma_{\mathbf{L}}$ is the model covariance matrix of the latents and $\mathbf{\Psi}$ the covariance matrix of the error terms, usually a diagonal matrix. A common choice of latent covariance matrix is the identity matrix, based on the assumption that latents are independent. This can be represented as graphical model where variables in $\mathbf{X}$ are children of variables in $\mathbf{L}$, as illustrated by Figure 1.6(a), repeated in Figure 2.1 for convenience. If latent variables are arbitrarily dependent (e.g., as a distribution faithful to a DAG), this can be represented by a graphical model connecting latents, as shown in Figure 1.6(b). By definition, the absence of an edge $L_j \rightarrow X_i$ is equivalent to assuming $\lambda_{ij} = 0$.

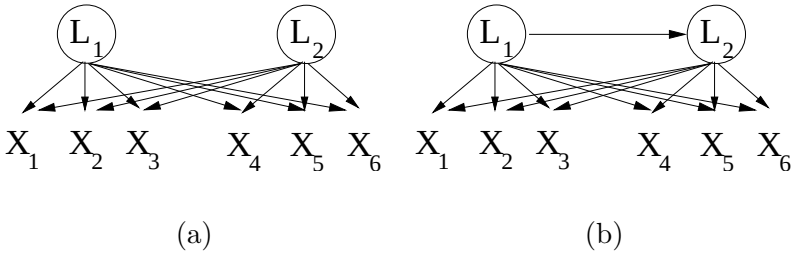


Figure 2.1: Two examples of graphical representations of factor analysis models.

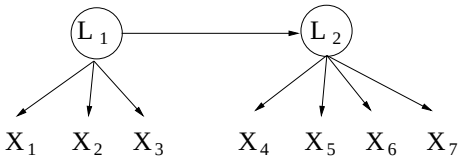


Figure 2.2: A "simple structure," in factor analysis terminology, is a latent graphical model where each observed variable has a single parent.

2.1.1 Identifiability and rotation

When learning latent structure from data under the absence of reliable prior knowledge, one does not want to restrict a priori how latents are connected to their respective measures. That is, in principle the matrix $\mathbf{\Lambda}$ of coefficients (sometimes called the *loading* matrix) does not contain any a priori zeroes specified. This creates a problem, since any linear transformation of $\mathbf{\Lambda}$ will generate an undistinguishable covariance matrix. This can be constructed as follows. Let matrix $\mathbf{\Lambda}_R = \mathbf{\Lambda R}$, where the rotation matrix $\mathbf{R}$ is non-singular. One can then verify that

$$\Sigma_{\mathbf{X}} = \Lambda \Sigma_{\mathbf{L}} \Lambda^T + \mathbf{\Psi} = \Lambda_R \Sigma'_{\mathbf{L}} \Lambda_R^T + \mathbf{\Psi}$$

where $\Sigma'_{\mathbf{L}} = \mathbf{R}^{-1} \Sigma_{\mathbf{L}} \mathbf{R}^{-T}$. This is independent on what the true latent covariance matrix $\Sigma_{\mathbf{L}}$ is.

Since $\mathbf{\Lambda}_R$ can be substantially different from $\mathbf{\Lambda}$, one cannot learn the proper causal connections between $\mathbf{L}$ and $\mathbf{X}$ by using the empirical covariance matrix. This matter can in principle be solved by using higher order moments of the distribution function (see Section 2.1.4 for a brief discussion on independent component analysis). However, this is not the case for Gaussian distributions, the typical case in applications of factor analysis. Moreover, estimating higher order moments is more difficult than estimating covariances, which can compromise any causal discovery analysis. If one wants or needs to use only covariance information, a rotation criterion is necessary.

The most common rotation criteria attempt to rotate the loading matrix to obtain something close to a "simple structure" (Harman, 1967; Johnson and Wichern, 2002; Bartholomew and Knott, 1999; Bartholomew et al., 2002). A FA model with "simple structure" is a model where each observed variable has a single latent parent. Structures "close to a simple structure" are those where one or few of the edges into a specific node X_i have a high absolute value, while all the other edges into X_i have coefficients close to zero. In real world applications, it is common practice to ignore loadings with absolute values smaller than some threshold, which may be set according to a significance test. Figure 2.2 illustrates a simple structure.

Variable	L_1	L_2	L_3	L_4
100-m run	.167	**.857**	.246	-.138
Long jump	.240	**.477**	**.580**	.011
Shot put	**.966**	.154	.200	-.058
High jump	.242	.173	**.632**	.113
400-m run	.055	**.709**	.236	.330
110-m hurdles	.205	.261	**.589**	-0.071
Discus	**.697**	.133	.180	-0.009
Pole vault	.137	.078	**.513**	.116
Javelin	**.416**	.019	.175	.002
1500-m run	-0.055	0.056	.113	**.990**

Table 2.1: Decathlon data modeled with factor analysis.

In practice, the following steps are performed in a factor analysis application.

- choose the number k of latents. This can be done by testing models with 1, 2, ..., n latents, choosing the one that maximizes some score function. For instance, choosing the smallest k such that a factor analysis model of k independent latents and a fully unconstrained loading matrix $\mathbf{L}$ has a p-value of at least 0.05 according to some test such as χ^2 (Bartholomew and Knott, 1999);
- fit the model with k latents (e.g., by maximum likelihood estimation, Bartholomew and Knott, 1999) and apply a rotation method to achieve something close to a "simple structure" (e.g., the OBLIMIN method, Bartholomew and Knott, 1999);
- remove edges from latents to observed variables according to their statistical significance.

The literature on how to find connections between the latents themselves is much less developed. Bartholomew and Knott (1999) present a brief discussion, but it relies heavily on the use of domain knowledge, which lead to the quote given at the beginning of Chapter 1.

2.1.2 An example

The following example is described in Johnson and Wichern (2002), a factor analytic study of Olympic decathlon scores since World War II. The scores for all 10 decathlon events were standardized. Four latent variables were chosen using a method based on the analysis of the eigenvalues of the empirical correlation matrix. Sample size is 160. Results after rotation are shown in Table 2.1. Latent variables were treated as independent in this analysis. Statistically significant loadings (which would correspond to edges in a graphical model) are shown in bold. There is an intuitive separation of factors, with a clear component for jumping, another for running, another for throwing and a component for the longer running competition. In this case, components were well-separated. In many cases, the separation is not clear, as in the examples given in Chapter 3.

Several multivariate analysis books as (Johnson and Wichern, 2002) describe applications of factor analysis. More specialized books provide more detailed perspectives. For instance, Bartholomew et al. (2002) describe a series of case studies of factor analysis and related methods in social sciences. Malinowski (2002) describes applications in chemistry.

2.1.3 Remarks

Given the machinery described in the previous sections, factor analysis has been widely used to discover latent variables, despite the model identification shortcomings that require rather ad-hoc matrix rotation methods.

One the the fundamental ideas used to motivate factor analysis with rotation as a method for finding meaningful hidden variables is that a group of random variables can be clustered according to the strength of their correlations. As put by a traditional textbook in multivariate analysis (Johnson and Wichern, 2002, 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. For example, consider the following simple structure with three latents (L_1, L_2 and L_3) and with four indicators per latent. Let $L_2 = 2L_1 + \epsilon_{L_2}$, $L_3 = 2L_2 + \epsilon_{L_3}$, where L_1, ϵ_{L_2} and ϵ_{L_3} are all standard normal variables. Let the first and fourth indicator of each latent have a loading of 9, and the second and third have a loading of 1. This means, for example, that the first indicator of L_1 is more strongly correlated with the first indicator of L_2 than with the second indicator of L_1. Factor analysis with rotation methods will be mislead, typically clustering indicators of L_2 and L_3 together.

Because of identifiability problems, many techniques to learn hidden variables from data are confirmatory, i.e., they start with a conjecture about a possible latent. Domain knowledge is used for selecting the initial set of indicators to be tested as a factor analysis model. Statistical and theoretical tools here aim at achieving *validity* and *reliability* assessment (Bollen, 1989) of hypothesized latent concepts. A model for a single latent 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 "reasonably" small. Since these criteria rely on unobservable quantities, they are not easy to evaluate.

Latents confirmed with FA in principle do not rule out other possible models that might fit the data as well. Moreover, when the model does not fit the data, finding the reason for the discrepancy between theory and evidence can be difficult. Consider the case of testing a theoretical factor analysis model for a single latent. 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.

The criticism is on determining the number of factors based merely in a criterion of statistical fitness. In their 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.

In contrast to Carmines and Zeller, the methods described in this thesis are data-driven. Some problems will be ultimately irreducible to a single or few models. While background knowledge will always be essential in practice, we will show that our approach at the very least attempts to produce submodels that can be justified on the grounds of a few domain-independent assumptions and the data.

Unlike factor analysis, our methods have theoretical justifications. If the true model is a simple structure, the method described in Section 2.1.1 is a reliable way of reconstructing the actual structure from data despite the counter-example described earlier in this section. However, if the true model is not a simple structure, even if it is an approximate one, this method is likely to generate unpredictable results. In Chapter 3 we perform some empirical tests using exploratory factor analysis. The conclusion is that FA is largely unreliable as a method for finding simple structures. Also, unlike the pessimistic conclusions of Bartholomew and Knott (1999), we show that it is possible to find causal structures among latents, depending on how decomposable the real model is, without requiring background knowledge.

2.1.4 Other variants

A variety of methodologies were created in order to generalize standard FA for other distributions. 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. Instead, some measure of independence is maximized without adopting strong assumptions concerning 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 on the original idea of factor analysis, interpreting observed variables as joint measurements of a set of independent latents. Some extensions, such as tree-based component analysis (Bach and Jordan, 2003), attempt to relax this assumption by allowing a tree-structured model among latents. This approach, however, is difficult to generalize to more flexible graphical models due to its computational cost and identifiability problems. For a few problems such as blind source separation such an assumption may be reasonable, but it is more often the case that it is not. Most variations of factor analysis, while useful for density estimation and data visualization (Minka, 2000; Bishop, 1998; Ghahramani and Beal, 1999; Buntine and Jakulin, 2004), insist on the assumption of independent latents.

2.1.5 Discrete models and item-response theory

While several variations of factor analysis concentrate on continuous models, there is also a large literature on discrete factor analysis. Some concern models with discrete latents and measures, such as latent class analysis (Bartholomew et al., 2002), discrete PCA (Buntine and Jakulin, 2004) and latent Dirichlet allocation (Blei et al., 2003). This thesis concerns models with continuous latents only. A discussion on the suitability of continuous latents can be found in (Bartholomew and Knott, 1999; Bartholomew et al., 2002).

Factor analysis models with continuous latents and discrete indicators are generally known as *latent trait models*. A discussion of latent trait models for ordinal and binary indicators is given in Chapter 5. In the rest of this section, we discuss latent trait models under the context of *item-response theory* (IRT). The field of IRT consists on the analysis of multivariate (usually binary) data as measurements of underlying "abilities" of an individual. This is the case of research on educational testing, whose goal is to design tests to measure the skills of a student according to determined factors as "mathematical skills" or "language skills." Once one models each desired ability as a latent variable, such random variables can be used to rank individuals and provide information about the distribution of such individuals in the latent space.

Much of the research on IRT consists on designing *tests of unidimensionality*. That is, a statistical procedure to determine if a set of questions are indicators of a single latent factor. Conditioned on such a factor, indicators should be independent. Besides testing for the dimensionality of a set of observed variables, estimating the *response functions* (i.e., the conditional distribution of each indicator given its latent parents) is part of the core research in IRT.

Parametric models of IRT are basically latent trait models. For the purposes of learning latent structure, they are not essentially different from generic latent trait models as explained in Chapter 5. A more distinctive aspect of IRT research is on *nonparametric* models (Junker and Sijtsma, 2001), where no finite dimensional set of parameters is assumed in the description of the response functions. Instead, the assumption of *monotonicity* of all response functions is used: this means that for a particular indicator X_i and a vector of latent variables $\mathbf{\Theta}$, $P(X_i = 1|\mathbf{\Theta})$ is non-decreasing as a function of (the coordinates of) $\mathbf{\Theta}$.

Some approaches allow mild violations of independence conditioned on the latents, as long as estimation of the latent values can be consistenly done when the number of questions (i.e., indicators) goes to infinite (see, e.g., Stout, 1990). Many non-parametric IRT approaches use a statistic as a proxy for the latent factors (such as the number of "correctly answered" questions) in order to estimate non-parametric associations due to common hidden factors. Junker and Sijtsma (2001) and Habing (2001) briefly review some of these approaches. Althought this thesis does not follow the non-parametric IRT approach in any direction, it might provide future extensions to our framework.

2.2 Graphical models

Beyond variations of factor analysis, there is a large literature in learning the structure of 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 called *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 be achieved (at least in a more general sense) by means of probability calculus only (Spirtes et al., 2000; Pearl, 2000).

2.2.1 Independence models

We described the PC algorithm in Chapter 1, stressing that such an algorithm assumes that no pair of observed variables have a hidden common cause. The Fast Causal Inference algorithm (FCI) (Spirtes et al., 2000) is an alternative algorithm for learning Markov equivalence classes of a special class of graphs called mixed ancestral graphs (MAGs) by Richardson and Spirtes (2002). MAGs allow the expression of which pairs of observed variables have hidden common causes. The FCI algorithm returns a representation of the Markov equivalence class of MAGs given the conditional independence statements that are known to be true among the observed variables. This representation shares many similarities with the pattern graphs used to represent Markov equivalence classes of DAGs.

Consider Figure 2.3(a) representing a true model with three hidden variables H_1, H_2 and H_3. The marginal distribution for W, X, Y, Z is faithful to several Markov equivalent MAGs. All equivalent graphs can be represented by the graph shown in Figure 2.3(b). Although describing such a representation in detail is out of the scope of this thesis, it suffices to say that, e.g., the edge X $\circ\!\!\rightarrow Y$ means that it is possible that X and Y have a hidden common cause, and that we know for sure that Y is not a cause of X. Edge $Z \rightarrow W$ means that Z causes W, and there is no hidden common cause between them.

Since only observed conditional independencies are used by FCI, any model where most observed variables are connected by hidden common causes will be problematic. For instance, consider the true model given in Figure 2.3(c), where H is a hidden common cause of all observed variables. Since no observed independencies exist, the output of FCI will be the sound, but minimally informative, graph of Figure 2.3(d). Such graphs do not attempt to represent latents explicitly. In contrast, this thesis provides an algorithm able to reconstruct Figure 2.3(c) when observed variables are linear functions of H.

An algorithm such as FCI is still necessary in models where observed independencies do not exist. Ultimately, even an algorithm able to explicitly represent latents still needs to describe how latent nodes are connected. Since explicit latent nodes might have hidden common causes that are not represented in the graphical model, a representation such as a MAG can be used to account for these cases.

2.2.2 General models

Many standard models can be recast in graphical representations (e.g., factor analysis as a graph where edges are oriented from latents to observed variables). Under the graphical modeling literature, there are several approaches for dealing with latent variables beyond models of Markov

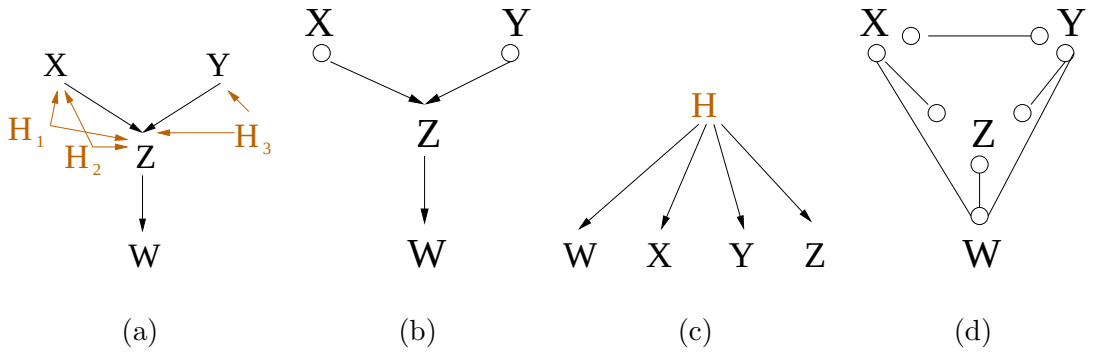


Figure 2.3: Figure (b) represents the Markov equivalence class of MAGs compatible with the marginal distribution of $\{W, X, Y, Z\}$ represented in Figure (a). Figure (d) represents the Markov equivalence class of MAGs compatible with the marginal distribution of $\{W, X, Y, Z\}$ represented in Figure (c)

equivalence classes. 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 (Minka, 2000).

Elidan et al. (2000) empirically evaluate heuristics for introducing latent variables. These heuristics were independently suggested in several occasions (e.g., Heckerman, 1998) and consist on the observation that if two variables are conditionally independent given a set of other observed variables, then given the faithfulness condition they should not have hidden common causes. Given a DAG representing probabilistic dependencies among observed variables, a clique of nodes might be the result of hidden common causes that explain such associations. The specific implementation of Elidan et al. (2000) introduces latent variables as a hidden common causes of sets of densely connected nodes (not necessarily cliques, in order to account for statistical mistakes in the original DAG). Since we are going to use FindHidden in some of our experiments, we describe the variation we used in Table 2.2. It also serves as an illustration of a score-based search algorith, as suggested in Chapter 1.

This algorithm uses as a sub-routine a StandardHillClimbing(G_{start}) procedure. Starting from a graph G_{start}, this is simply a greedy search algorithm among DAGs: given a current DAG G, all possible variations of G generated by either

- adding one edge to G
- deleting one edge from G
- reversing an edge in G

are evaluated. Given a dataset $\mathbf{D}$, the candidate that achieves the highest score according to a given score function $\mathcal{F}(G, \mathbf{D})$ is chosen as a new DAG, unless the current graph G has a higher score. In this case, the algorithm halts and returns G. Although simple, such a heuristic is quite effective for learning DAGs without latent variables (Chickering, 2002; Cooper, 1999), especially if enriched with standard techniques in combinatorial optimization for escaping local minima, such

Algorithm FindHidden
Input: a dataset $\mathbf{D}$

1. Let G_{null} be a graph over the variables in $\mathbf{D}$ with no edges.
2. $G \leftarrow$StandardHillClimbing(G_{null})
3. Do
4. Let $\mathbf{C}$ be the set of semicliques in G
5. Let $C_i \in \mathbf{C}$ be the semiclique that maximizes $\mathcal{F}$(IntroduceHidden(G, C_i))
6. $G_{new} \leftarrow$StandardHillClimbing(IntroduceHidden$(G, C_i), \mathbf{D}$)
7. If $\mathcal{F}(G_{new}, \mathbf{D}) > \mathcal{F}(G, \mathbf{D})$
8. $G \leftarrow G_{new}$
9. While G changes
10. Returns G

Table 2.2: One of the possible variations of FindHidden (Elidan et al., 2000; Heckerman, 1995), which iteratively introduces one latent at a time and attempts to learn a directed graph structure given such hidden nodes.

as tabu lists, beam search and annealing. FindHidden extends this idea by introducing latent variables into "dense" regions of G.

Such "dense" regions are denominated *semicliques*, which are basically groups of nodes where each node is adjacent to at least half of the other members of this group. Heuristics for enumerating the semicliques of a graph are given by Elidan et al. (2000).

Given a semiclique C_i, the operation InsertHidden(G, C_i) returns a modification of a graph G by introducing a new latent L_i, removing all edges into elements of C_i, and making L_i a common parent to all elements in C_i. Moreover, for each parent P_j of a node in C_i, we set P_j to be a parent of L_i unless that creates a cycle. According to Step 5 of Table 2.2, in our implementation we choose among the possible semicliques to start the next cycle of FindHidden by picking the one that has the best initial score.

But heuristic methods such as FindHidden have as their main goal reducing the number of parameters in a Bayesian network. The idea is reducing the variance of the resulting density estimator, achieving better probabilistic predictions. They do not provide any formal interpretation of what the resulting structure actually is, no explicit assumptions on how such latents should interact with the observed variables, no analysis of possible equivalence classes, and consequently no search algorithm that can account for equivalence classes. For probabilistic modeling, the results described by Elidan et al. (2000) are a convincing demonstration of the suitability of this approach, which is intuitively sound. For causality discovery under the assumption that all observed variables have hidden common causes (such as in the problems we discussed in Chapter 1), they are a unsatisfying solution.

The introduction of proper assumptions on how latents and measures interact makes learning the proper structure a more realistic possibility. By assuming a discrete distribution of latent variables and observed measurements in a hidden Markov model (HMM), Beal et al. (2001) present algorithms for learning the transition and emission probabilities with 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 model selection was not the goal. However, if one wants to have qualitative information of independence (as necessary in our axiomatic causality calculus), such analysis has to be carried on. This is also necessary in order to scale this approach for models with a large number of latent variables.

As another example, Zhang (2004) provides a sound representation for latent variable models of discrete variables (both observed and latent) with a multinomial probabilistic model. The model is constrained to be a tree, however, and every observed variable has one and only (latent) parent and no child. Similar to factor analysis, no observed variable can be a child of another observed variable or a parent of a latent. Instead of searching for variables that satisfy this assumption, Zhang (2004) assumes the variables measured satisfy it. To some extent, an equivalence class of graphs is described, which limits the number of latents and the possible number of states each categorical latent variable can have without being empirically indistinguishable from another graph with less latents or less states per latent. Under these assumptions, the set of possible latent variable models is therefore finite.

Approaches such as (Zhang, 2004) and (Elidan et al., 2000) are score-based search algorithms for learning DAGs with latent variables. Therefore, they require scoring thousands of candidate models, which can be a very computationally expensive operation since calculating the most common score functions requires solving non-convex optimization problems. More important, in principle they also require re-evaluation of the whole model for each score evaluation. The cost of such re-evaluation is prohibitive in all but very small problems.

However, the Structural EM framework (Friedman, 1998) can highly simplify the problem. Structural EM algorithms introduce a graphical search module into an expectation-maxization algorithm (Mitchell, 1997) besides parameter learning. If the score function to be optimized (usually the posterior distribution of the graph or penalized log-likelihood) is linear in the expected moments of the hidden variables, such moments are initially calculated (the expectation step), fixed as if they were observed data, and structural search proceeds as if there were no hidden variables (the maximization step). If the score function does not have this linearity property, some approximations might be used instead (Friedman, 1998).

There are different variations of Structural EM. For instance, we make use of the following variation:

1. choose an initial graph and an initial set of parameter values. It will be clear in our context which initial graphs are chosen.
2. maximize the score function with respect to the parameters
3. use the parameter values to obtain all required expected sufficient statistics (in our case, first and second order moments of the joint distribution of the completed data, i.e., including observed and hidden data points)
4. apply the respective structure search algorithm to maximize the score function as if the expected sufficient statistics were observed data

5. if the graphical structure changed, return to Step 2

STRUCTURAL EM-based algorithms are usually much faster than straighforward implementations, which might not be feasible at all otherwise. However, this framework is not without its shortcomings. A bad choice of initial graph, for instance, might easily result in a bad local maxima. Some guidelines for proper application of STRUCTURAL EM are given by (Friedman, 1998).

Glymour et al. (1987) and Spirtes et al. (2000) describe algorithms for modifying a latent variable model using constraints on the covariance matrix of the observed variables. These approaches are also either heuristic or require strong background knowledge and do not generate new latents from data. Pearl (1988) discuss a complementary approach that generates new latents, but requires the true model to be a tree, similarly to Zhang (2004). This thesis can be seen as a generalization of these approaches with formal results of consistency. Spirtes et al. (2000) present a sound test of conditional independence among latents, but it requires knowing in advance which observed variables measure which latents. We discuss this in detail in Chapter 3.

A recurring debate in the structural equation modeling literature is whether one should learn models from data by the following 2-step approach: 1. find which latents exist and which observed variables are their corresponding indicators; 2. given the latents, find the causal connections among them. The alternative is trying to achieve both at the same time (Fornell and Yi, 1992; Hayduk and Glaser, 2000; Bollen, 2000). As we will see, this thesis strongly supports a two-step procedure. A good deal of criticism on two-step approaches concerns the use of methods that suffer from non-identifiability shortcomings, such as factor analysis. In fact, we do not claim we can find the true model. Our solution, explained in Chapter 3, is trying to discover only features that *can* be identified, and reporting ignorance about what we cannot identify. The structural equation modeling literature offers no alternative. Instead, current "two-step" approachs are naive in the sense they do not account for equivalence classes (Bollen, 2000), and any "one-step" approach is hopeless: the arguments for this approach show an unhealthy obsession on using extensive background knowledge (Hayduk and Glaser, 2000), i.e., they mostly avoid solving the problem they are supposed to solve, which is learning from data. Although we again stress that assumptions concerning the true structure of the problem at hand are always necessary, we favor a more data-driven solution.

2.3 Summary

Probabilistic modeling through latent variables is a mature field. Causal modeling with latent variable models is still a fertile field, mostly due to the fact that researchers on this field are usually not concerned about equivalence classes.

Carreira-Perpinan (2001) has an extended review of probabilistic modeling with latent variables. Glymour (2002) offers a more detailed discussion on the shortcomings of factor analysis. The journals *Psychometrika* and *Structural Equation Modeling* are primary sources of research in latent variable modeling via factor analysis and SEMs.

Chapter 3

Learning the structure of linear latent variable models

The associations among a set of measured variables can often be explained by hidden common causes. Discovering such variables, and the relations among them, is a pressing challenge for machine learning. This chapter describes an algorithm for discovering hidden variables in linear models and the relations between them. Under the Markov and faithfulness conditions, we prove that our algorithm achieves Fisher consistency: in the limit of infinite data, all causal claims made by our algorithm are correct in a sense we make precise. In order to evaluate our results, we perform simulations and three case studies with real-world data.

3.1 Outline

This chapter concerns linear models, a very important class of latent variable models. It is organized as follows:

- **Section 2.2: The setup** formally defines the problem and makes explicit the assumptions we adopt;
- **Section 2.3: Learning measurement models** describes an approach to deal with half of the given problem, i.e., discovering latent common causes and which observed variables measure them;
- **Section 2.4: Learning the structure of the unobserved** describes an algorithm to learn a Markov equivalence class of causal graphs over latent variables given a measurement model;
- **Section 2.5: Empirical results** discusses series of experiments with simulated data and three real-world data sets, along with criteria of success;
- **Section 2.6: Conclusion** wraps up the contributions of this chapter.

3.2 The setup

We adopt the framework of causal graphical models. More background material in graphical causal models can be found in Spirtes et al. (2000) or Pearl (2000) and Chapter 1.

3.2.1 Assumptions

The goal of our work is to reconstruct features of the structure of a latent variable graphical model from i.i.d. observational data sampled from a subset of the variables in the unknown model. These features should be sound and informative. We assume that the true causal graph G generating the data has the following properties:

A1. there are two types of nodes: observed and latent.

A2. no observed node is an ancestor of any latent node. We call this property the *measurement assumption*;

A3. G is acyclic;

We call such objects *latent variable graphs*. Further, we assume that G is quantitatively instantiated as a semi-parametric probabilistic model with the following properties:

A4. G satisfies the causal Markov condition;

A5. each observed node O is a linear function of its parents plus an additive error term of positive finite variance;

A6. let $\mathbf{V}$ be the set of random variables represented as nodes in G, and let $f(\mathbf{V})$ be their joint distribution. We assume that $f(\mathbf{V})$ is faithful to G: that is, a conditional independence relation holds in $f(\mathbf{V})$ if and only if it is entailed in G by d-separation.

Without loss of generalization, we will assume all random variables have zero mean. We call such an object a *linear latent variable model*, or simply *latent variable model*. A single symbol, such as G, will be used to denote both a latent variable model and the corresponding latent variable graph. Notice that Zhang (2004) does not require latent variable models to be linear, but he requires the entire graph to be a tree, besides relying on the measurement assumption. We do not need to assume any special constraints in the graphical structure of our models besides being a directed acyclic graph (DAG).

Linear latent variable models are ubiquitous in econometric, psychometric, and social scientific studies (Bollen, 1989), where they are usually known as structural equation models. The methods we describe here rely on statistical constraints for continuous variables that are well known for such models. In theory, it is straightforward to extend it to model binary or ordinal discrete variables, as discussed in Chapter 5. The method of Zhang (2004) is applicable to discrete sample spaces only.

Two important definitions will be used throughout this chapter (Bollen, 1989):

Definition 3.1 (Measurement model) *Given a latent variable model G, the submodel containing the complete set of nodes, and all and only those edges that point into observed nodes, is called the measurement model of G.*

Definition 3.2 (Structural model) *Given a latent variable model G, its submodel containing all and only its latent nodes and respective edges is the structural model of G.*

3.2.2 The Discovery Problem

The discovery problem can loosely be formulated as follows: *given a data set with variables* **O** *that are observed variables in a latent variable model G satisfying the above conditions, learn a partial description of the measurement and structural models of G that is as informative as possible.*

Since we put very few restrictions on the graphical structure of the unknown model G, we will not be able to uniquely determine G's full structure. For instance, suppose there are many more latent common causes than observed variables, and every latent is a parent of every observed variable: no learning procedure can realistically be expected to identify such a structure. However, instead of making extra assumptions about the unknown graphical structure (e.g., assume the number of latents in bounded by a known constant, its causal model is tree-structured, etc.), we adopt a data-driven approach: if there are features that cannot be identified, then we just report ignorance.

We can further break up the discovery problem into three sub-problems:

DP1. Discover the number of latents in G.

DP2. Discover which observed variables measure each latent G.

DP3. Discover the Markov equivalence class among the latents in G.

The first two sub-problems involve discovering the measurement model, and the third discovering the structural model. Accordingly, our algorithm takes a two-step approach: in stage 1 it learns as much as possible about features of the measurement model of G, and in stage 2 it learns as much about the features of the structural model as possible using the measurement features discovered in stage 1. Exploratory factor analysis (EFA) can be viewed as an alternative algorithm for stage 1: finding the measurement model. In our simulation studies, we compare our procedure against EFA on several desiderata relevant to this task.

More specifically, we will focus on learning a special type of measurement model, called *pure measurement model*.

Definition 3.3 (Pure measurement model) *A pure measurement model is a measurement model in which each observed variable has only one latent parent, and no observed parent. That is, it is a tree beneath the latents.*

A pure measurement model implies a *clustering* of observed variables: each cluster is a set of observed variables that share a common (latent) parent, and the set of latents defines a partition over the observed variables.

There are several reasons for justifying the focus on pure instead of general measurement models. First, as it is explained in Section 3.4, this provides enough information concerning the Markov equivalence class of the structural model.

The second reason is motivated by a more practical reason: the equivalence class of general measurement models that are undistinguishable can be very hard to represent. While, for instance, a Markov equivalence class for models with no latent variables can be neatly represented by a single graphical object known as pattern (Pearl, 2000; Spirtes et al., 2000), the same is not true for latent variable models. For instance, the models in Figure 3.1 differ not only in the direction of the edges, but also in the adjancencies themselves ($\{X_1, X_2\}$ adjacent in one case, but not $\{X_3, X_4\}$; $\{X_3, X_4\}$ adjacent in another case, but not $\{X_1, X_2\}$) and the role of the latent variables (ambiguity

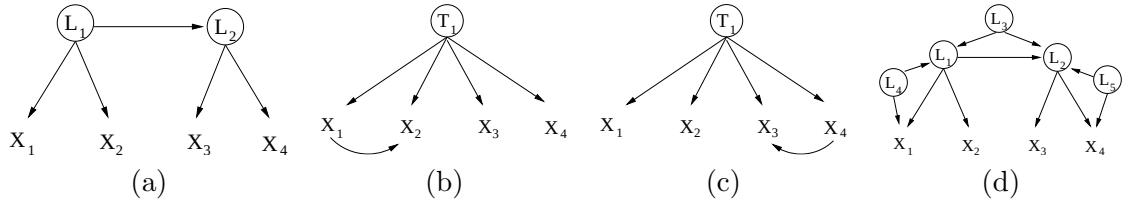


Figure 3.1: All of these four models can be undistinguishable by information contained in the covariance matrix.

about which latent d-separates which observed variables, how they are connected, etc. Notice that, in Figure 3.1(d), there is no latent that d-separates three observed variables, unlike in Figures (a), (b) and (c)). Just representing the class of this very small example can be cumbersome and uninformative.

In the next section, we describe a solution to the problem of learning pure measurement models by dividing it into two main steps:

1. find an intermediate representation, called *measurement pattern*, which implicitly encodes all the necessary information to find a pure measurement model. This is done in Section 3.3.2.

2. "purify" the measurement pattern by choosing a subset of the observed variables given in the pattern, such that this subset can be partitioned according to the latents in the true graph. This is done in Section 3.3.3.

Concerning the example given in Figure 3.1, if the input is data generated by any of the models given by this Figure, our algorithm will be conservative and return an empty model. The equivalence class is too broad to provide information about latents and their causal connections.

3.3 Learning pure measurement models

Given the covariance matrix of four random variables $\{A, B, C, D\}$ we have that zero, one or three of the following *tetrad constraints* may hold (Glymour et al., 1987):

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

where σ_{XY} represents the covariance of X and Y. Like conditional independence constraints, different latent variable models can entail different tetrad constraints, and this was explored heuristically by Glymour et al. (1987). Therefore, a given set of observed tetrad constraints will restrict the set of possible latent variable graphs.

The key to solve the problem of structure learning is a graphical characterization of tetrad constraints. Consider Figure 3.2(a). A single latent d-separates four observed variables. When this graphical model is linearly parameterized as

$$\begin{aligned} X_1 &= \lambda_1 L + \epsilon_1 \\ X_2 &= \lambda_2 L + \epsilon_2 \\ X_3 &= \lambda_3 L + \epsilon_3 \\ X_4 &= \lambda_4 L + \epsilon_4 \end{aligned}$$

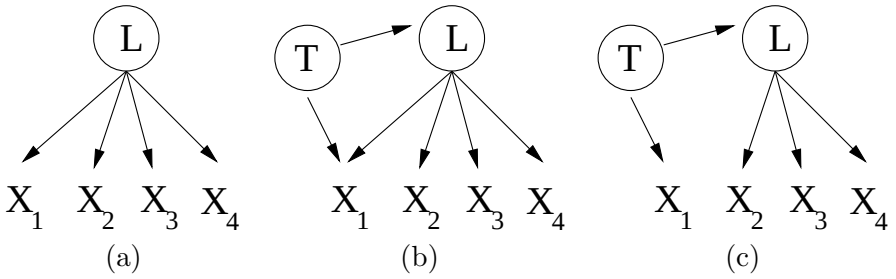


Figure 3.2: A linear variable model with any of the graphical structures above entails all possible tetrad constraints in the marginal covariance matrix of $X_1 - X_4$.

it entails all three tetrad constraints among the observed variables. That is, any choice of values for coefficients $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ and error variances implies

$$\begin{aligned} \sigma_{X_1X_2}\sigma_{X_3X_4} &= (\lambda_1\lambda_2\sigma_L^2)(\lambda_3\lambda_4\sigma_L^2) = (\lambda_1\lambda_3\sigma_L^2)(\lambda_2\lambda_4\sigma_L^2) = \sigma_{X_1X_3}\sigma_{X_2X_4} \\ &= (\lambda_1\lambda_2\sigma_L^2)(\lambda_3\lambda_4\sigma_L^2) = (\lambda_1\lambda_4\sigma_L^2)(\lambda_2\lambda_3\sigma_L^2) = \sigma_{X_1X_4}\sigma_{X_2X_3} \end{aligned}$$

where σ_L^2 is the variance of latent variable L.

While this result is straightforward, the relevant result for a structure learning algorithm is the converse, i.e., establishing equivalence classes from observable tetrad constraints. For instance, Figure 3.2(b) and (c) are different structures with the same entailed tetrad constraints that should be accounted for. One of the main contributions of this thesis is to provide several of such identification results, and sound algorithms for learning causal structure based on them. Such results require elaborate proofs that are left to the Appendix. What follows are descriptions of the most significant lemmas and theorems, and illustrative examples.

We start with one of the most basic lemmas, used as a building block for the more evolved results. It is basically the converse of the observation above. Let ρ_{AB} be the Pearson correlation coefficient of random variables A and B, and let G be a linear latent variable model with observed variables $\mathbf{O}$:

Lemma 3.4 *Let* $\{X_1, X_2, X_3, X_4\} \subset \mathbf{O}$ *be such that* $\sigma_{X_1X_2}\sigma_{X_3X_4} = \sigma_{X_1X_3}\sigma_{X_2X_4} = \sigma_{X_1X_4}\sigma_{X_2X_3}$. *If* $\rho_{AB} \neq 0$ *for all* $\{A, B\} \subset \{X_1, X_2, X_3, X_4\}$, *then there is a node* P *that d-separates all elements* $\{X_1, X_2, X_3, X_4\}$ *in* G.

It follows that, if no observed node d-separates $\{X_1, X_2, X_3, X_4\}$, then node P has to be a latent node.

In order to learn a pure measurement model, we basically need two pieces of information: i. which sets of nodes are d-separated by a latent; ii. which sets of nodes do not share any common hidden parent. The first piece of information can provide possible indicators (children/descendants) of a specific latent. However, this is not enough information, since a set $\mathbf{S}$ of observed variables can be d-separated by a latent L, and yet $\mathbf{S}$ might contain non-descendants of L (one of the nodes might have a common ancestor with L and not be a descendant of L, for instance). This is the reason why we need to *cluster* observed variables into different sets when it is possible to show they cannot share a common hidden parent. We will show that most non-descendants nodes can be removed if we are able to separate nodes in such a way.

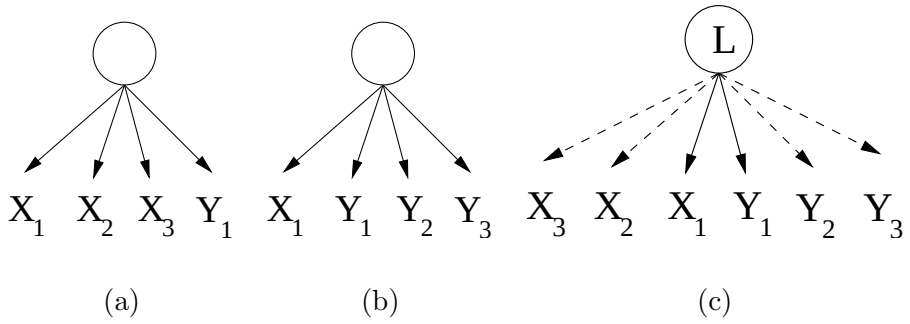


Figure 3.3: If sets $\{X_1, X_2, X_3, Y_1\}$ and $\{X_1, Y_1, Y_2, Y_3\}$ are each d-separated by some node (e.g., as in Figures (a) and (b) above), the existence of a common parent L for X_1 and Y_1 implies a common node d-separating $\{X_1, Y_1\}$ from $\{X_2, Y_2\}$, for instance (as exemplified in Figure (c)).

There are several possible combinations of observable tetrad constraints that allow one to identify such a clustering. Consider, for instance, the following case. Suppose we have a set of six observable variables, X_1, X_2, X_3, Y_1, Y_2 and Y_3 such that:

1. there is some latent node that d-separates all pairs in $\{X_1, X_2, X_3, Y_1\}$ (Figure 3.3(a));
2. there is some latent node that d-separates all pairs in $\{X_1, Y_1, Y_2, Y_3\}$ (Figure 3.3(b));
3. there is no tetrad constraint $\sigma_{X_1X_2}\sigma_{Y_1Y_2} - \sigma_{X_1Y_2}\sigma_{X_2Y_1} = 0$;
4. no pairs in $\{X_1, \ldots, Y_3\} \times \{X_1, \ldots, Y_3\}$ have zero correlation;

Notice that is possible to empirically verify the first two conditions by using Lemma 3.4. Now suppose, for the sake of contradiction, that X_1 and Y_1 have a common hidden parent L. One can show that L should d-separate all elements in $\{X_1, X_2, X_3, Y_1\}$, and also in $\{X_1, Y_1, Y_2, Y_3\}$. With some extra work (one has to consider the possibility of nodes in $\{X_1, X_2, Y_1, Y_2\}$ having common parents with L, for instance), one can show that this implies that L d-separates $\{X_1, Y_1\}$ from $\{X_2, Y_2\}$. For instance, Figure 3.3(c) illustrates a case where L d-separates all of the given observed variables.

However, this contradicts the third item in the hypothesis (such a d-separation will imply the forbidden tetrad constraint, as we show in the formal proof) and, as a consequence, no such L should exist. Therefore, the items above correspond to an *identification rule* for discovering some d-separations concerning observed and hidden variables (in this case, we show that X_1 is independent of all latent parents of Y_1 given some latent ancestor of X_1). This rule only uses constraints that can be tested from the data.

We restrict our algorithm to search for measurement models that entail the observed tetrad constraints and vanishing partial correlations judged to hold in the population. However, since these constraints ignore any information concerning the joint distribution besides its second moments, this might seem too restrictive.

Figure 3.4 helps to understand the limitations of tetrad constraints. Similarly to the example given in Figure 3.1, here we have several models that can represent the same tetrad constraint, $\sigma_{WY}\sigma_{XZ} = \sigma_{WZ}\sigma_{XY}$, and no other. However, this is much less of a problem when learning

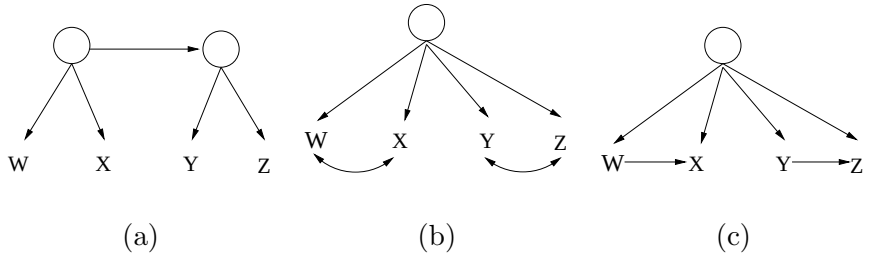


Figure 3.4: Three different latent variable models that can explain a tetrad constraint $\sigma_{WY}\sigma_{XZ} = \sigma_{WZ}\sigma_{XY}$. Bi-directed edges represent independent hidden common causes.

pure models. Moreover, trying to distinguish among such models using higher order moments of the distribution will increase the chance of committing statistical mistakes, a major concern for automated structure discovery algorithms.

We claim that what can be learned from pure models alone can still be substantial. This is supported by the empirical results discussed in Section 6, and by various results on factor analysis that empirically demonstrate that, under an appropriate rotation, it is often the case that many observed variables have a single or few significant parents (Bartholomew et al., 2002), with a reasonably large pure measurement submodel. Substantive causal information can therefore be learned in practice using only pure models and the observed covariance matrix.

3.3.1 Measurement patterns

We say that a linear latent variable graph G *entails* a constraint if and only if the constraint holds in every distribution with covariance matrix parameterized by Θ, the set of linear coefficients and error variances that defines the conditional expectation and variance of a node given its parents. A *tetrad equivalence class* $T(\mathcal{C})$ is a set of latent variable graphs T, each member of which entails the same set of tetrad constraints $\mathcal{C}$ among the measured variables. An equivalence class of measurement models $\mathcal{M}(\mathcal{C})$ for $\mathcal{C}$ is the union of the measurement models in $T(\mathcal{C})$. We now introduce a graphical representation of common features of all elements of $\mathcal{M}(\mathcal{C})$.

Definition 3.5 (Measurement pattern) *A measurement pattern, denoted $MP(\mathcal{C})$, is a graph representing features of the equivalence class $\mathcal{M}(\mathcal{C})$ satisfying the following:*

- *there are latent and observed nodes;*
- *the only edges allowed in an MP are directed edges from latents to observed nodes, and undirected edges between observed nodes. Every observed node in a MP has at least one latent parent;*
- *if two observed nodes X and Y in a $MP(\mathcal{C})$ do not share a common latent parent, then X and Y do not share a common latent parent in any member of $\mathcal{M}(\mathcal{C})$;*
- *if X and Y are not linked by an undirected edge in $MP(\mathcal{C})$, then X is not an ancestor of Y in any member of $\mathcal{M}(\mathcal{C})$.*

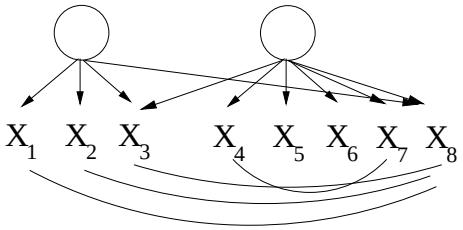


Figure 3.5: An example of a measurement pattern.

A measurement pattern does not make any claims about the connections between latents. We show an example in Figure 3.5. By the definition of measurement pattern, this graph claims that nodes X_1 and X_4 do not have any hidden common parent in common in any member of its equivalence class, which implies they do not have common hidden parents in the true unknown graph that generated the observable tetrad constraints. The same holds for any pair in $\{X_1, X_2\} \times \{X_4, X_5, X_6, X_7\}$.

It is also the case that, by the measurement pattern shown in Figure 3.5, X_1 cannot be an ancestor of X_2 in the true graph; X_1 cannot be an ancestor of X_4, and so on for all pairs that are not linked by an undirected edge.

Still in this measurement pattern, X_1 and X_2 *might* have a common hidden parent in the true graph. X_3 and X_4 might have a common hidden parent and so on. Also, X_4 might be an ancestor of X_7. X_1 might be an ancestor of X_8. It does not mean, however, that this is actually the case. Later in this chapter we show an example of a graph that generates this pattern by the algorithm given in the next section.

3.3.2 An algorithm for finding measurement patterns

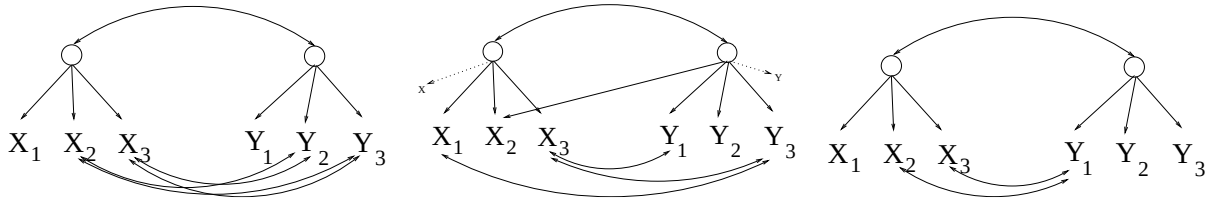
Assume for now that the population covariance matrix is known[1]. FindPattern, given in Table 3.1, is an algorithm to learn a measurement pattern. The first stage of FindPattern searches for subsets of $\mathcal{C}$ that will guarantee that two observed variables do not have any latent parents in common.

Let G be the latent variable graph for a linear latent variable model with a set of observed variables $\mathbf{O}$. Let $\mathbf{O}' = \{X_1, X_2, X_3, Y_1, Y_2, Y_3\} \subset \mathbf{O}$ such that for all triplets $\{A, B, C\}$, $\{A, B\} \subset \mathbf{O}'$ and $C \in \mathbf{O}$, we have $\rho_{AB} \neq 0, \rho_{AB.C} \neq 0$. Let τ_{IJKL} represent the tetrad constraint $\sigma_{IJ}\sigma_{KL} - \sigma_{IK}\sigma_{JL} = 0$ and $\neg\tau_{IJKL}$ represent the complementary constraint $\sigma_{IJ}\sigma_{KL} - \sigma_{IK}\sigma_{JL} \neq 0$. The following lemma is a formal description of the example given in Figure 3.3:

Lemma 3.6 (CS1 Test) *If constraints* $\{\tau_{X_1Y_1X_2X_3}, \tau_{X_1Y_1X_3X_2}, \tau_{Y_1X_1Y_2Y_3}, \tau_{Y_1X_1Y_3Y_2}, \neg\tau_{X_1X_2Y_2Y_1}\}$ *all hold, then* X_1 *and* Y_1 *do not have a common parent in* G.

"CS" here stands for "constraint set," the premises of a rule that can be used to test if two nodes do not share a common parent. Other sets of observable constraints can be used to reach the same conclusion. Let the predicate $F_1(X, Y, G)$ be true if and only if there exist two nodes W and Z in latent variable graph G such that τ_{WXYZ} and τ_{WXZY} are both entailed, all nodes in $\{W, X, Y, Z\}$ are correlated, and there is no observed C in G such that $\rho_{AB.C} = 0$ for $\{A, B\} \subset \{W, X, Y, Z\}$:

[1]Appendix A.3 describes how to deal with finite samples.



(a) (b) (c)

Figure 3.6: Three examples with two main latents and several independent latent common causes of two indicators (represented by double-directed edges). In (a), CS1 applies, but not CS2 nor CS3 (even when exchanging labels of the variables); In (b), CS2 applies (assuming the conditions for X_1, X_2 and Y_1, Y_2), but not CS1 nor CS3. In (c), CS3 applies, but not CS1 nor CS2.

Lemma 3.7 (CS2 Test) *If constraints* $\{\tau_{X_1Y_1Y_2X_2}, \tau_{X_2Y_1Y_3Y_2}, \tau_{X_1X_2Y_2X_3}, \neg\tau_{X_1X_2Y_2Y_1}\}$ *all hold such that* $F_1(X_1, X_2, G) = true$, $F_1(Y_1, Y_2, G) = true$, X_1 *is not an ancestor of* X_3 *and* Y_1 *is not an ancestor of* Y_3, *then* X_1 *and* Y_1 *do not have a common parent in* G.

Lemma 3.8 (CS3 Test) *If constraints* $\{\tau_{X_1Y_1Y_2Y_3}, \tau_{X_1Y_1Y_3Y_2}, \tau_{X_1Y_2X_2X_3}, \tau_{X_1Y_2X_3X_2}, \tau_{X_1Y_3X_2X_3}, \tau_{X_1Y_3X_3X_2}, \neg\tau_{X_1X_2Y_2Y_3}\}$ *all hold, then* X_1 *and* Y_1 *do not have a common parent in* G.

These rules are illustrated in Figure 3.6. Notice that those rules are not redundant: only one can be applied on each situation. For CS2 (Figure 3.6(b)), nodes X and Y are depicted as auxiliary nodes that can be used to verify predicates F_1. For instance, $F_1(X_1, X_2, G)$ is true because all three tetrads in the covariance matrix of $\{X_1, X_2, X_3, X\}$ hold.

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

Lemma 3.9 *If for some set* $\mathbf{O}' = \{X_1, X_2, X_3, X_4\} \subseteq \mathbf{O}$, $\sigma_{X_1X_2}\sigma_{X_3X_4} = \sigma_{X_1X_3}\sigma_{X_2X_4} = \sigma_{X_1X_4}\sigma_{X_2X_3}$ *and for all triplets* $\{A, B, C\}$, $\{A, B\} \subset \mathbf{O}', C \in \mathbf{O}$, *we have* $\rho_{AB.C} \neq 0$ *and* $\rho_{AB} \neq 0$, *then no element* $A \in \mathbf{O}'$ *is a descendant of an element of* $\mathbf{O}'\backslash\{A\}$ *in* G.

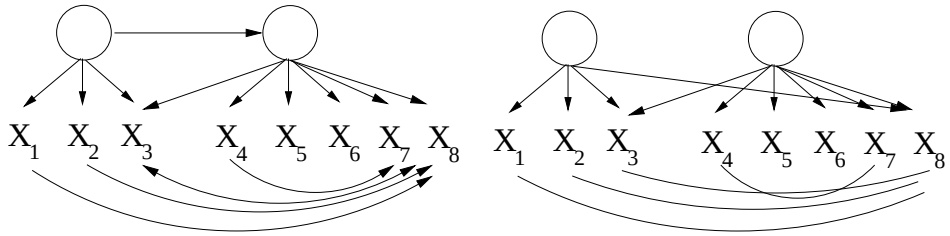
This lemma is a straightforward consequence of Lemma 3.4 and the assumption that no observed node is an ancestor of a latent node. For instance, in Figure 3.6(b) the existence of the observed node X (linked by a dashed edge to the parent of X_1) will allow us to infer that X_1 is not an ancestor of X_3, since all three tetrad constraints hold in the covariance matrix of $\{X, X_1, X_2, X_3\}$. Node Y plays a similar role with respect to Y_1 and Y_3.

Algorithm FindPattern has the following property:

Theorem 3.10 *The output of* FindPattern *is a measurement pattern* $MP(\mathcal{C})$ *with respect to the tetrad and zero/first order vanishing partial correlation constraints* $\mathcal{C}$ *of* Σ.

Figure 3.7 illustrates an application of FindPattern[2]. A full example of the algorithm is given

[2]Notice we do not make use of vanishing partial correlations where the size of the conditioning set is never greater than 1. We are motivated by problems where there is a strong belief that every pair of observed variables has at least one common hidden cause. Using such higher order constraints would just lead to higher possibility of commiting statistical mistakes.



(a) (b)

Figure 3.7: In (a), a model that generates a covariance matrix Σ. In (b), the output of FindPattern given Σ. Pairs in $\{X_1, X_2\} \times \{X_4, \ldots, X_7\}$ are separated by CS2. Notice that the presence of an undirected edge does not mean that adjacent nodes in the pattern are actually adjacent in the true graph (e.g., X_3 and X_8 share a common parent in the true graph, but are not adjacent). Observed nodes adjacent in the output pattern always share at least one parent in the pattern, but only sometimes they are actually children of a same parent (e.g., X_4 and X_7) in the true graph. Nodes sharing a common parent in the pattern might not share a parent in the true graph (e.g., X_1 and X_8).

in Figure 3.8.

3.3.3 Identifiability and purification

The FindPattern algorithm is sound, but not necessarily *complete*. That is, there might be graphical features shared by all members of the measurement model equivalence class that are not discovered by FindPattern. In general, a measurement pattern might not be informative enough, and this is the motivation for discovering pure measurement models: we would like to know in more detail how the latents in the output are related to the ones in the true graph. This is essential in order to find a corresponding structural model.

The output of FindPattern cannot, however, reliably be turned into a pure measurement model in the obvious way, by removing from it all nodes that have more than one latent parent and one of every pair of adjacent nodes, as attemped by the following algorithm:

- Algorithm TrivialPurification: remove all nodes that have more than one latent parent, and for every pair of adjacent observed nodes, remove an arbitrary node of the pair.

TrivialPurification is not correct. To see this, consider Figure 3.9(a), where with the exception of pairs in $\{X_3, \ldots, X_7\}$, every pair of nodes has more than one hidden common cause. Giving the covariance matrix of such model to FindPattern will result in a pattern with one latent only (because no pair of nodes can be separated by CS1, CS2 or CS3), and all pairs that are connected by a double directed edge in Figure 3.9(a) will be connected by an undirected edge in the output pattern. One can verify that if we remove one node from each pair connected by an undirected edge in this pattern, the output with the maximum number of nodes will be given by the graph in Figure 3.9(b).

There is no clear relation between the latent in the pattern and the latents in the true graph. While it is true that all nodes in $\{X_3, \ldots, X_7\}$ have a latent common cause (the parent of $\{X_4, X_5, X_6\}$)

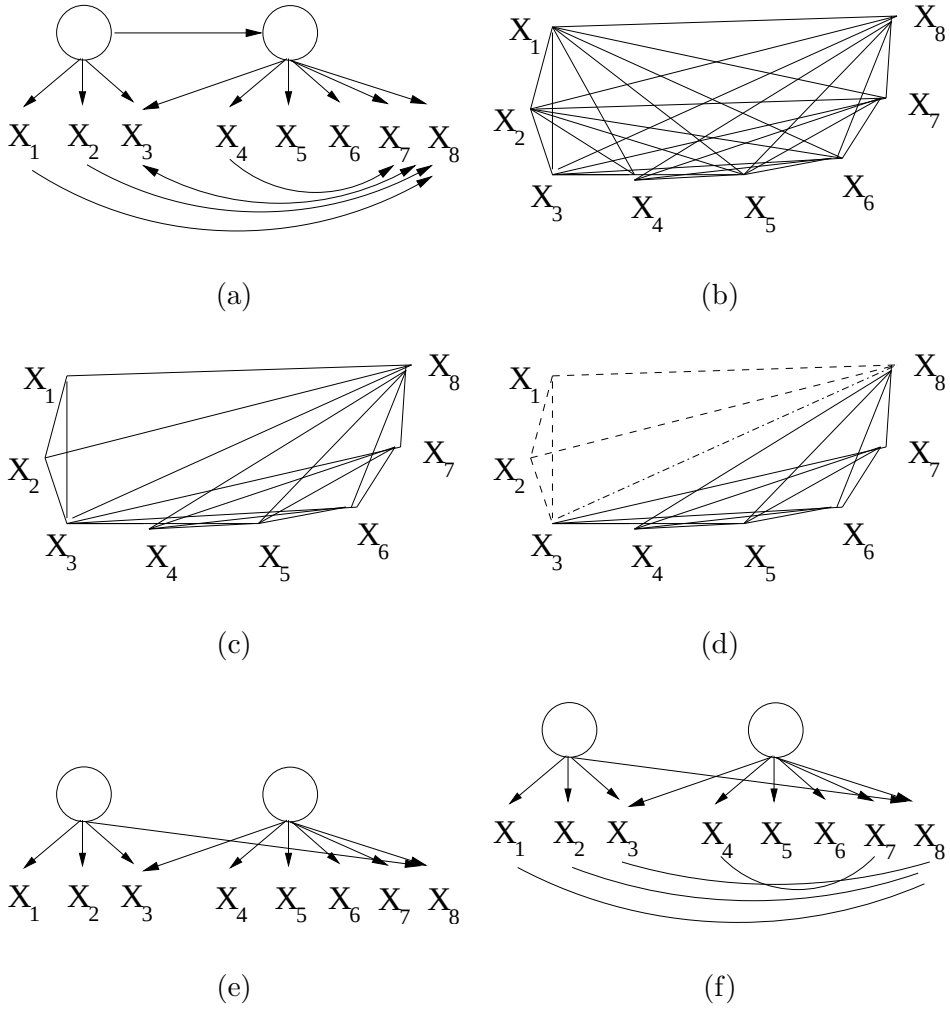


Figure 3.8: A step-by-step example on how a measurement pattern for the model given in (a) can be learned by FindPattern. Suppose we are given only the observed covariance matrix of the model in (a). We start with a fully connected graph among the observables (b), and remove some of the edges according to CS1-CS3. For instance, the edge $X_1 - X_4$ is removed by CS2 applied to the tuple $\{X_1, X_2, X_3, X_4, X_5, X_6\}$. This results in graph (c). In (d), we highlight the two different (and overlapping) maximal cliques found in this graph (edge $X_3 - X_8$ belongs to both cliques). The two cliques are transformed into two latents in (e). Finally, in (f) we add the required undirected edges (since, e.g., X_1 and X_8 are not part of any foursome where all three tetrad constraints hold).

Algorithm FindPattern
Input: a covariance matrix Σ

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

Table 3.1: Returns a measurement pattern corresponding to the tetrad and first order vanishing partial correlations of Σ.

in the true graph, such observed nodes cannot be causally connected by a linear model as suggested by Figure 3.9(b). In that graph, all three tetrads constraints among $\{X_3, X_4, X_5, X_7\}$ are entailed. This is not the case in the true graph.

Consider instead the algorithm BuildPureClusters of Table 3.2, which initially builds a measurement pattern using FindPattern. Variables are removed whenever some tetrad constraints are not satisfied, which corrects situations exemplified by Figure 3.9. Some extra adjustments concern clusters with proper subsets that are not consistently correlated to another variable (Steps 6 and 7) and a final merging of clusters (Step 8). We explain the necessity of these steps in Appendix A.1.

Notice that we leave out some details in the description of BuildPureClusters, i.e., there are severals ways of performing choices of nodes in Steps 2, 4, 5 and 9. We suggest an explicit way of performing these choices in Appendix A.3. There are two reasons why we present a partial description of the algorithm. The first is that, *independently of how such choices are made*, one can make several claims about the relationship of an output graph and the true measurement model. The graphical properties of the output of BuildPureClusters are summarized by the following theorem.

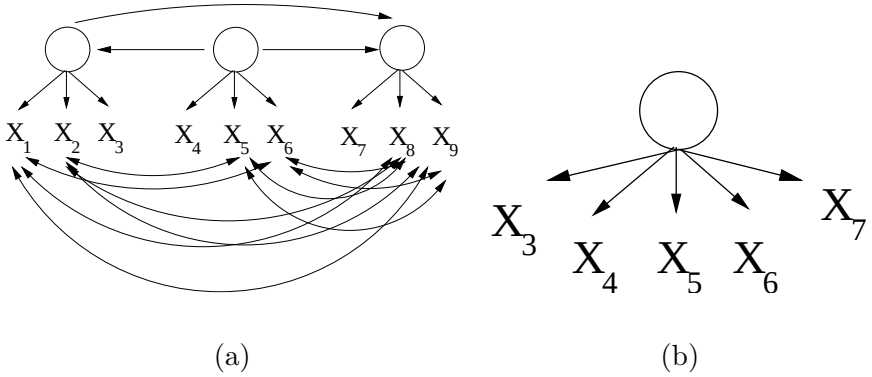
Theorem 3.11 *Given a covariance matrix Σ assumed to be generated from a linear latent variable model G with observed variables $\mathbf{O}$ and latent variables $\mathbf{L}$, let G_{out} be the output of* BuildPureClusters(Σ) *with observed variables $\mathbf{O_{out}} \subseteq \mathbf{O}$ and latent variables $\mathbf{L_{out}}$. Then G_{out} is a measurement pattern, and there is an unique injective mapping $M : \mathbf{L_{out}} \rightarrow \mathbf{L}$ with the following properties:*

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

Algorithm BuildPureClusters
Input: a covariance matrix Σ

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

Table 3.2: A general strategy to find a pure MP that is also a linear measurement model of a subset of the latents in the true graph. As explained in the body of the text, steps 2, 4, 5 and 9 are not described algorithmically in this Section.



(a) (b)

Figure 3.9: In (a), a model that generates a covariance matrix Σ. The output of FindPattern given Σ contains a single latent variable that is a parent of all observed nodes. In (b), the pattern with the maximum number of nodes that can be obtained by removing one node from each adjacent pair of observed nodes. This model is incorrect, since there is no latent that d-separates all of the nodes in (b) in a linear model.

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

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

Informally, there is a labeling of latents in G_{out} according to the latents in G, and in this relabeled output graph any d-separation between a measured node and some other node will hold in the true graph, G. This is illustrated by Figure 3.10. Given the covariance matrix generated by the true model in Figure 3.10(a), BuildPureClusters generates the model shown in Figure 3.10(b). Since the labeling of the latents is arbitrary, Theorem 3.11 is a formal description that latents in the output should correspond to latents in the true model up to a relabeling.

For each group of correlated observed variables, we can guaranteee that at most one edge from a latent into an observed variable is incorrectly directed. By "incorrectly directed," we mean the condition defined in the third item of Theorem 3.11: although all observed variables are children of latents in the output graph, one of these edges might be misleading, since in the true graph one of the observed variables might not be a descendant of the respective latent. This is illustrated by Figure 3.11.

Notice also that we cannot guarantee that an observed node X with latent parent L_{out} in G_{out} will be d-separated from the latents in G not in G_{out}, given $M(L_{out})$: if X has a common cause with $M(L_{out})$, then X will be d-connected to any ancestor of $M(L_{out})$ in G given $M(L_{out})$. This is also illustrated by Figure 3.11.

Let an DAG G be an *I-map* of a distribution D if and only if all independencies entailed in G by the Markov condition also hold in D (the faithfulness condition explained in Chapter 1 includes the converse) (Pearl, 1988). Using the notation from the previous theorem, the parametrical properties of the output of BuildPureClusters are described as follows:

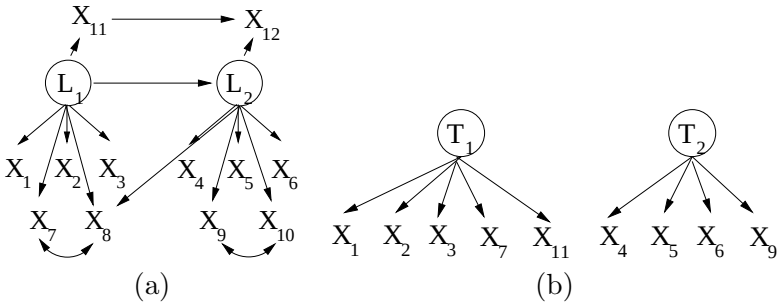


Figure 3.10: Given as input the covariance matrix of the observable variables $X_1 - X_{12}$ connected according to the true model shown in Figure (a), the BuildPureClusters algorithm will generate the graph shown in Figure (b). It is clear there is an injective mapping $M(.)$ from latents $\{T_1, T_2\}$ to latents $\{L_1, L_2\}$ such that $M(T_1) = L_1$ and $M(T_2) = L_2$ and the properties described by Theorem 3.11 hold.

Theorem 3.12 *Let $M(\mathbf{L_{out}}) \subseteq \mathbf{L}$ be the set of latents in G obtained by the mapping function $M()$. Let $\Sigma_{\mathbf{O_{out}}}$ be the population covariance matrix of $\mathbf{O_{out}}$. Let the DAG G^{aug}_{out} be G_{out} augmented by connecting the elements of $\mathbf{L_{out}}$ such that the structural model of G^{aug}_{out} is an I-map of the distribution of $M(\mathbf{L_{out}})$. Then there exists a linear latent variable model using G^{aug}_{out} as the graphical structure such that the implied covariance matrix of $\mathbf{O_{out}}$ equals $\Sigma_{\mathbf{O_{out}}}$.*

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

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

Theorem 3.13 *Problem $\mathcal{MP}^3$ is NP-complete.*

By presenting the high-level description of BuildPureClusters as in Table 3.2, we show there is no need to solve a NP-hard problem in order to have the same theoretical guarantees of interpretability of the output. For example, there is a stage in FindPattern where it appears necessary to find all maximal cliques, but, in fact, it is not. Identifying more cliques increases the chance of having a larger output (which is good) by the end of the algorithm, but it is not required for the algorithm's correctness. Stopping at Step 5 of FindPattern after a given amount of time will not affect Theorems 3.11 or 3.12.

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

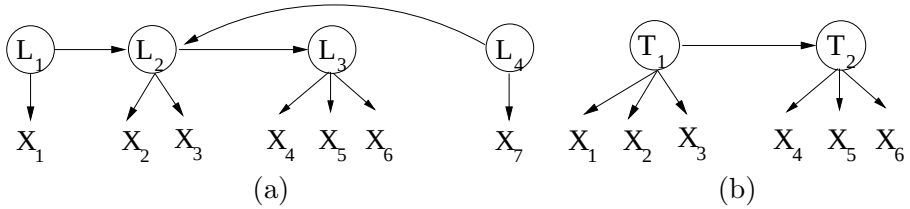


Figure 3.11: Given as input the covariance matrix of the observable variables $X_1 - X_7$ connected according to the true model shown in Figure (a), one of the possible outputs of BuildPureClusters algorithm is the graph shown in Figure (b). It is clear there is an injective mapping $M(.)$ from latents $\{T_1, T_2\}$ to latents $\{L_1, L_2, L_3, L_4\}$ such that $M(T_1) = L_2$ and $M(T_2) = L_3$. However, in (b) the edge $T_1 \rightarrow X_1$ does not express the correct causal direction of the true model. Notice also that X_1 is not d-separated from L_4 given $M(T_1) = L_2$ in the true graph.

Corollary 3.14 *The output of* BuildPureClusters *retains its guarantees even when rules CS1, CS2 and CS3 are applied an arbitrary number of times in* FindPattern *for any arbitrary subset of nodes and an arbitrary number of maximal cliques is found.*

3.3.4 Example

In this section, we illustrate how BuildPureClusters works given the population covariance matrix of a known latent variable model. Suppose the true graph is the one given in Figure 3.12(a), with two unlabeled latents and 12 observed variables. This graph is unknown to BuildPureClusters, which is given only the covariance matrix of variables $\{X_1, X_2, ..., X_{12}\}$. The task is to learn a measurement pattern, and then a purified measurement model.

In the first stage of BuildPureClusters, the FindPattern algorithm, we start with a fully connected graph among the observed variables (Figure 3.12(b)), and then proceed to remove edges according to rules CS1, CS2 and CS3, giving the graph shown in Figure 3.12(c). There are two maximal cliques in this graph: $\{X_1, X_2, X_3, X_7, X_8, X_{11}, X_{12}\}$ and $\{X_4, X_5, X_6, X_8, X_9, X_{10}, X_{12}\}$. They are distinguished in the figure by different edge representations (dashed and solid - with the edge $X_8 - X_{12}$ present in both cliques). The next stage takes these maximal cliques and creates an intermediate graphical representation, as depicted in Figure 3.12(d). In Figure 3.12(e), we add the undirected edges $X_7 - X_8$, $X_8 - X_{12}$, $X_9 - X_{10}$ and $X_{11} - X_{12}$, finalizing the measurement pattern returned by FindPattern. Finally, Figure 3.12(f) represents a possible purified output of BuildPureClusters given this pattern. Another purification with as many nodes as in the graph in Figure 3.12(f) substitutes node X_9 for node X_{10}.

3.4 Learning the structure of the unobserved

Even given a correct measurement model, it might not be possible to identify the corresponding structural model. Consider the case of factor analysis again, applied to multivariate normal models. In Figure 1.6 we depicted two graphs that are both able to represent a same set of normal distributions.

One might argue that this is an artifact of the Gaussian distribution, and identifiability could be improved by assuming other distributions other than normal for the given variables. However,

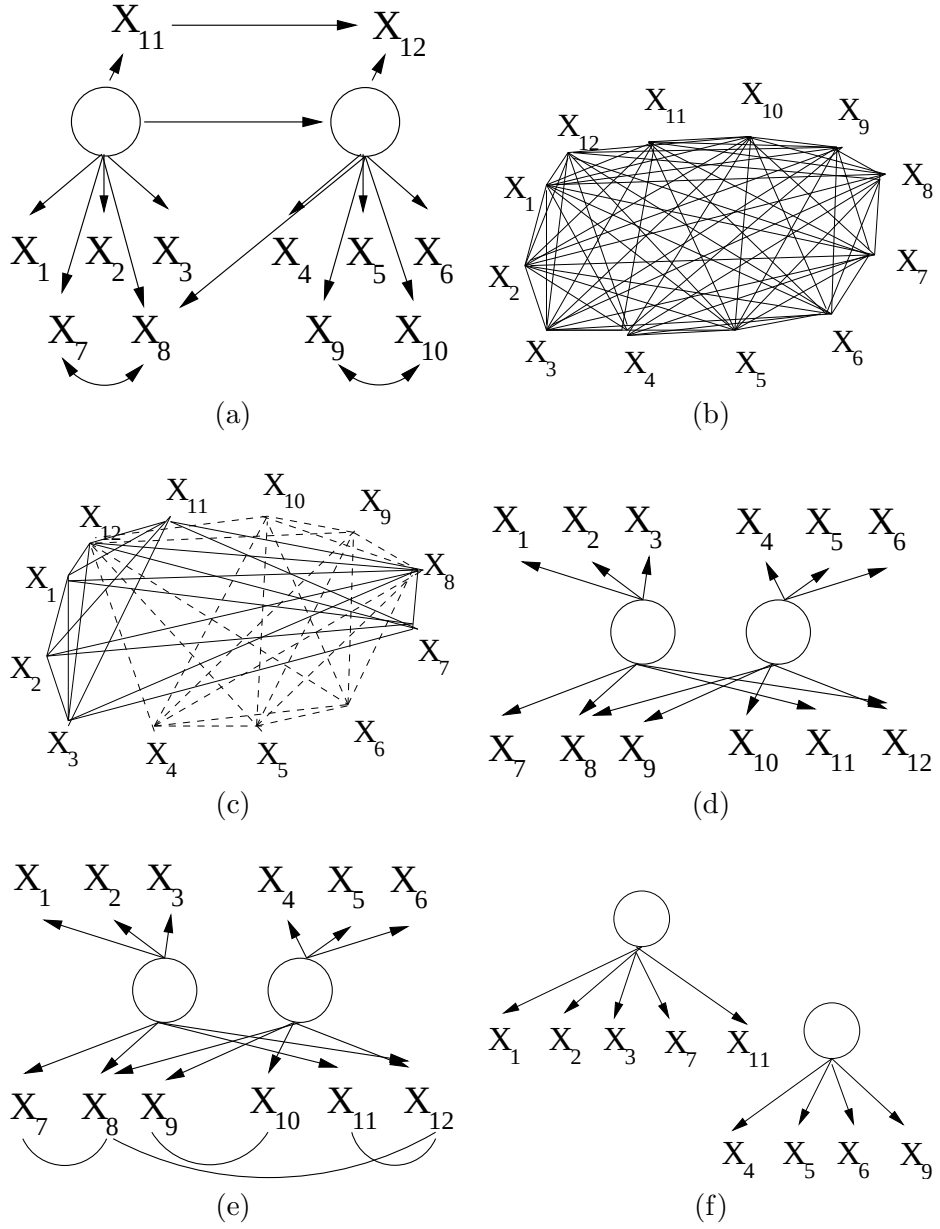


Figure 3.12: A step-by-step demonstration of how a covariance matrix generated by graph in Figure (a) will induce the pure measurement model in Figure (f).

for linear models, Gaussian distributions are an important case that cannot be ignored. Moreover, it is difficult to design identification criteria that are both computationally feasible (e.g., avoiding the minimization of a KL-divergence) and statistically realistic (how much fine-grained information about the distribution, such as high-order moments, could be reliably used in model selection?)

We take an approach that we believe to be much more useful in practice: to guarantee identifiability of the structural model by constraining the acceptable measurement models used as input, and do it without requiring high-order moments. We will from now assume the following condition

for our algorithm:

- the given measurement model has a pure measurement submodel with at least two measures per latent;

Notice that does not mean that the given measurement model has to be pure, but only a subset of it[3] has to be pure. The intuition about the suitability of this assumption is as follows: in pure measurement models, d-separation among latents entails d-separation among pure observed measures, and that has immediate consequences on the rank of the covariance matrix of the d-separated observed variables.

3.4.1 Identifying conditional independences among latent variables

The following theorem is due to Spirtes et al. (2000):

Theorem 3.15 *Let G be a pure linear latent variable model. Let L_1, L_2 be two latents in G, and $\mathbf{Q}$ a set of latents in G. Let X_1 be a measure of L_1, X_2 be a measure of L_2, and $X_{\mathbf{Q}}$ be a set of measures of $\mathbf{Q}$ containing at least two measures per latent. Then L_1 is d-separated from L_2 given $\mathbf{Q}$ in G if and only if the rank of the correlation matrix of $\{X_1, X_2\} \cup \mathbf{X_Q}$ is less than or equal to $|\mathbf{Q}|$ with probability 1 with respect to the Lebesgue measure over the linear coefficients and error variances of G.*

We can then use this constraint to identify[4] conditional independencies among latents provided we have the correct pure measures.

3.4.2 Constraint-satisfaction algorithms

Given Theorem 3.15, conditional independence tests can then be used as an oracle for constraint-satisfaction techniques for causality discovery in graphical models, such as the PC algorithm (Spirtes et al., 2000) which assumes the variables being tested to have no "unmeasured hidden common causes" (i.e., in our case, no pair of latents in our system can have another latent as a common cause that is not measured by some observed variable). An alternative is the FCI algorithm (Spirtes et al., 2000), which makes no such an assumption.

We define the algorithm PC-MIMBuild[5] as the algorithm that takes as input a measurement model satisfying the assumption of purity mentioned above and a covariance matrix, and returns the Markov equivalence of the structural model among the latents in the measurement model according to the PC algorithm. A FCI-MIMBuild algorithm is defined analogously. In the limit of infinite data, the following result follows from Theorems 3.11 and 3.15 and the consistency of PC and FCI algorithms (Spirtes et al., 2000):

[3]The definition of measurement submodel has to preserve all ancestral relationships. So, if measure X is not a parent of Y, but a chain $X \to K \to Y$ exists, any submodel that includes X and Y but not K has to include the edge $X \to Y$.

[4]One way to test if the rank of a covariance matrix in Gaussian models is at most q is to fit a factor analysis model with q latents and assess its significance (Bartholomew and Knott, 1999).

[5]MIM stands for "multiple indicator model", a term in structural equation model literature describing latent variable models with multiple measures per latent.

Corollary 3.16 *Given a covariance matrix Σ assumed to be generated from a linear latent variable model G, and G_{out} the output of* BuildPureClusters *given Σ, the output of* PC-MIMBuild *or* FCI-MIMBuild *given (Σ, G_{out}) returns the correct Markov equivalence class of the latents in G corresponding to latents in G_{out} according to the mapping implicit in* BuildPureClusters.

An example of the PC algorithm in action is given in Chapter 1. Exactly the same procedure could be applied to a graph consisted of latent variables, as illustrated in Figure 3.13. This example corresponds to the one given in Figure 1.5.

3.4.3 Score-based algorithms

Given Theorem 3.15, conditional independence constraints can then be used as search operators for score-based techniques for causality discovery in graphical models. Score-based approaches for learning the structure of Bayesian networks, such as GES (Meek, 1997), are usually more robust to variability on small samples than PC or FCI. If one is willing to assume that there are no extra hidden common causes connecting variables on its causal system, then GES should be a more robust choice than the PC algorithm.

We know of no consistent score function for linear latent variable models that can be easily computed. As a heuristic, we suggest using the Bayesian Information Criterion (BIC) function. Using BIC along with Structural EM (Friedman, 1998) and GES results in a very computationally efficient way of learning structural models, where the measurement model is fixed and GES is restricted to modify edges among latents only. Assuming a Gaussian distribution, the first step of Structural EM uses a fully connected structural model in order to estimate the first expected latent covariance matrix. We call this algorithm GES-MIMBuild and use it as the structural model search component in all the algorithms we now compare.

3.5 Evaluation

We evaluate our algorithm on simulated and real data. In the simulation studies, we draw samples of three different sizes from 9 different latent variable models involving three different structural models and three different measurement models. We then consider the algorithm's performance on three empirical datasets: one involving stress, depression, and spirituality; one concerning attitude of single mothers with respect to their children; and one involving test anxiety previously analyzed with factor analysis in (Bartholomew et al., 2002).

3.5.1 Simulation studies

We compare our algorithm against two versions of exploratory factor analysis, and measure the success of each on the following discovery problems, as previously defined:

DP1. Discover the number of latents in G.

DP2. Discover which observed variables measure each latent G.

DP3. Discover causal structure among the latents in G.

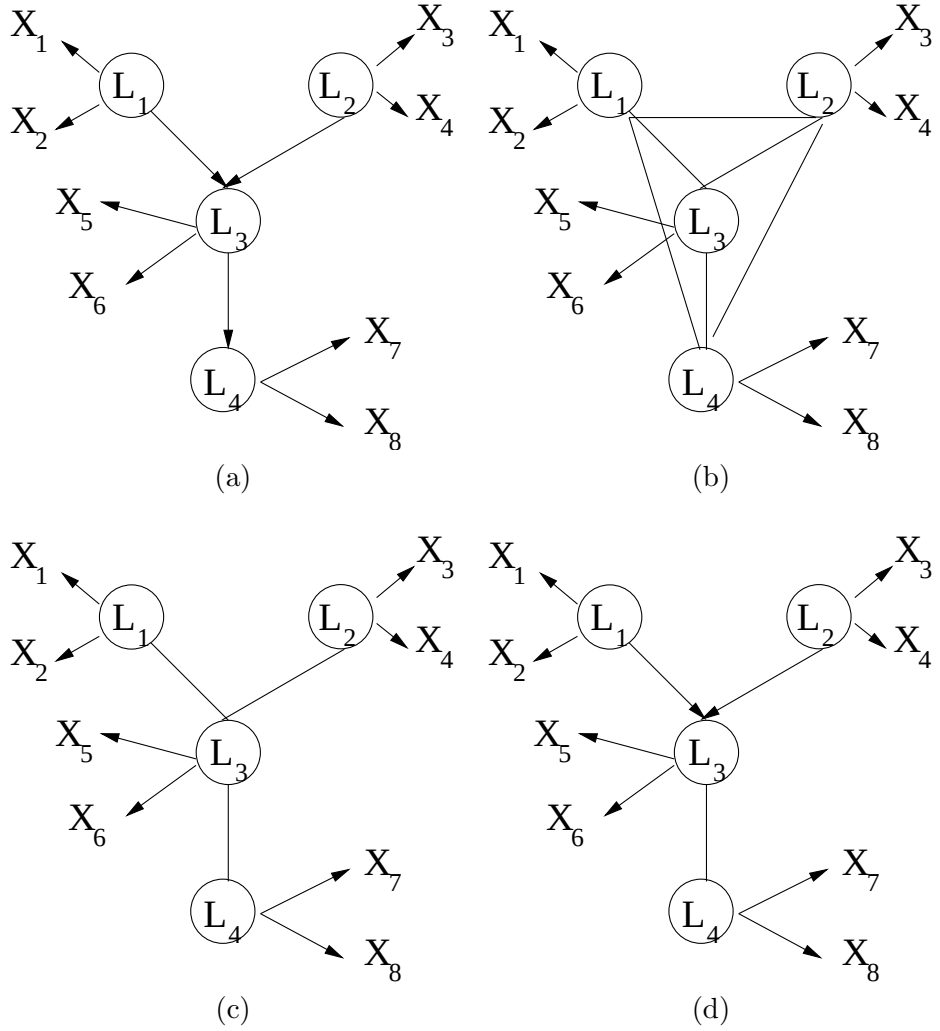


Figure 3.13: A step-by-step demonstration of the PC-MIMBuild algorithm. The true model is given in (a). We start with a full undirected graph among latents (b) and remove edges according to the independence tests described in Section 3.4.1, obtaining graph (c). By orienting unshielded colliders, we get graph (d). Extra steps of orientation will recreate the true graph. An identical example of the PC algorithm for the case where the variables of interest are observed is given in Figure 1.5.

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

We hypothesized that three features of the problem would affect the performance of the algorithms compared. First, the sample size should be important. Second, the complexity of the

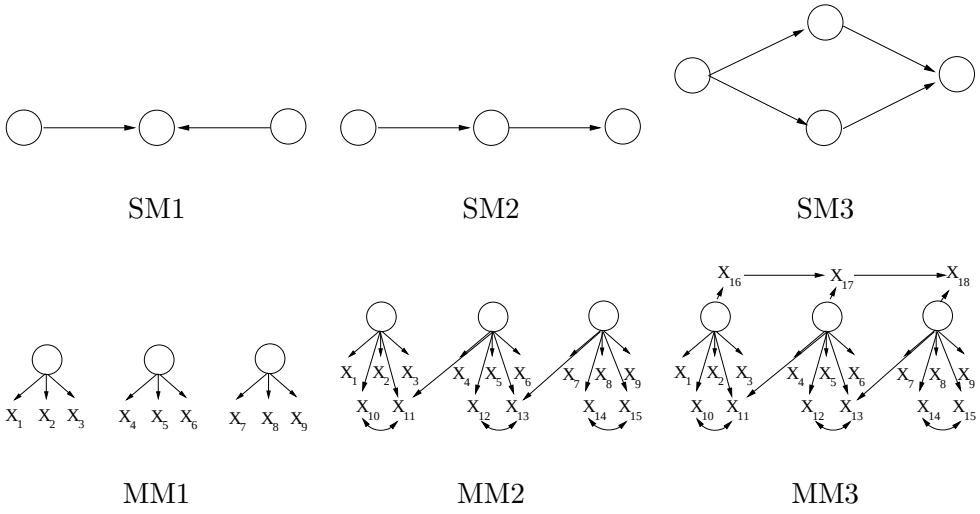


Figure 3.14: The structural and measurement models used in our simulation studies. When combining the 4-latent structural model SM3 with any measurement model, we add edges out of the fourth latent respecting the pattern used in the measurement model.

structural model might matter, and third, the complexity and level of impurity in the generating measurement model might matter. We used three different sample sizes for each study: 200, 1,000, and 10,000. We constructed nine generating latent variable graphs by using all combinations of the three structural models and three measurement models we show in Figure 3.14.

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

SM1 entails one unconditional independence among the latents: L1 is independent L3. SM2 entails one first order conditional independence: L1 $\perp$ L3 | L2, and SM3 entails one first order conditional independence: L2 $\perp$ L3 | L1, and one second order conditional independence relation: L1 $\perp$ L4 | {L2, L3}. Thus the statistical complexity of the structural models increases from SM1 to SM3.

Clearly any discovery procedure ought to be able to do very well on samples of 10,000 drawn from a generating model involving SM1 and MM1. Not as clear is how well a procedure can do on samples of size 200 drawn from a generating model involving SM3 and MM3.

Generating Samples

For each generating latent variable graph, we used the Tetrad IV program[6] with the following procedure to draw 10 multivariate normal samples of size 200, 10 at size 1,000, and 10 at size 10,000.

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

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

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

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

Algorithms Studied

We used three algorithms in our studies:

1. BPC: BuildPureClusters + GES-MIMBuild

2. FA: factor analysis + GES-MIMBuild

3. P-FA: factor analysis + Purify + GES-MIMBuild

BPC is the implementation of BuildPureClusters and GES-MIMBuild described in Appendix A.3. FA involves combining standard factor analysis to find the measurement model with GES-MIMBuild to find the structural model. For standard factor analysis, we used `factanal` from R 1.9 with the oblique rotation promax. FA and variations are still widely used and are perhaps the most popular approach to latent variable modeling (Bartholomew et al., 2002). We choose the number of latents by iteratively increasing its number till we get a significant fit above 0.05, or till we have to stop due to numerical instabilities[7].

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

Success on finding latents and a good measurement model

In order to compare the output of BPC, FA, and P-FA on discovery tasks DP1 (finding the correct number of underlying latents) and DP2 (measuring these latents appropriately), we must map the latents discovered by each algorithm to the latents in the generating model. That is, we must define a mapping of the latents in the G_{out} to those in the true graph G. Although one could do this in many ways, for simplicity we used a majority voting rule in BPC and P-FA. If a majority of the indicators of a latent L^i_{out} in G_{out} are measures of a latent node L^j in G, then we map L^i_{out} to L^j. Ties were in fact rare, and broken randomly. In this case, the latent that did not get the new label keeps a random label unrelated to latents in G. At most one latent in G_{out} is mapped to a fixed latent L in G, and if a latent in G had no majority in G_{out}, it was not represented in G_{out}.

The mapping for FA was done slightly differently. Because the output of FA is typically an extremely impure measurement model with many indicators loading on more than one latent, the simple minded majority method generates too many ties. For FA we do the mapping not by

[7]That is, where Heywood cases (Bartholomew and Knott, 1999) happened during fitting for 20 random re-starts. In this case, we just used the previous number of latents where Heywood cases did not happen.

majority voting of indicators according to their true clusters, but by verifying which true latent corresponds to the highest sum of absolute values of factor loadings for a given output latent. For example, let L_{out} be a latent node in G_{out}. Suppose S_1 is the sum of the absolute values of the loadings of L_{out} on measures of the true latent L_1 only, and S_2 is the sum of the absolute values of the loadings of L_{out} on measures of the true latent L_2 only. If $S_2 > S_1$, we rename L_{out} as L_2. If two output latents are mapped to the same true latent, we label only one of them as the true latent by choosing the one that corresponds to the highest sum of absolute loadings. The remaining latent receives a random label.

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

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

To be generous to factor analysis, we considered in FA outputs only latents with at least three indicators[8]. Again, to be conservative, we calculate the **misclustered indicators** error in the same way as in BuildPureClusters or P-FA, but here an indicator is not counted as mistakenly clustered if it is a child of the correct latent, even if it is *also* a child of a wrong latent.

Simulation results are given in Tables 3.3 and 3.4, where each number is the average error across 10 trials with standard deviations in parenthesis. Notice there are at most two maximal pure measurement models for each setup (there are two possible choices of which measures to remove from the last latent in MM_2 and MM_3) and for each G_{out} we choose our gold standard G as a maximal pure measurement submodel that contains the most number of nodes found in G_{out}. Each result is an average over 10 experiments with different parameter values randomly selected for each instance and three different sample sizes (200, 1000 and 10000 cases).

Table 3.3 evaluates all three procedures on the first two discovery tasks: DP1 and DP2. As predicted, all three procedures had very low error rates in rows involving MM1 and sample sizes of 10,000. In general, FA has very low rates of latent omission, but very high rates of latent commission, and P-FA, not surprisingly, does the opposite: very high rates of latent omission but very low rates of commission. In particular, FA is very sensitive to the purity of the generating measurement model. With MM2, the rate of latent commission for FA was moderate; with MM3 it was disastrous. BPC does reasonably well on all measures in Tables 3.3 at all sample sizes and for all generating models.

[8]Even with this help, we still found several cases in which latent commission errors were more than 100%, indicating that there were more spurious latents in the output graphs than latents in the true graph.

Table 3.4 gives results regarding indicator ommissions and commission, which, because FA keeps the original set of indicators it is given, only make sense for BPC and P-FA. P-FA omits far too many indicators, a behavior that we hypothesize will make it difficult for GES-MIMBuild on the measurement model output by P-FA.

Success on finding the structural model

As we have said from the outset, the real goal of our work is not only to discover the latent variables that underly a set of measures, but also the causal relations among them. In the final piece of the simulation study, we applied the best causal model search algorithm we know of, GES, modified for this purpose as GES-MIMBuild, to the measurement models output by BPC, FA, and P-FA.

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

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

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

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

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

and SM_3, 0 in SM_2);

- **orientation commission (OC)**, the number of arrows in the structural model of G_{out} that do not exist in G divided by the number of edges in the structural model of G_{out};

We have bent over, not quite backwards, to favor variations of factor analysis. Tables 3.5 and 3.6 summarize the results. Along with each average we provide the number of trials where no errors of a specific type were made. Althought it is clear from Tables 3.5 and 3.6 that factor analyis works well when the true models are pure, in general factor analysis commits way more errors of edge commission, since the presence of several spurious latents create spurious dependence paths. As a consequence, several orientation omissions follow. Under the same statistics, P-FA seems to work better than FA, but this is an artifact of P-FA having less latents on average than the other methods.

Figures 3.15 and 3.16 illustrate. Each picture contains a plot of the average edge error of each algorithm (i.e., the average of all four error statistics from Tables 3.5 and 3.6) with several points per algorithm representing different sample sizes or different measurement models, and is evaluated for a specific combination of structural model (SS_2). The pattern for the other two simulated structural models is similar.

The optimal performance is the bottom left. It is clear that P-FA achieves relatively high accuracy solely because of high percentage of latent omission. This pattern is similar across all structural models. Notice that FA is quite competitive when the true model is pure. BuildPureClusters tends to get lower latent omission error with the more complex measurement models (Figure 3.15) because the higher number of pure indicators in those situations helps the algorithm to identify each latent.

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

3.5.2 Real-world applications

We now discuss results obtained in three different domains in social sciences and psychology. Even though data collected from such domains (usually through questionnaires) may pose significant problems for exploratory data analysis since sample sizes are usually small and noisy, nevertheless they have a very useful property for our empirical evaluation: questionnaires are designed to target specific latent factors (such as "stress", "job satisfaction", and so on) and a theoretical measurement model is developed by experts in the area to measure the desired latent variables, thus providing a basis for comparison with the output of our algorithm. The chance that various observed variables are not pure measures of their theoretical latents is high. Measures are usually discrete, but often ordinal with a Likert-scale that can be treated as normally distributed measures with little loss (Bollen, 1989).

The theoretical models contain very few latents, and therefore are not as useful to evaluate MIMBuild as they are to BuildPureClusters.

Student anxiety factors: A survey of test anxiety indicators was administered to 335 grade 12 male students in British Columbia. The survey consisted of 20 measures on symptoms of anxiety

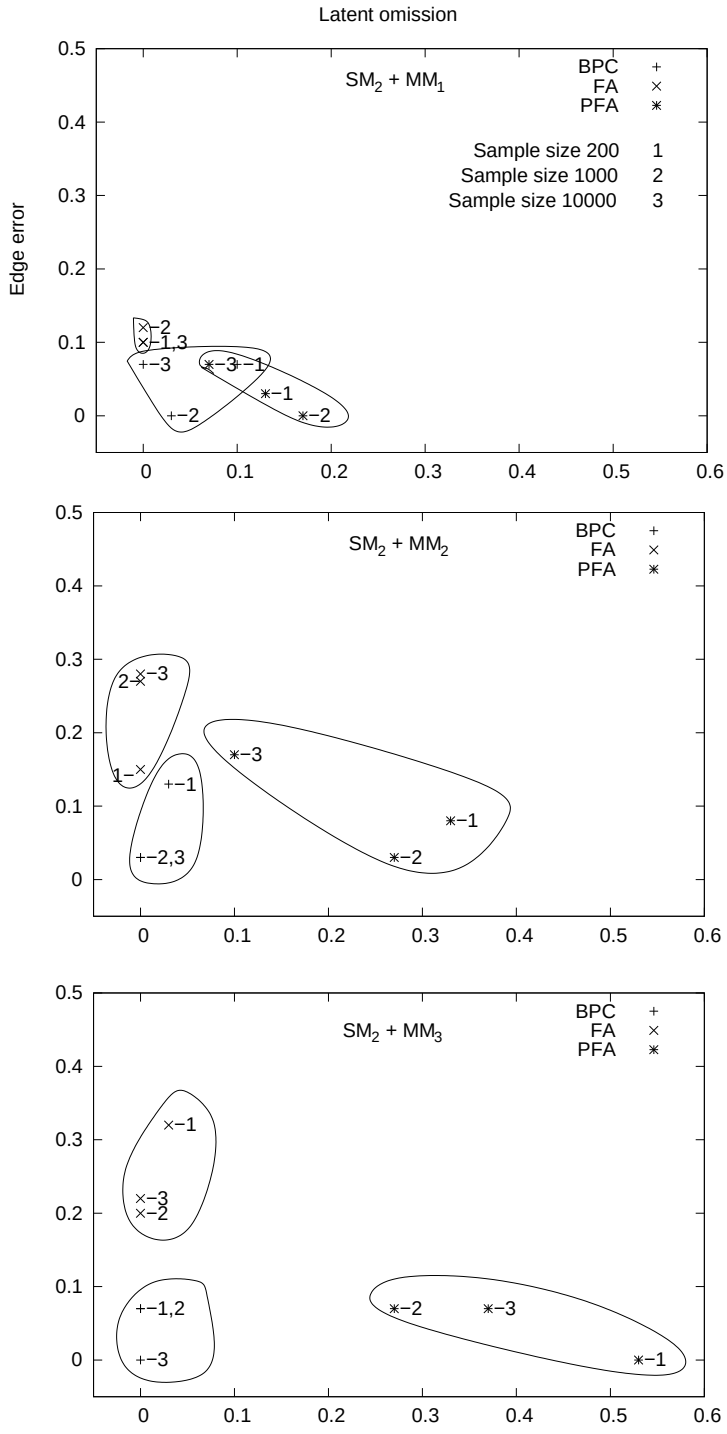


Figure 3.15: Comparisons of methods on measurement models of increasing complexity (from MM_1 to MM_3). While BPC tends to have low error on both dimensions (latent omission and edge error), the other two methods fail on either one.

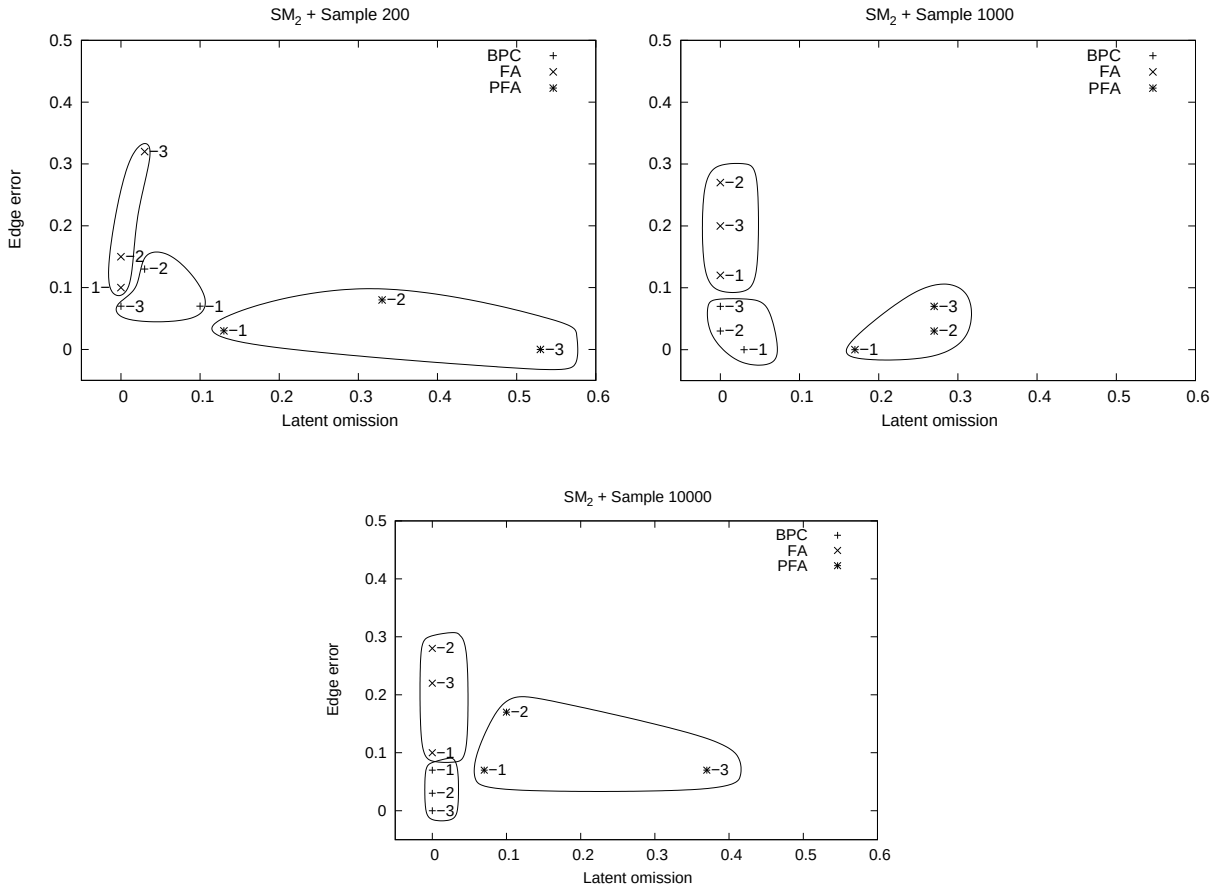


Figure 3.16: Comparisons of methods on increasing sample sizes. BPC has low error even at small sample sizes, while the other two methods show an apparent bias that does not go away with very large sample size.

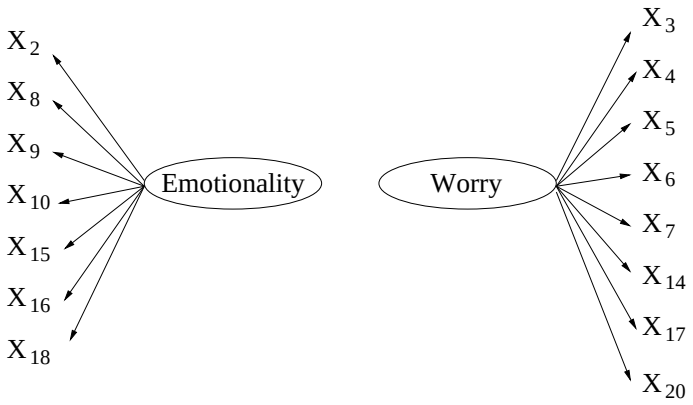


Figure 3.17: A theoretical model for psychological factors of test anxiety.

under test conditions. The covariance matrix as well as a description of the variables is given by Bartholomew et al. (2002)[9].

Using exploratory factor analysis, Bartholomew concluded that two latent common causes underly the variables in this data set, agreeing with previous studies. The original study identified items $\{x_2, x_8, x_9, x_{10}, x_{15}, x_{16}, x_{18}\}$ as indicators of an "emotionality" latent factor (this includes physiological symptoms such as jittery and faster heart beatting), and items $\{x_3, x_4, x_5, x_6, x_7, x_{14}, x_{17}, x_{20}\}$ as indicators of a more psychological type of anxiety labeled "worry" by Bartholomew et al. No further description is given about the remaining five variables. Bartholomew et al.'s factor analysis with oblique rotation roughly matches this model. Bartholomew's exploratory factor analysis model for a subset of the variables is shown in Figure 3.17. This model is not intended to be pure. Instead, the figure represents which of the two latents is more "strongly" connected to each indicator. The measurement model itself is not constrained.

We ran our algorithm 10 times with different random orderings of variables and we got always the same following measurement model:

1. $x_2, x_8, x_9, x_{10}, x_{11}, x_{16}, x_{18}$
2. x_3, x_5, x_7
3. x_6, x_{14}

Interestingly, the largest cluster closely corresponds to the "emotionality" factor as described by previous studies. The remaining two clusters are a split of "worry" into two subclusters with some of the original variables eliminated. Variables in the second cluster are only questions that explicitly describe "thinking" about sucess/failure (the only other question in the survey with the same characteristic was x_{17} which was eliminated). Variables x_6 and x_{14} are can be interpreted as indicating self-defeat.

The BuildPureClusters measurement model, with our interpretation of the latents based on the questionnaire contents, is shown in Figure 3.18(a).

If we treat Bartholomew's model as a pure model (as in Figure 3.17 with correlated latents, the result is a model that fails a chi-square test, p = 0. The full factor analysis of this dataset fits the

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

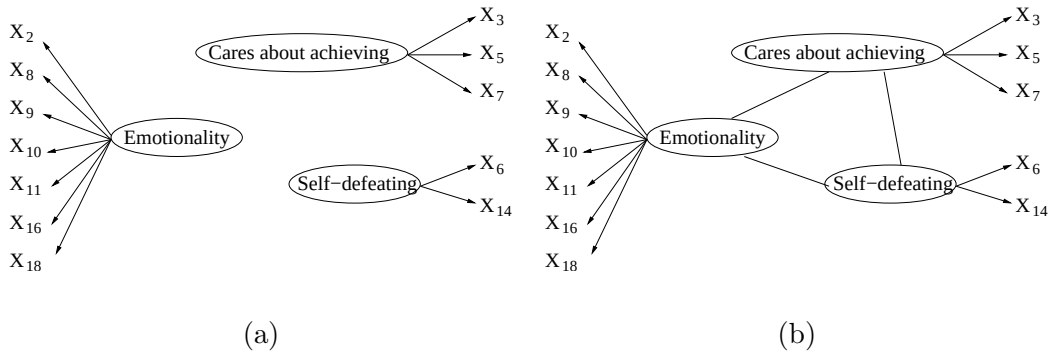


Figure 3.18: The output of BPC and GES-MIMBuild for the test anxiety study.

data with a p-value greater than 0.05, but it requires that many of the indicators have significant loadings on both latents. There might be no simple principled way to explain why such loadings are necessary. They may be due to direct effects of one variable on another, or due to other latent factors independent of the two conjectured. Besides that, the significance of such coefficients is tied to the ad-hoc rotation method employed in order to obtain a "simple structure".

Applying GES-MimBuild to the BPC measurement model of Figure 3.18(a) we obtain Figure 3.18(b). The model passes a chi square test handily, p = 0.47.

To summarize, by dropping only 3 out of 15 previously classified variables (among a total of 20 variables), our algorithm built a better fitting measurement model that is simpler to understand. The algorithm used absolutely no domain-specific prior knowledge, and did not rely in any way on ad-hoc rotation methods.

Well-being and spiritual coping: Bongjae Lee from the University of Pittsburgh conducted a study of religious/spiritual coping and stress in graduate students. In December of 2003, 127 Masters in Social Works students answered a questionnaire intendent to measure three main factors:

- *stress*, measured with 21 items, each using a 7-point scale (from "not all stressful" to "extremely stressful") according to situations such as: "fulfilling responsabilities both at home and at school"; "meeting with faculty"; "writing papers"; "paying monthly expenses"; "fear of failing"; "arranging childcare";

- *well-being*, measured with 20 items, each using a 4-point scale (from "rarely or none" to "most or all the time") according to indicators such as: "my appetite was poor"; "I felt fearful"; "I enjoyed life" "I felt that people disliked me"; "my sleep was restless";

- *religious/spiritual coping*, measured with 20 items, each using a 4-point scale (from "not at all" to "a great deal") according to indicators such as: "I think about how my life is part of a larger spiritual force"; "I look to God (high power) for strength in crises"; "I wonder whether God (high power) really exists"; "I pray to get my mind off of my problems";

As an illustration, the full questionnaire is given in Appendix A. Theoretical latents are not necessarily unidimensional, i.e., they might be partioned into an unknown set of sublatents and

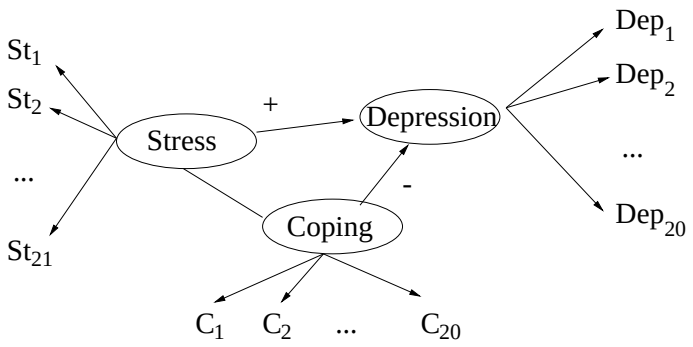


Figure 3.19: A theoretical model for the interaction of religious coping, stress and depression. The signs on the edges depicts the theoretical signs of the corresponding effects.

their indicators might be impure, but there was no prior knowledge about which impurities might exist.

The goal of the original study was to use graphical models to quantify how spiritual coping moderates the association of stress and well-being. Lee's model is shown in Figure 3.19. The undirected edge means lack of knowledge about causal directionality. This model fails a chi square test: its p-value is zero.

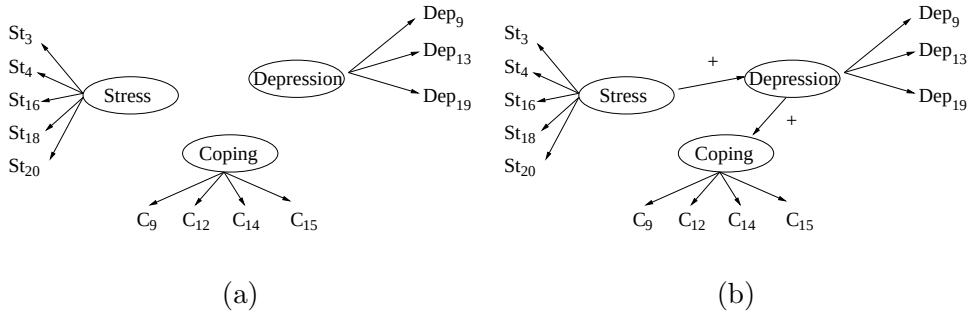
Our goal in this analysis is to verify if we get a clustering consistent with the theoretical measurement model (i.e., questions related to different topics will not end up in a same cluster), and analyse how questions are partioned within each theoretical cluster (i.e., how a group of questions related to the same theoretical latent ended up divided in different subclusters) using no prior knowledge.

The algorithm was applied 10 times with a different random choice of variable ordering each time. On average we got 18.2 indicators (standard deviation of 1.8). Clusters with only one variable were excluded. On average, 5.5 latents were discovered (standard deviation of 0.85). Counting only latents with at least three indicators, we had on average 4 latents (standard deviation of 0.67). In comparison, using the theoretical model as an initial model and by applying purification directly[10], i.e. without automated clustering, we obtained 15 variables (8 indicators of stress, 4 indicators of coping and 3 indicators of depression). We should not expect to do much better with an automated clustering method. This clustering is given below:

1. **Clustering C0 (p-value: 0.28):**
   STR03, STR04, STR16, STR18, STR20
   DEP09, DEP13, DEP19
   COP09, COP12, COP14, COP15

By comparing each result from an individual BPC run to the theoretical model and taking the proportion of indicators that were clustered differently from the theoretical model, we had an

[10]In order to save time, we first applied a constraint-based purification method described in (Spirtes et al., 2000) as a first step, using false discovery rates as a method for controlling to multiple hypothesis tests. Due to relatively large number of variables, this method is quite conservative and will tend to underprune the model, and therefore should not compromise the subsequent score-based purification that was applied. For instance, after the first step the model still had a p-value of zero according to a chi-square test.



(a) (b)

Figure 3.20: The output of BPC and GES-MIMBuild for the coping study.

average percentage of 0.05 (standard deviation of 0.05). The proportionally high standard deviation is a consequence of the small percentages: in 4 out of 10 cases there was no indicator mistakenly clustered with respect to the questionnaire, in 5 out of 10 we had only one mistake, and in only one case there were two mistakes.

The outputs with the highest number of indicators were also the ones with the highest number of latents: One full model automatically produced with GES-MIMBuild with the prior knowledge that STRESS is not an effect of other latent variables is given in Figure 3.20(b). This model passes a chi square test: p = 0.28.

Single-mothers' self-efficacy and children's development: Jackson and Scheines (2005) present a longitudinal study on single black mothers with one child in New York City from 1996 to 1999. The goal of the study was to detect the relationship among perceived self-efficacy, mothers' employement, maternal parenting and child outcomes. Overall, there were nine factors used in this study. Three of them, age, education and income, are represented directly by one indicator each (here represented as W2moage, W2moedu and W2faminc, respectively). The other six factors are latent variables measured by a varied number of indicators:

1. *financial strain* (3 indicators, represented by W2finan1, W2finan2, W2finan3)
2. *parenting stress* (26 indicators, represented by W2paroa - W2paroz)
3. *emotional support from family* (20 indicators, represented by W2suf01 - W2suf20)
4. *emotional support from friends* (20 indicators, W2sufr01 - S2sufr20)
5. *tangible support* (i.e., more material than psychological. 4 indicators, W2ssupta - W2ssuptd)
6. *problem behaviors of child* (30 indicators, W2mneg1 - W2mneg30)

We do not reproduce the original questionnaire here due to its size. The questionnaire is based on previous work on creating scales for such latents. As before, we evaluate how our algorithm output compares to the theoretical model. The extra difficulty here is that the distribution of the variables, which are ordinal categorical, are significantly skewed. Some of the categories are very rare, and we smoothed the original levels by collapsing values that were adjacent and represented

less than 5% of the total total number of cases. Several variables ended up binary by doing this transformation, which reduces the efficiency of models based on multivariate Gaussian distributions. 1 out of the 106 variables was also removed (W2sufr04) since 98% of the points fell into one of the two possible categories. The sample size is 178, relatively large for this kind of study, but it still considerably small for exploratory data analysis.

As before, the algorithm was applied 10 times with a different random choice of variable ordering each time. On average we got 21 indicators (standard deviation of 3.35) excluding clusters with only one variable. On average, 7.3 latents were discovered (standard deviation of 1.5). Counting only latents with at least three indicators, we had on average 4.3 latents (standard deviation of 0.86). Moreover, comparing each result to the theoretical model and taking the proportion of indicators that were wrongly clustered, we had an average percentage of 0.08, with standard deviation of 0.07.

It was noticeable that the small theoretical clusterings ("financial strain" and "tangible support") did not show up in the final models, but we claim that errors of omission are less harmful than those of commission, i.e., wrong clustering. However, it was relatively unexpected that the clusterings obtained in the first stage of our implementation (i.e., the output of FindInitialSelection, Appendix A.3) were larger in the number of indicators than the ones obtained at the end of the process. This can be explained by the fact that the initial step is a more constrained search, and therefore less prone to overfit. Since our data set is noisier than in the previous cases, we choose to evaluate only the three largest clusters obtained from FindInitialSelection. In this case, we had an average proportion of 0.037 wrongly clustered items (standard deviation: 0.025), 4.9 clusters (deviation: 0.33), 4.6 clusters of size at least three (deviation: 0.71) and 24.2 indicators (deviation: 2.8). Notice that the clusters were less fragmented than in the previous case, i.e., we had less clusters, more indicators per clustering, and a insignificant number of clusters with less than three indicators.

The largest clusters in this situations were the following:

1. **Cluster D1 (p-value: 0.46):**
   W2sufr02 W2sufr05 W2sufr08 W2sufr13 W2sufr14 W2sufr19 W2sufr20
   W2mneg14 W2mneg15 W2mneg2 W2mneg22 W2mneg26 W2mneg28 W2mneg29
   W2suf01 W2suf05 W2suf08
   W2paro2e W2paro2j W2paro2t W2paro2w
   W2suf07 W2suf12 W2suf17
2. **Cluster D2 (p-value: 0.22):**
   W2sufr01 W2sufr08 W2sufr10 W2sufr12 W2sufr13 W2sufr14 W2sufr19 W2sufr20
   W2suf04 W2suf05 W2suf10
   W2paro2e W2paro2j W2paro2t W2paro2w
   W2paro2k W2suf12 W2suf17
   W2mneg2 W2mneg5 W2mneg12 W2mneg14 W2mneg21 W2mneg22 W2mneg26
3. **Cluster D3 (p-value: 0.29):**
   W2mneg2 W2mneg10 W2mneg22 W2mneg26 W2mneg28 W2mneg29
   W2sufr01 W2sufr05 W2sufr08 W2sufr09 W2sufr12 W2sufr13 W2sufr14 W2sufr19
   W2suf02 W2suf04 W2suf05 W2suf11 W2suf13 W2suf20
   W2paro2e W2paro2j W2paro2t W2paro2w
   W2paro2k W2suf12 W2suf17

One can see that such models largely agree with those formed from prior knowledge. However, sucess in this domain is not as interesting as in the previous two cases: unlike in the test anxiety and spiritual coping models, the covariance matrix of the latent variables has a majority number of very small entries, resulting in a considerably easier clustering by just observing marginal independencies among items.

Still, the cases where theoretical clusters were split seem to be in accordance with the data: merging the W2suf indicators in a single pure cluster in D1 will result in a model with a p-value of 0.008. Merging the W2suf variables in D2 will also result in a low p-value (0.06) even when W2paro2k is removed. Unsurprinsingly, doing a similar merging in D3 gives a model with a p-value of 0.04. This is a strong indication that W2suf12 and W2suf17 should form a cluster on their own. In fact, these two items are formulated as two very similar indicators: "members of my family come to me for emotional support" and "members of my family seek me out for companionship". No other indicator for this latent seems to fall in the same category. Why this particular pair is singled out in comparison with other indicators for this latent is a question for future studies and a simple example of how our procedure can help in understanding the latent structure of the data.

3.6 Summary

We introduced a method to discover latents and identify their respective indicators. This generalizes the work on modifying measurement models described by Glymour et al. (1987) and complements the MIMBuild approach of Spirtes et al. (2000). What can be learned from our approach is that identifiability matters, that intuitively appealing heuristics (e.g., rotation methods) might fail when the goal is structure learning with causal interpretation, and that in many times is preferable to model the relationships of a subset of the given variables than trying to force a bad model over all of them (Kano and Harada, 2000).

Still, the full linearity assumption might be too strong. This will be relaxed in the next chapter.

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

Table 3.3: Results obtained with BuildPureClusters (BPC), factor analysis (FA) and purified factor analysis (P-FA) for the problem of learning measurement models. Each number is an average over 10 trials, with the standard deviation over these trials in parenthesis.

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

Table 3.4: Results obtained with BuildPureClusters (BPC) and purified factor analysis (P-FA) for the problem of learning measurement models. Each number is an average over 10 trials, with standard deviations in parenthesis.

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

Table 3.5: Results obtained with the application of GES-MIMBuild to the output of BPC, FA, and P-FA, with the number of perfect solutions over ten trials on the right of each average.

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

Table 3.6: Results obtained with the application of GES-MIMBuild to the output of BPC, FA, and P-FA, with the number of perfect solutions over ten trials on the right of each average.

Chapter 4

Learning measurement models of non-linear structural models

The assumption of full linearity, as discussed in Chapter 3, might be too strong. It turns out that much can still be discovered if one allows flexible functional relations among latent variables. Tetrad constraints in the observed covariance matrix continue to carry information concerning the measurement model. This chapter discusses how to identify some important features of the true model using tetrad constraints, even when latents are non-linearly related. Such identification rules justify applying an essentially unmodified BuildPureClusters algorithm to a much larger class of models. However, we do not have identification rules to detect d-separations among latents in this case. Moreover, the clusters we discover might not correspond to single latents in the true model. With the help of background knowledge, the modified BuildPureClusters is still a valuable tool in latent variable modeling.

4.1 Approach

We modify the assumptions introduced in Chapter 3. For now, we assume that the latent variable model to be discovered has a graphical structure and parameterization that obey the following conditions besides the causal Markov condition:

A1. no observed variable is a parent of a latent variable;

A2. any observed variable is a linear function of its parents with additive noise of finite positive variance;

A3. all latent variables have finite positive variance, and the correlation of any two latents lies strictly in the open interval (-1, 1);

A4. there are no cycles that include an observed variable;

This means that observed variables can have observed parents, that latents can be (noisy) non-linear functions of their parents, and that cycles are allowed among latents. No other structural assumptions are required (such as full acyclicity), and we do not require linearity among latents.

In the previous chapter, we made use of the faithfulness assumption, which states that a conditional independence holds in the joint distribution if and only if it is entailed in the respective

graphical model by the Markov condition. The movitation is that observed conditional independences should be the result of the graphical structure, not of an accidental choice of parameters defining the probability of a node given its parents.

The parametric assumptions in this chapter are not strong enough in order to test conditional independences between hidden variables, and therefore no graphical conditions for independence among latents will be assumed. In particular, faithfulness will not be assumed. Instead, since the structural model is treated as a black box, our results will have a measure-theoretic motivation. All results presented here have the following characteristics:

C1. they hold with probability 1 with respect to the Lebesgue measure over the set of linear coefficients and error variances that partially parameterize the density function of an observed variable given its parents;

C2. they hold for any distribution of the latent variables (that obeys the given assumptions);

One can show that the Lebesgue argument is no different from the faithfulness assumption for typical families of graphical models, such as multinomial and Gaussian (Spirtes et al., 2000).

Our goal is not to fully identify a graphical structure. The assumptions are too weak to reallistically accomplish this goal. Instead we will focus on a more restricted task:

- GOAL: *to identify d-separations between a pair of observed variables, or a pair of observed and latent variable, conditioned on sets of latents variables.*

Identifying d-separations between latents is a topic for future research, where specific assumptions concerning latent structure can be adopted according to the problem at hand. Instead, we focus in what can be achieved without any further assumptions.

As before, the strategy to accomplish our goal is to use constraints in the observed covariance matrix that will allow us to identify the following features of the unknown latent variable model:

F1. the existence of hidden variables;

F2. that observed variable X cannot be an ancestor of observed variable Y;

F3. that observed variable X cannot have a common parent with observed variable Y;

Section 4.2 will describe empirical methods that can in many cases identify the above features. In Section 4.3, we describe a variation of BuildPureClusters as a way of putting together these pieces of information to learn a measurement model for non-linear structural models.

4.2 Main results

Assume for now we know the true population covariance matrix. Without loss of generality, assume also that all variables have zero mean. Let $G(\mathbf{O})$ be the graph of a latent variable model with observed variables $\mathbf{O}$. The following lemma illustrates a simple result that is intuitive but does not follow immediately from correlation analysis, since observed nodes may have non-linear dependencies:

Lemma 4.1 *If for* $\{A, B, C\} \subseteq \mathbf{O}$ *we have* $\rho_{AB} = 0$ *or* $\rho_{AB.C} = 0$*, then* A *and* B *cannot share a common latent parent in* G*.*

Althought vanishing partial correlations can sometimes be useful, we are mostly motivated by problems where *all* observed variables have hidden common ancestors. In this case, vanishing partial correlations are useless. We will still use tetrad constraints detectable from the covariance matrix of the observed variables.

The following result allows us to learn that observed variable X cannot be an ancestor of observed variable Y in several situations:

Lemma 4.2 *For any set* $\mathbf{O}' = \{A, B, C, D\} \subseteq \mathbf{O}$*, if* $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BC}$ *such that for all triplets* $\{X, Y, Z\}$*,* $\{X, Y\} \subset \mathbf{O}', Z \in \mathbf{O}$*, we have* $\rho_{XY.Z} \neq 0$ *and* $\rho_{XY} \neq 0$*, then no element in* $X \in \mathbf{O}'$ *is an ancestor of any element in* $\mathbf{O}' \backslash X$ *in* G*.*

Notice that this result allows us to identify the non-existence of several ancestral relations even when no conditional independences are observed and latents are non-linearly related. A second way of learning such a relation is as follows: let $G(\mathbf{O})$ be a latent variable graph and $\{A, B\}$ be two elements of $\mathbf{O}$. Let the predicate $Factor_1(A, B, G)$ be true if and only there exists a set $\{C, D\} \subseteq \mathbf{O}$ such that the conditions of Lemma 4.2 are satisfied for $\mathbf{O}' = \{A, B, C, D\}$, i.e., $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BC}$ with the corresponding partial correlation constraints. The second approach for detecting lack of ancestral relations between two observed variables is given by the following lemma:

Lemma 4.3 *For any set* $\mathbf{O}' = \{X_1, X_2, Y_1, Y_2\} \subseteq \mathbf{O}$*, if* $Factor_1(X_1, X_2, G) = true$*,* $Factor_1(Y_1, Y_2, G) = true$*,* $\sigma_{X_1Y_1}\sigma_{X_2Y_2} = \sigma_{X_1Y_2}\sigma_{X_2Y_1}$*, and all elements of* $\{X_1, X_2, Y_1, Y_2\}$ *are correlated, then no element in* $\{X_1, X_2\}$ *is an ancestor of any element in* $\{Y_1, Y_2\}$ *in* G *and vice-versa.*

We define the predicate $Factor_2(A, B, G)$ to be true if and only it is possible to learn that A is not an ancestor of B in the unknown graph G that contains these nodes by using Lemma 4.3.

We now describe two ways of detecting if two observed variables have no (hidden) common parent in $G(\mathbf{O})$. Let first $\{X_1, X_2, X_3, Y_1, Y_2, Y_3\} \subseteq \mathbf{O}$. The following identification conditions are sound:

CS1. If $\sigma_{X_1Y_1}\sigma_{X_2X_3} = \sigma_{X_1X_2}\sigma_{X_3Y_1} = \sigma_{X_1X_3}\sigma_{X_2Y_1}, \sigma_{X_1Y_1}\sigma_{Y_2Y_3} = \sigma_{X_1Y_2}\sigma_{Y_1Y_3} = \sigma_{X_1Y_3}\sigma_{Y_1Y_2}, \sigma_{X_1X_2}\sigma_{Y_1Y_2} \neq \sigma_{X_1Y_2}\sigma_{X_2Y_1}$ and for all triplets $\{X, Y, Z\}, \{X, Y\} \subset \{X_1, X_2, X_3, Y_1, Y_2, Y_3\}$, $Z \in \mathbf{O}$, we have $\rho_{XY} \neq 0, \rho_{XY.Z} \neq 0$, then X_1 and Y_1 do not have a common parent in G.

CS2. If $Factor_1(X_1, X_2, G)$, $Factor_1(Y_1, Y_2, G)$, X_1 is not an ancestor of X_3, Y_1 is not an ancestor of Y_3, $\sigma_{X_1Y_1}\sigma_{X_2Y_2} = \sigma_{X_1Y_2}\sigma_{X_2Y_1}$, $\sigma_{X_2Y_1}\sigma_{Y_2Y_3} = \sigma_{X_2Y_3}\sigma_{Y_2Y_1}$, $\sigma_{X_1X_2}\sigma_{X_3Y_2} = \sigma_{X_1Y_2}\sigma_{X_3X_2}$, $\sigma_{X_1X_2}\sigma_{Y_1Y_2} \neq \sigma_{X_1Y_2}\sigma_{X_2Y_1}$ and for all triplets $\{X, Y, Z\}, \{X, Y\} \subset \{X_1, X_2, X_3, Y_1, Y_2, Y_3\}, Z \in \mathbf{O}$, we have $\rho_{XY} \neq 0, \rho_{XY.Z} \neq 0$, then X_1 and Y_1 do not have a common parent in G.

As in the previous chapter, "CS" here stands for "constraint set," a set of constraints in the observable joint that are empirically verifiable. In the same way, call CS0 the separation rule of Lemma 4.1. The following lemmas state the correctness of CS1 and CS2:

Lemma 4.4 *CS1 is sound.*

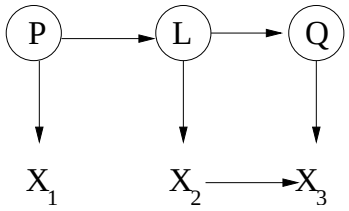


Figure 4.1: In this figure, L and Q are immediate latent ancestors of X_3, since there are directed paths from L and Q into X_3 that do not contain any latent node. Latent P, however, is not an immediate latent ancestor of X_3, since every path from P to X_3 contains at least one other latent.

Lemma 4.5 *CS2 is sound.*

We have shown before that such identification results also hold in fully linear latent variable models. One might conjecture that, as far as identifying ancestral relations among observed variables and hidden common parents goes, linear and non-linear latent variable models are identical. However, this is not true.

Theorem 4.6 *There are sound identification rules that allow one to learn if two observed variables share a common parent in a linear latent variable model that are not sound for non-linear latent variable models.*

In other words, one gains more identification power if one is willing to assume full linearity of the latent variable model. We will see more of the implications of assuming linearity.

Another important building block in our approach is the identification of which latents exist. Define an *immediate latent ancestor* of an observed node O in a latent variable graph G as a latent node L that is a parent of O or the source of a directed path $L \rightarrow V \rightarrow \cdots \rightarrow O$ where V is an observed variable. Notice that this implies that every element in this path, with the exception of L, is an observed node, since we are assuming that observed nodes cannot have latent descendants. Figure 4.1 illustrates the concept.

Lemma 4.7 *Let* $\mathbf{S} \subseteq \mathbf{O}$ *be any set such that, for all* $\{A, B, C\} \subseteq \mathbf{S}$*, there is a fourth variable* $D \in \mathbf{O}$ *where i.* $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BD}$ *and ii. for every set* $\{X, Y\} \subset \{A, B, C, D\}, Z \in \mathbf{O}$ *we have* $\rho_{XY.Z} \neq 0$ *and* $\rho_{XY} \neq 0$*. Then* $\mathbf{S}$ *can be partioned into two sets* $\mathbf{S_1}, \mathbf{S_2}$ *where*

1. *all elements in* $\mathbf{S_1}$ *share a common immediate latent ancestor, and no two elements in* $\mathbf{S_1}$ *have any other common immediate latent ancestor;*

2. *no element* $S \in \mathbf{S_2}$ *has any common immediate latent ancestor with any other element in* $\mathbf{S} \backslash S$*;*

3. *all elements in* $\mathbf{S}$ *are d-separated given the latents in* G*;*

Unlike the linear model case, a set of tetrad constraints $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BD}$ is not a sufficient condition (along with non-vanishing correlations) for the existence of a node d-separating nodes $\{A, B, C, D\}$. For instance, consider the graph in Figure 4.2(a), which depicts a latent variable graph with three latents L_1, L_2 and L_3, and four measured variables, W, X, Y, Z.

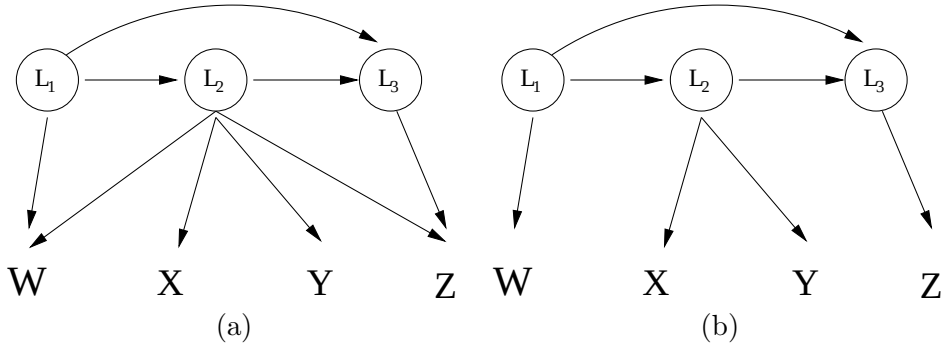


Figure 4.2: It is possible that $\rho_{L_1 L_3 . L_2} \neq 0$ even though L_2 does not d-separate L_1 and L_3. That happens, for instance, if $L_2 = \lambda_1 L_1 + \epsilon_2$, $L_3 = \lambda_2 L_1^2 + \lambda_3 L_2 + \epsilon_3$, where L_1, ϵ_2 and ϵ_3 are normally distributed with zero mean.

L_2 does not d-separate L_1 and L_3, but there is no constraint in the assumptions that precludes the partial correlation of L_1 and L_3 given L_2 of being zero. For example, in the additive model $L_2 = \lambda_1 L_1 + \epsilon_2$, $L_3 = \lambda_2 L_1^2 + \lambda_3 L_2 + \epsilon_3$, where $L_1, \epsilon_1, \epsilon_2$ are standard normals, we have that $\rho_{13.2} = 0$, which will imply all three tetrad constraints among $\{W, X, Y, Z\}$.

In this case Lemma, 4.7 says that, for $\mathbf{S} = \{W, X, Y, Z\}$, we have some special partition of $\mathbf{S}$. In Figure 4.2(a) given by $\mathbf{S_1} = \{W, X, Y, Z\}$, $\mathbf{S_2} = \emptyset$. In Figure 4.2(b), $\mathbf{S_1} = \{X, Y\}$, $\mathbf{S_2} = \{W, Z\}$. However, tetrad constraints, and actually no covariance constraint, can distinguish both graphs from a model where a single latent d-separates all four indicators.

We will see an application of our results in the next section, where they are used to identify interesting clusters of indicators, i.e., disjoint sets of observed variables that measure disjoint sets of latents.

4.3 Learning a semiparametric model

The assumptions and identification rules provided in the previous section can be used to learn a partial representation of the unknown graphical structure that generated the data, as suggested in Chapter 3. Given a set of observed variables $\mathbf{O}$, let $\mathbf{O'} \subseteq \mathbf{O}$, and let $\mathbf{C}$ be a partition of $\mathbf{O'}$ into sets $\{\mathbf{C_1}, \ldots, \mathbf{C_k}\}$ such that

SC1. for any $\{X_1, X_2, X_3\} \subset \mathbf{C_i}$, there is some $X_4 \in \mathbf{O'}$ such that $\sigma_{X_1 X_2}\sigma_{X_3 X_4} = \sigma_{X_1 X_3}\sigma_{X_2 X_4} = \sigma_{X_1 X_4}\sigma_{X_2 X_3}$, $1 \leq i \leq k$ and X_4 is correlated with all elements in $\{X_1, X_2, X_3\}$;

SC2. for any $X_1 \in \mathbf{C_i}, X_2 \in \mathbf{C_j}$, $i \neq j$, we have that X_1 and X_2 are separated by CS0, CS1 or CS2;

SC3. for any $X_1, X_2 \in \mathbf{C_i}$, $Factor_1(X_1, X_2, G) = true$ or $Factor_2(X_1, X_2, G) = true$;

SC4. for any $\{X_1, X_2\} \subset \mathbf{C_i}$, $X_3 \in \mathbf{C_j}$, $\rho_{X_1 X_3} \neq 0$ if and only if $\rho_{X_2 X_3} \neq 0$;

Any partition with structural conditions SC1-SC4 has the following properties:

Theorem 4.8 *If a partition* $\mathbf{C} = \{\mathbf{C_1}, \ldots, \mathbf{C_k}\}$ *of* $\mathbf{O'}$ *respects structural conditions SC1-SC4, then the following holds in the true latent variable graph* G *that generated the data:*

1. *for all* $X \in \mathbf{C_i}, Y \in \mathbf{C_j}, i \neq j$, X *and* Y *have no common parents, and* X *is d-separated from the latent parents of* Y *given the latent parents of* X*;*

2. *for all* $X, Y \in \mathbf{O'}$, X *is d-separated from* Y *given the latent parents of* X*;*

3. *every set* $\mathbf{C_i}$ *can be partitioned into two groups according to Lemma 4.7;*

The BuildPureClusters algorithm of Chapter 3 can be adapted to generate a partition with these properties. In this case, condition CS3 is not used. We describe a possible implementation of BuildPureClusters in Section 4.4. This variation can in principle allow some sets $\mathbf{C_i}$ of size 1 and 2. In those cases, the properties established by Lemma 4.7 hold vacuously. This algorithm cannot identify how each set $\mathbf{C_i}$ can be further partitioned into such two subsets, one where every node has an unique common immediate latent ancestor, and one where each node has no common immediate latent ancestor with any other node. It might be the case that no two nodes in $\mathbf{C_i}$ have a common immediate latent ancestor. It might be the case that all nodes in $\mathbf{C_i}$ have an unique common immediate latent ancestor. The combination of Lemma 4.7 and domain knowledge can be useful to find the proper sub-partition.

These are weaker results than the ones obtained for linear models. There, each set $\mathbf{C_i}$ is associated with an unique latent variable L_i from G (as long as $|\mathbf{C_i}| > 2$). Furthermore, conditioned on L_i each node in $\mathbf{C_i}$ is d-separated from all other nodes in $\mathbf{O'}$, as well as from their respective latent parents. There might be no latent node in the non-linear case with these properties, as previously illustrated in Figure 4.2. In this case, the trivial partition $\mathbf{C} = \{\{W, X, Y, Z\}\}$, with a single element, will satisfy the structural conditions SC1-SC4, and therefore the properties of Theorem 4.8. However, there is no unique latent variable in this system that d-separates all elements of $\{W, X, Y, Z\}$. This would not be the case in a linear system.

This the fundamental difference between the work presented here and the one developed by Silva et al. (2003). There, it was assumed that each latent d-separated at least three unique measures, i.e., each latent was assumed to have three observed children that were d-separated by it. In this way, it was possible to use CS1 and Lemma 4.2 to identify all latents and an unique map between a set $\mathbf{C_i}$ and a latent. Although one might adopt this assumption in studies where one already has a strong idea of which latents exist, this is in general an untestable assumption. This chapter explores what is possible to achieve when minimal assumptions about the graphical structure are adopted, and expands it with extra identification rules. Lemmas 4.3 and 4.5 are new identification rules. Lemma 4.7, Theorems 4.6, 4.8 and Theorem 4.9 below are all new results that are necessary: with the stronger assumptions of Silva et al. (2003), all latents could be identified, which highly simplified the problem. This is not the case here.

As in the linear case, it is still possible to parameterize a latent variable model using the partition $\mathbf{C} = \{\mathbf{C_1}, \ldots, \mathbf{C_k}\}$ of a subset $\mathbf{O'}$ of the given observed variables such that the first two moments of the distribution of $\mathbf{O'}$ can still be represented. Given a graph G, a *linear parameterization* of G associates a parameter with each edge and two parameters with each node, such that each node V is functionally represented as a linear combination of its parents plus an additive error: $V = \mu_V + \Sigma_i \lambda_i Pa_{V_i} + \epsilon_V$, where $\{Pa_{V_i}\}$ is the set of parents of V in G, and ϵ_V is a random variable with zero mean and variance ζ_V (μ_V and ζ_V are the two extra parameters by node). Notice that this parameterization might not be enough to represent all moments of a given family of probability distributions.

A linear latent variable model is a latent variable graph with a particular instance of a linear parameterization. The following result mirrors the one obtained for linear models:

Theorem 4.9 *Given a partition* $\mathbf{C}$ *of a subset* $\mathbf{O}'$ *of the observed variables of a latent variable graph* G *such that* $\mathbf{C}$ *satisfies structural constraints SC1-SC4, there is a linear latent variable model for the first two moments of the density of* $\mathbf{O}'$.

Consider the graph G_{linear} constructed by the following algorithm:

1. initialize G_{linear} with a node for each element in $\mathbf{O}'$;
2. for each $\mathbf{C_i} \in \mathbf{C}$, add a latent L_i to G, and for each $V \in \mathbf{C_i}$, add an edge $L_i \rightarrow V$
3. fully connect the latents in G_{linear} to form an arbitrary directed acyclic graph;

For instance, the G_{linear} graph associated with Figures 4.2(a) and 4.2(b) would be a one-factor model where a single latent L is the common parent of $\{W, X, Y, Z\}$, and L d-separates its children.

The constructive proof of Theorem 4.9 (see Appendix B) shows that G_{linear} can be used to parameterize a model of the first two moments of $\mathbf{O}'$. This has an important heuristic implication: if the joint distribution of the latents and observed variables can be reasonably approximated by a mixture of Gaussians, where each component has the same graphical structure, one can fit a mixture of G_{linear} graphical models. This can be motivated by assuming each mixture component represents a different subpopulation with its own probabilitistic model, where the same causal structures hold, and the distributions are close to normal (e.g., a drug might have different quantitative effects on different genders but with the same qualitative causal structure). Each model will approximate the mean and covariance of the observed variables for a particular component of the mixture: since each component has the same graphical structure, the same required constraints in the component covariance matrix hold, and therefore the same parametric formulation can be used.

Notice this is less stringent than assuming that the causal model is fully linear. Assuming the distribution is fully linear can result in a wrong structure that might not be approximated well (e.g., if one applies unsound identification rules, as suggested by Theorem 4.6). Here, at least in principle the structure can be correctly induced. The joint distribution is approximated, and the quality of approximation will be dependendent on the domain.

As a future work, it would be interesting to empirically assess the robustness of MIMBuild with respect to small deviations from linearity, or with respect to monotone non-linear functions.

4.4 Experiments

In this section we evaluate an alternative implementation of BuildPureClusters on the tasks of causality discovery and density estimation.

This alternate implementation is a simple variation of algorithm RobustBuildPureClusters of Table A.2 (Appendix A). One difference is that we do not use identification rule CS3. The main difference is that we do not use RobustPurify, which requires a test of fitness (in Chapter 3 we adopted the multivariate Gaussian family). Instead, we adapt the original formulation of BuildPureClusters (Table 3.2), which makes use of tetrad constraints, as follows: first, remove all variables that appear in more than one cluster. For any pair of variables $\{X, Y\}$ in the pre-purified graph, try to find another pair $\{W, Z\}$, in the same graph (i.e., do not use removed indicators) such that all three tetrad hold among $\{W, X, Y, Z\}$. If X and Y are in the same cluster, then W has to be in the same cluster, but not necessarily Z. If X and Y are in different clusters, then W and Z are either in the same cluster as X, or in the same cluster ar Y. Unless otherwise

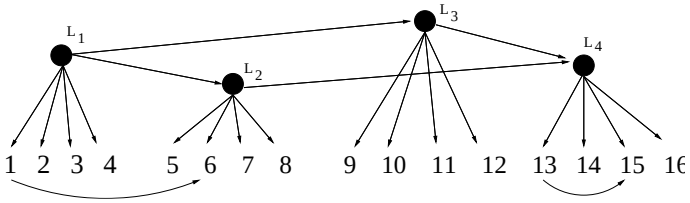


Figure 4.3: An impure model with a diamond-like latent structure. Notice there are two ways to purify this graph: by removing 6 and 13 or removing 6 and 15.

specified, we used the test of tetrad constraints described by Bollen (1990), which is an asymptotic distribution-free test of such constraints.

4.4.1 Evaluating nonlinear latent structure

In this section we perform an experiment with a non-linear latent structure and non-normally distributed data. The graph in Figure 4.3 is parameterized by the following nonlinear structural 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 4.4 depicts histograms for each variable in a specific sample. We show a randomly selected correlation matrix from a sample of size 5000 in Table 4.1.

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 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 (Spirtes et al., 2000; Wishart, 1928), which assumes multivariate normality[1]. Samples of size 1000, 5000 and 50000 were used. Results are given in Table 4.2. Such

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

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 4.1: An example of a sample correlation matrix of a sample of size 5000.

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 4.2: 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.

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 10 of FindInitialSelection in Table A.3). 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 4.2 with samples smaller than 1000. That happens because Step 10 of FindInitialSelection 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.

4.4.2 Experiments in density estimation

In this section, we will concentrate on evaluating our procedure as a way of finding submodels with a good fit. The goal is to show that causally motivated algorithms can be also suitable for density estimation. We run our algorithm over some datasets from the UCI Machine Learning Repository to obtain a graphical structure analogous to G_{linear} described in the previous section. We then fit the data to such a structure by using a mixture of Gaussian latent DAGs with a standard EM algorithm. Each component has a full parameterization: different linear coefficients and error variances for each variable on each mixture component. The number of mixture components is chosen by fitting the model with 1 to up to 7 components and choosing the one that maximizes the BIC score.

We compare this model against the mixture of factor analyzers (MofFA) (Ghahramani and Hinton, 1996). In this case, we want to compare what can be gained by fitting a model where latents are allowed to be dependent, even when we restrict the observed variables to be children of a single latent. Therefore, we fit mixtures of factor analyzers using the same number of latents we find with our algorithm. The number of mixture components is chosen independently, using the same BIC-based procedure. Since BPC can return only a model for a subset of the given observed variables, we run MofFA for the same subsets output by our algorithm.

In practice, our approach can be used in two ways. First, as a way of decomposing the full joint of a set $\mathbf{O}$ of observed variables by splitting it into two sets: one set where variables $\mathbf{X}$ can be modeled as a mixture of G_{linear} models, and another set of variables $\mathbf{Y} = \mathbf{O} \backslash \mathbf{X}$ whose conditional probability $f(\mathbf{Y}|\mathbf{X})$ can be modeled by some other representation of choice. Alternatively, if the observed variables are redundant (i.e., many variables are intended to measure the same latent concept), this procedure can be seen as a way of choosing a subset whose marginal is relatively easy to model with simple causal graphical structures.

As a baseline, we use a standard mixture of Gaussians (MofG), where an unconstrainted multivariate Gaussian is used on each mixture component. Again, the number of mixture components is chosen independently by maximizing BIC. Since the number of variables used in our experiments are relatively small, we do not expect to perform significantly better than MofG in the task of density estimation, but a similar performance is an indication that our highly constrained models provide a good fit, and therefore our observed rank constraints can be reasonably expected to hold in the population.

We ran a 10-fold cross-validation experiment for each one the following four UCI datasets: iono, specft, water and wdbc, all of which are measured over continuous or ordinal variables. We tried also the small dataset wine (13 variables), but we could not find any structure using our method. The other datasets varied from 30 to 40 variables. The results given in Table 6.9 show the average log-likelihood per data point on the respective test sets, also averaged over the 10 splits. These results are subtracted from the baseline established by MofG. We also show the average percentage of variables that were selected by our algorithm. The outcome is that we can represent the joint of a significant portion of the observed variables as a simple latent variable model where observed variables have a single parent. Such models do not lose information comparing to the full mixture of Gaussians. In one case (iono) we were able to significantly improve over the mixture of factor analyzers when using the same number of latent variables.

In the next chapter we show how these results can be improved by using Bayesian search algorithms which also allow the insertion of more observed variables, and not only those that have a single parent in a linearized graph.

Dataset	BPC	MofFA	% variables
iono	1.56 ± 1.10	-3.03 ± 2.55	0.37 ± 0.06
spectf	-0.33 ± 0.73	-0.75 ±0.88	0.34 ± 0.07
water	-0.01 ± 0.74	-0.90 ± 0.79	0.36 ± 0.04
wdbc	-0.88 ± 1.40	-1.96 ± 2.11	0.24 ± 0.13

Table 4.3: The difference in average test log-likelihood of BPC and MofFA with respect to a multivariate mixture of Gaussians. Positive values indicate that a method gives a better fit that the mixture of Gaussians. The statistics are the average of the results over a 10-fold cross-validation. A standard deviation is provided. The average number of variables used by our algorithm is also reported.

4.5 Completeness considerations

So far, we have emphasized the soundness BuildPureClusters in both its linear and non-linear versions. However, an algorithm that always returns an empty graph is vacuously sound. BuildPureClusters is of interest only if it can return useful information about the true graph. In Chapter 3, we only briefly described issues concerning *completeness* of this algorith, i.e., how many of the common features of all tetrad-equivalent models can be discovered.

It has to be stressed that there is no guarantee of how large the set of indicators in the output of BuildPureClusters will be for any problem. It can be an empty set, for instance, if all observed variables are children of several latents. Variations of BuildPureClusters are still able to asymptotically find the submodel with the largest number of latents that can be identified with CS rules. To accomplish that, one has to apply the following algorithm in place of Step 2 of Table 3.2:

Algorithm MaximumLatentSelection

1. Create an empty graph G_L, where each node correspond to a latent
2. Add an undirected edge $L_i - L_j$ if and only if L_i has three pure indicators that L_j does not have, and vice-versa
3. Return a maximum clique of G_L

An interesting implication is: if there is a pure submodel of the true measurement model where each latent has at least three indicators, then this algorithm will identify all latents (Silva et al., 2003). This assumption is not testable, however. Moreover, because of the maximum clique step, this algorithm is exponential in the number of latents, in the worst case.

In principle, much of the identifiability limitations here described can be solved if one explores constraints that uses information besides the second moments of the observed variables. Still, it is of considerable interest to know what can be done with covariance information only, since using higher order moments highly increases the chance of commiting statistical mistakes. This is especially difficult concerning learning the structure of latent variable models.

Although we do not provide a complete characterization of the tetrad equivalence class, we can provide a necessary condition in order to identify if two nodes have no common latent parent when no marginal vanishing correlations are observed:

Lemma 4.10 *Let $G(\mathbf{O})$ be a latent variable graph where no pair in $\mathbf{O}$ is marginally uncorrelated, and let $\{X, Y\} \subset \mathbf{O}$. If there is no pair $\{P, Q\} \subset \mathbf{O}$ such that $\sigma_{XY}\sigma_{PQ} = \sigma_{XP}\sigma_{YQ}$ holds, then there is at least one graph in the tetrad equivalence class of G where X and Y have a common latent parent.*

Notice this does not mean one cannot distinguish between models where X and Y have and do not have a common hidden parent. We are claiming that for tetrad equivalence classes only. For instance, in some situations this can be done by using only conditional independencies, which is the base of the FAST CAUSAL INFERENCE algorithm of Spirtes et al. (2000). Figure 4.5 illustrates a case.

In practice, it is not of great interest having identification rules that require the use of many variables. The more variables are necessary, the more computationally expensive any search algorithm gets, as well as less statistically reliable. Our CS rules, for instance, require 6 variables, which is already a considerably high number. However, as far as using tetrad constraints goes, one cannot expect to extend BUILDPURECLUSTERS with identification rules that are computationally simpler than CS1, CS2 or CS3. The following result shows that in the general case (i.e., where marginal independencies are not observed), one does not have a criterion for clustering indicators that uses less than six variables using tetrad constraints:

Theorem 4.11 *Let $\mathbf{X} \subseteq \mathbf{O}$ be a set of observed variables, $|\mathbf{X}| < 6$. Assume $\rho_{X_1X_2} \neq 0$ for all $\{X_1, X_2\} \subseteq \mathbf{X}$. There is no possible set of tetrad constraints within $\mathbf{X}$ for deciding if two nodes $\{A, B\} \subset \mathbf{X}$ do not have a common parent in a latent variable graph $G(\mathbf{O})$.*

Notice again that it might be the case a combination of tetrad and conditional independence constraints might provide an identification rule that uses less than 6 variables (in a case where conditional independencies alone are not enough). This result is for tetrad constraints only.

4.6 Summary

We presented empirically testable conditions that allows one to learn structural features of latent variable models where latents are non-linearly related. These results can be used in an algorithm for learning a measurement model for some latents without making any assumptions about the true graphical structure, besides the fairly general assumption by which observed variables cannot be parents of latent variables.

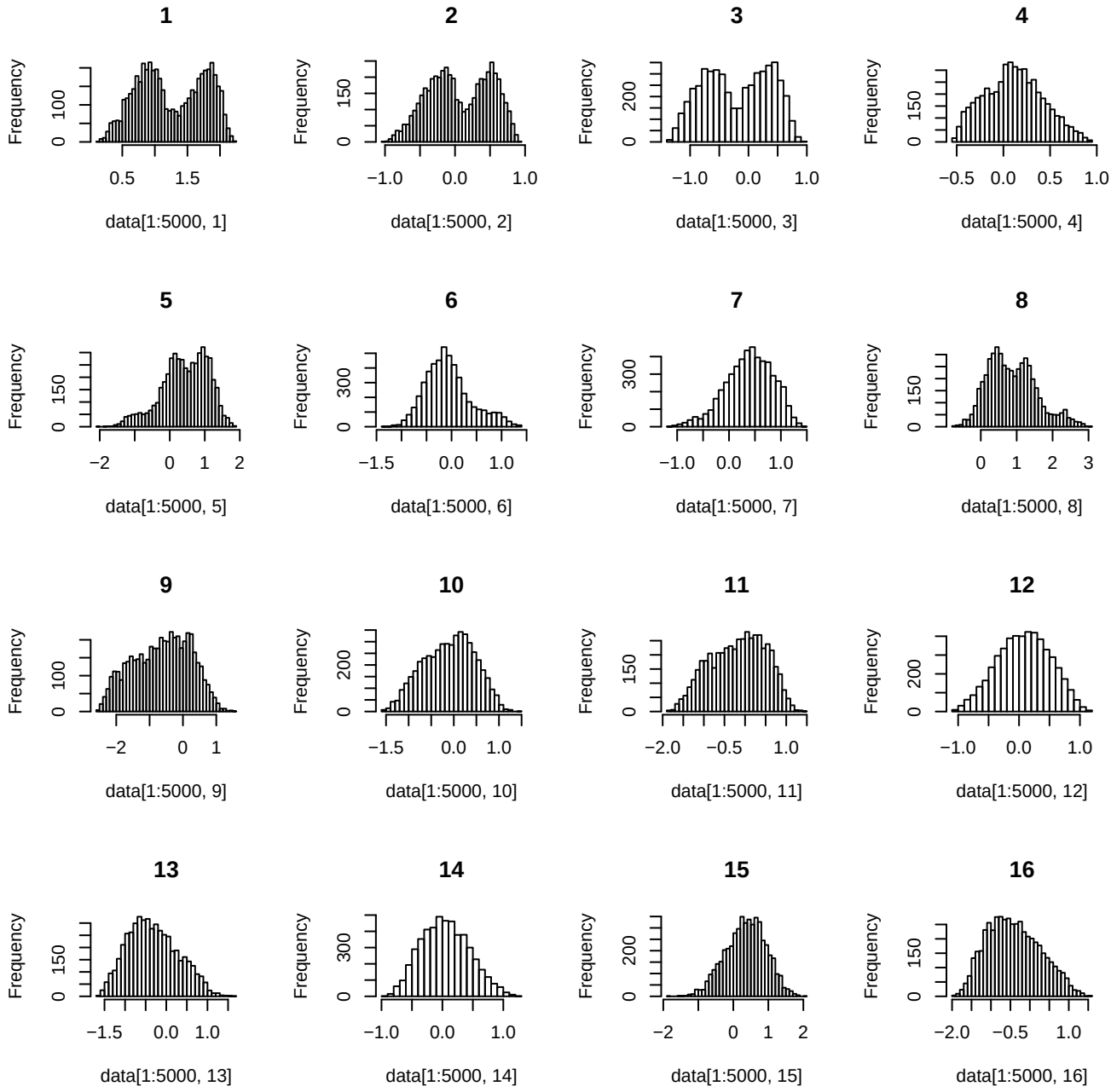


Figure 4.4: Univariate histograms for each of the 16 variables (organized by row) from a data set of 5000 observations sampled from the graph in Figure 4.3. 30 bins were used.

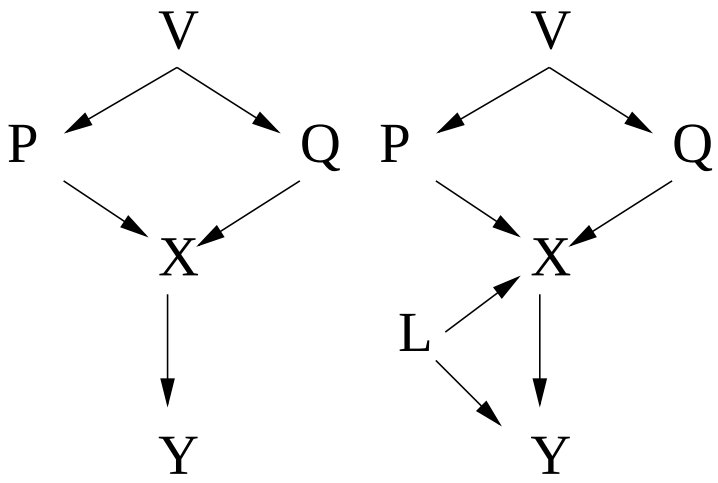


Figure 4.5: These two models (L is the only hidden variable) can be distinguished by using conditional independence constraints, but not through tetrad constraints only.

Chapter 5

Learning local discrete measurement models

The BuildPureClusters algorithm (BPC) constructs a global measurement model: a single model composed of several latents. In linear models, it provides sufficient information for the application of sound algorithms for learning latent structure. As defined, BPC can be applied only to continuous (or approximately continuous) data.

However, one might be interested in a *local model*, which we define as a *set* of several *small models* covering a few variables each, where their respective set of variables might overlap. A local model is usually not *globally consistent*: in probabilistic terms, this means the marginal distribution for a given set of variables differs according to different elements of the local model. In causal terms, this means conflicting causal directions. The two main reasons why one would use a local model instead of a global one are: 1. ease of computation, especially for high dimensional problems; 2. there might be no good global model, but several components of a local model might be of interest.

In this chapter, we develop a framework for learning local measurement models of *discrete* data using BuildPureClusters and compare it to one of the most widely used local model formulations: association rules.

5.1 Discrete associations and causality

Discovering interesting associations in discrete databases is a key task in data mining. Defining interestingness is, however, an elusive task. One can informally describe interesting (conditional) associations as those that allow one to create *policies* that maximize a measure of success, such as profit in private companies or increase of life expectancy in public health. Ultimately, many questions phrased as "find interesting associations" in data mining literature are nothing but causal questions with observational data (Silverstein et al., 2000).

A canonical example is the following hypothetical scenario: one where baby diapers and beer are products with a consistent association across several market basket databases. From this previously unknown association and extra prior knowledge, an analyst was able to infer that this association is due to the causal process where fathers, when assigned to the duty of buying diapers, indulge on buying some beer. One possible policy that makes use of this information is displaying beer and diapers in the same aisle to convince parents to buy beer more frequently when buying diapers.

In this case, the link from association to causality came from prior knowledge about a hidden

variable. The interpretation of the hidden variable, however, came from the nature of the two items measuring it, and without the knowledge of the statistical support for this association, it would be unlikely that the analyst would conjecture the existence of such a latent.

Association rules (Agrawal and Srikant, 1994) are a very common tool for discovering interesting associations. A standard association rule is simply a propositional rule of the type "If A, then B", or simply $A \Rightarrow B$, with two particular features:

- the support of the rule: the number of cases in the database where events A and B jointly
- the confidence of the rule: the proportion of cases that have B, counting only among those that have A

Searching for association rules requires finding a good trade-off between these two features. With extra assumptions, association rule mining inspired by algorithms such as the PC algorithm can be used to reveal causal rules (Silverstein et al., 2000; Cooper, 1997).

However, in many situations the causal explanation for the observed associations is due to latent variables, such as in our example above. The number of rules can be extremely large even in relatively small data sets. More recent algorithms may dramatically reduce the number of rules when compared to classical alternatives (Zaki, 2004), but even there the set of rules can be unmanageable. Although rules can describe specific useful knowledge, they do not take in account that hidden common causes might explain several patterns not only in a much more succint way, but in a way on which leaping from association to causation would require less background knowledge. How to introduce hidden variables in causal association rules is the goal of the algorithm described in this chapter.

5.2 Local measurement models as association rules

Association rules are local models by nature. That is, the output of an association rule analysis consists on a set of rules covering only some variables in the original domain. Such rules might be contradictory: the probability $P(B|A)$ might be different according to rules $A \Rightarrow \{B, C\}$ and $A \Rightarrow \{B, D\}$, for instance, depending on which model is used to represent these conditional distributions[1].

One certainly loses statistical power by using local models instead of global ones. This is one of the main reasons why an algorithm such as GES usually performs better than the PC search (Spirtes et al., 2000), for instance (although PC outputs a global model, it does so by merging several local pieces of information derived nearly independently). However, searching for local models can be orders of magnitude faster than searching for global models. For large problems, it might be simply impossible to find a global model. Pieces of a local model can still be useful, as in causal association rules compared to full graphical models (Silverstein et al., 2000; Cooper, 1997).

Moreover, not requiring global consistency can in some sense be advantageous: for instance, it is well known that the PC algorithm might return cyclic graphs even though it is not supposed to. This happens because the PC algorithm builds a model out of local pieces of information, and such pieces might not fit globally as a DAG. Although such an output might be the result of statistical mistakes, they can also indicate failure of assumptions, such as the non-existence of hidden variables. An algorithm such as GES will always return a DAG, and therefore it is less

[1]This will not be the case if this probability is the standard maximum likelihood estimator of an unconstrained multinomial distribution.

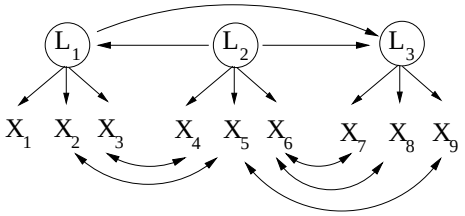


Figure 5.1: Latent L_2 will never be identified by any implementation of BPC that attempts to include L_1 and L_3, althought individually it has a pure measurement model.

robust to failures of the assumptions. When datasets are very large, running PC might be a better alternative than, or at least a complement to, GES.

This is especially interesting for the problem of finding pure measurement models. BuildPureClusters will return a pure model if one exists. However, one might lose information that could be easily derived using the same identification conditions of Chapter 3. Consider the model in Figure 5.1. Latent L_2 cannot exist in the same pure model as the other two latents, since it requires deleting too many indicators of L_1 and L_3. However, one can verify there is a pure measurement model with at least four (direct and indirect) indicators for L_2 (X_1, X_2, X_3, X_4), which could be derived independently.

Learning a full model with impurities might be statistically difficult, as discussed in previous chapters: in simulations, estimated measurement patterns are considerably far off from the real ones. Listing all possible combinations of pure models might be intractable. Instead, an interesting compromise for finding measurement models can be described in three steps:

1. find one-factor models only;
2. filter such models;
3. use the selected one-factor models according to the problem at hand.

The first step can be used to generate *local* models, i.e., sets of one-factor models generated independently, without the necessity of being globally coherent. This means that in principle one might generate a one-factor model for $\{X_1, X_2, X_3, X_4\}$, $\{X_1, X_2, X_3, X_5\}$, $\{X_2, X_3, X_4, X_5\}$, but fail to generate a one-factor model using $\{X_1, X_2, X_3, X_4, X_5\}$, althought the first three logically imply the latter. This could not happen if assumptions hold and data is infinite, but it is possible for finite samples and real-world data.

Since the local model might have many one-factor elements, one might need to filter elements considered irrelevant by some criteria. By following this framework, we will introduce a variation of BuildPureClusters for discrete data that performs Steps 1 and 2 above. We leave Step 3 to be decided according to the application. For instance, one might select one-factor models, learn which impurities might hold among them, and then use the final result to learn the structure among latents. This, however, can be very computationally expensive, orders of magnitude more costly than the case for continuous variables (Bartholomew and Knott, 1999; Buntine and Jakulin, 2004).

Another alternative is a more theory-driven approach, where latents are just labeled by an expert in the field but no causal model for the latents is automatically derived from data. They can be derived by theory or ignored: in this case, each one-factor model itself is the focus of the

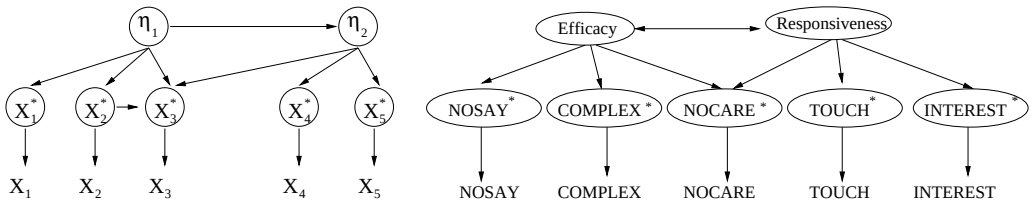


Figure 5.2: Graphical representations of two latent trait models with 5 ordinal observed variables.

analysis. This is similar to performing data analysis with association rules, where rules themselves are taken as independent pieces of knowledge. Each one-factor model can then be seen as a causal association rule with a latent variable as an antecedent and a probabilistic model where observed variables are independent given this latent.

The rest of the chapter is organized as follows: in Section 5.3, we discuss the parametric formulation we will adopt for the problem of learning discrete measurement models. In Section 5.4, we formulate the problem more precisely. Section 5.4.1 describes the variation of BuildPureClusters for local models. Finally, in Section 5.5 we evaluate the method with synthetic and real-world data.

5.3 Latent trait models

Factor analysis and principal component analysis (PCA) are classical latent variable models for continuous measures. For discrete measures, several variations of discrete principal component analysis exist (Buntine and Jakulin, 2004), but they all rely on the assumption that latents are independent. There is little reason, if any at all, to make such an artificial assumption if the goal is causal analysis among the latents.

Several approaches exist for learning models with correlated latent variables. For instance, Pan et al. (2004) present a scalable approach for discovering dependent hidden variables in a stream of continuous measures. While such type of approach might be very useful in practice, they are still not clear on which causal assumptions are being made in order to interpret the latents. In contrast, in the previous chapters we presented a set of well-defined assumptions that are used to infer and justify the choice of latent variables that are generated, based on the axiomatic causality calculus of Pearl (2000); Spirtes et al. (2000). This chapter is on how to extend them to discrete ordinal (or binary) data based on the framework of latent trait models and local models.

Latent trait models (Bartholomew and Knott, 1999) are models for discrete data that in general do not make the assumption of latent independence. However, they usually rely on distributional assumptions, such as a multivariate Gaussian distribution for the latents. We consider such assumptions to be much less harmful for causal analysis than the assumption of full independence, and in several cases acceptable, such as in variables used in social sciences and psychology (Bollen, 1989; Bartholomew et al., 2002).

The main idea in latent trait models is to model the joint latent distribution as a multivariate Gaussian. However, in this model the observed variables are not direct measures of the latents. Instead, the latents in the trait model have other *hidden, continuous measures*. Such extra hidden measures are quantitative indicators of the latent feature of interest. To distinguish these "latent indicators" from the "target latents," we will refer to the former as *underlying variables* (Bartholomew and Knott, 1999; Bartholomew et al., 2002).

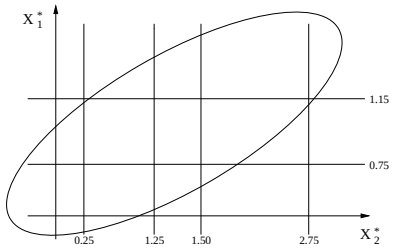


Figure 5.3: Two ordinal variables X_1 and X_2 can be seen as discretizations of two continuous variables X_1^* and X_2^*. The lines in the graph above represent thresholds that define the discretization. The ellipse represents a countourplot of the joint Gaussian distribution of the two underlying continuous variables. Notice that the degree of correlation of the underlying variables has testable implications on the joint distribution of the observed ordinal variables.

This model is more easily understood through a graphical representation. As a graphical model, a latent trait model has three layers of nodes: the first layer corresponds to the latent variables; in the second layer, underlying variables are children of latents and other underlying variables; in the third layer, each discrete measure has a single parent in the underlying variable layer. Consider Figure 5.2(a), for example. The top layer corresponds to our target latents, η_1 and η_2. These targets have underlying measures $X_1^* - X_5^*$. The underlying measures are observed as discrete ordinal variables $X_1 - X_5$.

As another example, consider the following simplified political action survey data set discussed in detail by Joreskog (2004). It consists on a questionnaire intended to gauge how citizens evaluate the political efficacy of their governments. The variables used in this study correspond to questions to which the respondent has to give his/her degree of agreement on a discrete ordinal scale of 4 values. The given variables are the following:

- NOSAY: "People like me have no say on what the government does"
- VOTING: "Voting is the only way that people like me can have any say about how the government runs things"
- COMPLEX: "Sometimes politics and government seem so complicated that a person like me cannot really understand what is going on"
- NOCARE: "I don't think that public officials care much about what people like me think"
- TOUCH: "Generally speaking, those we elect to Congress in Washington lose touch with people pretty quickly"
- INTEREST: "Parties are only interested in people's votes but not in their opinion"

In (Joreskog, 2004), a theoretical model consisting of two latents, one with measures NOSAY, COMPLEX and NOCARE, and another with measures NOCARE, TOUCH and INTEREST is given. This is represented in Figure 5.2(b). The first latent would correspond to a previously established theoretical trait of *Efficacy*, "individuals self-perceptions that they are capable of understanding politics and competent enough to participate in political acts such as voting" (Joreskog, 2004, p. 21). The second latent would be the pre-established trait of *Responsiveness*, "belief that

the public cannot influence political outcomes because government leaders and institutions are unresponsive." VOTING is discarded by Joreskog (2004) for this particular data under the argument that the question is not clearly phrased.

Under this framework, our goal is to discover pieces of the measurement model of the latent variable model. The mapping from an underlying variable X^* to the respective observed discrete variable X is defined as follows. Let X be an ordinal variable of n values, $\{1, \ldots, n\}$. Let $\{\tau_1^X, \ldots, \tau_{n-1}^X\}$ be a set of real numbers such that $\tau_1^X < \tau_2^X < \cdots < \tau_{n-1}^X$. Then:

$$X = \begin{cases} 1 & \text{if } X^* < \tau_1^X; \\ 2 & \text{if } \tau_1^X \leq X^* < \tau_2^X; \\ \ldots & \\ n & \text{if } \tau_{n-1}^X \leq X^*; \end{cases}$$

where the underlying variable X^* with parents $\{z_1^X, \ldots, z_k^X\}$ is given by

$$\begin{array}{rcl} X^* & = & \sum_{i=1}^k \lambda_i^X z_i^X + \epsilon^X; \\ \epsilon & \sim & \mathrm{N}(0, \sigma_X^2); \end{array}$$

where each λ_i^X corresponds to the linear effect of parent z_i^X on X^*, and z_i^X is either a "target latent" or an underlying variable. Latents and underlying variables are centered at zero without loss of generality.

Since the underlying variables can be correlated, this imposes constraints on the observed joint distribution of the ordinal variables. Figure 5.3 illustrates this case for two ordinal variables X_1 and X_2 of 3 and 5 values respectively. The correlation of the two underlying variables corresponding to two ordinal variables in a latent trait model is called the *polychoric* correlation (or *tetrachoric*, if the two variables are binary, Basilevsky, 1994; Bartholomew and Knott, 1999).

Therefore, the fitness of a latent trait model depends on how well the covariance matrix of polychoric correlations fit the respective factor analysis model composed of the latents η and underlying variables $\mathbf{X}^*$. Several estimators and algorithms exist for estimating polychoric correlations (Bartholomew and Knott, 1999) and evaluating latent trait models. They will be essential in our approach for learning measurement models of latent traits as discussed in the next section.

5.4 Learning latent trait measurement models as causal rules

BuildPureClusters was designed to find *pure measurement models*. There were three main reasons why we focused on pure, instead of general, measurement models:

- a pure measurement model with two measures per latent is enough information to learn dependencies among latents;
- a pure measurement model can be estimated much more reliably from data than general models. This will be of special importance in this chapter, where learning models of discrete variables require large samples;
- general, unrestricted, models are not fully identifiable. That is, in general a large number of structures might be compatible with the data. It is not known which equivalence classes exists, or even if a simple representation for such equivalence classes exist;

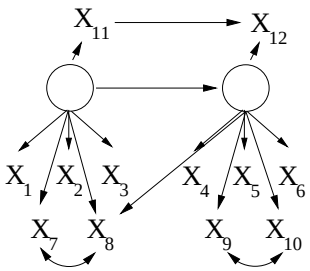


Figure 5.4: Several sets of observed variables can measure the same latent variable. In this case, $\{X_1, X_2, X_3\}$ and any othe observable variable can be used to measure the latent on the left in a way that is detectable by tetrad constraints.

Since in this chapter we are focusing on multivariate Gaussian latents, we will also make the assumption of full linearity, as in Chapter 3. We will adapt this approach for measurement models with discrete binary and ordinal variables. However, we will relax the requirement for pure measurement models. Instead, our algorithm will return a set $\mathcal{S}$ of sets of observed variables $\{\mathbf{S_1}, \ldots, \mathbf{S_k}\}$ by assuming there is a latent trait model G that generated the data. Each set $\mathbf{S} \in \mathcal{S}$ has the following properties:

- there is an unique latent variable L in the true unknown latent trait model where conditioned on L all elements of $\mathbf{S}$ are independent;
- at most one element in $\mathbf{S}$ is not a descendant of L in G;

Furthermore, it is desirable to make each set $\mathbf{S}$ maximal, i.e., no element can be added to it and still make it comply with the two properties above. One can think of each set $\mathbf{S}$ as an "causal association rule" where the antecedent of the rule is a latent variable and the rule is a naive Bayes model where observations are independent given the latent. Since the number of sets with this property might be very large, we further filter such sets as follows:

- sometimes it is possible to find out that two observed variables cannot share any common hidden parent in G. When this happens, we will not consider sets containing such a pair. This can drastically reduce the number of rules and computational time;
- we eliminate some sets in $\mathcal{S}$ that are measuring the same latent as some other set;

In the next section we first describe a variation of BuildPureClusters based on these principles.

5.4.1 Learning measurement models

In order to learn measurement models, one has to discover the following pieces of information concerning the unknown graph that represents the model:

- which latent nodes exist;
- which pairs of observed variables are known not to have any hidden common parent;

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

1. $(\mathbf{Selection}, C, C_0) \leftarrow$FINDINITIALSELECTION$(\Sigma)$.
2. For every pair of nonadjacent nodes $\{N_1, N_2\}$ in C where at least one of them is not in *Selection* and an edge $N_1 - N_2$ exists in C_0, add a RED edge $N_1 - N_2$ to C.
3. For every pair of adjacent nodes $\{N_1, N_2\}$ in C linked by a YELLOW edge, add a RED edge $N_1 - N_2$ to C.
4. For every pair of nodes linked by a RED edge in C, apply successively rules CS1 and CS2. Remove an edge between every pair corresponding to a rule that applies.
5. Let $\mathbf{H}$ be the set of maximal cliques in C.
6. $\mathbf{P_C} \leftarrow$ PURIFYINDIVIDUALCLUSTERS$(\mathbf{H}, C_0, \Sigma)$.
7. Return FILTERREDUNDANT$(\mathbf{P_C})$.

Table 5.1: An algorithm for learning locally pure measurement models. It requires information returned in graphs $\mathcal{C}$ and $\mathcal{C}_0$, which are generated in algorithm FINDINITIALSELECTION, described in Table 5.2.

- which sets of observed variables are independent conditioned on some latent variable;

Using the same assumptions from Chapter 3, it is still the case that the following holds for the *underlying variables*:

Corollary 5.1 *Let G be a latent trait model, and let $\{X_1, X_2, X_3, X_4\}$ be underlying variables such that $\sigma_{X_1X_2}\sigma_{X_3X_4} = \sigma_{X_1X_3}\sigma_{X_2X_4} = \sigma_{X_1X_4}\sigma_{X_2X_3}$. If $\rho_{AB} \neq 0$ for all $\{A, B\} \subset \{X_1, X_2, X_3, X_4\}$, then there is a node P that d-separates all elements in $\{X_1, X_2, X_3, X_4\}$.*

Since no two underlying variables are independent conditional on an observed variable, then node P has to be a latent variable (possibly an underlying variable).

This is not enough information. In Figure 5.4 (repeated from 3.12(a)), for instance, the latent node on the left d-separates $\{X_1, X_2, X_3, X_4\}$, and the latent on the right d-separates $\{X_1, X_4, X_5, X_6\}$. Although these one-factor models are sound, we would rather not include X_1 and X_4 in a same rule since they are not children of the same latent. We accomplish this by detecting as many observed variables that cannot (directly) measure any common latent as possible. In this case, pairs in $\{X_1, X_2, X_3, X_7, X_{11}\} \times \{X_4, X_5, X_6, X_9, X_{10}\}$ can be separated using the CS rules of Chapter 3.

The algorithm BUILDSINGLEPURECLUSTERS (BSPC, Table 5.1) makes use of such results in order to learn latents with respective sets of pure indicators. However, we need an initial step called FINDINITIALSOLUTION (Table 5.2) due to the same reasons explained in Appendix A.3: to reduce the number of false positives when applying the CS rules.

The goal of FINDINITIALSELECTION is to find a pure submodels using only DISJOINTGROUP (defined in Appendix A.3) instead of CS1 or CS2 (CS3 is not used in our implementation because it tends to commit many more false positive mistakes). These pure submodels are then used as an starting point for learning a more complete model in the remaining stages of BUILDSINGLEPURECLUSTERS.

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

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

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

   Otherwise, if there is no node N_3 that forms a BLUE clique with $\{N_1, N_2\}$ in C, and no BLUE clique in $\{N_4, N_5, N_6\}$ such that all six nodes form a clique in C_0 and DisjointGroup$(N_1, N_2, N_3, N_4, N_5, N_6; \Sigma) = true$, then change the color of the edge $N_1 - N_2$ to YELLOW.
7. Remove all GRAY and YELLOW edges from C.
8. $\mathbf{List_C} \leftarrow$FindMaximalCliques$(C)$.
9. $\mathbf{P_C} \leftarrow$PurifyIndividualClusters$(\mathbf{List_C}, C_0, \Sigma)$.
10. $\mathbf{F_C} \leftarrow$ FilterRedundant$(\mathbf{P_C})$.
11. Let **Selection** be the set of all elements in $\mathbf{P_C}$.
12. Add all GRAY and YELLOW edges back to C.
13. Return $(\mathbf{Selection}, C, C_0)$.

Table 5.2: Selects an initial pure model.

The definition of FindInitialSelection in Table 5.2 is slightly different from the one in Appendix A.3. It is still the case that if a pair $\{X, Y\}$ cannot be separated into different clusters, but also does not participate in any true instantiation of DisjointGroup in Step 6 of Table Table 5.2, then this pair will be connected by a GRAY or YELLOW edge: this indicates that these two nodes cannot be in a pure submodel with two latents and three indicators per latent. Otherwise, these nodes are "compatible," meaning that they *might* be in such a pure model. This is indicated by a BLUE edge.

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

Algorithm	PURIFYINDIVIDUALCLUSTERS
Inputs:	**Clusters**, a set of subsets of some set **O**; an undirected graph G_0; Σ, a sample covariance matrix of **O**.

1. **Output** $\leftarrow \emptyset$
2. Repeat Steps 3-8 below for all $Cluster \in$ **Clusters**
3. If $Cluster$ has two variables X, Y only, verify if there are two other variables W and Z in **O** such that: $\sigma_{XY}\sigma_{WZ} = \sigma_{XW}\sigma_{YZ} = \sigma_{XZ}\sigma_{WY}$ and all variables in $\{W, X, Y, Z\}$ are adjacent in G_0. If true, add $Cluster$ to **Output**.
4. If $Cluster$ has three variables X, Y, Z only, verify if there is a fourth variable W in **O** such that: $\sigma_{XY}\sigma_{WZ} = \sigma_{XW}\sigma_{YZ} = \sigma_{XZ}\sigma_{WY}$ and all variables in $\{W, X, Y, Z\}$ are adjacent in G_0. If true, add $Cluster$ to **Output**.
5. If $Cluster$ has more than three variables
6. For each pair of variables $\{X, Y\}$ in $Cluster$, if there is no pair of nodes $W, Z \in Cluster$ such that $\sigma_{XY}\sigma_{WZ} = \sigma_{XW}\sigma_{YZ} = \sigma_{XZ}\sigma_{WY}$, add a GRAY edge $X - Y$ to $Cluster$.
7. While there are GRAY edges in $Cluster$, remove the node with the largest number of adjacent nodes
8. If $Cluster$ has more than three variables, add it to **Output**. Otherwise, add it to **Output** if and only if the

   criteria in Steps 3 or 4 can be applied.
9. Return **Output**.

Table 5.3: Identifying the pure measures per cluster.

After we find pure one-factor models, we filter those that are judge to be redundant. For instance, if two sets in $\mathbf{P_C}$ have a common intersection of at least three variables, we known that theoretically they are related to the same latent (follows from Corollary 5.1). We order the elements in $\mathbf{P_C}$ by size[2] and remove sets that either have a large enough intersection with a previously added set or where all (but possibly one) of its elements are contained in the union of the previously added sets. Table 5.4 describes this process in more detail.

5.4.2 Statistical tests for discrete models

It is clear that the same tetrad constraints used in the continuous case can be applied to underlying latent variables in the respective latent trait model. The difference lies on how to test such constraints. For the continuous case, there are fast Gaussian and large-sample distribution free tests of tetrad constraints, but for latent trait models tests are relatively expensive.

To test if a tetrad constraint $\sigma_{XZ}\sigma_{WY} = \sigma_{XW}\sigma_{YZ}$ holds, we fit a latent trait model with two latents η_1, η_2, where η_1 is a parent of η_2[3]. Each latent has two underlying variables as children: X^* and Y^* for η_1; W^* and Z^* for η_2. Each underlying variable has the respective observed indicator.

[2]Ties are broken randomly in our implementation. Instead, one can implement different criteria, such as the sum of the absolute value of the polychoric correlations within each set in $\mathbf{P_C}$.

[3]This model is probabilistically identical to the one where the edge $\eta_1 \rightarrow \eta_2$ is reversed.

Algorithm FILTERREDUNDANT
Inputs: **Clusters**, a set of subsets of some set **O**;

1. **Output** $\leftarrow \emptyset$.
2. Sort **Clusters** by size in descending order
3. For all elements $Cluster \in$ **Clusters** according to the given order
4. If there is some element in **Output** that intersects *Clusters* in three or more variables, skip to the next element of **Clusters**
5. Let N be the number of elements of *Cluster*
6. If at least $N-1$ elements of *Cluster* are present in the union of the elements of **Output**, skip to the next element of **Clusters**
7. Otherwise, add *Clusters* to **Output**.
8. Return **Output**.

Table 5.4: Filtering redundant clusters.

The tetrad will judged to hold in the population if the model passes a χ^2 test at a pre-defined significance level (Bartholomew and Knott, 1999). Testing if all three tetrads hold is analogous, using a single latent η.

Ideally, one would like to use full-information methods, i.e., methods where all parameters are fit simultaneously, such as the maximum likelihood estimator (MLE). However, finding the MLE is relatively computationally expensive even for a small model of four variables. Since our algorithm might require thousands of such estimations, this is not a feasible method.

Instead, we use a three-stage approach. Similar estimators are used, for instance, in commercial systems such as LISREL (Joreskog, 2004). Testing a latent trait model is done by the following steps:

1. let X be an ordinal variable taking values in the space $\{1, 2, \ldots, m(X)\}$. Estimate the threshold parameters $\{\tau_1^X, \ldots, \tau_{m(X)}^X\}$ by direct inversion of the normal cumulative distribution function Φ using the empirical counts. That is, given the marginal empirical counts $\{n_1^X, \ldots, n_{m(X)}^X\}$ corresponding to the values of X in a sample of size N, estimate τ_1^X as $\Phi^{-1}(n_1^X/N)$. Estimate $\tau_{m(X)}^X$ as $\Phi^{-1}(1 - n_{m(X)}^X/N)$. Estimate τ_i^X, $1 < i < m(X)$, as $\Phi^{-1}((n_i^X - n_{i-1}^X)/N)$.

2. in this step we estimate the polychoric correlation independently for each pair. This is done by maximum likelihood. Let the model loglikelihood function for a pair X, Y be given by

$$L = \sum_{i=1}^{m(X)} \sum_{j=1}^{m(Y)} n_{ij} \log \pi_{ij}(\rho) \tag{5.1}$$

where $\pi_{ij}(\rho)$ is the population probability of the event $\{X = i, Y = j\}$ with polychoric correlation ρ and n_{ij} is the corresponding empirical count. Probability $\pi_{ij}(\rho)$ is given by

$$\pi_{ij}(\rho) = \int_{\tau_i^X}^{\tau_{i+1}^X} \int_{\tau_j^Y}^{\tau_{j+1}^Y} \phi_2(u, v, \rho) du dv \tag{5.2}$$

where ϕ_2 is the bivariate normal density function with zero mean and correlation coefficient ρ. Thresholds are fixed according to the previous step. We therefore optimize (5.1) with respect to ρ only. Gradient-based optimization methods can be used here.

3. given all estimates of polychoric correlations, we have an estimate of the correlation matrix of the underlying variables, $\Sigma(\Theta)$. To test the corresponding latent trait model, we fit $\Sigma(\Theta)$ to the factor analysis model corresponding to the latents and underlying variables to get an estimate of the coefficient parameters. We then calculate the expected cell probabilities and return the p-value corresponding to the χ^2 test.

The drawback of this estimator is that is not as statistically efficient as the MLE. This means that our method is unreliable with small sample sizes. We recommend a sample size of at least 1,000 data points, even for binary variables. An open problem would be adjusting for the "actual sample size" used in the test, since the estimated covariance among underlying variables has more variance than the sample covariance matrix that would be estimated if such variables were observed. Therefore, this indirect test of tetrad constraint among latent variables has less power than the respective test for observed variables used in Chapter 3. However, false positives are still the main concern of any causal discovery algorithm that relies on hypothesis testing.

In our implementation, we use significance tests in two ways to minimize false positives/false negatives in rules CS1 and CS2. These rules have in their premises tetrad constraints that need to be true or need to be false in order for a rule to apply. For those constraints that need to be true, we require the corresponding p-value to be at least 0.10. For those constraints that need to be false, we require that the corresponding p-value to be at most 0.01. Those values were chosen by doing preliminary simulations.

5.5 Empirical evaluation

In the following sections we evaluate BSPC in a series of simulated experiments where the ground truth is known. We also report exploratory results in two real data sets. In the simulated cases, we report statistics about the number of association rules that the standard algorithm APRIORI (Agrawal and Srikant, 1994) returns on the same data. The goal is to provide evidence that in the presence of latent variables, association rules might produce thousands of rules, even though it fails to actually capture the causal processes that are essential in policy making.

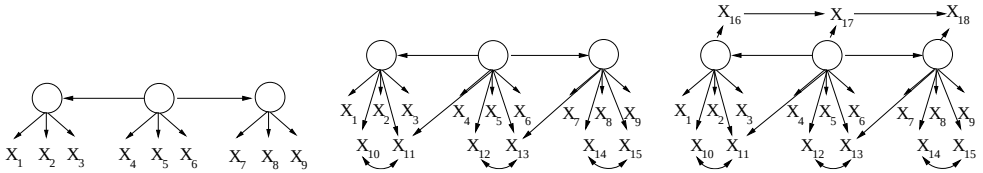
The APRIORI algorithm is an efficient search procedure that generates association rules in two phases. We briefly describe it for the case where variables are binary. In the first stage, all sets of variables that are of high support[4] are found. This search is made efficient by first constructing sets of small size and only looking for larger sets with by expanding small sets that are frequent enough. Notice that this only generates sets of positive association. Within each frequent set, APRIORI finds conditional probabilities that are of high confidence[5] .

5.5.1 Synthetic experiments

Let G be our true graph, from which we want to extract features of the measurement model as causal rules. The graph is known to us by simulation, but it is not known to the algorithm. The

[4]That is, they co-occur in a large enough number of database records, according to some given threshold.

[5]That is, given a frequent set of binary variables $\mathbf{X} = \{X_1, X_2, \ldots, X_k\}$, it attempts to find a partition of $\mathbf{X} = \mathbf{X_A} \cup \mathbf{X_B}$ such that $P(\mathbf{X_B} = \mathbf{1} | \mathbf{X_A} = \mathbf{1})$ is above some threshold.



MM1 MM2 MM3

Figure 5.5: The measurement models used in our simulation studies.

goal of experiments with synthetic data is to objectively measure the performance of BSPC in finding correct and informative latent causal rules of ordinal variables from G.

Correctness in our setup is measured by a *Precision* statistic. That is,

- given $\mathcal{S}$, a set of latent causal rules, and $S_i \in \mathcal{S}$ a particular rule, the *individual precision* of S_i is the proportion of observed variables in S_i that are d-separated given an unique latent in G. The *precision* of the set $\mathcal{S}$ is the average of the respective individual precisions.

For example, if $\mathcal{S} = \{S_1, S_2, S_3\}$, 4 out of 5 observed variables in S_1 are d-separated by latent L_x in G, 3 out of 3 observed variables in S_2 are d-separated by a latent L_y in G, 2 out of 3 observed variables in S_3 are d-separated by a latent L_z in G, then the precision of $\mathcal{S}$ is $(4/5+1+2/3)/3 \sim 0.82$.

Completeness in our setup is measured by a *Recall* statistic. That is,

- given $\mathcal{S}$, a set of latent causal rules, the *recall* of $\mathcal{S}$ is the proportion of latents $\{L_i\}$ in G such that there is at least one rule in $\mathcal{S}$ containing at least two children of L_i and at most one observed variable that is not a child[6] of L_i.

For example, if G has four latents, and three of them are represented by some rule in $\mathcal{S}$ as described above, then the recall of $\mathcal{S}$ is 0.75.

In our study we use the three graphs depicted in Figure 5.5, similar to some of the graphs used in Chapter 3. MM1 has already a pure measurement model. MM2 has two possible pure submodels: the one including $\{X_1, \ldots, X_{10}, X_{12}, X_{14}\}$, and another including X_{15} instead of X_{14}. MM3 has the same pure measurement models as MM2 with the addition of indicator X_{16}.

Notice that in our experiments all latents are potentially identifiable by *BSPC*. The goal is not to test its assumptions, but to evaluate how well it performs in finite samples.

Given each graph, we generated 20 parametric models. 10 of such models were used to generate samples of 1,000 cases. The remaining 10 were used to generate samples of 5,000 cases. The total number of runs of our algorithm is therefore 60. To facilitate comparison against Apriori, all observed variables are binary. The sampling scheme to generate synthetic models and data was as follows:

1. Pick coefficients for each edge in the model randomly from the interval $[-1.5, -0.5] \cup [0.5, 1.5]$ (all latents and underlying variables can have zero mean without loss of generalization).

[6]Since in some cases it is not theoretically possible to rule out this possibility (Chapter 3).

Evaluation of BSPC **output**			
Sample	**Precision**	**Recall**	**#Rules**
MM_1			
1000	1.00(.0)	0.97(.1)	3.2(.4)
5000	0.98(.05)	0.97(.1)	2.9(.3)
MM_2			
1000	0.94(.04)	1.00(.0)	3.2(1.03)
5000	0.94(.05)	1.00(.0)	3.4(0.70)
MM_3			
1000	0.90(.06)	0.90(.16)	4.2(.91)
5000	0.90(.08)	0.90(.22)	3.5(.52)

Table 5.5: Results obtained with BUILDSINGLEPURECLUSTERS for the problem of learning measurement models as causal rules. Each number is an average over 10 trials, with the standard deviation over these trials in parenthesis.

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

3. Normalize coefficients such that all underlying variables have variance 1;

4. For each of the two values of a given observed binary variable, generate a random integer in $\{1, \ldots, 5\}$ as the "weight" of the value. Normalize the weights to sum to 1. Set each threshold τ_k to $\Phi^{-1}(S_k)$, where Φ^{-1} is the inverse of the cumulative distribution function of a normal (0, 1) variable, and S_k is the sum of the weights of values $1, \ldots, k$.

A similar sampling scheme for continuous data was used in Chapter 3. It is not an easy setup, and some of the variables might have a high proportion of its variance due to the error terms. Factor analysis failed to produce meaningful results under this sampling scheme, for instance.

Results are displayed in Table 5.5 using the evaluating criteria introduced in the beginning of this section. We also display the number of rules that are generated. Ideally, in all cases we should generate exactly 3 rules. However, due to statistical mistakes, more or less than 3 rules can be generated. It is noticeable that there is a tendency to produce more rules than necessary as the measurement increases in complexity. It is also worthy to point out that without the filtering described in the previous section, we obtain around around 5 to 8 rules in most of the experiments, with a larger difference between the results at a sample size of 1,000 compared to 5,000.

As a comparison, we report the distribution of rules generated by APRIORI in Table 5.6. The implementation used is the one of Borgelt and Kruse (2002) with the default parameters. We report the maximum and minimum number of rules for each model and sample size across the 10 trials, as well as average and standard deviation. The outcome is that not only APRIORI generates a very large number of rules, the actual number per trial varies enormously. For MM1 at sample size 5000, we had a trial with as few as 9 rules, and one with as much as 546, even though the causal process that generated the data is the same across trials.

5.5.2 Evaluations on real-world data

This section describes the use of BSPC on two real-world data sets. Unlike the synthetic data study, we do not have objective measure of evaluation. Instead, we will use data sets whose results

APRIORI **statistics**				
Sample	**MIN**	**MAX**	**AVG**	**STD**
MM_1				
1000	15	159	81	59.4
5000	9	546	116	163.9
MM_2				
1000	243	2134	1070.4	681.2
5000	336	3565	1554.7	1072.2
MM_3				
1000	363	6036	2916.7	1968.7
5000	158	4434	2608.3	1214.6

Table 5.6: Results obtained with APRIORI. Our goal is to evaluate how many association rules are generated when hidden variables are the explanation of all observed associations. Not only the number of rules is overwhelming, but the algorithm depicts a high variability in the number. For each combination of model (MM_1, MM_2 and MM_3) and sample size (1000, 5000) we show the least number of rules (MIN) in 10 independent trials, the maximum number (MAX), the average (AVG) and standard deviation (STD).

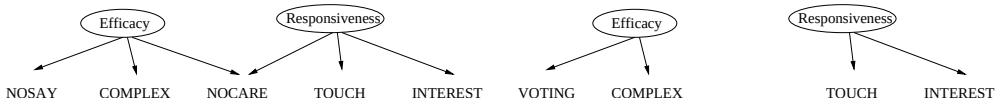


Figure 5.6: A theoretical model for the voting dataset is shown in (a), while BSPC output is shown in (b).

can be reasonably evaluated by using common-sense knowledge.

Political action survey

We start with a very simple example consisting of a data set of six variables only. The data set is the simplified polical action survey data set discussed in Section 5.3. We used a subset of the data where missing values were filled by a particular method given in (Joreskog, 2004). This data is available as part of the LISREL software for latent variable analysis. The model chosen by Joreskog (2004) is shown again, without the underlying variables and latent connection, in Figure 5.6(a). Recall that variable VOTING is discarded by Joreskog (2004) for this particular data under the argument that the question is not clearly phrased, an argument we believe to be unsubstantial. In our data-driven approach, we also found two latents: one corresponding to NOSAY and VOTING; another corresponding to TOUCH and INTEREST. This is shown in Figure 5.6(b). Our output partially matches the theoretical model without making any use of prior knowledge.

Freedom and tolerance data set: self-evaluation of social attitude

We applied BSPC to the data collected in a 1987 study[7] on freedom and tolerance in the United States (Gibson, 1991). This is a large study comprising 381 questions targeting political tolerance and perceptions of personal freedom in the United States. 1267 respondents completed the inter-

[7]Available at http://webapp.icpsr.umich.edu/cocoon/ICPSR-STUDY/09454.xml

view. Each question is an ordinal variable with 2 to 5 levels, often with an extra non-ordinal value corresponding to a "Don't know/No answer" reply.

However, several questions are explicity dependent on answers given to previous questions[8]. To simplify the task, in this empirical evaluation we will first focus on a particular section of this questionnaire, the Deck 6. Other subsection of the study is used in a separate experiment described in the next section.

This deck of questions is composed of a self-administred questionnaire of 69 items concerning an individual's attitude with respect to other people. Answers corresponding to "Don't know/No answer" usually amounted to 1% of all respondents for each question. We modified these answers on each question to correspond to the majority answer to avoid throwing away data.

The measurement model obtained by BSPC was a set of 15 clusters (i.e., causal latent rules) where 40 out of the 69 questions appear on at least on rule. All clusters with at least three observed variables are depicted in Tables 5.7 and 5.8.

There is a clear relation among items within most rules. For instance, items on Rule 1 of Table 5.7 correspond to measures of a latent trait of empathy and easiness of communication. This causal view of the associations among these questions makes more sense than a set of association rules without a latent variable.

Rule 2 has three items (X28, X30, X61) that clearly correspond to measures of a tendency of impulsive reaction. The fourth item (X41) is not clearly related to this trait, but the data supports the idea that this latent trait explains the associations between pushing oneself too much and reacting strongly to other people's actions and ideas.

Rule 3 is clearly a set of indicators of the trait of deciding when to change one's mind and plan of action. Rule 4 is apparently due to a more specific latent along the same lines: unwillingness to change according to other people's opinion. It is interesting to note that is theoretically plausible that different rules might correspond to different latents and share a same observed variable (X9 in this case).

Rule 5 overlaps with Rule 1, and again stress indicators of ability to communicate with other people and understand other people's ideas. Rule 6 is a set corresponding to a latent trait of attitude to risks. Rule 7 seems to be explained by a trait of being energic on implementing one's ideas. Rule 8 is a rule measuring the ability of remaining calm under difficult conditions, and seems to have some overlap with Rule 6. Rule 9 is not completely clear because of item X37, and conceptually appears to overlap with Rule 7. Finally, Rule 10 is a rule where the associations are apparently due to a latent variable concerning individualism.

It is also interesting to stress that each estimated rule is composed of questions that are not physically adjacent in the actual questionnaire. Rule 1, for example, is composed of questions scattered over the response form (X3, X7, X27, X31, X61). The respondents are not estimulated to respond in a similar pattern by trying to keep coherence or balance with respect to previous answers.

Althought this given set of causal latent rules might not be perfect, they do explain a lot concerning the mechanisms explaining observed associations using very few rules.

[8]For instance, opinions about a particular political group that was selected by the respondent on a previous question, or whole sets of answers where only a subset of the individuals are asked to fill out.

Rule 1	
X27	I feel it is more important to be sympathetic and understanding of other people than to be practical and tough-minded
X3	I like to discuss my experiences and feelings openly with friends instead of keeping them to myself
X31	People find it easy to come to me for help, sympathy, and warm understanding
X67	When I have to meet a group of strangers, I am more shy than most people
X7	I would like to have warm and close friends with me most of the time
Rule 2	
X28	I lose my temper more quickly than most people
X30	I often react so strongly to unexpected news that I say or do things that I regret
X41	I often push myself to the point of exhaustion or try to do more than I really can
X61	I find it upsetting when other people don't give me the support that I expect from them
Rule 3	
X9	I usually demand very good practical reasons before I am willing to change my old ways of doing things
X53	I see no point in continuing to work on something unless there is a good chance of success
X46	I like to think about things for a long time before I make a decision
Rule 4	
X9	I usually demand very good practical reasons before I am willing to change my old ways of doing things
X17	I usually do things my own way − rather than giving in to the wishes of other people
X11	I hate to change the way I do things, even if many people tell me there is a new and better way to do it
Rule 5	
X3	I like to discuss my experiences and feelings openly with friends instead of keeping them to myself
X40	I am slower than most people to get excited about new ideas and activities
X12	My friends find it hard to know my feelings because I seldom tell them about my private thoughts

Table 5.7: Clusters of variables obtained by BSPC on Deck 6 of the Freedom and Tolerance data set. On the left column, the question number according to the original questionnaire. On the right column, the respective textual description of the question.

Freedom and tolerance data set: tolerance concerning freedom of speech and government perception

We applied BSPC to data corresponding to Decks 4 and 5 of the same study described in the previous section. We removed two questions from Deck 4 that could be answered only by some respondents (questions 58B and 59B). We did the same in Deck 5, keeping only all subitems of questions 86, 87, 90-93. As in the data set from the previous section, every item should be answered according to an ordinal measure of agreement. Blank values and "don't know" answers were processed to reflect the opinion of the majority. The total number of items amounted to 70. The reason why we did not use any of the other decks in our experiments was mostly due to interdependence between answers (i.e., an answer in one question explicitly affecting other answers, or determining which other questions should skipped).

Questions in our 70 item dataset were mostly about attitude to tolerance of freedom of speech, how one interacts with other people to discuss sensitive issues, and how one perceives the role of the government in freedom of speech issues. 52 items out of the 70 appear in some rule given by the output of BPC. All rules as given in Tables 5.9, 5.10 and 5.11. Unfortunately, BSPC did not cluster such items into well separated causal rules as in the previous cases.

There is a considerable overlap between some rules. For instance, questions about one's attitude

Rule 6	
X51	Most of the time I would prefer to do something risky (like hanggliding or parachute jumping) − rather than having to stay quiet and inactive for a few hours
X47	Most of the time I would prefer to do something a little risky (like riding in a fast automobile over steep hills and sharp turns) − rather than having to stay quiet and inactive for a few hours
X29	I am usually confident that I can easily do things that most people would consider dangerous (such as driving an automobile fast on a wet or icy road)
Rule 7	
X52	I am satisfied with my accomplishments, and have little desire to do better
X54	I have less energy and get tired more quickly than most people
X57	I often need naps or extra rest periods because I get tired so easily
Rule 8	
X8	I nearly always stay relaxed and carefree, even when nearly everyone else is fearful
X1	I usually am confident that everything will go well, in situations that worry most people
X26	I usually stay calm and secure in situations that most people would find physically dangerous
Rule 9	
X59	I am more energetic and tire less quickly than most people
X49	I try to do as little work as possible, even when other people expect more of me
X37	I often avoid meeting strangers because I lack confidence with people I do not know
Rule 10	
X15	It wouldn't bother me to be alone all the time
X58	I don't go out of my way to please other people
X38	I usually stay away from social situations where I would have to meet strangers, even if I am assured that they will be friendly

Table 5.8: Continuation of Table 5.7.

about discussing polemical/sensitive opinions, better reflected by Rule 11 (Table 5.11), are scattered around other rules. Questions concerning the Supreme Court (Rule 1, Table 5.9) are not in a rule of their own, as one would expect a priori. Questions about expression of racist opinions are also scattered. Questions about indirect demonstrations of support (wearing buttons, putting a sign in front of one's house), as in Rule 6 (Table 5.10), are well-clustered, but still mixed with barely related questions. Although every rule (perhaps with the exception of Rule 3, Table 5.9) might be individually interpreted as measuring one broad latent concept concerning freedom of speech, from a global point of view some groups of questions are intuitively measuring a more specific trait (e.g., attitude with respect to the Supreme Court). This partially undermines the results, since the given clustering is not as informative as it could be. An interesting question for future research is if more statistically robust approaches for learning discrete measurement models could detect more fine-grained differences in this data set, or if the data itself is too noisy too allow further conclusions.

5.6 Summary

We introduced a novel algorithm for finding associations among discrete variables due to hidden common causes. It can be described as a method for clustering variables based on explicit causal assumptions.

Our emphasis in comparing BSPC with association rules is due to the fact that none of the approaches tries to find a global model that includes all variables, and both are primarily used for policy making. That is, they are used in the deduction of causal processes by a combination of data-driven submodels and prior knowledge. However, generic latent variable models are usually ad-hoc methods, unlike BSPC.

One method is not intended to substitute the other. Latent trait models rely on substantial parametric assumptions, while association rules do not. Association rules can also be much more scalable when the required rule supports are relatively high and data is sparse. However, standard association rules, or even causal rules, do not make use of latent variables, which might result in an very complicated and ultimately unusable model for policy making.

The assumption of a Gaussian distribution for latent variables was essential in the approach described here. Bartholomew and Knott (1999) argue that for domains such as social sciences and econometrics, such assumptions are not harmful if the goal is parameter estimation. However, two issues remain unclear: how well the method tetrad tests work with small deviations from normality; and which kind of output will be generated if the model deviates considerably from the assumptions (i.e., if a nearly empty model will be generated - which is good, or if a large spurious model will be the output instead). Work in non-parametric item response theory (Junker and Sijtsma, 2001) might provide more flexible causal models, although it is unclear how robust such methods could be.

Scalability is also a very important issue. Fast clustering procedures for discrete variables, as the one proposed by Chakrabarti et al. (2004), might be crucial as an initialization procedure, spliting the task of finding one-factor models on disjoint sets of variables.

Rule 1	
X7	Should we allow a speech extremely critical of the U. S. Constitution?
X12	It is better to live in an orderly society than to allow people so much freedom that they can become disruptive.
X14	Free speech is just not worth it if it means that we have to put up with the danger to society of radical and extremist political views.
X15	When the country is in great danger we may have to force people to testify against themselves in court even if it violates their rights.
X17	No matter what a person's political beliefs are, he is entitled to the same legal rights and protections as anyone else.
X19	Any person who hides behind the laws when he is questioned about his activities doesn't deserve much consideration.
X24	Would you say you engage in political discussions with your friends?
X31	Would you be willing to sign a petition that would be published in the local newspaper with your name on it supporting the unpopular political view?
X43	Do you think the government would allow you to organize a nationwide strike of all workers to oppose the actions of the government?
X46	If the Supreme Court continually makes decisions that the people disagree with, it might be better to do away with the Court altogether.
X48	It would not make much difference to me if the U.S. Constitution were rewritten so as to reduce the powers of the Supreme Court.
X49	The power of the Supreme Court to declare acts of Congress unconstitutional should be eliminated.
X50	The right of the Supreme Court to decide certain types of controversial issues should be limited by the Congress.
Rule 2	
X3	If such a person wanted to make a speech in your community claiming that Blacks are inferior, should he be allowed to speak, or not?
X6	Should such a person be allowed to organize a march in your community, and claim that Blacks are inferior?
X20	Because demonstrations frequently become disorderly and disruptive, radical and extremist political groups shouldn't be allowed to demonstrate.
X39	Would you be allowed to publish pamphlets to oppose the actions of the government?
X40	Would you be allowed to organize protest marches and demonstrations to oppose the actions of the government?
X43	Do you think the government would allow you to organize a nationwide strike of all workers to oppose the actions of the government?
X59	How likely is it that you would try to get the government to stop the demonstration (of an undesired group)?
X60	How likely is it that you would try to get people to go to the demonstration (of an undesired group) and stop it in any way possible, even if it meant breaking the law?
X62	Or would you do nothing to try to stop the demonstration from taking place?
Rule 3	
X6	Should such a person be allowed to organize a march in your community, and claim that Blacks are inferior?
X9	Do you believe it should be allowed... A speech advocating the overthrow of the U.S. Government.
X21	I believe in free speech for all, no matter what their views might be.
X65	How likely would you be to try to get the legislature's decision reversed by some other governmental body or court?

Table 5.9: Clusters of variables obtained by BSPC on Decks 4 and 5 of the Freedom and Tolerance data set. On the left column, the question number according to the order they appear in the original questionnaire. On the right column, a simplified textual description of the question. See Gibson (1991) for more details.

Rule 4	
X14	Free speech is just not worth it if it means that we have to put up with the danger to society of radical and extremist political views
X22	It is refreshing to hear someone stand up for an unpopular political view, even if most people find the view offensive.
X25	Would you say you engage in political discussions with casual acquaintances?
X43	Do you think the government would allow you to organize a nationwide strike of all workers to oppose the actions of the government?
X44	Now, on a different subject, some people pay attention to what the United States Supreme Court is doing most of the time. Others aren't that interested. Would you say that you pay attention to the Supreme Court most of the time, some of the time, or hardly at all?
X68	My local government council usually gives interested citizens an opportunity to express their views before making its decisions.
Rule 5	
X5	If some people in your community suggested that a book he wrote which said Blacks are inferior should be taken out of your public library, would you favor removing this book, or not?
X11	Do you believe a speech that might incite listeners to violence should be allowed?
X15	When the country is in great danger we may have to force people to testify against themselves in court even if it violates their rights.
X18	Do you agree strongly, agree, disagree, or disagree strongly with this: Free speech ought to be allowed for all political groups even if some of the things they say are highly insulting and threatening to some segments of society.
X19	Any person who hides behind the laws when he is questioned about his activities doesn't deserve much consideration.
X20	Because demonstrations frequently become disorderly and disruptive, radical and extremist political groups shouldn't be allowed to demonstrate.
X38	Do you think the government would allow you to organize public meetings to oppose the government?
X40	Would you be allowed to organize protest marches and demonstrations to oppose the actions of the government?
X59	How likely is it that you would try to get the government to stop the demonstration?
Rule 6	
X31	Would you be willing to sign a petition that would be published in the local newspaper with your name on it supporting the unpopular political view?
X32	Would you be willing to wear a button to work or in public in support of the unpopular view?
X33	Would you be willing to put a bumper sticker on your car in support of that position?
X34	Would you be willing to put a sign in front of your home or apartment in support of the unpopular view?
X35	Would you be willing to participate in a demonstration in support of that position?
X45	In general, would you say that the Supreme Court is too liberal or too conservative or about right in its decisions?
X49	The power of the Supreme Court to declare acts of Congress unconstitutional should be eliminated.
Rule 7	
X1	Should we allow a speech extremely critical of the U. S. Constitution?
X2	Do you think that a book he (a writer with racist views) wrote should be removed from a public library?
X8	Should we allow a speech extremely critical of various minority groups?
X67	The members of my local government council seldom consider the views of all sides to an issue before making a decision.

Table 5.10: Continuation of Table 5.9.

Rule 8	
X4	Should such a person (a person of racist position) be allowed to teach in a college or university, or not?
X6	Should such a person be allowed to organize a march in your community, and claim that Blacks are inferior?
X8	Should we allow a speech extremely critical of various minority groups?
X9	Should we allow a speech advocating the overthrow of the U.S. Government?
X10	Should we allow a speech designed to incite listeners to violence?
X59	How likely is it that you would try to get the government to stop the demonstration?
X65	How likely would you be to try to get the legislature's decision reversed by some other governmental body or court?
X66	How likely is it that you would do nothing at the moment but vote against the members of the local legislature at the next election?
Rule 9	
X23	Would you say you engage in political discussions with your family?
X29	Best not to say anything about (polemical issues) to casual acquaintances.
X30	Best not to say anything about (polemical issues) to your neighbors.
X44	Now, on a different subject, some people pay attention to what the United States Supreme Court is doing most of the time. Others aren't that interested. Would you say that you pay attention to the Supreme Court most of the time, some of the time, or hardly at all?
X46	If the Supreme Court continually makes decisions that the people disagree with, it might be better to do away with the Court altogether.
X48	It would not make much difference to me if the U.S. Constitution were rewritten so as to reduce the powers of the Supreme Court.
Rule 10	
X28	Have you ever had a political view that was so unpopular that you thought it best not to say anything about it to your friends?
X51	I am sometimes reluctant to talk about politics because it creates enemies.
X56	I am sometimes reluctant to talk about politics because I don't like arguments.
Rule 11	
X26	Would you say you engage in political discussions with your neighbors?
X27	Best not to say anything about (polemical issues) to your family.
X28	Best not to say anything about (polemical issues) to your friends.
X30	Best not to say anything about (polemical issues) to your neighbors.

Table 5.11: Continuation of Table 5.10.

Chapter 6

Bayesian learning and generalized rank constraints

BuildPureClusters is an algorithm for learning the causal structure of latent variable models by testing tetrad constraints at a given significance level. In Chapter 3, a large batch of experiments demonstrated that this algorithm is robust for multivariate Gaussian distributions. However, this will not be the case for more complicated distributions such as mixtures of Gaussians. In this chapter, we introduce a score-based algorithm based on the principles of BuildPureClusters that is more effective in handling mixture of Gaussians distributions.

Moreover, we evaluate how a modification of this algorithm can be used in the problem of density estimation. This is motivated by several algorithms based on factor analysis and its variants that are used in unsupervised learning (i.e., density estimation). Such algorithms have applications in, e.g., outlier detection and classification with missing data. In factor analysis for density estimation, the goal is to smooth the data by introducing rank constraints in the covariance matrix of the observed variables. Our modified algorithm searches for rank constraints in a relatively efficient way inspired by the clustering idea of BuildPureClusters. Experiments demonstrate the suitability of this approach.

6.1 Causal learning and non-Gaussian distributions

In Chapter 4, we performed experiments using BuildPureClusters to find a measurement model for a set of latents whose distribution deviated considerably from a multivariate Gaussian. Conditioned on the latents, however, the observed variables were still Gaussian. The performance of the algorithm was not as good as in the experiments of Chapter 3, where all variables were multivariate Gaussian, but still reasonable.

Results get considerably worse when the population follows a mixture of Gaussians distribution, where observed variables are not Gaussian given the latents. For instance, in the case where each conditional distribution of an indicator given its latent parents also depends on the mixture component. In this case, the number of false positive tests of tetrad constraints is high even for reasonable sample sizes. In simulation studies using the same graphs of Chapter 3 and a mixture of Gaussians model, one can show that BPC will return a mostly empty model.

This chapter describes alternative algorithms inspired by BuildPureClusters to learn a graphical structure using a mixture of Gaussians model. The focus on mixtures of Gaussians is due

to two main reasons:

- first, in causal models it is of interest to model a mixture of Gaussian-distributed populations that follow the same causal linear structure, but with different parameters (e.g., the distribution of physiological measurements given the latent factors of interest might differ in different genders, and yet the graphical structure of the measurement model is the same). Since the variable determining the mixture component can be hidden, we need a mixture of Gaussians approach in this case;
- second, a mixture of Gaussians is a practical and flexible model for the multivariate distribution of a population (Roeder and Wasserman, 1997; Mitchell, 1997), especially when data is limited and more sophisticated models cannot be estimated reliably;

Instead of relying on an algorithm for constraint-satisfaction learning of causal graphs, we present an alternative score-based approach for the problem. In particular, Silva (2002) described the score-based WASHDOWN algorithm for learning pure measurement models with Gaussian data. The outline of the algorithm is as follows:

1. Start with a one-factor model using all observed variables

That is, create a model with a single latent that is the common parent of all observed variables. This is illustrated at the top of Figure 6.1.

2. Until the model passes a significance test (using the χ^2 test), remove from the model the indicator that will most increase the likelihood of the model

That is, given the latent variable model with k indicators, consider all submodels with $k-1$ indicators that are generated by removing one indicator. Choose the one with the highest likelihood[1] and iterate. This is illustrated in Figure 6.1.

3. If some node was removed in the previous step, add a new latent to the model, make it a children of all other latents, and re-insert all removed nodes as children of the next latent in the sequence. Go back to Step 2.

That is, suppose indicator X_i, that is a child of latent L_j, was removed in the previous step. We now introduce X_i back into the model, but as a child of latent L_{j+1}. If latent L_{j+1} does not exist, create it. There is a natural order for the latents in WASHDOWN, since one latent is created at each time. We move X_i to the next latent according to this order. Latents are fully connected to avoid introducing other constraints besides those that are a result of the given measurement model. Figure 6.3 illustrates a simple case of WASHDOWN, where the algorithm reconstructs a pure submodel of the true model shown in Figure 6.2.

The motivation for this algorithm is as follows: in Step 2, if there is some tetrad constraint that is entailed by the candidate model but that does not hold in the true model, we expect that removing one of the nodes that participate in this invalid constraint will increase the fit of the model. Heuristically, one expects that the node that "most violates" the implied tetrad constraints

[1]This is analogous to the purification step in BUILDPURECLUSTERS as described in Appendix A.3.

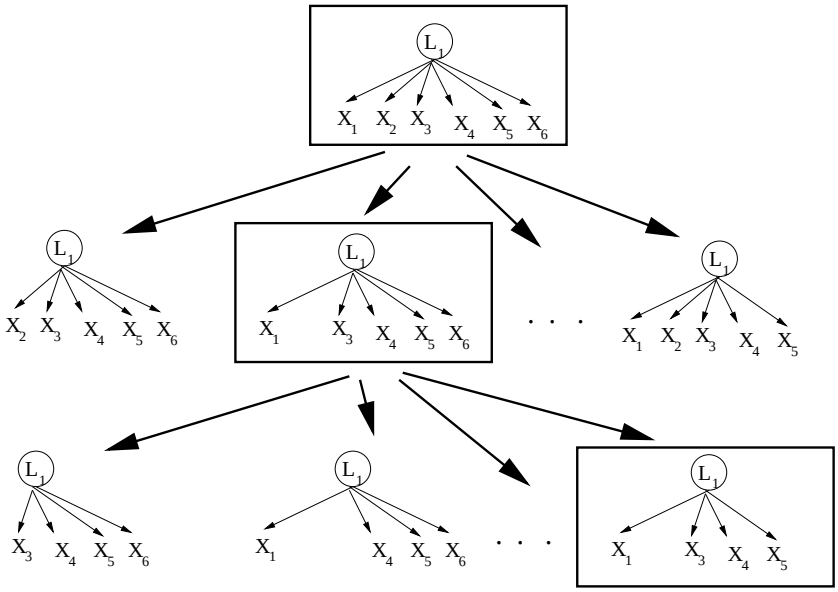


Figure 6.1: WASHDOWN iteratively removes one indicator at a time by choosing the submodel with the highest likelihood. In this example, we start with the model on the top, evaluate 6 possible candidates, and choose to remove X_2. Given this new graph, we evaluate 5 possible candidates, and decide to remove X_6.

according to the data will be the one chosen in Step 2. This is a heuristic and it is not guaranteed to return a pure model even if one exists. See the results in Appendix C.2 for an explanation.

However, if some pure model is returned, and it passes a statistical test, then at least asymptotically one can guarantee that the tetrad constraints in the model should hold in the population. By the theoretical results from Chapter 3, if the returned pure model has three indicators per cluster, the implied constraints are equivalent to a causal model with the corresponding latents and causal directions. The bottom line is that WASHDOWN is not guaranteed to return a structure, but if it returns one, then it should be correct.

In Section 6.2 we introduce our parametric formulation of a mixture of Gaussians. In Section 6.3, we will present a Bayesian version of WASHDOWN for mixtures of Gaussians. Experiments with the Bayesian WASHDOWN are reported in Section 6.4, where we observe that this problem can still be quite difficult to solve. Based on WASHDOWN, we provide a generalization of the algorithm for the problem of density estimation in Section 6.5, with the corresponding experiments in Section 6.7.

6.2 Probabilistic model

We assume the population distribution is a finite mixture of Gaussians. Our generative model will follow closely previous work in mixture of factor analysers (Ghahramani and Beal, 1999).

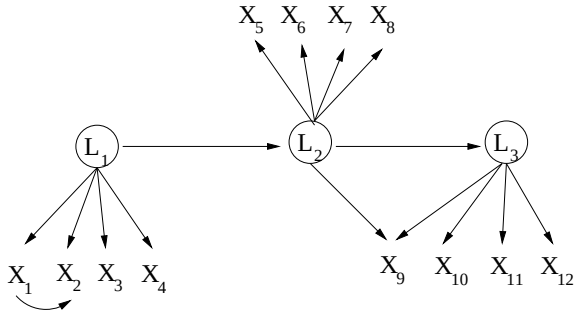


Figure 6.2: Graph that generates the data used in the example of Figure 6.3.

6.2.1 Parametric formulation

Let s be a discrete variable with a finite sample space $1, \ldots, S$. Variable s is modeled as a multinomial with parameter π:

$$s \sim Multinomial(\pi) \tag{6.1}$$

Let $L^{(k)} \in \mathbf{L}$ be a latent variable such that $L^{(k)}$, conditioned on s, is a linear function of its parents with additive noise that follows a Gaussian distribution. That is

$$L^{(k)}|s \sim N(\Sigma_{j \in P_L^{(k)}} \beta_{kjs} L^{(j)}, 1/\zeta_{ks}) \tag{6.2}$$

where $P_L^{(k)}$ is the index set corresponding to the parents of $L^{(k)}$ in G, β_{kjs} corresponds to the coefficient of $L^{(j)}$ in $L^{(k)}$ on component s, and ζ_{ks} is the inverse of the error variance of $L^{(k)}$ given s and its parents.

Let $\mathbf{X}$ be our observed variables, and define $\mathbf{Z} = \mathbf{L} \cup \mathbf{X} \cup \{1\}$. Analogously,

$$X^{(k)}|s \sim N(\Sigma_{j \in P_X^{(k)}} \lambda_{kjs} Z^{(j)}, 1/\psi_k) \tag{6.3}$$

Let the constant 1 be a parent of all $X \in \mathbf{X}$. The role of $\{1\}$ in $\mathbf{Z}$ is to create an intercept term for the linear regression of $X^{(k)}$ on its parents. Notice that the precision parameter ψ_k is not dependent on s.

6.2.2 Priors

A useful metric for ranking graphs is their posterior probability given the data. For this purpose, we should first specify priors over the graphs and parameters.

Our prior for π is the following Dirichlet:

$$\pi \sim Dirichlet(a^* \mathbf{m}^*) \tag{6.4}$$

where $\mathbf{m}^* = [1/S, \ldots, 1/S]$ is a fixed vector and a^* is a hyperparameter. The hyperparameter is a measure of deviation from an uniform mixture proportion.

We adopt a simple prior for the parameters $\mathbf{B} = \{\beta_{jps}\}$. As in empirical Bayesian analysis, we plan to optimize the hyperparameters of our model. It is therefore important to minimize the computational time of this optimization. Thus, we adopt the same prior for all parameters $\theta \in \mathbf{B}$:

$$\theta \sim N(0, 1/\upsilon_{\mathbf{L}}^*) \tag{6.5}$$

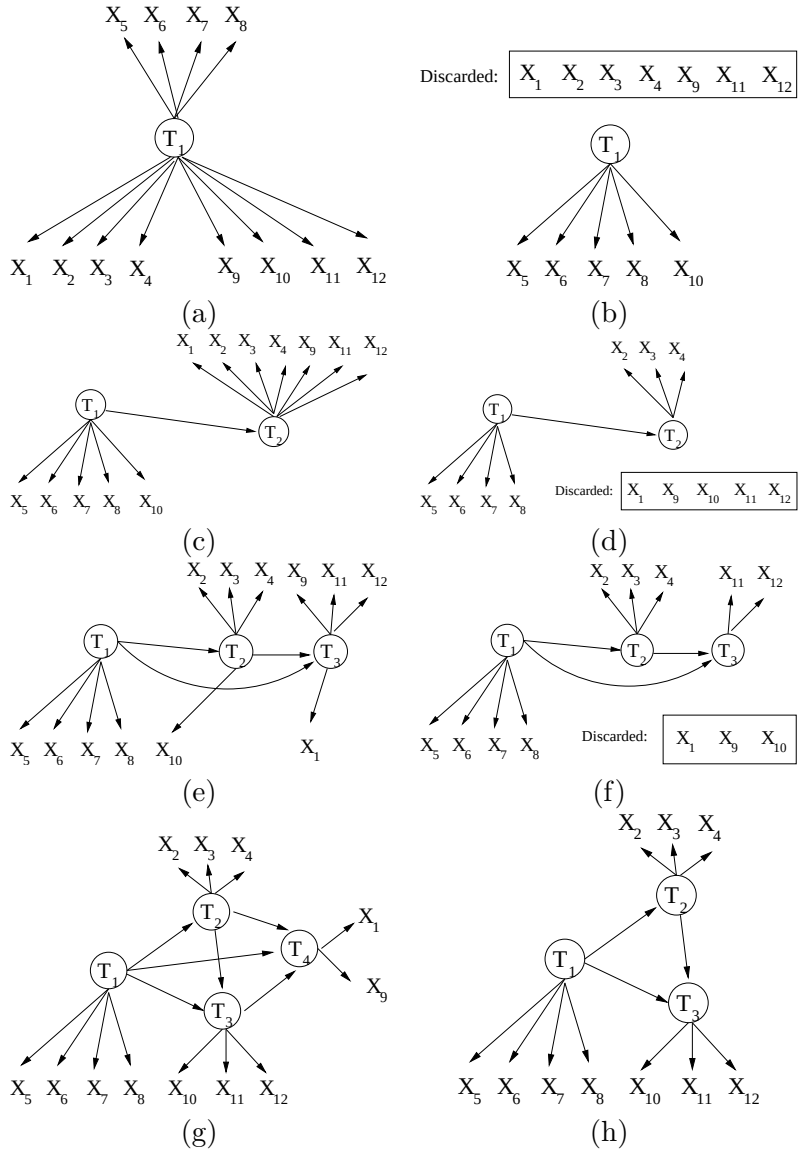


Figure 6.3: A run of WASHDOWN for data generated from the model in Figure 6.2. We start with the one-factor model in (a), and by using the process of node elimination, we generate the graph in (b), where nodes $\{X_1, X_2, X_3, X_4, X_9, X_{11}, X_{12}\}$ are eliminated. We wash down such discarded nodes to a new cluster, corresponding to latent L_2 (c). Another round of node elimination generates the graph in (d) with the respective discarded nodes. Such nodes are washed down to the next latents (X_{10} moves to L_2, the others move to L_3) as depicted in (e). Nodes are eliminated again generate graph (f). The eliminated nodes are clustered under latent L_4, as in Figure (g). Because this latent has too few indicators, we eliminate it, arriving at the final graph in (h). Notice that the label of the latents is arbitrary and corresponds only to the order of creation.

That is, we have a single hyperparameter $\upsilon^*_{\mathbf{L}}$, which can be optimized by a closed formula given all other parameters.

We will not define a prior over the error precisions for $\mathbf{L}$ and $\mathbf{X}$, the set $\{\zeta, \psi\}$. The number of error precisions for $\mathbf{X}$ does not increase with model complexity. Therefore, no penalization for complexity is needed for these parameters. Concerning the precisions for $\mathbf{L}$, they do not introduce extra degrees of freedom since the scale of the latent variables can be adjusted according to an arbitrary number. Therefore, we will also treat them as hyperparameters to be fitted.

Concerning elements in $\mathbf{\Lambda} = \{\lambda_{kjs}\}$, we also adopt a single prior for the parameters $\theta \in \mathbf{\Lambda}$. For each observed variable $X^{(k)}$, and each $\Theta \in \mathbf{\Lambda}^k$:

$$\Theta \sim N(0, 1/\upsilon^*_{X^{(k)}}), \tag{6.6}$$

if Θ is does not correspond to an intercept term, and

$$\Theta \sim N(0, 1/\upsilon^t_{X^{(k)}}), \tag{6.7}$$

if Θ *does* correspond to an intercept term. That is, the number of hyperparameters $\{\{\upsilon^*_{X^{(k)}}\}, \upsilon^t_{X^{(k)}}\}$ neither increases with the number of mixture components nor with the number of parents of variable $X^{(k)}$.

This model can be interpreted as a mixture of causal models of different subpopulations, where each subpopulation has the same causal structure, but different causal effects. The measurement error, represented by Ψ, the matrix of precision parameters for the observed variables, is the same across subpopulations.

Another motivation for making Ψ independent of s is computational: first, estimation can get much more unstable if Ψ is allowed to vary with s. Second, a prior for Ψ is not strictly necessary, and therefore we will not need to fit the corresponding hyperparameters. Usual prior distributions for precision parameters, such as gamma distributions, have hyperparameters that cannot be fit by a closed formula (see, e.g, Beal and Ghahramani, 2003). This could slow down the procedure considerably.

The natural question to make is what happens to the entailment of tetrad constraints in finite mixtures of linear models. Again, a constraint is entailed if and only if it holds for all parameter values of the mixture model. We can appeal to a measure theoretical argument, not unlike the one used in Chapter 4, to argue that observed tetrad constraints that are not entailed by the graphical structure require coincidental cancellation of parameters, and therefore are ruled out as unlikely. This argument is less convincing when the number of mixtures approaches infinite. Nevertheless, we will be implicitly assuming that the number of mixture components is not high. That is, high to the point where constraints are judged to hold in the population by finite sample scoring, and yet they are not graphically entailed.

6.3 A Bayesian algorithm for learning latent causal models

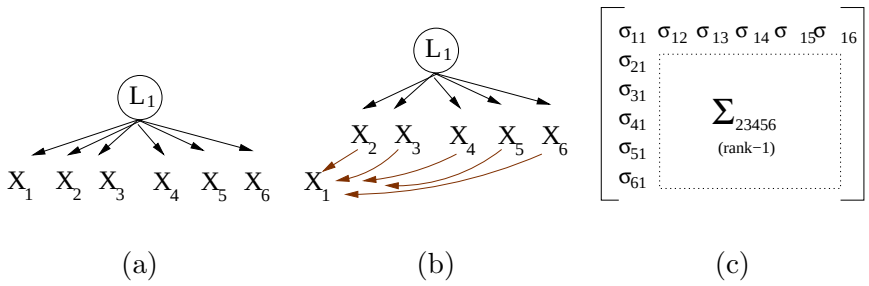
.

The original WASHDOWN of Silva (2002) was based on a χ^2 test. We introduce a variation of this algorithm using a Bayesian score function. Based on the success of Bayesian score functions in other structure learning algorithms (Cooper, 1999), we conjecture that in general it should be a better alternative than χ^2 tests for small sample sizes. Moreover, the χ^2 stopping criterion of

Algorithm WASHDOWN
Input: a data set **D** of observed variables **O**
Output: a DAG

1. Let G be an empty graph
2. $G_0 \leftarrow G$
3. Do
4. $G \leftarrow$INTRODUCELATENTCLUSTER$(G, G_0, \mathbf{O})$
5. Do
6. Let $O \leftarrow argmax_{O \in G} \mathcal{F}(G_{\setminus O}, \mathbf{D})$
7. If $\mathcal{F}(G_{\setminus O}, \mathbf{D}) > \mathcal{F}(G, \mathbf{D})$
8. Remove O from G
9. While G is modified
10. If GRAPHIMPROVED(G, G_0)
11. $G_0 \leftarrow G$
12. While G_0 is modified
13. Return G_0

Table 6.1: Build a latent variable model where observed variables either share the same parents or no parents.



(a) (b) (c)

Figure 6.4: Deciding if X_1 should be excluded of the one-factor model in (a) is done by comparing models (a) and (b). Equivalently, removing X_1 generates a model where the entries corresponding to the covariance of X_1 and X_i (σ_{1i}) are not constrained, while the remaining covariance matrix Σ_{23456} is a rank-1 model, as illustrated by (c).

the original WASHDOWN function depended on a pre-specified significance value that can be quite arbitrary, while our suggested score function does not have any special parameters to be set a priori.

Let $\mathcal{F}(G, \mathbf{D})$ be a function that scores graph G using dataset **D**. Our goal with WASHDOWN will be finding local maxima for $\mathcal{F}$ in the space of pure measurement models. Section 6.3.1 describes the algorithm. Several implementation details are left to Appendix C.3. A proposed score function $\mathcal{F}$ is described only in Section 6.3.2.

6.3.1 Algorithm

The modified WASHDOWN algorithm is shown in Table 6.1. We will explain it step by step.

In Table 6.1, graph G is our candidate graph, the one that will have indicators removed and latents added to. Graph G_0 represents the candidate graph in the previous iteration of the algorithm. Moving to the next iteration in WASHDOWN only happens when graph G is better to G_0

Algorithm IntroduceLatentCluster
Input: two graphs G, G_0; a set of observed variables $\mathbf{O}$;
Output: a DAG

1. Let **NodeDump** be the set of observed nodes in $\mathbf{O}$ that are not in G
2. Let T be the number of latents in G
3. Add a latent L_T to G and form a complete DAG among latents in G.
4. For all $V \in$ **NodeDump**
5. If $V \in G_0$
6. Let L_i be the parent of V in G_0
7. Set L_{i+1} to be the parent of V in G
8. Else
9. Set L_T to be the parent of V in G
10. If L_T does not have any children
11. Remove L_T from G
12. Return G

Table 6.2: Introduce a new latent by moving nodes down the latent layer.

according to the function GraphImproved, shown in Table 6.3 and explained in detail later in this section.

G starts without any nodes. Function IntroduceLatentCluster, described in Table 6.2, adds a new latent node to G (connecting it to all other latents) and moves around observed variables that are not in G. As in the original Washdown, illustrated in Figure 6.3, latents in G are numbered (L_1, L_2, L_3, etc.). Any node removed from G that was originally child of latent L_i will be assigned to be an indicator of latent L_{i+1}. It is this flow of indicators, "downstream" the latent layer, that justifies the name "washdown."

After the addition of a new latent, we proceed to the cycle of indicator removal. This is represented by steps 5-9 in Table 6.1. The way this removal is implemented is one of the main differences between the original algorithm of Silva (2002) and the new Washdown. Let $G_{\backslash O}$ be a modification of graph G generated by removing all edges into O, and adding an edge from every observed node in G into O. By definition, $G_{\backslash \emptyset} = G$. We will select the observable node O in G that maximizes $\mathcal{F}(G_{\backslash O}, \mathbf{D})$.

The intuition for this comparison is as follows. For example, consider a latent variable model with a single latent, where this latent is the common parent of all observed variables and no other edges exist. Figure 6.4(a) illustrates this type of model. To simplify the exposition, we will consider a model with only one Gaussian component. The covariance matrix Σ of $X_1 - X_6$ can be represented as

$$\Sigma = \Lambda\Lambda' + \Psi \tag{6.8}$$

where Λ is the vector corresponding to the edge coefficients relating the latent to each of the observed variables and Ψ is the respective matrix of residuals. That is, this one-factor model imposes a rank constraint in the first term of this sum: $\Lambda\Lambda'$ is by construction a matrix of rank 1 (i.e., tetrad constraints hold for each foursome of observed variables). However, it may be the case that in the true model not all variables are independent conditioned on L_1. Consider dropping X_1 out of this one-factor model. That is:

1. variables $\{X_2, \ldots, X_6\}$ still form a one-factor model, and therefore their marginal covariance matrix (minus the residuals) is still constrained to have rank 1

Algorithm GRAPHIMPROVED
Input: two graphs G_1, G_2; a dataset $\mathbf{D}$
Output: a boolean

1. Let $\mathbf{O}_i$ be the set of observed variables in graph G_i
2. $\mathbf{O}_{All} \leftarrow \mathbf{O}_1 \cup \mathbf{O}_2$
3. For $i = 1, 2$
4. Initialize $G'_i \leftarrow G_i$
5. Let $\mathbf{O}^C \leftarrow \mathbf{O}_{All} \backslash \mathbf{O}_i$ and them to G_i
6. Add edges $V \rightarrow O$ to G_i for all $(V, O) \in \mathbf{O}_i \times \mathbf{O}^C$
7. Form a full DAG among elements $\mathbf{O}^C$ in G'_i
8. If $\mathcal{F}(G'_1, \mathbf{D}) > \mathcal{F}(G'_2, \mathbf{D})$
9. Return true
10. Else
11. Return false

Table 6.3: Compare two graphs that initially might have different sets of observed variables.

2. the conditional distribution $p(X_1|\{X_2, \ldots, X_6\})$ is unconstrained. This can be done by as shown in Figure 6.4(b)

That is, we modify the implied joint distribution $p_0(X_1, \ldots, X_6)$ into a new joint $p_1(X_1|\{X_2, \ldots, X_6\}) \times p_1(X_2, \ldots, X_6)$ where $p_1(X_1|\ \{X_2, \ldots, X_6\})$ is saturated (no further constraints imposed). This operation will remove any rank constraints that include X_1. This idea is largely inspired by the search procedure described by Kano and Harada (2000). The algorithm of Kano and Harada (2000) adds and removes nodes in a factor analysis graph by doing an analogous comparison of nested models. That approach, however, was intended to modify a factor analysis graph given a priori, i.e., it was a *purification* procedure for a pre-defined clustering. We use it as a step to build clusters from data.

Empirically, this procedure for selecting which indicator to remove worked better in preliminary experiments than simply choosing among models that differ from G by having one less indicator, as used in (Silva, 2002). This is intuitive, because it measures not only how well the remaining indicators fit the data, but also how much is gained in representing the covariance between the removed indicator and the other variables without imposing constraints.

At Step 10 of Table 6.1, we have to decide if we proceed to the next iteration or if we halt. In the original WASHDOWN formulation, we would always start the next iteration and not proceed if the new model passed a statistical test at a given significance level[2]. This has two major drawbacks: it requires a choice of significance level, which is many times arbitrary; it requires the test to have significant power. For more complex distributions as mixtures of Gaussians, having a test of acceptable power might be difficult.

Instead, we use the criterion defined by function GRAPHIMPROVED (Table 6.3). Both the current candidate graph, G, and the previous graph, G_0, embody a set of tetrad constraints. The score function is expected to reflect how well such constraints are supported by the data: in this case, the better the score, the better supported are the tetrad constraints. However, due to variable elimination, G and G_0 might differ with respect to their set of observed variables. Comparing them directly is meaningless: for instance, if G equals G_0 with some indicators removed, then the likelihood of G will be higher than G_0.

[2]In the original WASHDOWN, clusters with 1 or 2 indicators would just be removed in the end.

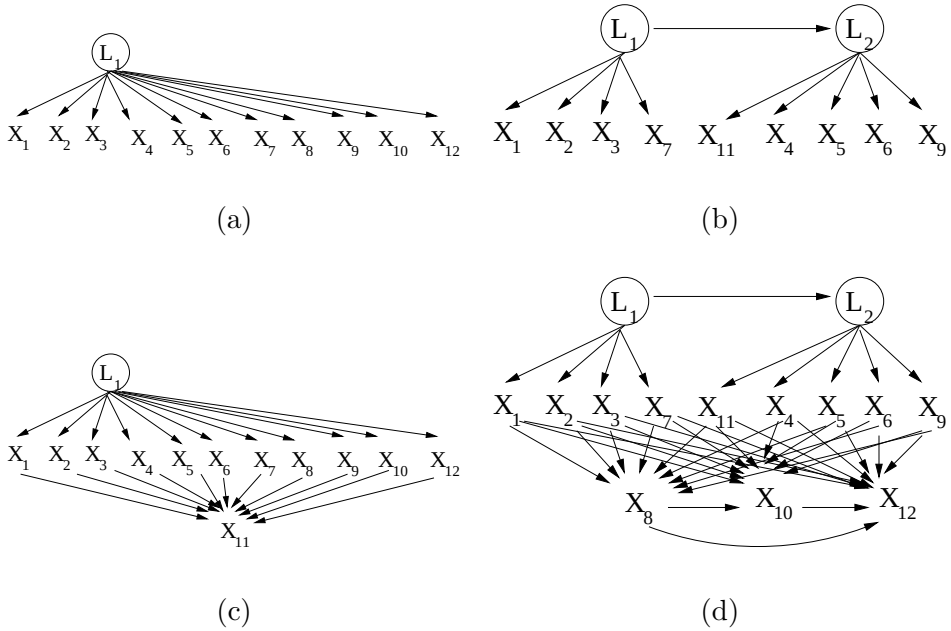


Figure 6.5: Graphs in (a) and (b) are transformed into graphs (c) and (d) before comparison in method GraphComparison.

Instead, we "normalize" G and G_0 in GraphImproved before making the comparison. Nodes in G that are not in G_0 are added to G_0. Nodes in G_0 not in G are added to G. Such nodes are connected to the pre-existing nodes by adding all possible edges from the original nodes into the new nodes. The goal is to include the new nodes without imposing any constraints on how they are mutually connected and connected with respect to the existing nodes.

For example, consider Figure 6.5. Graph G_0 has a single cluster (Figure 6.5(a)). Graph G has two clusters (Figure 6.5(b)). Graph G_0 has nodes X_8, X_{10} and X_{12} that are not present in G_0. Node X_{11} is in G but not in G_0. Therefore, we normalize both graphs with respect to each other, obtaining G_0' in Figure 6.5(c) and G' in Figure 6.5(d). If G' scores better than G_0', we accept G as our new graph and proceed to the next iteration.

Figure 6.3, used to illustrate the algorithm described by Silva (2002), also illustrates the new Washdown algorithm. Most modifications are in the internal evaluations, but the overall structure of the algorithm remains the same. The only difference, in this example, is that we choose the model in Figure 6.3(g) instead of Figure 6.3(h) not because we eliminate clusters with less than 2 indicators, but because the score of the former is higher than the score of the latter.

6.3.2 A variational score function

We adopt the posterior distribution of a graph as its score function. Our prior over graph structures will be uniform, which implies that the score function of a graph G amounts to the marginal likelihood $p(\mathbf{D}|G)$, $\mathbf{D}$ being the data set. Since calculating this posterior is intractable for any practical search algorithm, we adopt a variational approximation for it that is similar to the Bayesian variational mixture of factor analysers (Ghahramani and Beal, 1999).

In fact, Beal and Ghahramani (2003) show by an heuristic argument that asymptotically this variational approximation is "equivalent" to the BIC score. However, for finite samples we have the power of fitting the hyperparameters and choosing a more suitable penalization function than the one given by BIC. In experiments on model selection described by Beal and Ghahramani (2003), this variational framework was able to give better results than BIC at roughly the same computational cost.

Let the posterior probability of the parameters and hidden variables be approximated as follows:

$$p(\pi, \mathbf{B}, \mathbf{\Lambda}, \{s_i, \mathbf{L}_i\}_{i=1}^n | \mathbf{X}) \approx q(\pi) q(\mathbf{B}) q(\mathbf{\Lambda}) \prod_{i=1}^n q(s_i, \mathbf{L}_i)$$

where $p(.)$ is the density function, $q(.)$ are the variational approximations, n is the sample size. The main approximation assumption is the conditional decoupling of parameters and latent variables.

Given the logarithm of marginal distribution of the data

$$\begin{aligned} \mathcal{L} \equiv \ln p(\mathbf{X}) = \ln \; & (\int d\pi \; p(\pi | a^* \mathbf{m}^*) \int d\mathbf{B} \; p(\mathbf{B} | \upsilon_{\mathbf{L}}^*) \int d\mathbf{\Lambda} \; p(\mathbf{\Lambda} | \upsilon_{\mathbf{X}}^*) \times \\ & \prod_{i=1}^N [\sum_{s_i=1}^S p(s_i | \pi) \int d\mathbf{L}_i \; p(\mathbf{L}_i | s_i, \mathbf{B}, \zeta) p(\mathbf{X}_i | \mathbf{Z}_i, s_i, \mathbf{\Lambda}, \mathbf{\Psi})]) \end{aligned}$$

we introduce our variational approximation by using Jensen's inequality

$$\begin{aligned} \mathcal{L} \geq \; & \int d\pi \; d\mathbf{B} \; d\mathbf{\Lambda} (\ln \tfrac{p(\pi | a^* \mathbf{m}^*) p(\mathbf{B} | \upsilon_{\mathbf{L}}^*) p(\mathbf{\Lambda} | \upsilon_{\mathbf{X}}^*)}{q(\pi) q(\Lambda L) q(\mathbf{\Lambda})} \\ & + \sum_{i=1}^n [\sum_{s_i=1}^S \int d\mathbf{L}_i \; q(s_i, \mathbf{L}_i) \; (\ln \tfrac{p(s_i | \pi) p(\mathbf{L}_i | s_i, \mathbf{B}, \zeta)}{q(s_i, \mathbf{L}_i)} + \ln \; p(\mathbf{X}_i | \mathbf{Z}_i, s_i, \mathbf{\Lambda}, \mathbf{\Psi})])) \end{aligned}$$

Therefore, our score function is

$$\begin{aligned} \mathcal{F}(G, \mathbf{D}) = \; & \int d\pi \; \ln \tfrac{p(\pi | a^* \mathbf{m}^*)}{q(\pi)} + \sum_{s=1}^S [\int d\mathbf{B}^{\mathbf{s}} q(\mathbf{B}^{\mathbf{s}}) \ln \tfrac{p(\mathbf{B}^{\mathbf{s}})}{q(\mathbf{B}^{\mathbf{s}})} + \int d\mathbf{\Lambda}^{\mathbf{s}} q(\mathbf{\Lambda}^{\mathbf{s}}) \ln \tfrac{p(\mathbf{\Lambda}^{\mathbf{s}})}{q(\mathbf{\Lambda}^{\mathbf{s}})}] \\ & \sum_{i=1}^n \sum_{s_i=1}^S q(s_i) [\int d\pi q(\pi) \ln \tfrac{p(s_i | \pi)}{q(\mathbf{s_i})} \\ & + \int d\mathbf{B}^{\mathbf{s}} d\mathbf{L}_i q(\mathbf{B}^{\mathbf{s}}) q(\mathbf{L}_i | s_i) \ln \tfrac{p(\mathbf{L}_i | s_i, \mathbf{B}^{\mathbf{s}}, \zeta^{\mathbf{s}})}{q(\mathbf{L}_i | s_i)} \\ & + \int d\mathbf{\Lambda}^{\mathbf{s}} d\mathbf{L}_i q(\mathbf{\Lambda}^{\mathbf{s}}) q(\mathbf{L}_i | s_i) \ln p(\mathbf{X}_i | s_i, \mathbf{Z}_i, \mathbf{\Lambda}^{\mathbf{s}}, \mathbf{\Psi})] \end{aligned}$$

In this function, the first three lines correspond to the negative KL-divergence between the priors and the approximate posteriors. The fourth line is the expected log-likelihood of the data by the approximate posteriors. Although this variational score is not guaranteed to consistently rank models, it is a natural extension of the BIC score (also inconsistent for latent variable models), where penalization term increases not with the number of parameters, but by how much they deviate from a given prior.

Optimizing our variational bound is a non-convex optimization problem. To fit our variational parameters, we alternate between optimization steps where we find the value of one parameter, or hyperparameter, while fixing all the others. The steps are given in Appendix C, Section C.1.

6.3.3 Choosing the number of mixture components

So far we mentioned how to search for a graph for a given probabilistic model, but we did not mention how to choose the number of Gaussian components to begin with. A principled alternative for choosing the number of mixture components would be running the WASHDOWN algorithm for varying numbers and choosing the one with the best score. Since this is computationally expensive, we instead heuristically choose the number of components according to the output of the variational mixture of factor analyzers Ghahramani and Beal (1999) and fix it as an input for WASHDOWN.

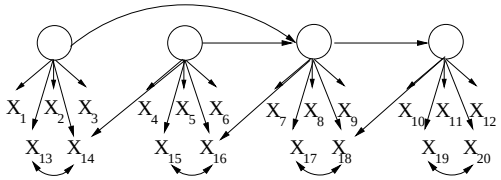


Figure 6.6: The network used in the experiments throughout Section 6.4.

Evaluation of output measurement models					
Trial	**Latent omission**	**Latent commission**	**Misclustered indicator**	**Impurities**	
1	1	0	0	0	0
2	0	0	2	1	0
3	0	0	2	0	0
4	3	0	12	0	0
5	2	0	9	1	0
6	2	0	8	0	4
7	3	0	12	0	0
8	3	0	13	0	2
9	0	0	2	0	0
10	0	2	0	4	0
average	1.4	0.2	6	0.6	0.6

Table 6.4: Results on structure learning for WASHDOWN for samples of size 200.

6.4 Experiments on causal discovery

.

In this section we perform simulated experiments using the same type of graphical models described in the experimental section of Chapter 3. The goal is to analyse how well WASHDOWN is able to reconstruct the true graph. We will use the graphical structure in Figure 6.6 to generate synthetic datasets. In all of our experiments, we generated data from a mixture of 3 Gaussians. Within each Gaussian, we sampled the parameters following the same procedure of Chapter 3. The probability of each Gaussian component was chosen by selecting uniformly an integer in $\{1, 2, 3, 4, 5\}$ and normalizing. Distributions where one of the components had probability less than 0.15 were discarded.

The criteria of success are the same of Chapter 3, using counts instead of percentuals. Table 6.4 shows the results for 10 independent trials using sample size of 200. Table 6.5 shows the results for 10 independent trials using sample size of 1000.

The results, especially for sample size of 200 (on which variance is high), might not be as good as in the Gaussian case presented in Chapter 3. However, these problems are much harder, and BUILDPURECLUSTERS, for instance, does not provide reasonable outputs (it returns a mostly empty graph). WASHDOWN, while much more computationally expensive, is still able to return mostly correct outputs for the given problem at reasonable sample sizes.

Evaluation of output measurement models					
Trial	**Lat. omission**	**Lat. commission**	**Ind. omission**	**Impurities**	**Misclustered ind.**
1	0	0	1	0	0
2	0	0	3	1	2
3	0	0	0	0	0
4	0	0	0	0	1
5	0	0	3	2	1
6	0	0	0	0	0
7	1	0	0	0	0
8	0	0	1	0	0
9	0	1	0	2	1
10	1	0	0	0	0
average	0.2	0.1	0.8	0.5	0.5

Table 6.5: Results on structure learning for WASHDOWN for samples of size 1000.

6.5 Generalized rank constraints and the problem of density estimation

.

As discussed in the first chapter of this thesis, latent variable models are also important tools in density estimation. For instance, Bishop (1998) discusses variations of factor analysis and mixtures of factor analysers (probabilistic principal component analysis, to be more specific) for the problem of density estimation. One of the applications discussed by Bishop was in digit recognition from images, which can be used for automated ZIP code identification from scanned envelopes. Instead of building a discriminative model that would classify each image according to the set $\{0, 1, \ldots, 9\}$, his proposed model calculates the posterior probability of each digit using 10 different density models, one for each digit. In this way, it is possible to raise a flag when none of the density models recognizes a digit with high probability, so that human classification is required and a better trade-off between automation and cost of human intervention can be achieved. Outlier detection, as in the digit recognition example, is a common application of density models. Latent variable models, usually variations of factor analysis, are among the most common tools for this task. Bishop (1998) and Carreira-Perpinan (2001) describe several other applications.

Most approaches based on factor analysis are also computationally appealing. The problem of structure learning is in many cases reduced to the problem of choosing the number of latents. Maximum likelihood estimation of a few latent variable models is computationally feasible even for very large problems. If one wants a Bayesian criterion for model selection, in practice one could use the BIC score which only requires a maximum likelihood estimator. Other approximate Bayesian scores are computationally feasible, such as the variational score discussed here. Minka (2000) provides other examples of approximation methods to compute the posterior of a factor analysis model.

The common factor analysis model consists on a fully connected measurement model with disconnected latents, i.e., a model where every indicator is a child of every latent, and where there are no edges connecting latents. However, this space of factor analysis graphs might not be the best choice for a probabilistic model. For instance, assume the true linear model that generated our data is shown in Figure 6.7(a). In the usual space of factor analysis graphs, we would need the

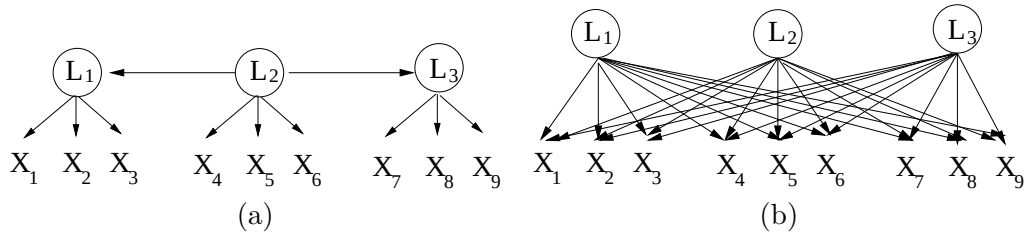


Figure 6.7: If fully connected measurement models with disconnected latents are used to represent the joint density function of model (a), the result is the model shown in (b).

graph represented in Figure 6.7(b). The relatively high number of parameters in this case might lead to inefficient statistical estimation, requiring more data than we would need if we used the right structure.

Alternatively, a standard hill-climbing method with hidden variables, such as FindHidden (Elidan et al., 2000, discussed in Chapter 2), might find a better structure, as measured by how well the model fits the data. However, this type of approach has two disadvantages:

- it is computationally expensive, since at every step we have to decide which edge to remove/add/reverse. Each search operator scales quadratically with the number of variables;
- its search space is naive. Single edge modifications might be appropriate for some spaces of latent variable models (e.g., if the true model and all models in the search space have likelihood functions with a single global optimum). In general, however, if the model has many unidentifiable parameters, or if sample sizes are relatively small, models that differ by a single edge might be undistinguishable or nearly undistinguishable and easily misguide the algorithm. Instead of operating on single edges, a search algorithm should operate on graphical modifications that entail different relevant constraints in the observed marginal;

If the given observed variables have many hidden common causes, as in the problems that motivate this thesis, it might be more appropriate to discard the general FindHidden approach and adopt an algorithm that operates directly in the space of factor analysis graphs. Such an algorithm should have the following desirable features:

- unlike standard model selection in factor analysis, its search space should include a large class of latent variable graphs, not only fully connected measurement models;
- unlike FindHidden, its search space should have operators that scale at least linearly with the number of observed variables;
- unlike FindHidden, any two neighbors in the search space should imply different constraints in the observed marginal, and such different constraints should be relatively easy to distinguish given reasonable sample sizes;

Washdown satisfies these criteria. Its search space of pure measurement models with connected latents can represent several distributions using sparse graphs, where fully connected measurement models with disconnected latents would require many edges, as in the example of Figure 6.7. Each

search operator scales linearly with the number of observed variables[3]. Pure measurement models that differ according to tetrad constraints can be expected to be easier to distinguish with small samples than dense graphs that differ on a single edge.

However, WASHDOWN has one essential limitation that precludes it of being directly applicable to density estimation problems: it might discard an unpredictable number of observed variables. For instance, in Chapter 4 we analysed the behavior of BUILDPURECLUSTERS in some real-world data. Approximately two thirds of the observed variables where eliminated.

There are two main ways of modifying WASHDOWN. One is to somehow force all variables to be indicators in a pure measurement model, as in the approach described by Zhang (2004) for learning latent trees of discrete variables. The drawback is that such an approach can sometimes (or perhaps even frequently) underfit the data, as Zhang acknowledges.

Another way is to adopt a hybrid approach, as hinted in the end of Chapter 4. For instance, by using an algorithm where different pure measurement models are learnt for different subsets of variables, and combined in the end by introducing the required impurities. Consider, for example, the true model shown in Figure 6.8(a). An algorithm that learns pure measurement models only cannot generate the cluster represented by latent L_3 if it includes L_1 and L_2, and vice-versa.

Now imagine we run WASHDOWN and obtain a model that includes latents L_1 and L_2, as well all of their respective indicators, as in Figure 6.8(b). We can run WASHDOWN again with the discarded indicators $X_9, X_{10}, X_{11}, X_{12}$ and obtain an independent one-factor model, as in Figure 6.8(c). We could then merge these two marginally pure models into a single latent variable model, as in Figure 6.8(d). Starting from this model, we could apply a standard greedy search approach to introduce bi-directed edges among indicators if necessary.

This is the main idea behind the generalized WASHDOWN approach we introduce in the next section. However, there are a few other issues to be solved if we want a model that will include all observed variables.

First, building pure models where observed variables have a single parent might not be enough. If, for instance, the true model is the one in Figure 6.7(b), this generalized WASHDOWN algorithm will not work: it should still return an empty graph.

The proposed solution is to embed WASHDOWN in a even more general framework: we first try to build several disjoint pure models with one latent per cluster, until an empty graph is returned. When this happens, and we still have unclustered indicators, we attempt to build several disjoint pure models with *two* latents per cluster, and so on. This approach explores *general* rank constraints. That is, a cluster with k latents imposes a rank constraint in the covariance matrix of its p indicators, namely, that it can be decomposed into two matrices as follows:

$$\Sigma = \Lambda \mathbf{\Phi} \Lambda' + \Psi$$

where $\mathbf{\Phi}$ is the covariance matrix of the latents; Λ is the set of edge coefficients connecting indicators to their latent parents (a $p \times k$ matrix); and Ψ is diagonal matrix representing the covariance matrix of the residuals. That is, Σ is constrained to be the sum of a matrix of rank k ($\Lambda \mathbf{\Phi} \Lambda'$) and a diagonal matrix (Ψ). Figure 6.9 presents a type of output that could be generated using this algorithm.

The problem is not completely solved yet. It would not make sense, for instance, to have clusters with a single indicator, as in Figure 6.10: instead of having each latent connected to the

[3]This is not to say that WASHDOWN is more computationally efficient than FINDHIDDEN in general. If the graph found in the first stage of FINDHIDDEN, which does not require latent variables, is quite sparse, then FINDHIDDEN is likely to be faster than WASHDOWN. However, for problems where the initial graph is found to be somewhat dense, FINDHIDDEN can be slow compared to an approach such as WASHDOWN.

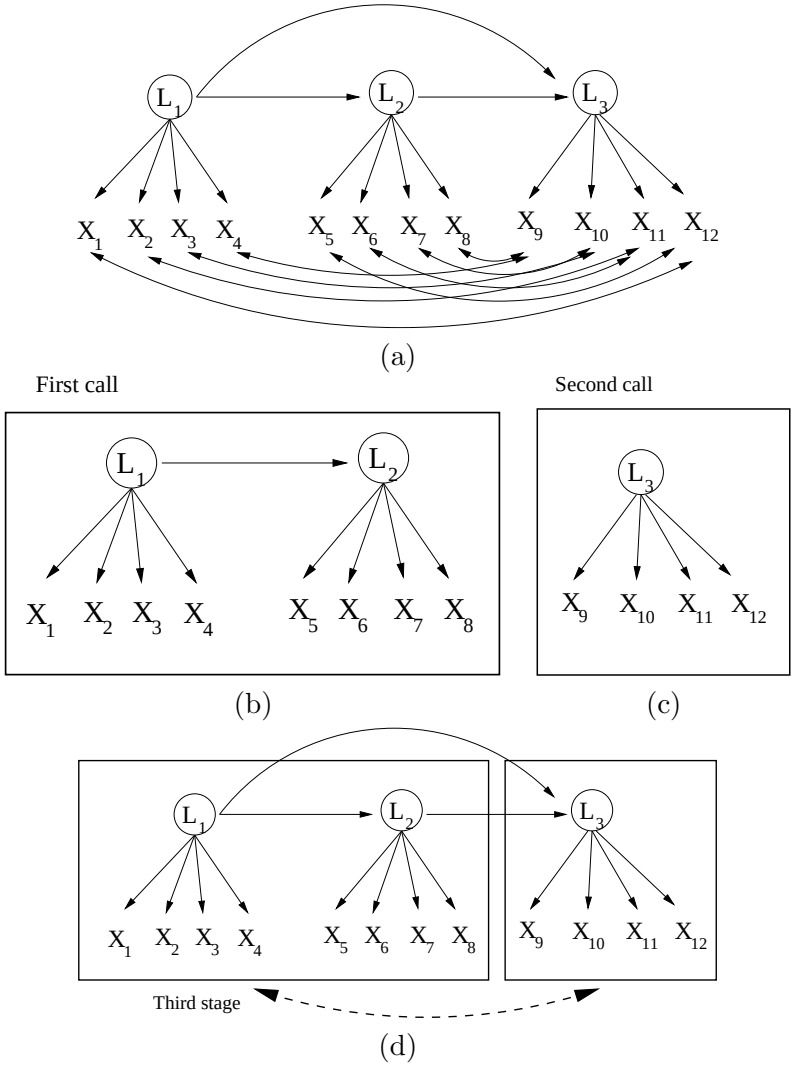


Figure 6.8: Given data sampled from the model in (a), one variation of WASHDOWN could be used to first generate the model in (b) and then, independently, the model in (c). Both models would then be merged, and bi-directed edges could be later added by some greedy search (d).

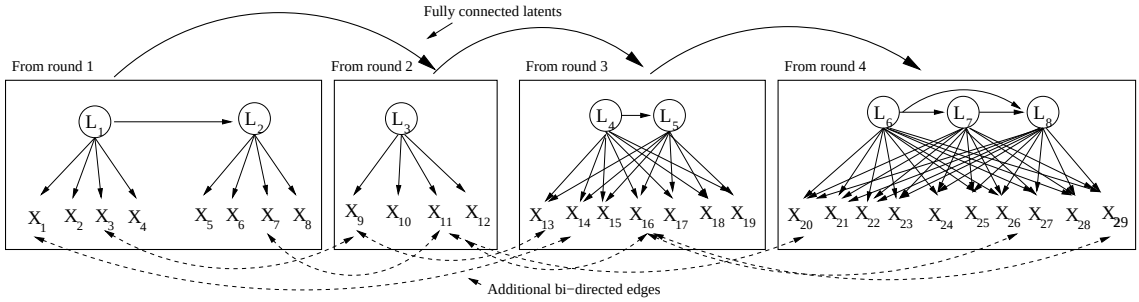


Figure 6.9: A possible outcome of a generalized WASHDOWN that allows multiple latent parents per indicator and impurities. In this case, four calls to a clustering procedure were made. In the first two calls, models with one latent per cluster were build. In the third call, two latents per cluster. In the fourth call, three latents. We merge all models and fully connect all latents, which is represented by bold edges between subgraphs in the figure above. Bi-directed edges are then added by an additional greedy search for impurities.

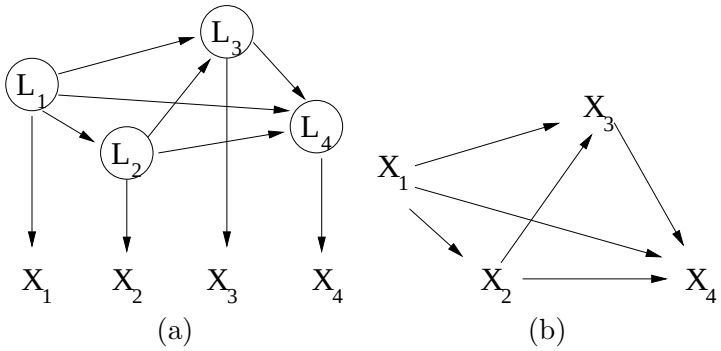


Figure 6.10: For density estimation it makes little sense to have a model of fully connected latents with one indicator per latent, as in (a), even if this is the true causal model. The same distribution could be represented without latent variables, as in (b), with less parameters.

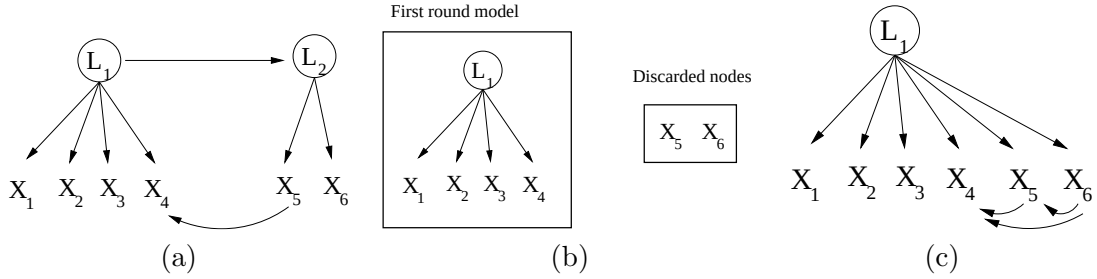


Figure 6.11: Suppose the true model is the one in (a). A call to WASHDOWN can return the model in (b), where indicators X_5 and X_6 are discarded. These indicators cannot be used to form any one-factor model (nor any other factor model) since a one-factor model does not impose any constraint on their joint distribution. Our solution is just to add X_5 and X_6 to the latent variable model and proceed with a standard greedy search method. A possible outcome is the one shown in (c).

other latents as in this example, we could just connect all the indicators directly and eliminate the latents. Although this is an extreme example, it illustrates that "small" clusters are undesirable. In general, a k-factor model (i.e., a factor model with k latents) is only statistically meaningful if there is a minimal number of indicators in this model. For instance, a one-factor model needs at least four indicators, since any covariance matrix of three or less variables can be represented by a one-factor model. We do not want small clusters.

If, after attempting to create pure models with $1, 2, \ldots, k$ latents per cluster, we end up with a set of unclustered indicators that is not large enough for a k-factor model, we will not attempt to create a new cluster. Instead, we will just add these remaining nodes to the model and do a standard greedy search to connect them to the clustered ones. Figure 6.11 illustrates a case where we have only two remaining indicators, and a possible resulting model where these two indicators are included in the latent variable model.

To summarize, we propose the following extension of WASHDOWN for density estimation problems. This proposal is motivated by the necessity of including all observed variables and by the computational convenience of clustering nodes instead of finding arbitrary factor analysis graphs:

- attemp to create pure models with one latent per cluster (as in WASHDOWN). After finding one such model, try to create a new one with the indicators that were discarded. Each new model is generated independently of the previous models. Iterate until no such a pure model is returned.
- attemp to create now pure models with *two* latents per cluster. Iterate.
- after no new k-factor model can be constructed with the remaining indicators, merge all pure models create so far into a single global model. Add bi-directed edges among indicators using a greedy search algorithm, if necessary. We explain how to parameterize such edges in Appendix C.
- add all the unclustered indicators to this global model, and connect them to the other nodes by using a standard greedy search.

Inspired by the strategy used for learning causal models with WASHDOWN, we expect this algorithm to find first sets of indicators whose marginal is a sparse measurement model of a few

Algorithm K-LatentClustering
Input: a data set $\mathbf{D}$ of observed variables $\mathbf{O}$, an integer k
Output: a DAG

1. Let G be an empty graph
2. $G_0 \leftarrow G$
3. Do
4. $G \leftarrow$IntroduceKLatentCluster$(G, G_0, \mathbf{O}, k)$
5. Do
6. Let $O \leftarrow argmax_{O \in G} \mathcal{F}(G_{\setminus O}, \mathbf{D})$
7. If $\mathcal{F}(G_{\setminus O}, \mathbf{D}) > \mathcal{F}(G, \mathbf{D})$
8. Remove O from G
9. While G is modified
10. If GraphImproved(G, G_0)
11. $G_0 \leftarrow G$
12. While G_0 is modified
13. Return G_0

Table 6.6: Build a latent variable model where observed variables either share the same k latent parents or no parents.

latent variables, and only increase the number of required latent parents for the remaining indicators if the data says so. We conjecture this provides a good trade-off between learning latent variable models that are relatively sparse and the required computational cost of this search.

6.5.1 Remarks

We do not have theoretical results concerning equivalence classes of causal models for combinations of rank-r ($r > 1$) models. Wegelin et al. (2005) describe an equivalence class of some types of rank-r models. They do not provide an equivalence class of all graphs that are undistinguishable given arbitrary combinations of different rank-r constraints. However, for density estimation problems it is not necessary to describe such an equivalence class, as long as the given procedure provides a better estimation of the joint than other methods. The generalized variation of Washdown presented in the next section and the results of Wegelin et al. (2005) might be used as a starting point to new approaches for causal discovery.

6.6 An algorithm for density estimation

.

We first introduce a slightly modified Washdown algorithm that takes an input not only a dataset, but also an integer parameter indicating how many latent parents each indicator should have (i.e., how many latents per cluster). We call this variation the K-LatentClustering algorithm, as shown in Table 6.6. This algorithm is identical to Washdown, with the exception of introducing k latents within each cluster, as made explicit by algorithm IntroduceKLatentCluster (Table 6.7).

Finally, we only need to formalize how K-LatentClustering will be used to generate several disjoint pure measurement models, and how such models are combined. This is detailed by

Algorithm IntroduceKLatentCluster
Input: two graphs G, G_0; a set of observed variables $\mathbf{O}$;
an integer k defining the cluster size
Output: a DAG

1. Let **NodeDump** be the set of observed nodes in $\mathbf{O}$ that are not in G
2. Let T be the number of clusters in G
3. Add a new cluster of k latents $\mathbf{LC_T}$ to G and form a complete DAG among latents in G.
4. For all $V \in$ **NodeDump**
5. If $V \in G_0$
6. Let $\mathbf{LC_i}$ be the parent set of V in G_0
7. Set $\mathbf{LC_{i+1}}$ to be the parent set of V in G
8. Else
9. Set $\mathbf{LC_T}$ to be the parent set of V in G
10. If $\mathbf{LC_T}$ d-separates an insufficient number of nodes
11. Remove $\mathbf{LC_T}$ from G and add its observed children back to **NodeDump**
12. Return G

Table 6.7: Introduce a new latent set by moving nodes down the latent layer.

algorithm FullLatentClustering given in Table 6.8. Notice that, in step 16 of FullLatentClustering, we initialize our final greedy search by making all latents be the parents of the last nodes added to our graph, the set $\mathbf{O}^C$. In step 17, we never add edges from $\mathbf{O}^C$ into a previously clustered node. This simplification of the search space is justified as follows, and illustrated in Figure 6.12: any two previously clustered nodes participate in some rank constraint in the marginal covariance matrix (e.g., nodes X_1 and X_2 in the Figure 6.12(a) participate in a rank-1 constraint with nodes X_3 and X_4). If some other node is set to be a parent of two clustered nodes, this constraint is destroyed (e.g., making X_5 a parent of both X_1 and X_2 would destroy the rank-1 constraint int the covariance matrix of $\{X_1, X_2, X_3, X_4\}$). Although allowing two clustered nodes to have an observed common parent might correct some previous statistical mistake, to simplify the search space we just forbid edges from $\mathbf{O}^C$ into clustered nodes.

More implementation details, concerning for instance the nature of the bi-directed edges used in K-LatentClustering, are given in Appendix C.3.

Finally, we complement the search by looking for structure among the latents, exactly as in the GES-MIMBuild algorithm of Chapter 3. We call the combination FullLatentClustering + GES-MIMBuild the RankBasedAutomatedSearch algorithm (RBAS).

6.7 Experiments on density estimation

We evaluate our algorithm against the mixture of factor analysers (MofFa), one of the approaches most closely related to RBAS, and against FindHidden, a standard algorithm for learning graphical models with latent variables (Elidan et al., 2000). Both RBAS and MofFA intend to be applied to the same kind of data (observed variables with many hidden common causes) using the same type of probabilistic model (finite mixture of Gaussians). FindHidden is best suited when many conditional independencies among observed variables are present in the true model.

The data are normalized to a multivariate standard Normal distribution. We evaluate a model

Algorithm FullLatentClustering
Input: a data set $\mathbf{D}$
Output: a DAG

1. $i \leftarrow 0$; $\mathbf{D}_0 \leftarrow \mathbf{D}$; **Solutions** $\leftarrow \emptyset$; $k \leftarrow 1$
2. Do
3. $\quad G_i \leftarrow$ K-LatentClustering$(\mathbf{D}_i, k)$
4. $\quad$ If G_i is not empty
5. $\quad\quad$ **Solutions** $\leftarrow$ **Solutions** $\cup\, G_i$
6. $\quad\quad \mathbf{D}_{i+1} \leftarrow \Pi_{\mathbf{D}_i \setminus G_i}(\mathbf{D}_i)$
7. $\quad\quad i \leftarrow i + 1$
8. While **Solutions** changes
9. Increase k by 1 and repeat Steps 2-8 till the covariance matrix of $\mathbf{D}_i$ does not have enough entries to justify a k-factor model
10. Let G_{full} be the graph composed by merging all graphs in **Solutions**, where latents are fully connected as an arbitrary DAG
11. For every pair $\{G_i, G_j\} \subseteq$ **Solutions**
12. $\quad$ Let $G_{partial}$ be the respective merge of G_i, G_j
13. $\quad$ Do a standard greedy search, adding bi-directed edges $X_i \leftrightarrow X_j$ to $G_{partial}$
14. $\quad$ Do a standard greedy search, deleting bi-directed edges $X_i \leftrightarrow X_j$ from $G_{partial}$
15. $\quad$ Add all bi-directed edges in $G_{partial}$ to G_{full}
16. Let $\mathbf{O}^C$ be the set of all nodes in $\mathbf{O}$ that are not in G_{full}. Add $\mathbf{O}^C$ to G_{full} and make all latent nodes be parents of all nodes $\mathbf{O}^C$ in G_{full}
17. Do a standard hill climbing procedure do add edges or delete edges into $\mathbf{O}^C$ in G_{full}, or reverse edges connecting two elements of $\mathbf{O}^C$
18. Return G_{full}

Table 6.8: Merge the solutions of multiple K-LatentClustering calls.

by its average log-likelihood on a test set. We perform model selection by using Bayesian criteria. The variational Bayesian mixture of factor analysers (Ghahramani and Beal, 1999) is used to get the number of mixture distributions. For MofFA, we chose the number of factors by using BIC and a grid search from 1 to 15 latents[4]. For FindHidden, we use the implementation with Structural EM described in Chapter 2, but where we also re-evaluate the full model after each modification in order to avoid bad local optima due to the Structural EM approximation. We used exactly the same probabilistic model and variational approximation as in RBAS. Once a model is selected by RBAS, MofFA or FindHidden using a training set, we estimate its parameters by maximum likelihood over the training set and test it with an independent test set.

The datasets used in the experiments are as follows. All datasets and their descriptions can be obtained from the UCI Repository (Blake and Merz, 1998). We basically chose datasets with a large number of continuous eletronic measurements of some natural phenomena, plus a synthetic dataset (wave). Discrete variables were removed. Instances with missing values were removed.

- ionosphere (iono): 351 instances / 34 variables
- heart images (spectf): 349 / 44
- water treatment plant (water): 380 / 38

[4]The available software for variational mixture of factor analysers does not perform model selection for the number of factors. We used the same number of factors per component, which in this study is not a real issue, since in all datasets the number of chosen components was 2.

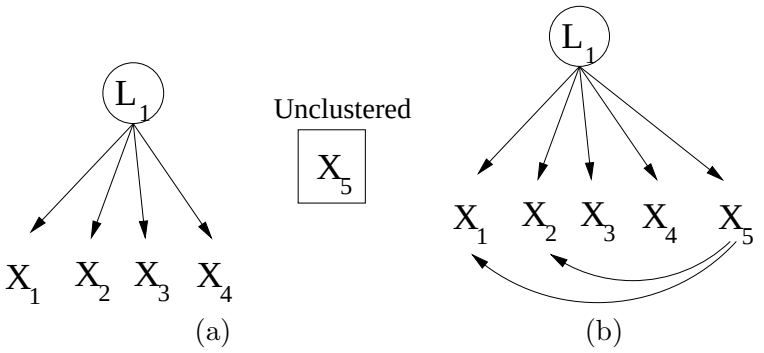


Figure 6.12: Suppose $X_1 - X_4$ are clustered as an one-factor model as in (a), and an unclustered node X_5 has to be added to this model. If X_5 is set to be a common parent of $X_1 - X_4$ as in (b), this contradicts the previously established rank-1 constraint in the covariance matrix of $X_1 - X_4$ implied by the clustering. A simple solution is to avoid adding any edges at all from X_5 into nodes in $X_1 - X_4$.

- waveform generator (wave): 5000 / 21

Table 6.9 shows the results. We use 5-fold cross-validation, and report the results for each partition. Results for RBAS and for the difference RBAS − MofFA (R − M) and RBAS - FindHidden (R − F) are given.

As a baseline, we also report results for the fully connected DAG over the observed variables using no latent variables. This provides an indication on how much the fit can increase by searching for the proper rank constraints. The resuls are given in Table 6.10.

In three datasets, we obtained a clear advantage over MofFA. We outperform FindHidden in *iono* and *spectf* according to a sign test, and *iono* according to a t-test at a 0.01 significance level. One of the reasons RBAS and MofFA did not perform better in the *water* dataset is due to the presence of several ordinal variables. The variational score function was especially unstable in this case, where different starting points would frequently lead to quite different scores. Since FindHidden relies less on latent variables, this might be an explanation of why it gave more stable results across all data partitions. In the dataset *wave*, all three methods gave basically the same result, but in this case even the fully connected model performs as well.

It is interesting to notice that *iono* is the dataset that generated the DAG with the highest number of edges per node in FindHidden before the introduction of any latent. The DAGs generated with the *spectf* dataset are much more sparse, but RBAS consistently outperform the standard FindHidden approach. The dataset *water* is used to illustrate an interesting phenomenon: RBAS does not work well with a dataset which has several discrete ordinal variables, being very unstable.

It should also be obvious that we do not claim that RBAS can be expected to outperform an algorithm such as FindHidden if a finite mixture of Gaussians is a bad probabilistic model for the given problem or if few rank constraints that are useful for clustering variables hold in the population. Datasets such as *iono*, in which observed variables are connected by many hidden common causes, represent the ideal type of problem for this type of approach. Due to the higher computational cost of RBAS, one might want to use a MofFA model to evaluate how well it fits the data compared to some method as FindHidden before trying our algorithm. If MofFA is of

Table 6.9: Evaluation of the average test log-likelihood of the outcomes of three algorithms. Each line is the result of a single split in a 5-fold cross-validation. The entry R − M is the difference between RBAS and MofFA. The entry R − F is the difference between RBAS and FindHidden. The table also provides the respective averages (avg) and standard deviations (dev).

	iono			**spectf**			**water**			**wave**		
Set	RBAS	R - M	R - F	RBAS	R - M	R - F	RBAS	R - M	R - F	RBAS	R - M	R - F
1	-34.65	4.84	9.54	-47.60	1.48	2.33	-30.91	6.33	5.10	-24.11	-0.06	0.80
2	-25.60	6.06	11.58	-45.76	4.72	4.66	-29.69	2.48	4.36	-23.97	-0.05	-0.61
3	-28.30	7.05	11.53	-47.93	-0.01	0.21	-40.76	7.57	-1.74	-23.87	-0.10	0.96
4	-32.90	4.25	6.73	-43.42	2.31	4.64	-42.57	4.97	-2.77	-24.10	-0.09	0.97
5	-32.87	7.72	9.89	-41.52	3.01	5.13	-44.4	8.08	-9.63	-24.24	-0.05	-0.04
avg	-30.86	5.98	9.86	-45.24	2.30	3.40	-39.21	5.88	-0.94	-24.06	-0.07	0.43
dev	3.77	1.46	1.98	2.74	1.75	2.08	8.82	2.25	6.00	0.14	0.02	0.69

Table 6.10: Evaluation of the average test log-likelihood of the outcomes of three algorithms. A fully connected DAG was used in this case as a baseline.

	iono	**spectf**	**water**	**wave**
Set	Full DAG	Full DAG	Full DAG	Full DAG
1	-63.27	-54.09	-52.61	-24.06
2	-42.78	-58.46	-39.05	-23.95
3	-55.15	-55.43	-60.70	-23.84
4	-52.93	-51.60	-57.57	-24.08
5	-64.75	-53.44	-52.48	-24.24
avg	-55.78	-54.60	-54.33	-24.03
dev	8.86	2.56	9.25	0.15

at least competitive performance compared to FindHidden, one might want to apply RBAS to the given problem.

6.8 Summary

We introduced a new Bayesian search algorithm for learning latent variable models. This approach is shown to be especially interesting for density estimation problems. For causality discovery, it can provide provide models where BuildPureClusters fail. The new algorithm also motivates new problems in identification of linear latent variable models using generalized rank constraints and score-based search algorithms that try to achieve a better trade-off between computational cost and quality of the results.

Chapter 7

Conclusion

This thesis introduced several new techniques for learning the structure of latent variables models. The fundamental point of this thesis is that common and appealing heuristics (e.g., factor rotation methods) fail when the goal is structure learning with a causal interpretation. In many cases it is preferable to model the relationships of a subset of the given variables than trying to force a bad model over all of them (Kano and Harada, 2000).

Its main contributions are:

- identifiability results for learning different types of d-separations in a large class of continuous latent variable models;
- algorithms for discovering causal latent variable structures in linear, non-linear and discrete cases, using such identification rules;
- empirical evaluation of causality discovery algorithms, including a study of the shortcomings of the most common method, factor analysis;
- an algorithm for heuristic Bayesian learning of probabilistic models, one of the few methods with arbitrarily connected latents, motivated by results in causal analysis;

The procedures described in this thesis are not meant to discover causal relations when the true measurement model is far from a pure model. This includes, for instance:

- modeling text documents as a mixture of a large number of latent topics (Blei et al., 2003);
- chemometrics studies where observed variables are a mixture of many hidden components (Malinowski, 2002);
- in general, blind source separation problems, where measures are linear combinations of all latents in the study (Hyvarinen, 1999);

A number of open problems invite further research. They can be divided into three main classes.

New identifiability results in covariance structures

- completeness of the tetrad equivalence class of measurement models: can we identify all the common features of measurement models in the same tetrad equivalence class? A simpler, and practical, result would be finding all possible identification rules using no more than six observed variables. Anything more than that might be of limited applicability due to the computational cost and lack of statistical reliability of such criteria;

- the graphical characterization of tetrad constraints in linear DAGs with faithful distributions was fully developed by Spirtes et al. (2000) and Shafer et al. (1993) and provided the main starting point for this thesis. Can we provide a graphical characterization for *conditional* tetrad constraints that could be used to learn directed edges among indicators?

- more generally, a graphical characterization of rank constraints and other type of covariance constraints to learn latent variable models, possibly identifying the nature of some impure relationships. Steps towards such results can be found, e.g., in (Grzebyk et al., 2004; Stanghellini and Wermuth, 2005; Wegelin et al., 2005);

Improving discrete models

- new heuristics to increase the scalability of the causal rule learner of Chapter 5, including special treatment of sparse data such as market basket data, one of the main motivations behind association rule algorithms (Agrawal and Srikant, 1994);

- computationally tractacle approximations for global models with discrete measurement models. Estimating latent trait models with a large number of latents is hard. Even finding the maximum likelihood estimator requires high-dimensional integration (Bartholomew and Knott, 1999; Bartholomew et al., 2002). Monte Carlo approximation algorithms (e.g., Wedel and Kamakura, 2001) are out of question for our problem of model search due to their extremely demanding computational cost. Deterministic approximations, such as the one described by Chu and Ghahramani (2004) to solve the problem of Bayesian ordinal regression, are the only viable alternatives. Finding suitable approximations that can be integrated with model search is an open problem;

Learning non-linear latent structure

- using constraints generated by higher order moments of the observed distribution. Although it was stressed throughout this thesis that such constraints are more problematic in model selection problems to the increased difficulty on statistical estimation, they nevertheless can be useful in practice for small model selection problems. For example, in problems partially solved by covariance constraints. An example of the use of higher order constraints in linear models for non-Gaussian distributions is given by Kano and Shimizu (2003). Several parametric formulations of factor analysis models with non-linear relations exist (Bollen and Paxton, 1998; Wall and Amemiya, 2000; Yalcin and Amemiya, 2001), but no formal description of equivalence classes or systematic search procedures exist to the best of our knowledge;

- in special, finding non-linear causal relationships among latent variables given a fixed linear measurement model can be seen as a problem of regression with measurement error and

instrumental variables (Carroll et al., 2004). Our techniques for learning measurement models for non-linear structural models as a way of finding instrumental variables could also be adapted to this specific problem. Moreover, research in non-parametric item response theory (Junker and Sijtsma, 2001) can also provide ideas for the discrete case;

- moreover, since our algorithms are basically using information concerning dot products of vectors of random variables (i.e., covariance information), it can be adapted to non-linear spaces by means of the "kernel trick" (Scholkopf and Smola, 2002; Bach and Jordan, 2002). This basically consists on mapping the input space to some feature space by a non-linear transformation. In this feature space, algorithms designed for linear models (e.g., principal component analysis, Scholkopf and Smola, 2002) can be applied in a relatively straightforward and computationally unexpensive way. This might be problematic if one is interested in a causal description of the data generating process, but not as much if the goal is density estimation.

This thesis was concluded roughly a hundred years after Charles Spearman published what is usually acknowledged as the first application of factor analysis (Spearman, 1904). Much has been done concerning estimation of latent variable models (Bartholomew et al., 2002; Loehlin, 2004; Jordan, 1998), but little progress on automated search of causal models with latent variables was achieved. Few problems in automated learning and discovery are as difficult and fundamental as learning causal relations among latent variables without background knowledge and experimental data. Better methods are available now, and further improvements will surely come from machine learning research.

Appendix A

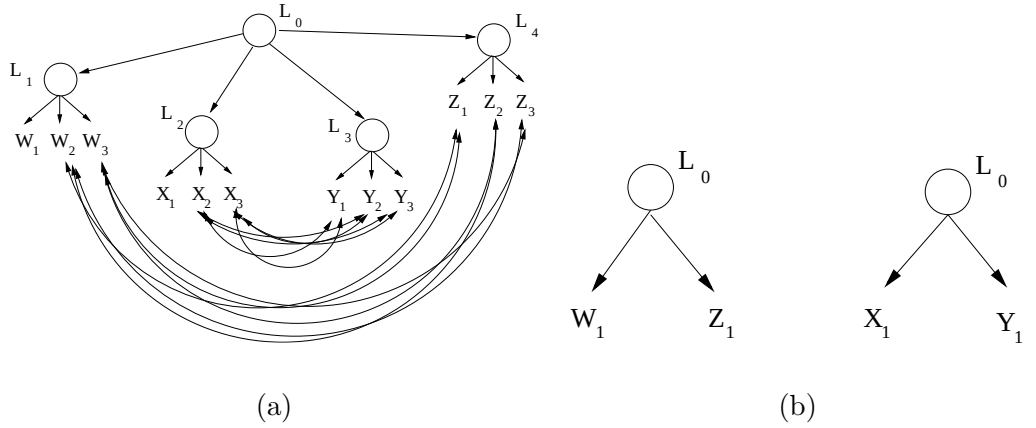
Results from Chapter 3

A.1 BuildPureClusters: refinement steps

Concerning the final steps of Table 3.2, it might be surprising that we merge clusters of variables that we know cannot share a common latent parent in the true graph. However, we are not guaranteed to find a large enough number of pure indicators for each of the original latent parents, and as a consequence only a subset of the true latents will be represented in the measurement pattern. It might be the case that, with respect to the variables present in the output, the observed variables in two different clusters might be directly measuring some ancestor common to all variables in these two clusters. As an illustration, consider the graph in Figure A.1(a), where double-directed edges represent independent hidden common causes. Assume any sensible purification procedure will choose to eliminate all elements in $\{W_2, W_3, X_2, X_3, Y_2, Y_3, Z_2, Z_3\}$ because they are directly correlated with a large number of other observed variables (extra edges and nodes not depicted).

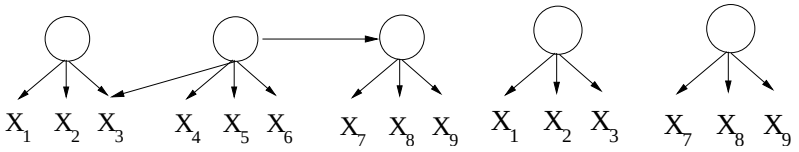
Meanwhile, one can verify that all three tetrad constraints hold in the covariance matrix of $\{W_1, X_1, Y_1, Z_1\}$, and therefore there will be no undirected edges connecting pairs of elements in this set in the corresponding measurement pattern. Rule CS1 is able to separate W_1 and X_1 into two different clusters by using $\{W_2, W_3, X_2, X_3\}$ as the support nodes, and analogously the same happens to Y_1 and Z_1, W_1 and Y_1, X_1 and Z_1. However, no test can separate W_1 and Z_1, nor X_1 and Y_1. If we do not merge clusters, we will end up with the graph seen in Figure A.1(b) as part of our output pattern. Although this is a valid measurement pattern, and in some situations we might want to output such a model, it is also true that W_1 and Z_1 measure a same latent L_0 (as well as X_1 and Y_1). It would be problematic to learn a structural model with such a measurement model. There is a deterministic relation between the latent measured by W_1 and Z_1, and the latent measured by X_1 and Y_1: they are the same latent! Probability distributions with deterministic relations are not faithful, and that causes problems for learning algorithms.

Finally, we show examples where Steps 6 and 7 of BuildPureClusters are necessary. In Figure A.2(a) we have a partial view of a latent variable graph, where two of the latents are marginally independent. Suppose that nodes X_4, X_5 and X_6 are correlated to many other measured nodes not in this figure, and therefore are removed by our purification procedure. If we ignore Step 6, the resulting pure submodel over $\{X_1, X_2, X_3, X_7, X_8, X_9\}$ will be the one depicted in Figure A.2(b) ($\{X_1, X_2\}$ are clustered apart from $\{X_7, X_8, X_9\}$ because of marginal zero correlation, and X_3 is clustered apart from $\{X_7, X_8, X_9\}$ because of CS1 applied to $\{X_3, X_4, X_5\} \times \{X_7, X_8, X_9\}$). However, no linear latent variable model can be parameterized by this graph: if we let the two



(a) (b)

Figure A.1: The true graph in (a) will generate at some point a purified measurement pattern as in (b). It is desirable to merge both clusters.



(a) (b)

Figure A.2: Suppose (a) is our true model. If for some reason we need to remove nodes X_4, X_5 and X_6 from our final pure graph, the result will be as shown in Figure (b), unless we apply Step 6 of BuildPureClusters. There are several problems with (b), as explained in the text.

latents to be correlated, this will imply X_1 and X_7 being correlated. If we make the two latents uncorrelated, X_3 and X_7 will be uncorrelated.

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

A.2 Proofs

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

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

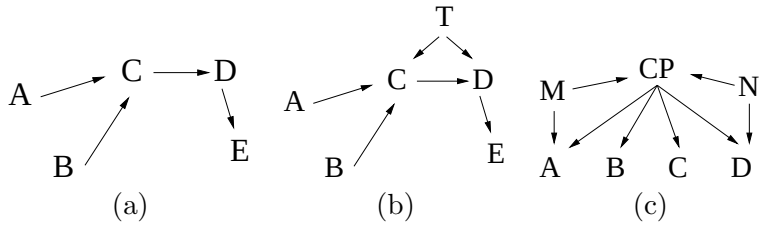


Figure A.3: In (a), C is a choke point for sets $\{A, B\} \times \{D, E\}$, since it lies on all treks connecting nodes in $\{A, B\}$ to nodes in $\{D, E\}$ and lies also on the $\{D, E\}$ side of all of such treks. For instance, C is on the $\{D, E\}$ side of $A \rightarrow C \rightarrow D$, where A is the source of such a trek. Notice also that this choke point d-separates nodes in $\{A, B\}$ from nodes in $\{D, E\}$. Analogously, D is also a choke point for $\{A, B\} \times \{D, E\}$ (there is nothing on the definition of a choke point $\mathbf{I} \times \mathbf{J}$ that forbids it of belonging $\mathbf{I} \cup \mathbf{J}$). In Figure (b), C is a choke point for sets $\{A, B\} \times \{D, E\}$ that does not d-separate such elements. In Figure (c), CP is a node that lies on all treks connecting $\{A, C\}$ and $\{B, D\}$ but it is not a choke point, since it does not lie on the $\{A, C\}$ side of trek $A \leftarrow M \rightarrow CP \rightarrow B$ and neither lies on the $\{B, D\}$ side of $D \leftarrow N \rightarrow CP \rightarrow A$. The same node, however, is a $\{A, D\} \times \{B, C\}$ choke point.

into X_1 (or X_n) if the arrow of the edge $\{X_1, X_2\}$ is into X_1 ($\{X_{n-1}, X_n\}$ into X_n);

- a *collider* on a path $\{X_1, \ldots, X_n\}$ is a node X_i, $1 < i < n$, such that X_{i-1} and X_{i+1} are parents of X_i;
- a *trek* is a path that does not contain any collider;
- the *source* of a trek is the unique node in a trek to which no arrows are directed;
- the I *side* of a trek between nodes I and J with source X is the subpath directed from X to I. It is possible that $X = I$, and the I side is just node I;
- a *choke point* CP between two sets of nodes $\mathbf{I}$ and $\mathbf{J}$ is a node that lies on every trek between any element of $\mathbf{I}$ and any element of $\mathbf{J}$ such that CP is either (i) on the $\mathbf{I}$ side of every such trek [1] or (ii) on the $\mathbf{J}$ side or every such trek.

With the exception of choke points, all other concepts are well known in the literature of graphical models (Spirtes et al., 2000; Pearl, 1988, 2000). What is interesting in a choke point is that, by definition, such a node is in all treks linking elements in two sets of nodes. Being in all treks connecting a node X_i and a node X_j is a necessary condition for a node to d-separate X_i and X_j, although this is not a sufficient condition.

Consider Figure A.3, which illustrates several different choke points. In some cases, the choke point will d-separate a few nodes. The relevant fact is that even when the choke point is a latent variable, this has an implication on the observed marginal distribution, as stated by the *Tetrad Representation Theorem*:

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

Theorem A.1 (The Tetrad Representation Theorem) *Let G be a linear latent variable model, and let I_1, I_2, J_1, J_2 be four variables in G. Then $\sigma_{I_1J_1}\sigma_{I_2J_2} = \sigma_{I_1J_2}\sigma_{I_2J_1}$ if and only if there is a choke point between $\{I_1, I_2\}$ and $\{J_1, J_2\}$.*

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

Shafer et al. (1993) also provide more details on the definitions and several examples.

Therefore, unlike a partial correlation constraint obtained by conditioning on a given set of variables, where such a set should be observable, *some d-separations due to latent variables can be inferred using tetrad constraints*. We will use the Tetrad Representation Theorem to prove most of our results. The challenge lies on choosing the right combination of tetrad constraints that allows us to identify latents and d-separations due to latents, since the Tetrad Representation Theorem is far from providing such results directly.

In the following proofs, we will frequently use the symbol $G(\mathbf{O})$ to represent a linear latent variable model with a set of observed nodes $\mathbf{O}$. A choke point between sets $\mathbf{I}$ and $\mathbf{J}$ will be denoted as $\mathbf{I} \times \mathbf{J}$. We will first introduce a lemma that is going to be useful to prove several other results. The lemma is a slightly reformulated version of the one given in Chapter 3 to include a result on choke points:

Lemma 3.4 *Let $G(\mathbf{O})$ be a linear latent variable model, and let $\{X_1, X_2, X_3, X_4\} \subset \mathbf{O}$ be such that $\sigma_{X_1X_2}\sigma_{X_3X_4} = \sigma_{X_1X_3}\sigma_{X_2X_4} = \sigma_{X_1X_4}\sigma_{X_2X_3}$. If $\rho_{AB} \neq 0$ for all $\{A, B\} \subset \{X_1, X_2, X_3, X_4\}$, then an unique choke point P entails all the given tetrad constraints, and P d-separates all elements in $\{X_1, X_2, X_3, X_4\}$.*

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

Assume $P \neq Q$. Because there is a trek that links X_1 and X_4 throught P (since $\rho_{X_1X_4} \neq 0$), we have that Q should also be on that trek. Suppose T is a trek connecting X_1 to X_4 through P and Q, and without loss of generality assume this trek follows an order that defines three subtreks: T_0, from X_1 to P; T_1, from P to Q; and T_2, from Q to X_4, as illustrated by Figure A.4(a). In principle, T_0 and T_2 might be empty, i.e., we are not excluding the possibility that $X_1 = P$ or $X_4 = Q$.

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

Case 1: T_{Q2} includes P. T_{Q2} has to be into P, and $P \neq X_1$, or otherwise there will be a trek connecting X_2 to X_1 through a (possibly empty) trek T_0 that does not include Q, contrary to our hypothesis. For the same reason, T_0 has to be into P. This will imply that T_1 is a directed path from P to Q, and T_2 is a directed path from Q to X_4 (Figure A.4(b)).

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

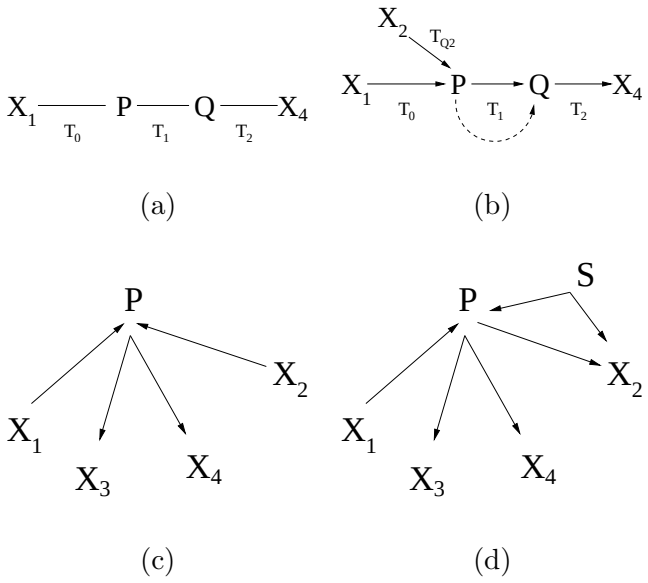


Figure A.4: In (a), a depiction of a trek T linking X_1 and X_4 through P and Q, creating three subtreks labeled as T_0, T_1 and T_2. Directions in such treks are left unspecified. In (b), the existence of a trek T_{Q2} linking X_2 and Q through P will compel the directions depicted as a consequence of the given tetrad and correlation constraints (the dotted path represents any possible continuation of T_{Q2} that does not coincide with T). The configuration in (c) cannot happen if P is a choke point entailing all three tetrads among marginally dependent nodes $\{X_1, X_2, X_3, X_4\}$. The configuration in (d) cannot happen if P is a choke point for $\{X_1, X_3\} \times \{X_2, X_4\}$, since there is a trek $X_1 - P - X_2$ such that P is not on the $\{X_1, X_3\}$ side of it, and another trek $X_2 - S - P - X_3$ such that P is not on the $\{X_2, X_4\}$ side of it.

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

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

To finish the proof, we only have to show that there are no two elements $\{A, B\} \subset \{X_1, X_2, X_3, X_4\}$ such that A and B are both connected to P through treks that are both into P.

We will prove that by contradiction, that is, assume without loss of generality that there is a trek connecting X_1 and P that is into P, and a trek connecting X_2 and P that is into P. If there is no trek connecting X_1 and P that is out of P neither any trek connecting X_2 and P that is out of P, then there is no trek connecting X_1 and X_2, since P is on every trek connecting these two elements

according to the Tetrad Representation Theorem. But this implies $\rho_{X_1X_2} = 0$, a contradiction, as illustrated by Figure A.4(c).

Consider the case where there is also a trek out of P and into X_2. Then there is a trek connecting X_1 to X_2 through P that is not on the $\{X_1, X_3\}$ side of pair $\{X_1, X_3\} \times \{X_2, X_4\}$ to which P is a choke point. Therefore, P should be on the $\{X_2, X_4\}$ of every trek connecting elements pairs in $\{X_1, X_3\} \times \{X_2, X_4\}$. Without loss of generality, assume there is a trek out of P and into X_3 (because if there is no such trek for either X_3 and X_4, we fall in the previous case by symmetry). Let S be the source of a trek into P and X_2, which should exist since X_2 is not an ancestor of P. Then there is a trek of source S connecting X_3 and X_2 such that P is not on the $\{X_2, X_4\}$ side of it as shown in Figure A.4(d). Therefore P cannot be a choke point for $\{X_1, X_3\} \times \{X_2, X_4\}$. Contradiction. □

Lemma 4.2 *Let $G(\mathbf{O})$ be a linear latent variable model. If for some set $\mathbf{O}' = \{X_1, X_2, X_3, X_4\} \subseteq \mathbf{O}$, $\sigma_{X_1X_2}\sigma_{X_3X_4} = \sigma_{X_1X_3}\sigma_{X_2X_4} = \sigma_{X_1X_4}\sigma_{X_2X_3}$ and for all triplets $\{A, B, C\}$, $\{A, B\} \subset \mathbf{O}', C \in \mathbf{O}$, we have $\rho_{AB.C} \neq 0$ and $\rho_{AB} \neq 0$, then no element $A \in \mathbf{O}'$ is a descendant of an element of $\mathbf{O}'\backslash\{A\}$ in G.*

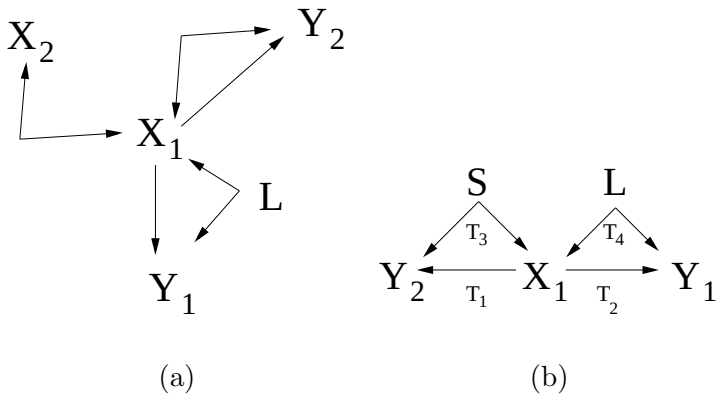
Proof: Without loss of generality, assume for the sake of contradiction that X_1 is an ancestor of X_2. From the given tetrad and correlation constraints and Lemma 3.4, there is a node P that lies on every trek between X_1 and X_2 and d-separates these two nodes. Since P lies on the directed path from X_1 to X_2, P is a descendant of X_1, and therefore an observed node. However, this implies $\rho_{X_1X_2.P} = 0$, contrary to our hypothesis. □

Lemma 4.4 *Let $G(\mathbf{O})$ be a linear latent variable model. Assume $\mathbf{O}' = \{X_1, X_2, X_3, Y_1, Y_2, Y_3\} \subseteq \mathbf{O}$. If constraints $\{\tau_{X_1Y_1X_2X_3}, \tau_{X_1Y_1X_3X_2}, \tau_{Y_1X_1Y_2Y_3}, \tau_{Y_1X_1Y_3Y_2}, \neg\tau_{X_1X_2Y_2Y_1}\}$ all hold, and that for all triplets $\{A, B, C\}, \{A, B\} \subset \mathbf{O}'$, $C \in \mathbf{O}$, we have $\rho_{AB} \neq 0, \rho_{AB.C} \neq 0$, then X_1 and Y_1 do not have a common parent in G.*

Proof: We will prove this result by contradiction. Suppose that X_1 and Y_1 have a common parent L in G. Suppose L is not a choke point for $\{X_1, X_2\} \times \{Y_1, X_3\}$ corresponding to one of the tetrad constraints given by hypothesis. Because of the trek $X_1 \leftarrow L \rightarrow Y_1$, then either X_1 or Y_1 is a choke point. Without loss of generality, assume X_1 is a choke point in this case. By Lemma 4.2 and the given constraints, X_1 cannot be an ancestor of either X_2 or X_3, and by Lemma 3.4 it is also the choke point for $\{X_1, Y_1\} \times \{X_2, X_3\}$. That means that all treks connecting X_1 and X_2, and X_1 and X_3 should be into X_1. Since there are no treks between X_2 and X_3 that do not include X_1, and all paths between X_2 and X_3 that include X_1 collide at X_1, that implies $\rho_{X_2X_3} = 0$, contrary to our hypothesis. By symmetry, Y_1 cannot be a choke point. Therefore, L is a choke point for $\{X_1, Y_1\} \times \{X_2, X_3\}$ and by Lemma 3.4, it also lies on every trek for any pair in $\mathbf{S_1} = \{X_1, X_2, X_3, Y_1\}$.

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

Remember that predicate $F_1(X, Y, G)$ is true if and only if there exist two nodes W and Z in G such that τ_{WXYZ} and τ_{WXZY} are both entailed, all nodes in $\{W, X, Y, Z\}$ are correlated, and



(a) (b)

Figure A.5: Figure (a) illustrates necessary treks among elements of $\{X_1, X_2, Y_1, Y_2, L\}$ according to the assumptions of Lemma 4.5 if we further assume that X_1 is a choke point for pairs $\{X_1, X_2\} \times \{Y_1, Y_2\}$ (other treks might exist). Figure (b) rearranges (a) by emphasizing that Y_1 and Y_2 cannot be d-separated by a single node.

there is no observed C in G such that $\rho_{AB.C} = 0$ for $\{A, B\} \subset \{W, X, Y, Z\}$.

Lemma 4.5 *Let $G(\mathbf{O})$ be a linear latent variable model. Assume $\mathbf{O}' = \{X_1, X_2, X_3, Y_1, Y_2, Y_3\} \subseteq \mathbf{O}$, such that $F_1(X_1, X_2, G)$ and $F_1(Y_1, Y_2, G)$ hold, Y_1 is not an ancestor of Y_3 and X_1 is not an ancestor of X_3. If constraints $\{\tau_{X_1Y_1Y_2X_2}, \tau_{X_2Y_1Y_3Y_2}, \tau_{X_1X_2Y_2X_3}, \neg\tau_{X_1X_2Y_2Y_1}\}$ all hold, and that for all triplets $\{A, B, C\}, \{A, B\} \subset \mathbf{O}', C \in \mathbf{O}$, we have $\rho_{AB} \neq 0, \rho_{AB.C} \neq 0$, then X_1 and Y_1 do not have a common parent in G.*

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

Step 1: If there is a common parent L to X_1 and Y_1, then L is a $\{X_1, X_2\} \times \{Y_1, Y_2\}$ choke point. For the sake of contradiction, assume X_1 is a choke point in this case. By Lemma 4.2 and assumption $F_1(X_1, X_2, G)$, we have that X_1 is not an ancestor of X_2, and therefore all treks connecting X_1 and X_2 should be into X_1. Since $\rho_{X_2Y_2} \neq 0$ by assumption and X_1 is on all treks connecting X_2 and Y_2, there must be a directed path out of X_1 and into Y_2. Since $\rho_{X_2Y_2.X_1} \neq 0$ by assumption and X_1 is on all treks connecting X_2 and Y_2, there must be a trek into X_1 and Y_2. Because $\rho_{X_2Y_1} \neq 0$, there must be a trek out of X_1 and into Y_1. Figure A.5(a) illustrates the configuration.

Since $F_1(Y_1, Y_2, G)$ is true, by Lemma 3.4 there must be a node d-separating Y_1 and Y_2 (neither Y_1 nor Y_2 can be the choke point in $F_1(Y_1, Y_2, G)$ because this choke point has to be latent, according to the partial correlation conditions of F_1). However, by Figure A.5(b), treks $T_2 - T_3$ and $T_1 - T_4$ cannot both be blocked by a single node. Contradiction. Therefore X_1 cannot be a choke point for $\{X_1, X_2\} \times \{Y_1, Y_2\}$ and, by symmetry, neither can Y_1.

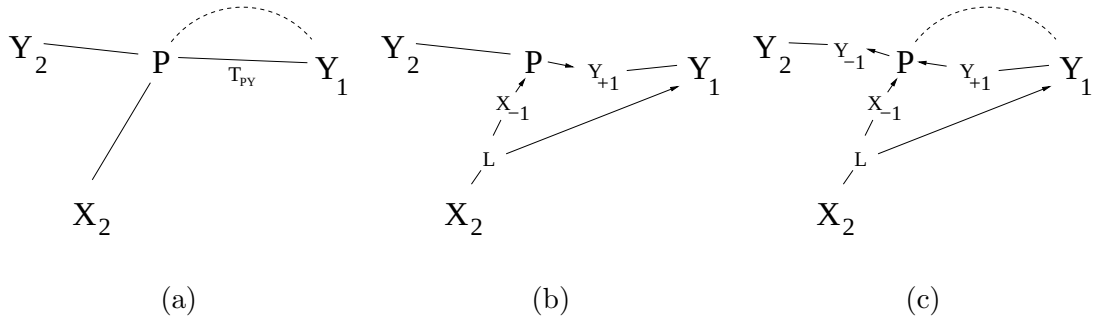


Figure A.6: In (a), a depiction of T_Y and T_X, where edges represent treks (T_X can be seen more generally as the combination of the solid edge between X_2 and P concatenated with a dashed edge between P and Y_1 representing the possibility that T_Y and T_X might intersect multiple times in T_{PY}, but in principle do not need to coincide in T_{PY} if P is not a choke point.) In (b), a possible configurations of edges $< X_{-1}, P >$ and $< P, Y_{+1} >$ that do not collide in P, and P is a choke point (and $Y_{+1} \neq Y$). In (c), the edge $< Y_{-1}, P >$ is compelled to be directed away from P because of the collider with the other two neighbors of P.

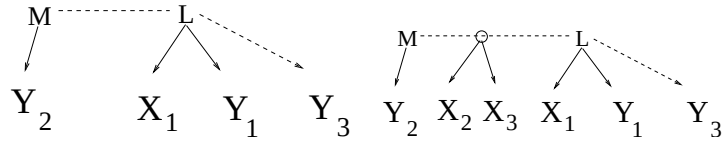
Step 2: L is on every trek connecting Y_1 and Y_2 and on every trek connecting X_1 and X_2. Let L be the choke point for pairs $\{X_1, X_2\} \times \{Y_1, Y_2\}$. As a consequence, all treks between Y_2 and X_1 go through L. All treks between X_2 and Y_1 go through L. All treks between X_2 and Y_2 go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

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

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

Let T_{PY} be the subtrek of T_Y starting on P and ending one node before Y_1. Any choke point $\{X_2, Y_2\} \times \{Y_1, Y_3\}$ should lie on T_{PY} (Figure A.6(a)). (Y_1 cannot be such a choke point, since all treks connecting Y_1 and Y_2 are into Y_1, and by hypothesis all treks connecting Y_1 and Y_3 are into Y_1. Since all treks connecting Y_2 and Y_3 would need to go through Y_1 by definition, then there would be no such trek, implying $\rho_{Y_2Y_3} = 0$, contrary to our hypothesis.)

Assume first that $X_2 \neq P$ and $Y_2 \neq P$. Let X_{-1} be the node before P in T_X starting from X_2. Let Y_{-1} be the node before P in T_Y starting from Y_2. Let Y_{+1} be the node after P in T_Y starting from Y_2 (notice that it is possible that $Y_{+1} = Y_1$). If X_{-1} and Y_{+1} do not collide on P (i.e., there is no structure $X_{-1} \to P \leftarrow Y_{+1}$), then there will be a trek connecting X_2 to Y_1 through T_{PY} after P. Since L is not in T_{PY}, L should be before P in T_X. But then there will be a trek connecting X_2 and Y_1 that does not intersect T_{PY}, which is a contradiction (Figure A.6(b)). If the collider does exist, we have the edge $P \leftarrow Y_{+1}$. Since no collider $Y_{-1} \to P \leftarrow Y_{+1}$ can exist because T_Y is a trek, the edge between Y_{-1} and P is out of P. But that forms a trek connecting X_2 and Y_2 (Figure



(a) (b)

Figure A.7: In (a), Y_2 and X_1 cannot share a parent, and because of the given tetrad constraints, L should d-separate M and Y_3. Y_3 is not a child of L either, but there will be a trek linking L and Y_3. In (b), an (invalid) configuration for X_2 and X_3, where they share an ancestor between M and L.

A.6(c)), and since L is in every trek between X_2 and Y_2 and T_Y does not contain L, then T_X should contain L before P, which again creates a trek between X_2 and Y_1 that does not intersect T_{PY}.

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

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

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

Lemma 3.8 *Let* $G(\mathbf{O})$ *be a linear latent variable model. Let* $\mathbf{O}' = \{X_1, X_2, X_3, Y_1, Y_2, Y_3\} \subseteq \mathbf{O}$. *If constraints* $\{\tau_{X_1Y_1Y_2Y_3}, \tau_{X_1Y_1Y_3Y_2}, \tau_{X_1Y_2X_2X_3}, \tau_{X_1Y_2X_3X_2}, \tau_{X_1Y_3X_2X_3}, \tau_{X_1Y_3X_3X_2}, \neg\tau_{X_1X_2Y_2Y_3}\}$ *all hold, and that for all triplets* $\{A, B, C\}, \{A, B\} \subset \mathbf{O}', C \in \mathbf{O}$, *we have* $\rho_{AB} \neq 0, \rho_{AB.C} \neq 0$, *then* X_1 *and* Y_1 *do not have a common parent in* G.

Proof: We will prove this result by contradiction. Suppose X_1 and Y_1 have a common parent L in G. Since all three tetrads hold in the covariance matrix of $\{X_1, Y_1, Y_2, Y_3\}$, by Lemma 3.4 the choke point that entails these constraints d-separates the elements of $\{X_1, Y_1, Y_2, Y_3\}$. The choke point should be in the trek $X_1 \leftarrow L \rightarrow Y_1$, and since it cannot be an observed node because by hypothesis no d-separation conditioned on a single node holds among elements of $\{X_1, Y_1, Y_2, Y_3\}$, L has to be a latent choke point for all pairs of pairs in $\{X_1, Y_1, Y_2, Y_3\}$.

It is also given that $\{\tau_{X_1Y_2X_2X_3}, \tau_{X_1Y_2X_3X_2}, \tau_{X_1Y_1Y_2Y_3}, \tau_{X_1Y_1Y_3Y_2}\}$ holds. Since it is the case that $\neg\tau_{X_1X_2Y_2Y_3}$, by Lemma 4.4 X_1 and Y_2 cannot share a parent. Let T_{ML} be a trek connecting some parent M of Y_2 and L. Such a trek exists because $\rho_{X_1Y_2} \neq 0$.

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

ancestor of L. If V is an ancestor of M, then there will be a trek connecting Y_2 and Y_3 that is not through L, which is a contradiction. If V is an ancestor of L but not M, then both Y_2 and Y_3 are d-connected to a node V is a collider at the intersection of such d-connecting treks. However, V is an ancestor of L, which means L cannot d-separate Y_2 and Y_3, a contradiction. Finally, if the trek connecting V and Y_3 is out of V, then Y_2 and Y_3 will be connected by a trek that does not include L, which again is not allowed. We therefore showed there is no node with the properties of V. This configuration is illustrated by Figure A.7(a).

Since all three tetrads hold among elements of $\{X_1, X_2, X_3, Y_2\}$, then by Lemma 3.4, there is a single choke point P that entails such tetrads and d-separates elements of this set. Since T_{ML} is a trek connecting Y_2 to X_1 through L, then there are three possible locations for P in G:

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

Case 2: P lies between M and L in T_{ML}. This configuration is illustrated by Figure A.7(b). As before, no choke point exists for pairs $\{X_1, X_3\} \times \{X_2, Y_3\}$, contrary to our hypothesis.

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

Theorem 3.10 *The output of* FindPattern *is a measurement pattern with respect to the tetrad and vanishing partial correlation constraints of* Σ

Proof: Two nodes will not share a common latent parent in a measurement pattern if and only if they are not linked by an edge in graph C constructed by algorithm FindPattern and that happens if and only if some partial correlation vanishes or if any of rules CS1, CS2 or CS3 applies. But then by Lemmas 4.4, 4.5, 3.8 and the equivalence of vanishing partial correlations and conditional independence in linearly faithful distributions (Spirtes et al., 2000) the claim is proved. The claim about undirected edges follows from Lemma 4.2. □

Theorem 3.11 *Given a covariance matrix* Σ *assumed to be generated from a linear latent variable model* $G(\mathbf{O})$ *with latent variables* $\mathbf{L}$*, let* G_{out} *be the output of* BuildPureClusters(Σ) *with observed variables* $\mathbf{O_{out}} \subseteq \mathbf{O}$ *and latent variables* $\mathbf{L_{out}}$*. Then* G_{out} *is a measurement pattern, and there is an injective mapping* $M : \mathbf{L_{out}} \rightarrow \mathbf{L}$ *with the following properties:*

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

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

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

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

Let Cl_i be a cluster in a non-empty G_{out}. Cl_i has three elements X, Y and Z, and there is at least some W in G_{out} such that all three tetrad constraints hold in the covariance matrix of $\{W, X, Y, Z\}$, where no pair of elements in $\{X, Y, Z\}$ is marginally d-separated or d-separated by an observable variable. By Lemma 3.4, it follows that there is an unique latent L_i d-separating X, Y and Z. If Cl_i has more than three elements, it follows that since no node other than L_i can d-separate all three elements in $\{X, Y, Z\}$, and any choke point for $\{W', X, Y, Z\}$, $W' \in Cl_i$, will d-separate all elements in $\{W', X, Y, Z\}$, then there is an unique latent L_i d-separating all elements in C_i. An analogous argument concerns the d-separation of any element of Cl_i and observed nodes in other clusters.

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

If there is no trek connecting X to L_i that is into L_i, then L_i d-separates X and L_j. The same holds for L_j and A with respect to L_i. If there is a trek T connecting X and L_i that is into L_i, and since all three tetrad constraints hold in the covariance matrix of $\{X, Y, Z, A\}$ by construction, then there is no trek connecting A and L_i that is into L_i (Lemma 3.4). Since there are treks connecting L_i and L_j, they should be all out of L_i and into L_j. This means that L_i d-separates X and L_j. But this also creates a trek connecting X and L_j that is into L_j. Since all three tetrad constraints hold in the covariance matrix of $\{X, A, B, C\}$ by construction, then there is no trek connecting A and L_j that is into L_j (by the d-separation implied by Lemma 3.4). This means that L_j d-separates A from L_i. This also means that the existance of such a trek T out of X and into L_i forbids the existance of any trek connecting a variable correlated to X that is into L_i (since all treks connecting L_i and some L_j are out of L_i), which proves the third claim of the theorem.

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

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

Theorem 3.12 *Given a covariance matrix Σ assumed to be generated from a linear latent variable model $G(\mathbf{O})$ with latent variables $\mathbf{L}$, let G_{out} be the output of* BuildPureClusters(Σ) *with*

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

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

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

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

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

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

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

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

The relation $\sigma_{\epsilon_X L_X} = 0$ follows directly from the fact that λ_X is defined by the regression coefficient of X on L_X. Notice that if X and L_X do not have a common ancestor, λ_X is the direct effect of L_X in X with respect to G_{out}. As we know, by Theorem 3.11, at most one variable in any set of correlated variables will not fulfill this condition.

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

Theorem 3.13 *Problem* $\mathcal{MP}^3$ *is NP-complete.*

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

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

Corollary 3.14 *The output of* BUILDPURECLUSTERS *retains its guarantees even when rules CS1, CS2 and CS3 are applied an arbitrary number of times in* FINDPATTERN *for any arbitrary subset of nodes and an arbitrary number of maximal cliques is found.*

Proof: Independently of the choice made on Step 2 of BUILDPURECLUSTERS and which nodes are not separated into different cliques in FINDPATTERN, the exhaustive verification of tetrad constraints by BUILDPURECLUSTERS provides all the necessary conditions for the proof of Theorem 3.11. □

Corollary 3.16 *Given a covariance matrix Σ assumed to be generated from a linear latent variable model G, and G_{out} the output of* BUILDPURECLUSTERS *given Σ, the output of* PC-MIMBUILD *or* FCI-MIMBUILD *given (Σ, G_{out}) returns the correct Markov equivalence class of the latents in G corresponding to latents in G_{out} according to the mapping implicit in* BUILDPURECLUSTERS

Proof: By Theorem 3.11, each observed variable is d-separated from all other variables in G_{out} given its latent parent. By Theorem 3.12, one can parameterize G_{out} as a linear model such that the observed covariance matrix as a function of the parameterized G_{out} equals its corresponding marginal of Σ. By Theorem 3.15, the rank test using the measurement model of G_{out} is therefore a consistent independence test of latent variables. The rest follows immediately from the consistency property of PC and FCI given a valid oracle for conditional independencies. □

A.3 Implementation

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

A.3.1 Robust purification

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

Suppose we are given a clustering of variables (not necessarily disjoint clusters) and a undirect graph indicating which variables might be ancestors of each other, analogous to the undirect edges generated in FindPattern. We purify this clustering not by testing multiple tetrad constraints, but through a greedy search that eliminates nodes from a linear measurement model that entails tetrad constraints. This is iterated till the current model fits the data according to a chi-square test of significance (Bollen, 1989) and a given acceptance level. Details are given in Table A.1.

This implementation is used as a subroutine for a more robust implementation of BuildPureClusters described in the next section. However, it can be considerably slow. An alternative is using the approximation derived by Kano and Harada (2000) to rapidly calculate the fitness of a factor analysis model when a variable is removed. Another alternative is a greedy search over the initial measurement model, freeing correlations of pairs of measured variables. Once we found which variables are directly connected, we eliminate some of them till no pair is impure. Details of this particular implementation are given by Silva and Scheines (2004). In our experiments with synthetic data, it did not work as well as the iterative removal of variables described in Table A.1. However, we do apply this variation in the last experiment described in Section 6, because it is computationally cheaper. If the model search in RobustPurify does not fit the data after we eliminate too many variables (i.e., when we cannot statistically test the model) we just return an empty model.

A.3.2 Finding a robust initial clustering

The main problem of applying FindPattern directly by using statistical tests of tetrad constraints is the number of false positives: accepting a rule (CS1, CS2, or CS3) as true when it does not hold in the population. One can see that might happen relatively often when there are large groups of observed variables that are pure indicators of some latent: for instance, assume there is a latent L_0 with 10 pure indicators. Consider applying CS1 to a group of six pure indicators of L_0. The first two constraints of CS1 hold in the population, and so assume they are correctly identified by the statistical test. The last constraint, $\sigma_{X_1X_2}\sigma_{Y_1Y_2} \neq \sigma_{X_1Y_2}\sigma_{X_2Y_1}$, should not hold in the population, but will not be rejected by the test with some probability. Since there are $10!/(6!4!) = 210$ ways of CS1 being wrongly applied due to a statistical mistake, we *will* get many false positives in all certainty.

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

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

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

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

Table A.1: A score-based purification.

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

Based on DisjointGroup, we propose here a modification to increase the robustness of BuildPureClusters, the RobustBuildPureClusters algorithm, as given in Table A.2. It starts with a first step called FindInitialSelection (Table A.3). The goal of FindInitialSelection is to find a pure model using only DisjointGroup instead of CS1, CS2 or CS3. This pure model is then used as an starting point for learning a more complete model in the remaining stages of RobustBuildPureClusters.

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

In FindInitialSelection we then find cliques of compatible nodes (Step 8)[2]. Each clique is a candidate for a one-factor model (a latent model with one latent only). We purify every clique found to create pure one-factor models (Step 9). This avoids using clusters that are large not because they are all unique children of the same latent, but because there was no way of separating its elements. This adds considerably more computational cost to the whole procedure.

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

ChooseClusteringClique is a heuristic designed to find a large set of one-factor models

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

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

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

Table A.2: A modified BuildPureClusters algorithm.

(nodes of H) that can be grouped into a pure model with three indicators per latent (we need a heuristic since finding a maximum clique in H is NP-hard). First, we define the *size* of a clustering $H_{candidate}$ (a set of nodes from H) as the number of variables that remain according to the following elimination criteria: 1. eliminate all variables that appear in more than one one-factor model inside $H_{candidate}$; 2. for each pair of variables $\{X_1, X_2\}$ such that X_1 and X_2 belong to different one-factor models in $H_{candidate}$, if there is an edge $X_1 - X_2$ in C, then we remove one element $\{X_1, X_2\}$ from $H_{candidate}$ (i.e., guarantee that no pair of variables from different clusters which were not shown to have any common latent parent will exist in $H_{candidate}$). We eliminate the one that belongs to the largest cluster, unless the smallest cluster has less than three elements to avoid extra fragmentation; 3. eliminate clusters that have less than three variables.

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

A.3.3 Clustering refinement

The next steps in RobustBuildPureClusters are basically the FindPattern algorithm of Table 3.1 with a final purification. The main difference is that we do not check anymore if pairs of nodes in the initial clustering given by *Selection* should be separated. The intuition explaining the usefulness of this implementation is as follows: if there is a group of latents forming a pure subgraph of the true graph with a large number of pure indicators for each latent, then the initial step should identify such group. The consecutive steps will refine this solution without the risk of splitting the large clusters of variables, which are exactly the ones most likely to produce false positive decisions. RobustBuildPureClusters has the power of identifying the latents with large sets of pure indicators and refining this solution with more flexible rules, covering also cases where DisjointGroup fails.

Notice that the order by which tests are applied might influence the outcome of the algorithms,

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

A.4 The spiritual coping questionnaire

The following questionnaire is provided to facilitate understanding of the religious/spiritural coping example given in Section 3.5.2. It can also serve as an example of how questionnaires are actually designed.

Section I This section intends to measure the level of stress of the subject. In the actual questionnaire, it starts with the following instructions:

Circle the number next to each item to indicate how stressful each of these events has been for you since entered your graduate program. If you have never experienced one of the events listed below, then circle number 1. If one of the events listed below has happened to you and has caused you a great deal of stress, rate that event toward the "Extremely Stressful" end of the rating scale. If an event has happened to you while you have been in graduate school, but has not bothered you at all, rate that event toward the lower end of the scale ("Not at all Stressful").

The student then chooses the level of stress by circling a number on a 7 point scale. The questions of this section are:

1. Fulfilling responsibilities both at home and at school
2. Trying to meet peers of your race/ethnicity on campus
3. Taking exams
4. Being obligated to participate in family functions
5. Arranging childcare
6. Finding support groups sensitive to your needs
7. Fear of failing to meet program expectations
8. Participating in class
9. Meeting with faculty
10. Living in the local community
11. Handling relationships
12. Handling the academic workload
13. Peers treating you unlike the way they treat each other
14. Faculty treating you differently than your peers
15. Writing papers
16. Paying monthly expenses
17. Family having money problems

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

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

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

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

Table A.3: Selects an initial pure model.

18. Adjusting to the campus environment
19. Being obligated to repay loans
20. Anticipation of finding full-time professional work
21. Meeting deadlines for course assignments

Section II This section intends to measure the level of depression of the subject. In the actual questionnaire, it starts with the following instructions:

Below is a list of the ways you might have felt or behaved. Please tell me how often you have felt this way during the past week.

The student then chooses the level of frequency that some events happened to him/her by circling a number on a 4 point scale. The scale is "Rarely or None of the Time (less than 1 day)", "Some or Little of the Time (1 - 2 days)", "Occasionally or a Moderate Amount of the Time (3 - 4 days)" and "Most or All of the Time (5 - 7 days)". The events are as follows:

1. I was bothered by things that usually don't bother me
2. I did not feel like eating; my appetite was poor
3. I felt that I could not shake off the blues even with help from my family or friends
4. I felt that I was just as good as other people
5. I had trouble keeping my mind on what I was doing
6. I felt depressed
7. I felt that everything I did was an effort
8. I felt hopeful about the future
9. I thought my life had been a failure
10. I felt fearful
11. My sleep was restless
12. I was happy
13. I talked less than usual
14. I felt lonely
15. People were unfriendly
16. I enjoyed life
17. I had crying spells
18. I felt sad
19. I felt that people disliked me
20. I could not get "going"

Section III This section intends to measure the level of spiritual coping of the subject. In the actual questionnaire, it starts with the following instructions:

Please think about how you try to understand and deal with major problems in your life. These items ask what you did to cope with your negative event. Each item says something about a particular way of coping. To what extent is your religion or higher power involved in the way you cope?

The student then chooses the level of importance of some spiritual guideline by circling a number on a 4 point scale. The scale is "Not at all", "Somewhat", "Quite a bit", "A great deal". The guidelines are:

1. I think about how my life is part of a larger spiritual force
2. I work together with God (high power) as partners to get through hard times
3. I look to God (high power) for strength, support, and guidance in crises
4. I try to find the lesson from God (high power) in crises
5. I confess my sins and ask for God (high power)'s forgiveness
6. I feel that stressful situations are God (high power)'s way of punishing me for my sins or lack of spirituality
7. I wonder whether God has abandoned me
8. I try to make sense of the situation and decide what to do without relying on God (high power)
9. I question whether God (high power) really exists
10. I express anger at God (high power) for letting terrible things happen
11. I do what I can and put the rest in God (high power)'s hands
12. I do not try much of anything; simply expect God (high power) to take my worries away
13. I pray for a miracle
14. I pray to get my mind off of my problems
15. I ignore advice that is inconsistent with my faith
16. I look for spiritual support from clergy
17. I disagree with what my religion wants me to do or believe
18. I ask God (high power) to help me find a new purpose in life
19. I try to find a completely new life through religion
20. I seek help from God (high power) in letting go of my anger

Appendix B

Results from Chapter 4

All of the following proofs hold with probability 1 with respect to the Lebesgue measure taken over the set of linear coefficients and error variances that partially parameterize the density function of an observed variable given its parents. In all of the following proofs, G is a latent variable graph with a set $\mathbf{O}$ of observable variables. In some of these proofs, we use the term "edge label" as a synonym of the coefficient associated with an edge that is into an observed node. Without loss of generality, we will also assume that all variables have zero mean, unless specified otherwise. The symbol $\{X_t\}$ will stand for a finitely indexed set of variables.

Lemma 4.1 *If for* $\{A, B, C\} \subseteq \mathbf{O}$ *we have* $\rho_{AB} = 0$ *or* $\rho_{AB.C} = 0$*, then* A *and* B *cannot share a common latent parent in* G*.*

Proof: We will prove this argument by contradiction. Assume A and B have a common parent L, i.e., let A, B, C be defined according to the following linear functions

$$\begin{aligned} A &= aL + \textstyle\sum_p a_p A_p + \epsilon_A \\ B &= bL + \textstyle\sum_i b_i B_i + \epsilon_B \\ C &= \textstyle\sum_j c_j C_j + \epsilon_C \end{aligned}$$

where L is a common latent parent of A and B, $\{A_p\}$ represents parents of A, $\{B_i\}$ are parents of B, $\{C_j\}$ parents of C, and $\{a_p\} \cup \{b_i\} \cup \{c_j\} \cup \{a, b, \zeta_A, \zeta_B, \zeta_C\}$ are parameters of the graphical model, $\{\zeta_A, \zeta_B, \zeta_C\}$ being the variances of error terms $\{\epsilon_A, \epsilon_B, \epsilon_C\}$, respectively.

By the equations above, $\sigma_{AB} = ab\sigma_L^2 + K$, where K is a polynomial containing the remaining terms of the respective expression. We will show first that no term in K has a factor ab. For that to happen, either the symbol b appears in some σ_{LB_j}, or the symbol a appears in some σ_{LA_i}, or the symbol ab appears within some $\sigma_{A_pB_i}$. The symbol b will appear in some σ_{LB_j} only if there is path from L to B_j through B, but that cannot happen since B_j is a parent of B and the graph is acyclic beneath the latents. The arguments for a and σ_{LA_i}, and ab with respect to $\sigma_{A_pB_i}$ are analogous.

Consider first that the hypothesis $\rho_{AB} = 0$ is true. With probability 1 with respect to the Lebesgue measure over parameters $\{a_p\} \cup \{b_i\} \cup \{c_j\} \cup \{a, b, \zeta_A, \zeta_B, \zeta_C\}$, the polynomial identity $ab\sigma_L^2 + K = 0$ will hold. For this identity to hold, every term in the polynomial should vanish. Since the only term containing the expression ab is the one given above, we therefore need $ab\sigma_L^2 = 0$. However, by assumption, $ab \neq 0$ and latent variables have positive variance, which contradicts $ab\sigma_L^2 = 0$.

Assume now that $\rho_{AB.C} = 0$. This implies $\sigma_{AB}\sigma_C^2 - \sigma_{AC}\sigma_{BC} = 0$ where $\sigma_C^2 > 0$ by assumption. By expressing $\sigma_{AB}\sigma_C^2$ as a function of the given coefficients, we obtain $ab\sigma_L^2\sigma_C^2 + Q$, where Q is a polynomial that does not contain any term that includes some symbol in $\{c_j\} \cup \zeta_C$ (using arguments analogous to the previous case). Since C is not an ancestor of L (because L is latent) no term in $ab\sigma_L^2$ contains the symbol ζ_C, nor any coefficient $\{c_j\}$. Since every term in $\sigma_{AC}\sigma_{BC}$ that might contain ζ_C must also contain some $\{c_j\}$, then no term in $\sigma_{AC}\sigma_{BC}$ can cancel any term in $ab\sigma_L^2\zeta_C$ (which is contained in $ab\sigma_C^2\sigma_C^2$). This implies $ab\sigma_L^2\zeta_C = 0$, a contradiction. □

Lemma 4.2 *For any set* $\mathbf{O}' = \{A, B, C, D\} \subseteq \mathbf{O}$*, if* $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BC}$ *such that for all triplets* $\{X, Y, Z\}$, $\{X, Y\} \subset \mathbf{O}', Z \in \mathbf{O}$*, we have* $\rho_{XY.Z} \neq 0$ *and* $\rho_{XY} \neq 0$*, then no element in* $X \in \mathbf{O}'$ *is an ancestor of any element in* $\mathbf{O}' \backslash X$ *in* G*.*

Since G is acyclic among observed variables, 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:

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

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, another indicators or, for now, the respective error term, but each indicator has at least one latent parent besides the error term. Let $\mathbf{L}$ be the set of latent variables in G. 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 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:

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

where on the right-hand side of each equation we have the respective immediate latent ancestors of A, B, C and D and λ parameters are functions of the original coefficients of the measurement model. Notice that in general the sets of immediate latent ancestors for each pair of elements in $\mathbf{O}'$ will overlap.

Since the graph is acyclic, at least one element of $\{A, B, C\}$ is not an ancestor of the other two. By symmetry, assume without loss of generality that C is such a node. Assume also C is an ancestor of D. We will prove by contradiction that this is not possible. 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

$$C = cL + \sum_j \lambda_{C_j} L_{C_j} \tag{B.1}$$

where by an abuse of notation we are keeping the same symbols λ_{C_j} and L_{C_j} to represent the other dependencies of C. Notice that it is possible that $L = L_{C_j}$ for some L_{C_j} if there is more

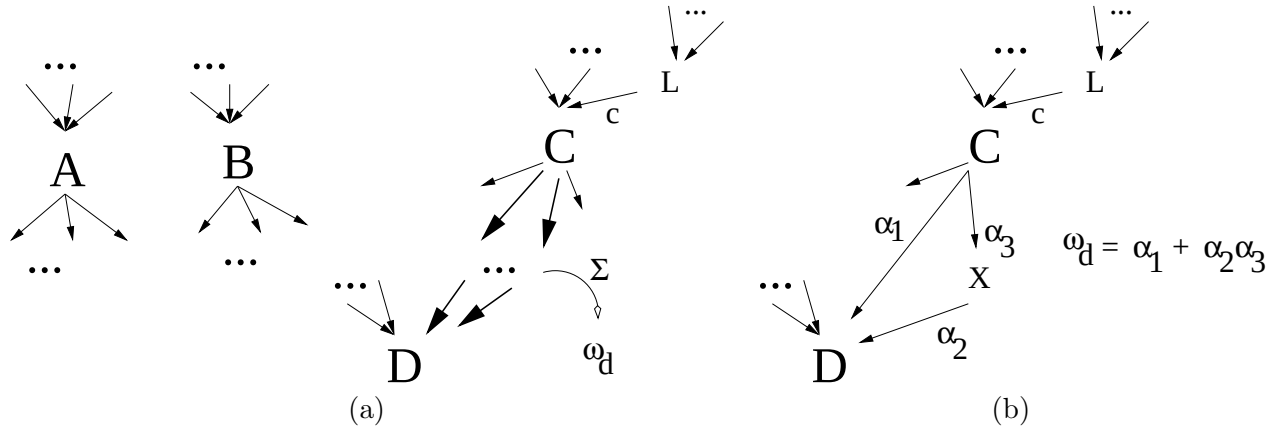


Figure B.1: (a) The symbol ω_d is defined as the sum over all directed paths from C to D of the product of the labels of each edge that appears in each path. Here the larger edges represent edges in such directed paths. (b) An example: we have two directed paths from C to D. The symbol ω_d then stands 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).

than one directed path from L to C, but this will not be relevant for our proof. In this case, the corresponding coefficient λ is modified by subtracting c. It should be stressed that the symbol c does not appear anywhere in the polynomial corresponding to $\sum_j \lambda_{C_j} L_{C_j}$, where in this case the variables of the polynomial are the original coefficients parameterizing the measurement model and the immediate latent ancestors of C.

By another abuse of notation, rewrite A, B and D as

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

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 B.1. The possible corresponding $\lambda_{X_{v_t}}$ coefficient for L is adjusted in the summation by subtracting $c\omega_{X_{v_t}}$ (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 in 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 factorize the expression corresponding to the tetrad constraints. Notice that, by assumption, $\omega_a = \omega_b = 0$, and $\omega_d \neq 0$. We keep ω_a and ω_b in our equations to account for the next cases, where we will prove that B and A cannot be ancestors of D. The reasoning will be analogous, but the respective ωs will be nonzero.

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:

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

Consider the polynomial identity $\sigma_{AB}\sigma_{CD} - \sigma_{AC}\sigma_{BD} = 0$ as a function of the parameters of the measurement model, i.e., the linear coefficients and error variances for the observed variables. Assume this constraint is entailed by G and its unknown latent covariance matrix. With a Lebesgue measure over the parameters, this will hold with probability 1, which follows from the fact that the solution set to non-trivial polynomial constraints has measure zero. See Meek (1997) and references within for more details. This also means that every term in this polynomial expression should vanish to zero with probability 1: i.e., the coefficients (functions of the latent covariance matrix) of every term in the polynomial should be zero. Therefore, the sum of all terms with a factor $\omega_{dt} = l_1 l_2 ... l_z$ at a given choice of exponents for each $l_1, ..., l_z$ should be 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, say, $c^2\omega_{dt}$, for some ω_{dt}. This can include elements from any term that will explicitly show $c^2\omega_d$ when multiplying the covariance equations above among others, but we have to consider the multiplicity of the factors that compose ω_{dt}. Let $\omega_{dt} = l_1 l_2 ... l_z$. We want to factorize our tetrad constraint according to terms that contain $l_1 l_2 ... l_z$ with multiplicity 1 for each label (i.e., our terms cannot include l_1^2, for instance, or some subset of $\{l_1, ..., l_z\}$). Since C does not have some descendant X that is a common ancestor of A and D or B and D, this means that no algebraic term ω_a, ω_b or $\lambda_{A_p}, \lambda_{B_i}$ can contain some symbol in $\{l_1, ..., l_z\}$. Notice that some λ_{D_k}s will be functions of ω_{dt}: every immediate latent ancestor of C is an immediate latent ancestor of D. Therefore, for each common immediate 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 (B.1). As discussed before, no factor of ω_{dt} can be a factor of any term in λ_{B_i}. The same holds for ω_a. Therefore, the multiplicity of each $l_1, ..., l_z$ in this term is exactly 1.

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 that can be factorized by $c^2\omega_{dt}$, where the power of c in such terms is 2, and the multiplicity of each $l_1, ..., l_z$ is 1. Since this has to be identically zero and $\omega_{dt} \neq 0$, we have the following relation:

$$f_1(G) = f_2(G) \tag{B.2}$$

where

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

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

Similarly, when we factorize terms that include $c\omega_{dt}$, where the respective powers of $c, l_1, ..., l_z$ in the term have to be 1, we get the following expression as an additive term of $\sigma_{AB}\sigma_{CD} - \sigma_{AC}\sigma_{BD}$:

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

for which we have:

$$g_1(G) = g_2(G) \tag{B.3}$$

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:

$$h_1(G) = h_2(G) \tag{B.4}$$

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 $\sigma_C^2\sigma_{AB}$ will result in:

$$\sigma_{AC}\sigma_{BC} = P(G) + f_2(G) + g_2(G) + h_2(G) \tag{B.5}$$

$$\sigma_C^2 \sigma_{AB} = P(G) + f_1(G) + g_1(G) + h_1(G) \quad \text{(B.6)}$$

where

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

By (B.2), (B.3), (B.4), (B.5) and (B.6), we have:

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

Contradiction. Therefore, C cannot be an ancestor of D, and more generally, of any element in $\mathbf{O}'\backslash C$.

Assume without loss of generality that B is not an ancestor of A. C is not an ancestor of any element in $\mathbf{O}'\backslash C$. If B does not have a descendant that is a common ancestor of C and D, then by analogy with the (C, D) case (where now more than one ω element will be nonzero as hinted before, since we have to consider the possibility of B being an ancestor of both C and D), B cannot be an ancestor of C nor D.

Assume then that B has a descendant X that is a common ancestor of C and D, where $X \neq C$ and $X \neq D$, since C is not an ancestor of D and vice-versa. Notice also that X is not an ancestor of A, since B is not an ancestor of A. Relations such as Equation B.2 might not hold, since we might be equating terms that have different exponents for symbols in $\{l_1, ..., l_z\}$. However, since now we have an observed intermediate term X, we can make use of its error variance parameter ζ_X corresponding to the error term ϵ_X.

No term in σ_{AB} can have ζ_X, since ϵ_X is independent of both A and B. There is at least one term in σ_{CD} that contains ζ_X as a factor. There is no term in σ_{AC} that contains ζ_X as a factor, since ϵ_X is independent of A. There is no term in σ_{BD} that contains ζ_X as a factor, since ϵ_X is independent of B. Therefore, in $\sigma_{AB}\sigma_{CD}$ we have at least one term that has ζ_X, while no term in $\sigma_{AC}\sigma_{BD}$ contains such term. That requires some parameters or the variance of some latent ancestor of B to be zero, which is a contradiction.

Therefore, B is not an ancestor of any element in $\mathbf{O}'\backslash B$. In a completely analogous way, one can show that A is not an ancestor of any element in $\mathbf{O}'\backslash A$. □

The following lemma will be useful to proof Lemma 4.2:

Lemma B.1 *For any set $\{A, B, C, D\} = \mathbf{O}' \subseteq \mathbf{O}$, if $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BD}$ such that for every set $\{X, Y\} \subset \mathbf{O}', Z \in \mathbf{O}$ we have $\rho_{XY.Z} \neq 0$ and $\rho_{XY} \neq 0$, then no pair of elements in $\mathbf{O}'$ has an observed common ancestor.*

Proof: Assume for the sake of contradiction that some pair in $\mathbf{O}'$ has an observed common ancestor. Let K be a common ancestor of some pair of elements in $\mathbf{O}'$ such that no descendant of K is also a common ancestor of some pair in $\mathbf{O}'$.

Without loss of generality, assume K is a common ancestor of A and B. Let α be the concatenation of edge labels in some directed path from K to A, and β the concatenation of edge labels in some directed path from K to B. That is,

$$\begin{aligned} A &= \alpha K + R_A \\ B &= \beta K + R_B \end{aligned}$$

where R_X is the remainder of the polynomial expression that describes node X as a function of its immediate latent ancestors and K.

By the given constraint $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD}$, we have $\alpha\beta(\sigma_K^2\sigma_{CD} - \sigma_{CK}\sigma_{DK}) + f(G) = 0$, where

$$f(G) = (\alpha\sigma_{KR_B} + \beta\sigma_{KR_A} + \sigma_{R_AR_B})\sigma_{CD} - \sigma_{CR_A} - \sigma_{DR_B}$$

However, no term in $f(G)$ can contain the symbol $\alpha\beta$: by Lemma 4.2 no element X in $\mathbf{O'}$ can be an ancestor of any element in $\mathbf{O'}\backslash X$; also, by construction no descendant of K (with the possible exception of K) can be an ancestor of C or D and therefore no sequence α or β can be generated from the polynomial f that is a function of $\sigma_{KR_B}, \sigma_{KR_A}, \sigma_{R_AR_B}, \sigma_{CD}, \sigma_{CR_A}$ or σ_{DR_B}.

It follows that with probability 1 we have $\alpha\beta(\sigma_K^2\sigma_{CD} - \sigma_{CK}\sigma_{DK}) = 0$, and since $\alpha\beta \neq 0$ by assumption, this implies $\sigma_K^2\sigma_{CD} - \sigma_{CK}\sigma_{DK} = 0 \Rightarrow \rho_{CD.K} = 0$. Contradiction. □

Lemma 4.3 *For any set* $\mathbf{O'} = \{X_1, X_2, Y_1, Y_2\} \subseteq \mathbf{O}$*, if* $Factor_1(X_1, X_2, G) = true$*,* $Factor_1(Y_1, Y_2, G) = true$*,* $\sigma_{X_1Y_1}\sigma_{X_2Y_2} = \sigma_{X_1Y_2}\sigma_{X_2Y_1}$*, and all elements of* $\{X_1, X_2, Y_1, Y_2\}$ *are correlated, then no element in* $\{X_1, X_2\}$ *is an ancestor of any element in* $\{Y_1, Y_2\}$ *in* G *and vice-versa.*

Proof: Assume for the sake of contradiction that X_1 is an ancestor of Y_1. Let P be an arbitrary directed path from X_1 to Y_1 of K edges such that the edge coefficients on this path are $\alpha_1 \dots \alpha_K$. One can write the covariance of X_1 and Y_1 as $\sigma_{X_1Y_1} = c\alpha_1\sigma_{X_1}^2 + F(G)$, where $F(G)$ is a polynomial (in terms of edge coefficients and error variances) that does not contain any term that includes the symbol α_1, and $c = \alpha_2 \dots \alpha_K$. Also, the polynomial corresponding to $\sigma_{X_1}^2$ cannot contain any term that includes the symbol α_1.

Also analogously, $\sigma_{X_2Y_1}$ can be written as $c\alpha_1\sigma_{X_1X_2} + F'(G)$, where $F'(G)$ does not contain α_1, since X_1 cannot be an ancestor of X_2 by the given hypothesis and Lemma 4.2.

By Lemma 4.2 and the given conditions, Y_2 cannot be an ancestor of Y_1 and therefore, not an ancestor of X_1. X_1 cannot be an ancestor of Y_2, by Lemma B.1 applied to pair $\{Y_1, Y_2\}$. This implies that $\sigma_{X_1Y_2}$ cannot contain any term that includes α_1. By the same reason, the polynomial corresponding to $\sigma_{X_2Y_2}$ cannot contain any term that includes α_1.

This means that the constraint $\sigma_{X_1Y_1}\sigma_{X_2Y_2} = \sigma_{X_1Y_2}\sigma_{X_2Y_1}$ corresponds to the polynomial identity $\alpha_1(\sigma_{X_1}^2\sigma_{X_2Y_2} - \sigma_{X_1Y_2}\sigma_{X_1X_2}) + F''(G) = 0$, where the polynomial $F''(G)$ does not contain any term that includes α_1, and neither does any term in the factor $(\sigma_{X_1}^2\sigma_{X_2Y_2} - \sigma_{X_1Y_2}\sigma_{X_1X_2})$. This will imply with probability 1 that $\sigma_{X_1}^2\sigma_{X_2Y_2} - \sigma_{X_1Y_2}\sigma_{X_1X_2} = 0$ (which is the same of saying that the partial correlation of X_2 and Y_2 given X_1 is zero).

The expression $\sigma_{X_1}^2\sigma_{X_2Y_2}$ contains a term that include ζ_{X_1}, the error variance for X_1, while $\sigma_{X_1Y_2}\sigma_{X_1X_2}$ cannot contain such a term, since X_1 is not an ancestor of either X_2 or Y_2. That will then imply the term $\zeta_{X_1}\sigma_{X_2Y_2}$ should vanish, which is a contradiction since $\zeta_{X_1} \neq 0$ by assumption and $\sigma_{X_2Y_2} \neq 0$ by hypothesis. □

Let $X = \lambda_{x0}L + \sum_{i=1}^{k} \lambda_{xi}\eta_i$ and Y be random variables with zero mean, as well as $\{L, \eta_1, ..., \eta_k\}$. Let $\{\lambda_{x0}, \lambda_{x_1}, ..., \lambda_{x_k}\}$ be real coefficients. We define σ_{XYL}, the "covariance of X and Y through L", as $\sigma_{XYL} \equiv \lambda_{x0}E[LY]$. The following lemma will be useful to show Lemma 4.4:

Lemma B.2 *Let* $\{A, B, C, D\} \subset \mathbf{O}$ *such that* A *is not an ancestor of* B*,* C *or* D *in* G *and* A *has a parent* L *in* G*, and no element of the covariance matrix of* A, B, C *and* D *is zero. If* $\sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BC}$*, then* $\sigma_{ACL} = \sigma_{ADL} = 0$ *or* $\sigma_{ACL}/\sigma_{ADL} = \sigma_{AC}/\sigma_{AD} = \sigma_{BC}/\sigma_{BD}$*.*

Proof: 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, error terms included.

Given the assumptions, we have:

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

Since A is not an ancestor of B, C or D, there is no trek among elements of $\{B, C, D\}$ containing both L and A, and therefore the symbol a cannot appear in $\sum_p \sum_j a_p c_j \sigma_{A_p C_j}\sigma_{BD} - \sum_p \sum_k a_p d_k \sigma_{A_p D_k}\sigma_{BC}$ 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:

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

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 4.4 *CS1 is sound.*

Proof: 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 4.2 and the given constraints, an element of $\{X_1, Y_1\}$ cannot be an ancestor of the other, and neither can be an ancestor in G of any element in $\{X_2, X_3, Y_2, Y_3\}$. By definition, $\sigma_{X_1 VL} = (a/b)\sigma_{Y_1 VL}$ for some variable V, and therefore $\sigma_{X_1 VL} = 0 \Leftrightarrow \sigma_{Y_1 VL} = 0$. Assume $\sigma_{Y_1 X_2 L} = \sigma_{X_1 X_2 L} = 0$. Since it is given that $\sigma_{X_1 Y_1}\sigma_{X_2 X_3} = \sigma_{X_1 X_2}\sigma_{Y_1 X_3}$, by Lemma B.2 we have $\sigma_{X_1 Y_1 L} = \sigma_{X_1 X_2 L} = 0$. Since $\sigma_{X_1 Y_1 L} = ab\sigma_L^2 + K$, where no term in K contains the factor ab, then if $\sigma_{X_1 Y_1 L} = 0$, with probability 1 $ab\sigma_L^2 = 0 \Rightarrow \sigma_L^2 = 0$, which is a contradiction of the assumptions. By repeating the argument, no element in $\{\sigma_{X_1 X_2 L}, \sigma_{X_1 X_3 L}, \sigma_{Y_1 X_2 L}, \sigma_{Y_1 X_3 L}, \sigma_{X_1 Y_2 L}, \sigma_{X_1 Y_3 L}, \sigma_{Y_1 Y_2 L}, \sigma_{Y_1 Y_3 L}\}$ is zero. Therefore, since $\sigma_{X_1 Y_1}\sigma_{X_2 X_3} = \sigma_{X_1 X_2}\sigma_{X_3 Y_1} = \sigma_{X_1 X_3}\sigma_{X_2 Y_1}$ by assumption, from Lemma B.2 we have

$$\frac{\sigma_{X_1 X_3}}{\sigma_{X_3 Y_1}} = \frac{\sigma_{X_1 X_3 L}}{\sigma_{X_3 Y_1 L}} \tag{B.7}$$

and from $\sigma_{X_1 Y_1}\sigma_{Y_2 Y_3} = \sigma_{X_1 Y_2}\sigma_{Y_1 Y_3} = \sigma_{X_1 Y_3}\sigma_{Y_1 Y_2}$

$$\frac{\sigma_{Y_1 Y_3}}{\sigma_{X_1 Y_3}} = \frac{\sigma_{Y_1 Y_3 L}}{\sigma_{X_1 Y_3 L}} \tag{B.8}$$

Since no covariance among the given variables is zero,

$$\begin{array}{rcll} \dfrac{\sigma_{X_1X_2}\sigma_{Y_1X_3}}{\sigma_{X_1Y_2}\sigma_{Y_1Y_3}} & = & \dfrac{\sigma_{X_1X_3}\sigma_{Y_1X_2}}{\sigma_{X_1Y_3}\sigma_{Y_1Y_2}} & \Rightarrow \\ \sigma_{X_1X_2}\sigma_{Y_1Y_2} & = & \sigma_{X_1Y_2}\sigma_{Y_1X_2}\dfrac{\sigma_{X_1X_3}\sigma_{Y_1Y_3}}{\sigma_{Y_1X_3}\sigma_{X_1Y_3}} & \end{array}$$

From (B.7), (B.8) it follows:

$$\begin{array}{rcl} \sigma_{X_1X_2}\sigma_{Y_1Y_2} & = & \sigma_{X_1Y_2}\sigma_{Y_1X_2}\dfrac{\sigma_{X_1X_3L}\sigma_{Y_1Y_3L}}{\sigma_{Y_1X_3L}\sigma_{X_1Y_3L}} \\ & = & \sigma_{X_1Y_2}\sigma_{Y_1X_2}\dfrac{(a/b)\sigma_{Y_1X_3L}(b/a)\sigma_{X_1Y_3L}}{\sigma_{Y_1X_3L}\sigma_{X_1Y_3L}} \\ & = & \sigma_{X_1Y_2}\sigma_{Y_1X_2} \end{array}$$

Contradiction. □

Lemma 4.5 *CS2 is sound.*

Proof: Suppose X_1 and Y_1 have a common parent L in G. Let $X_1 = aL + \sum_p a_pA_p$ and $Y_1 = bL + \sum_p b_iB_i$. To simplify the presentation, we will represent $\sum_p a_pA_p$ by random variable P_x and $\sum_p b_iB_i$ by P_y, such that $X_1 = aL + P_x$ and $Y_1 = bL + P_y$. We will assume that $E[P_xP]$ and $E[P_yP]$ are not zero, for $P \in \{X_1, X_2, Y_1, Y_2\}$ to simplify the proof, but the same results can be obtained without this condition in an analogous (and simpler) way.

With probability 1 with respect to a Lebesgue measure over the linear coefficients parameterizing the graph, the constraint $\sigma_{X_1Y_1}\sigma_{X_2Y_2} - \sigma_{X_1Y_2}\sigma_{X_2Y_1} = 0$ corresponds to a polynomial identity where some terms contain the product ab, some contain only a, some contain only b, and some contain none of such symbols. Since this is a polynomial identity, all terms containing ab should sum to zero. The same holds for terms containing only a, only b and not containing a or b. This constraint can be rewritten as

$$\begin{array}{ll} ab(E[L^2]\sigma_{X_2Y_2} - E[LY_2]E[LX_2]) & + \\ a(E[LP_y]\sigma_{X_2Y_2} - E[LY_2]E[X_2P_y]) & + \\ b(E[LP_x]\sigma_{X_2Y_2} - E[Y_2P_x]E[LX_2]) & + \\ (E[P_xP_y]\sigma_{X_2Y_2} - E[P_xY_2]E[P_yX_2]) & \end{array}$$

From Lemmas 4.2 and 4.3 and the given hypothesis, X_1 cannot be an ancestor of any element of $\{X_2, Y_1, Y_2\}$ and Y_1 cannot be an ancestor of any element in $\{X_1, X_2, Y_2\}$. Therefore, the symbols a and b cannot appear inside any of the polynomial expressions obtained when terms such as $\sigma_{X_2Y_2}$ or $E[Y_2P_x]$ are expressed as functions of the latent covariance matrix and the linear coefficients and error variances of the measurement model. All symbols a and b of $\sigma_{X_1Y_1}\sigma_{X_2Y_2} - \sigma_{X_1Y_2}\sigma_{X_2Y_1}$ were therefore factorized as above. Therefore, with probability 1 we have:

$$E[L^2]\sigma_{X_2Y_2} = E[LX_2]E[LY_2] \tag{B.9}$$

$$E[LP_y]\sigma_{X_2Y_2} = E[LY_2]E[X_2P_y] \tag{B.10}$$

$$E[LP_x]\sigma_{X_2Y_2} = E[Y_2P_x]E[LX_2] \tag{B.11}$$

$$E[P_x P_y]\sigma_{X_2Y_2} = E[Y_2 P_x]E[X_2 P_Y] \tag{B.12}$$

Analogously, the constraint $\sigma_{X_2Y_1}\sigma_{Y_2Y_3} - \sigma_{X_2Y_3}\sigma_{Y_2Y_1} = 0$ will force other identities. Since Y_1 is also not an ancestor of Y_3, we can split the polynomial expression derived from $\sigma_{X_2Y_1}\sigma_{Y_2Y_3} - \sigma_{X_2Y_3}\sigma_{Y_2Y_1} = 0$ into two parts

$$b\{E[LX_2]\sigma_{Y_2Y_3} - E[LY_2]\sigma_{X_2Y_3}\} + \{E[X_2P_Y]\sigma_{Y_2Y_3} - E[Y_2P_Y]\sigma_{X_2Y_3}\} = 0$$

where the second component, $E[X_2P_Y]\sigma_{Y_2Y_3} - E[Y_2P_Y]\sigma_{X_2Y_3}$, cannot contain any term that includes the symbol b, and neither can the second factor of the first component, $E[LX_2]\sigma_{Y_2Y_3} - E[LY_2]\sigma_{X_2Y_3}$. With probability 1, it follows that:

$$\begin{array}{rcl} E[LX_2]\sigma_{Y_2Y_3} & = & E[LY_2]\sigma_{X_2Y_3} \\ E[X_2P_Y]\sigma_{Y_2Y_3} & = & E[Y_2P_Y]\sigma_{X_2Y_3} \end{array}$$

Since we have that $\sigma_{Y_2Y_3} \neq 0$ and $\sigma_{X_2Y_3} \neq 0$, from the two equations above, we get:

$$E[LX_2]E[Y_2P_Y] = E[LY_2]E[X_2P_Y] \tag{B.13}$$

From the constraint $\sigma_{X_1X_2}\sigma_{X_3Y_2} = \sigma_{X_1Y_2}\sigma_{X_3X_2}$ and a similar reasoning, we get

$$E[LX_2]E[Y_2P_X] = E[LY_2]E[X_2P_X] \tag{B.14}$$

from which follows

$$E[X_2P_X]E[Y_2P_Y] = E[X_2P_Y]E[Y_2P_X] \tag{B.15}$$

Combining (B.10) and (B.13), we have

$$aE[LP_y]\sigma_{X_2Y_2} = aE[LX_2]E[Y_2P_Y] \tag{B.16}$$

Combining (B.11) and (B.14), we have

$$bE[LP_x]\sigma_{X_2Y_2} = bE[X_2P_X]E[LY_2] \tag{B.17}$$

Combining (B.12) and (B.15), we have

$$E[P_xP_y]\sigma_{X_2Y_2} = E[X_2P_X]E[Y_2P_Y] \tag{B.18}$$

From (B.9), (B.16), (B.17) and (B.18) and the given constraints:

$\sigma_{X_1X_2}\sigma_{Y_1Y_2} = abE[LX_2]E[LY_2] + aE[LX_2]E[Y_2P_x] + bE[X_2P_x]E[LY_2] + E[X_2P_X]E[Y_2P_Y] = abE[L^2]\sigma_{X_2Y_2} + E[LP_y]\sigma_{X_2Y_2} + E[LP_y]\sigma_{X_2Y_2} + E[P_xP_y]\sigma_{X_2Y_2} = \sigma_{X_1Y_1}\sigma_{X_2Y_2} = \sigma_{X_1Y_2}\sigma_{X_2Y_1}$

Contradiction. □

Theorem 4.6 *There are sound identification rules that allow one to learn if two observed variables share a common parent in a linear latent variable model that are not sound for non-linear latent variable models.*

Proof: Consider first the following test: let $G(\mathbf{O})$ be a linear latent variable model. Assume $\{X_1, X_2, X_3, Y_1, Y_2, Y_3\} \subseteq \mathbf{O}$ and $\sigma_{X_1Y_1}\sigma_{Y_2Y_3} = \sigma_{X_1Y_2}\sigma_{Y_1Y_3} = \sigma_{X_1Y_3}\sigma_{Y_1Y_2}$, $\sigma_{X_1Y_2}\sigma_{X_2X_3} = \sigma_{X_1X_2}\sigma_{Y_2X_3} = \sigma_{X_1X_3}\sigma_{X_2Y_2}$, $\sigma_{X_1Y_3}\sigma_{X_2X_3} = \sigma_{X_1X_2}\sigma_{Y_3X_3} = \sigma_{X_1X_3}\sigma_{X_2Y_3}$, $\sigma_{X_1X_2}\sigma_{Y_2Y_3} \neq \sigma_{X_1Y_2}\sigma_{X_2Y_3}$ and that for all triplets $\{A, B, C\}, \{A, B\} \subset \{X_1, X_2, X_3, Y_1, Y_2, Y_3\}, C \in \mathbf{O}$, we have $\rho_{AB} \neq 0, \rho_{AB.C} \neq 0$. Then X_1 and Y_1 do not have a common parent in G.

Call this test CS3. Test CS3 is sound for linear models: if its conditions are true, then X_1 and Y_1 do not have a common parent in G. The proof of this result is given by Silva et al. (2005). However, this is not a sound rule for the non-linear case. To show this, it is enough to come up with a latent variable model where X_1 and Y_1 have a common parent, and a latent covariance matrix such that, for any choice of linear coefficients and error variances, this test applies. Notice that the definition of a sound identification rule in non-linear graphs allows us to choose specific latent covariance matrices but the constraints should hold for any choice of linear coefficients and error variances (or, more precisely, with probability 1 with respect to the Lebesgue measure).

Consider the graph G with five latent variables $L_i, 1 \leq i \leq 5$, where L_1 has X_1 and Y_1 as its only children, X_2 is the only child of L_2, X_3 is the only child of L_3, Y_2 is the only child of L_4 and Y_3 is the only child of L_5. Also, $\{X_1, X_2, X_3, Y_1, Y_2, Y_3\}$, as defined in CS3, are the only observed variables, and each observed variable has only one parent besides its error term. Error variables are independent.

The following simple randomized algorithm will choose a covariance matrix Σ_L for $\{L_1, L_2, L_3, L_4, L_5\}$ that entails CS3. The symbol σ_{ij} will denote the covariance of L_i and L_j.

1. Choose positive random values for all $\sigma_{ii}, 1 \leq i \leq 5$
2. Choose random values for σ_{12} and σ_{13}
3. $\sigma_{23} \leftarrow \sigma_{12}\sigma_{13}/\sigma_{11}$
4. Choose random values for σ_{45}, σ_{25} and σ_{24}
5. $\sigma_{14} \leftarrow \sigma_{12}\sigma_{45}/\sigma_{25}$
6. $\sigma_{15} \leftarrow \sigma_{12}\sigma_{45}/\sigma_{24}$
7. $\sigma_{35} \leftarrow \sigma_{13}\sigma_{45}/\sigma_{14}$
8. $\sigma_{34} \leftarrow \sigma_{12}\sigma_{45}/\sigma_{15}$
9. Repeat from the beginning if Σ_L is not positive definite or if $\sigma_{14}\sigma_{23} = \sigma_{12}\sigma_{34}$

Table B.1 provides an example of such matrix. Notice that the intuition behind this example is to set the covariance matrix of the latent variables to have some vanishing partial correlations, even though one does not necessarily have any conditional independence. For linear models, both conditions are identical, and therefore this identification rule holds in such a case. □.

Lemma B.3 *For any set $\{A, B, C, D\} = \mathbf{O}' \subseteq \mathbf{O}$, if $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BD}$ such that for every set $\{X, Y\} \subset \mathbf{O}', Z \in \mathbf{O}$ we have $\rho_{XY.Z} \neq 0$ and $\rho_{XY} \neq 0$, then A and B do not have more than one common immediate latent ancestor in G.*

L_1	L_2	L_3	L_4	L_5
1.0				
0.4636804781967626	1.0			
0.31177237495755117	0.1445627639088577	1.0		
0.8241967922523632	0.6834605230188671	0.45954945371001815	1.0	
0.5167659523766029	0.428525239857415	0.28813447630828753	0.7617079965565864	1.0

Table B.1: A counterexample that can be used to prove Theorem 4.6.

Proof: Assume for the sake of contradiction that L_1 and L_2 are two common immediate latent ancestors of A and B in G. Let the structural equations for A, B, C and D be:

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

where α_1 is a sequence of labels of edges corresponding to some directed path connecting L_1 and A. Symbols α_2, β_1, β_2 are defined analogously. R_X is the remainder of the polynomial expression that describes node X as a function of its parents and the immediate latent ancestors L_1 and L_2.

Since the constraint $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD}$ is observed, we have $\sigma_{AB}\sigma_{CD} - \sigma_{AC}\sigma_{BD} = 0 \Rightarrow$ $(\alpha_1\beta_1\sigma^2_{L_1} + \alpha_1\beta_2\sigma_{L_1L_2} + \alpha_2\beta_1\sigma_{L_1L_2} + \alpha_2\beta_2\sigma^2_{L_2} + \alpha_1\sigma_{L_1R_B} + \alpha_2\sigma_{L_2R_B} + \beta_1\sigma_{L_1R_A} + \beta_2\sigma_{L_2R_A} + \sigma_{R_AR_B})\sigma_{CD} - (\alpha_1 \sum_j c_j\sigma_{C_jL_1} + \alpha_2 \sum_j c_j\sigma_{C_jL_2} + \sum_j c_j\sigma_{C_jR_A})$ $(\beta_1 \sum_k d_k\sigma_{D_kL_1} + \beta_2 \sum_k d_k\sigma_{D_kL_2} + \sum_k d_k\sigma_{D_kR_B})) = 0 \Rightarrow \alpha_1\beta_1(\sigma^2_{L_1}\sigma_{CD} - (\sum_j c_j\sigma_{C_jL_1})$ $(\sum_k d_k\sigma_{D_kL_1})) + f(G) = 0$, where

$$\begin{aligned} f(G) = {} & (\alpha_1\beta_2\sigma_{L_1L_2} + \alpha_2\beta_1\sigma_{L_1L_2} + \alpha_2\beta_2\sigma^2_{L_2} \\ & \alpha_1\sigma_{L_1R_B} + \alpha_2\sigma_{L_2R_B} + \beta_1\sigma_{L_1R_A} + \beta_2\sigma_{L_2R_A} + \sigma_{R_AR_B})\sigma_{CD} - \\ & \alpha_1 \textstyle\sum_j c_j\sigma_{C_jL_1}(\beta_2 \sum_k d_k\sigma_{D_kL_2} + \sum_k d_k\sigma_{D_kR_B})) - \\ & \alpha_2 \textstyle\sum_j c_j\sigma_{C_jL_2}(\beta_1 \sum_k d_k\sigma_{D_kL_1} + \beta_2 \sum_k d_k\sigma_{D_kL_2} + \sum_k d_k\sigma_{D_kR_B})) - \\ & \textstyle\sum_j c_j\sigma_{C_jR_A}(\beta_1 \sum_k d_k\sigma_{D_kL_1} + \beta_2 \sum_k d_k\sigma_{D_kL_2} + \sum_k d_k\sigma_{D_kR_B})) \end{aligned}$$

No element in $\mathbf{O}'$ is an ancestor of any other element in this set (Lemma 4.2) and no observed node in any directed path from $L_i \in \{L_1, L_2\}$ to $X \in \{A, B\}$ can be an ancestor of any node in $\mathbf{O}'\backslash X$ (Lemma B.1). That is, when fully expanding $f(G)$ as a function of the linear parameters of G, the product $\alpha_1\beta_1$ cannot possibly appear.

Therefore, since with probability 1 the polynomial constraint is identically zero and nothing in $f(G)$ can cancel the term $\alpha_1\beta_1$, we have:

$$\sigma^2_{L_1}\sigma_{CD} = \sum_j c_j\sigma_{C_jL_1} \sum_k d_k\sigma_{D_kL_1} \tag{B.19}$$

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

$$\sigma_{L_1L_2}\sigma_{CD} = \sum_j c_j\sigma_{C_jL_1} \sum_k d_k\sigma_{D_kL_2} \tag{B.20}$$

$$\sigma_{L_1L_2}\sigma_{CD} = \sum_j c_j \sigma_{C_jL_2} \sum_k d_k \sigma_{D_kL_1} \tag{B.21}$$

$$\sigma^2_{L_2}\sigma_{CD} = \sum_j c_j \sigma_{C_jL_2} \sum_k d_k \sigma_{D_kL_2} \tag{B.22}$$

From (B.19),(B.20), (B.21), (B.22), 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 $\sigma_{CD} - \sigma_{AC}\sigma_{AD}(\sigma^2_A)^{-1} = 0 \Rightarrow \rho_{CD.A} = 0$. Contradiction. □

Lemma B.4 *For any set $\{A, B, C, D\} = \mathbf{O}' \subseteq \mathbf{O}$, if $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BD}$ such that for every set $\{X, Y\} \subset \mathbf{O}', Z \in \mathbf{O}$ we have $\rho_{XY.Z} \neq 0$ and $\rho_{XY} \neq 0$, then if A and B have a common immediate latent ancestor L_1 in G, B and C have a common immediate latent ancestor L_2 in G, we have $L_1 = L_2$.*

Proof: Assume A, B and C are parameterized as follows:

$$\begin{array}{rcl}
A & = & aL_1 + \sum_p a_p A_p \\
B & = & b_1L_1 + b_2L_2 + \sum_i b_i B_i \\
C & = & cL_2 + \sum_j c_j C_j
\end{array}$$

where as before $\{A_p\} \cup \{B_i\} \cup \{C_j\}$ represents the possible other parents of A, B and C, respectively. Assume $L_1 \neq L_2$. We will show that $\rho_{L_1L_2} = 1$, which is a contradiction. From the given constraint $\sigma_{AB}\sigma_{CD} = \sigma_{AD}\sigma_{BC}$, and the fact that from Lemma 4.2 we have that for no pair $\{X, Y\} \subset \mathbf{O}'$ X is an ancestor of Y, if we factorize the constraint according to which terms include ab_1c as a factor, we obtain with probability 1:

$$ab_1c[\sigma^2_{L_1}\sigma_{L_2D} - \sigma_{L_1D}\sigma_{L_1L_2}] \tag{B.23}$$

If we factorize such constraint according to ab_2c, it follows:

$$ab_2c[\sigma_{L_1L_2}\sigma_{L_2D} - \sigma_{L_1D}\sigma^2_{L_2}] \tag{B.24}$$

From (B.23) and (B.24), it follows that $\sigma^2_{L_1}\sigma^2_{L_2} = (\sigma_{L_1L_2})^2 \Rightarrow \rho_{L_1L_2} = 1$. Contradiction. □

Lemma B.5 *For any set $\{A, B, C, D\} = \mathbf{O}' \subseteq \mathbf{O}$, if $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BD}$ such that for every set $\{X, Y\} \subset \mathbf{O}', Z \in \mathbf{O}$ we have $\rho_{XY.Z} \neq 0$ and $\rho_{XY} \neq 0$, then if A and B have a common immediate latent ancestor L_1 in G, C and D have a common immediate latent ancestor L_2 in G, we have $L_1 = L_2$.*

Proof: Assume for the sake of contradiction that $L_1 \neq L_2$. Let P_A be a directed path from L_1 to A, and α_1 the sequence of edge labels in this path. Analogously, define α_2 as the sequence of edge labels from L_1 to B by some arbitrary path P_B, β_1 a sequence from L_2 to C according to some path P_C and β_2 a sequence from L_2 to D according to some path P_D.

P_A and P_B cannot intersect, since it would imply the existance of an observed common cause for P_A and P_B, which is ruled out by the given assumptions and Lemma B.1. Similarly, no pair of paths in $\{P_A, P_B, P_C, P_D\}$ can intersect. By Lemma B.4, L_1 cannot be an ancestor of either C or D, or otherwise $L_1 = L_2$. Analogously, L_2 cannot be an ancestor of either A or B.

By Lemma 4.2 and the given constraints, no element X in $\mathbf{O'}$ can be ancestor of an element in $\mathbf{O'} \backslash X$.

It means that when expanding the given constraint $\sigma_{AB}\sigma_{CD} - \sigma_{AD}\sigma_{BC} = 0$, and keeping all and only the terms that include the symbol $\alpha_1\alpha_2\beta_1\beta_2$, we obtain $\alpha_1\alpha_2\beta_1\beta_2\sigma^2_{L_1}\sigma^2_{L_2} - \alpha_1\alpha_2\beta_1\beta_2\sigma^2_{L_1L_2} = 0$, which implies $\rho_{L_1L_2} = 1$ with probability 1. Contradiction. □

Lemma 4.7 *Let* $\mathbf{S} \subseteq \mathbf{O}$ *be any set such that, for all* $\{A, B, C\} \subseteq \mathbf{S}$*, there is a fourth variable* $D \in \mathbf{O}$ *where i.* $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BD}$ *and ii. for every set* $\{X, Y\} \subset \{A, B, C, D\}, Z \in \mathbf{O}$ *we have* $\rho_{XY.Z} \neq 0$ *and* $\rho_{XY} \neq 0$*. Then* $\mathbf{S}$ *can be partioned into two sets* $\mathbf{S_1}, \mathbf{S_2}$ *where*

1. *all elements in* $\mathbf{S_1}$ *share a common immediate latent ancestor, and no two elements in* $\mathbf{S_1}$ *have any other common immediate latent ancestor;*

2. *no element* $S \in \mathbf{S_2}$ *has any common immediate latent ancestor with any other element in* $\mathbf{S} \backslash S$

3. *all elements in* $\mathbf{S}$ *are d-separated given the latents in G;*

Proof: Follows immediately from the given constraints and Lemmas 4.2, B.4 and B.5. □

Theorem 4.8 *If a partition* $\{\mathbf{C_1}, \ldots, \mathbf{C_k}\}$ *of* $\mathbf{O'}$ *respects structural conditions SC1, SC2 and SC3, then the following holds in the true latent variable graph G that generated the data:*

1. *for all* $X \in \mathbf{C_i}, Y \in \mathbf{C_j}, i \neq j$*, X and Y have no common parents, and X is d-separated from the latent parents of Y given the latent parents of X;*

2. *for all* $X, Y \in \mathbf{O'}$*, X is d-separated from Y given the latent parents of X;*

3. *every set* $\mathbf{C_i}$ *can be partitioned into two groups according to Lemma 4.7;*

Proof: Follows immediately from the given constraints and Lemmas 4.1, 4.4, 4.5 and Lemmas 4.7. □

Before showing the proof of Theorem 4.9, the next two lemmas will be useful:

Lemma B.6 *Let set* $\{A, B, C, D\} = \mathbf{O'} \subseteq \mathbf{O}$ *be such that* $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BD}$ *and for every set* $\{X, Y\} \subset \mathbf{O'}, Z \in \mathbf{O}$ *we have* $\rho_{XY.Z} \neq 0$ *and* $\rho_{XY} \neq 0$*. If an immediate latent ancestor* L_X *of* $X \in \mathbf{O'}$ *is uncorrelated with some immediate latent ancestor* L_Y *of* $Y \in \mathbf{O'}$*, then* L_X *is uncorrelated with all immediate latent ancestors of all elements in* $\mathbf{O'} \backslash X$ *or* L_Y *is uncorrelated with all immediate latent ancestors of all elements in* $\mathbf{O'} \backslash Y$*.*

Proof: Since the immediate latent ancestors of $\mathbf{O}'$ are linked to $\mathbf{O}'$ in that set by directed paths that do not intersect (Lemma B.1) other than at the sources, and the model is linear below the latents, we can treat them as parents of $\mathbf{O}'$ without loss of generality. We will prove the theorem in two steps.

Step 1: let $X, Y \in \mathbf{O}'$. If a parent L_X of X is uncorrelated with all parents of Y, then L_X is uncorrelated with all parents of all elements in $\mathbf{O}' \backslash X$. To see this, without loss of generality let $A = aL_A + \sum_p a_p A_p$, and let L_A be uncorrelated with all parents of B. Let $C = cL_C + \sum_j c_j C_j$. This means that when expanding the polynomial $\sigma_{AB}\sigma_{CD} - \sigma_{AC}\sigma_{BD} = 0$, the only terms containing the symbol ac will be $ac\sigma_{L_A L_C}\sigma_{BD}$. Since $ac \neq 0, \sigma_{BD} \neq 0$, this will force $\sigma_{L_A L_C} = 0$ with probability 1. By symmetry, L_A will be uncorrelated with all parents of C and D.

Step 2: now we show the result stated by the theorem. Without loss of generality let $A = aL_A + \sum_p a_p A_p$, $B = bL_B + \sum_i b_i B_i$ and let L_A be uncorrelated with L_B. Then no term in the polynomial corresponding to $\sigma_{AB}\sigma_{CD}$ can contain a term with the symbol ab, since $\sigma_{L_A L_B} = 0$. If L_B is uncorrelated with all parents of D, then L_B is uncorrelated will all parents of all elements in $\mathbf{O}' \backslash B$, and we are done. Otherwise, assume L_B is correlated with at least one parent of D. Then at least one term in $\sigma_{AC}\sigma_{BD}$ will contain the symbol ab if there is some parent of C that is correlated with L_A (because σ_{BD} will contain some term with b). It follows that L_A has to be uncorrelated with every parent of D, and by the result in Step 1, with all parents of all elements in $\mathbf{O}' \backslash A$. □

Lemma B.7 *Let set $\{A, B, C, D\} = \mathbf{O}' \subseteq \mathbf{O}$ be such that $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BD}$ and for every set $\{X, Y\} \subset \mathbf{O}', Z \in \mathbf{O}$ we have $\rho_{XY.Z} \neq 0$ and $\rho_{XY} \neq 0$. Let $\{A_p\}$ be the set of immediate latent ancestors of A, $\{B_i\}$ be the set of immediate latent ancestors of B, $\{C_j\}$ be the set of immediate latent ancestors of C, $\{D_k\}$ be the set of immediate latent ancestors of D. Then $\sigma_{A_p B_i}\sigma_{C_j D_k} = \sigma_{A_p C_j}\sigma_{B_i D_k} = \sigma_{A_p D_k}\sigma_{B_i C_j}$ for all $\{A_p, B_i, C_j, D_k\} \in \{A_p\} \times \{B_i\} \times \{C_j\} \times \{D_k\}$.*

Proof: Since the immediate latent ancestors of $\mathbf{O}'$ are linked to $\mathbf{O}'$ in that set by directed paths that do not intersect (Lemma B.1) other than at the sources, and the model is linear below the latents, we can treat them as parents of $\mathbf{O}'$ without loss of generality. Let a_p be the coefficient linking A and A_p. Define b_i, c_j, d_k analogously. The lemma follows immediately by the same measure theoretical arguments of previous lemmas applied to the terms that include $a_p b_i c_j d_k$. □

Theorem 4.9 *Given a partition $\mathbf{C}$ of a subset $\mathbf{O}'$ of the observed variables of a latent variable graph G such that $\mathbf{C}$ satisfies structural constraints SC1-SC4, there is a linear latent variable model for the first two moments of $\mathbf{O}'$.*

Proof: We will assume that all elements of all sets in $\mathbf{C}$ are correlated. Otherwise, $\mathbf{C}$ can be partitioned into subsets with this property (because of the SC4 condition), and the parameterization given below can be applied independently to each member of the partition without loss of generality.

Let $\mathbf{An_i}$ be the set of immediate latent ancestors of the elements in $\mathbf{C_i} \in \mathbf{C} = \{\mathbf{C_1}, \ldots, \mathbf{C_k}\}$. Split every $\mathbf{An_i}$ into two disjoint sets $\mathbf{An_i^0}$ and $\mathbf{An_i^1}$, such that $\mathbf{An_i^0}$ contains all and only the those elements of $\mathbf{An_i^0}$ that are uncorrelated with all elements in $\mathbf{An_1} \cup \cdots \cup \mathbf{An_k}$. This implies that all elements in $\mathbf{An_1^1} \cup \cdots \cup \mathbf{An_k^1}$ are pairwise correlated by Lemma B.6.

Construct the graph G^L_{linear} as follows. For each set $\mathbf{An_i}$, add a latent L_{An_i} to G^L_{linear}, as well as all elements of $\mathbf{An^1_i}$. Add a directed edge from L_{An_i} to each element in $\mathbf{An^1_i}$. Let G^L_{linear} be also a linear latent variable model. We will define values for each parameter in this model.

Fully connected all elements in $\{L_{An_i}\}$ as an arbitrary directed acyclic graph (DAG). Instead of defining the parameters for the edges and error variances in the subgraph of G^L_{linear} induced by $\{L_{An_i}\}$, we will directly define a covariance matrix Σ_L among these nodes. Standard results in linear models can be used to translate this covariance matrix to the parameters of an arbitrary fully connected DAG (Spirtes et al., 2000). Set the diagonal of Σ_L to be 1.

Define the intercept parameters μ_x of all elements in G^L_{linear} to be zero. For each V in $\mathbf{An^1_i}$ we have a set of parameters for the local equations $V = \lambda_V L_{An_i} + \epsilon_V$, where ϵ_V is a random variable with zero mean and variance ζ_V.

Choose any three arbitrary elements $\{X, Y, Z\} \subseteq \mathbf{An^1_i}$. Since the subgraph $L_{An_i} \to X, L_{An_i} \to Y, L_{An_i} \to Z$ has six parameters $(\lambda_X, \lambda_Y, \lambda_Z, \zeta_X, \zeta_Y, \zeta_Z)$ and the population covariance matrix of X, Y and Z has six entries, these parameters can be assigned an unique value (Bollen, 1989) such that $\sigma_{XY} = \lambda_X \lambda_Y$ and $\zeta_X = \lambda^2_X - \sigma^2_X$. Let W be any other element of $\mathbf{An^1_i}$: set $\lambda_W = \sigma_{WX}/\lambda_X$, $\zeta_W = \sigma^2_W - \lambda^2_W$. From Lemma B.7, we have the constraint $\sigma_{WY}\sigma_{XZ} - \sigma_{WX}\sigma_{YZ} = 0$, from which one can verify that $\sigma_{WY} = \lambda_W \lambda_Y$. By symmetry and induction, for every pair P, Q in $\mathbf{An^1_i}$, we have $\sigma_{PQ} = \lambda_P \lambda_Q$.

Let T be some element in $\mathbf{An^1_j}$, $i \neq j$: set the entry σ_{ij} of Σ_L to be $\sigma_{TX}/(\lambda_T \lambda_X)$. Let R and S be another elements in $\mathbf{An^1_j}$. From Lemma B.7, we have the constraint $\sigma_{XT}\sigma_{RS} - \sigma_{XR}\sigma_{ST} = 0$, from which one can verify that $\sigma_{XR} = \lambda_X \lambda_R \sigma_{ij}$. Let Y and Z be another elements in $\mathbf{An^1_i}$. From Lemma B.7, we have the constraint $\sigma_{XT}\sigma_{YZ} - \sigma_{XY}\sigma_{ZT} = 0$ from which one can verify that $\sigma_{ZT} = \lambda_Z \lambda_t \sigma_{ij}$. By symmetry and induction, for every pair P, Q in $\mathbf{An^1_i} \times \mathbf{An^1_j}$, we have $\sigma_{PQ} = \lambda_P \lambda_Q \sigma_{ij}$.

Finally, let G_{linear} be a graph constructed as follows:

1. start G_{linear} with a node for each element in $\mathbf{O'}$;
2. for each $\mathbf{C_i} \in \mathbf{C}$, add a latent L_i to G, and for each $V \in \mathbf{C_i}$, add an edge $L_i \to V$
3. fully connect the latents in G_{linear} to form an arbitrary directed acyclic graph

Parameterize a linear latent model based on G as follows: let $V \in \mathbf{C_i}$ such that V has immediate latent ancestors $\{L_{V_i}\}$. In the true model, let $V = \mu^G_V + \Sigma_i \lambda^G_{iV} L_{V_i} + \epsilon^G_V$, where every latent is centered at its mean. Construct the equation $V = \mu_V + \lambda_V L_i + \epsilon_V$ by instantiating $\mu_V = \mu^G_V$ and $\lambda_V = \Sigma_i \lambda^G_{iV} \lambda_{L_{V_i}}$, where $\lambda_{L_{V_i}}$ is the respective parameter for L_{V_i} in G^L_{linear} if $L_{V_i} \in \mathbf{An^1_i}$, and 0 otherwise. The variance for ϵ_V is defined as $\sigma^2_V - \lambda^2_V$. The L_i variables have covariance matrix Σ_L as defined above. One can then verify that the covariance matrix generated by this model equals the true covariance matrix of $\mathbf{O'}$. □

Lemma 4.10 *Let $G(\mathbf{O})$ be a latent variable graph where no pair in $\mathbf{O}$ is marginally uncorrelated, and let $\{X, Y\} \subset \mathbf{O}$. If there is no pair $\{P, Q\} \subset \mathbf{O}$ such that $\sigma_{XY}\sigma_{PQ} = \sigma_{XP}\sigma_{YQ}$ holds, then there is at least one graph in the tetrad equivalence class of G where X and Y have a common latent parent.*

Proof: It will suffice to show the result for linear latent variable models, since they are more constrained than non-linear ones. Moreover, we will be able to make use of the Tetrad Representation

Theorem and the equivalence of d-separations and vanishing partial correlations, facilitating the proof.

If in all graphs in the tetrad equivalence graph of G we have that X and Y share some common hidden parent, then we are done. Assume then that there is at least one graph G_0 in this class such that X and Y have no common hidden parent. Construct graph G_0' by adding a new latent and edges $X \leftarrow L \rightarrow Y$. We will show that G_0' is in the same tetrad equivalence class, i.e., the addition of the substructure $X \leftarrow L \rightarrow Y$ to G_0 does not destroy any entailed tetrad constraint (it might, however, destroy some independence constraint).

Assume there is a tetrad constraint corresponding to some choke point $\{X, P\} \times \{T, Q\}$. If Y is not an ancestor of T or Q, then this tetrad will not be destroyed by the introduction of subpath $X \leftarrow L \rightarrow Y$, since no new treks connecting X or P to T or Q can be formed, and therefore no choke point $\{X, P\} \times \{T, Q\}$ will disappear.

Assume without loss of generality that Y is an ancestor of Q. Since there is a trek connecting X to Q through Y (because no marginal correlations are zero) in G, the choke point $\{X, P\} \times \{T, Q\}$ should be in this trek. Let X be the starting node of this trek, and Q the ending node. If the choke point is after Y on this trek, then this choke point will be preserved under the addition of $X \leftarrow L \rightarrow Y$. If the choke point is Y or is before Y on this trek, then there will be a choke point $\{X, P\} \times \{Y, Q\}$, a contradiction of the assumptions.

One can show that choke points $\{Y, P\} \times \{T, Q\}$ are also preserved by an analogous argument. □

Before proving Theorem 4.11, we will introduce several lemmas that will be used in the Theorem proof.

Lemma B.8 *Let $G(\mathbf{O})$ be a linear latent variable graph, and let $\mathbf{O}' = \{A, B, C, D\} \subseteq \mathbf{O}$. If all elements in $\mathbf{O}'$ are marginally correlated, and a choke point $CP = \{A, C\} \times \{B, D\}$ exists, and CP is in all treks connecting elements in $\{A, B, C, D\}$, then no two elements $\{X_1, X_2\}$, $X_1 \in \{A, C\}$, $X_2 \in \{B, D\}$, are both connected to CP in G by treks into CP.*

Proof: By the Tetrad Representation Theorem, CP should be either on the $\{A, C\}$ or the $\{B, D\}$ side of every trek connecting elements in these two sets. For the sake of contradiction, assume without loss of generality that A and B are connected to CP by some treks into CP. Since $\sigma_{AB} \neq 0$, CP has to be an ancestor of either A or B. Without loss of generality, let CP be an ancestor of B. Then there is at least one trek connecting A and B such that CP is not on the $\{A, C\}$ side of it: the one connecting CP and A that is into CP and continues into B.

If CP is an ancestor of C, then there is at least one trek connecting C and B such that CP is not in the $\{B, D\}$ side of it: the one connecting CP and B that is into CP and continues into C. But this cannot happen by the definition of choke point. If CP is not an ancestor of C, CP has to be an ancestor of A, or otherwise there would be no treks connecting A and C (since CP is in all treks connecting A and C by hypothesis, and at least one exists, because $\sigma_{AC} \neq 0$). This implies at least one trek connecting A and B such that CP is not on the $\{B, D\}$ side of it: the one connecting CP and B that is into CP and continues into A. Contradiction. □

Lemma B.9 *Let $G(\mathbf{O})$ be a linear latent variable graph, and let $\mathbf{O}' = \{A, B, C, D, E\} \subseteq \mathbf{O}$. If all elements in $\mathbf{O}'$ are marginally correlated, and constraints $\sigma_{AB}\sigma_{CD} = \sigma_{AD}\sigma_{BC}$, $\sigma_{AC}\sigma_{DE} = \sigma_{AE}\sigma_{CD}$ and $\sigma_{BC}\sigma_{DE} = \sigma_{BD}\sigma_{CE}$ hold, then all three tetrad constraints hold in the covariance matrix of $\{A, B, C, D\}$.*

Proof: By the Tetrad Representation Theorem, let CP_1 be a choke point $\{A, C\} \times \{B, D\}$, which is known to exist in G by assumption. Let CP_2 be a choke point $\{A, D\} \times \{C, E\}$, which is also assumed to exist. From the definition of choke point, all treks connecting C and D have to pass through both CP_1 and CP_2. We will assume without loss of generality that none of the choke points we introduce in this proof are elements of $\{A, B, C, D, E\}$.

First, we will show by contradiction that all treks connecting A to C should include CP_1. Assume that A is connected to C through a trek T that includes CP_2 but not CP_1. Let T_1 be the subtrek $A - CP_2$, i.e., the subtrek of T connecting A and CP_2. Let T_2 be the subtrek $CP_2 - C$. Neither T_1 or T_2 contain CP_1, and they should not collide at CP_2 by definition. Notice that a trek like T should exist, since CP_2 has to be in all treks connecting A and C, and at least one such trek exists because $\sigma_{AC} \neq 0$. Any subtrek connecting CP_2 to D that does not intersect T_2 elsewhere but in CP_2 has to contain CP_1. Let T_3 be the subtrek between CP_2 and CP_1. Let T_4 be a subtrek between CP_1 and B. Let T_5 be the subtrek between CP_1 and D. This is illustrated by Figure B.2(a). (B and D might be connected by other treks, simbolized by the dashed edge.)

Now consider the choke point $CP_3 = \{B, E\} \times \{C, D\}$. Since CP_3 is in all treks connecting B and C, CP_3 should be either on T_2, T_3 or T_4. If CP_3 is on T_4 (Figure B.2(b)), then there will be a trek connecting D and E that does not include CP_2, which contradicts the definition of choke point $\{A, D\} \times \{C, E\}$, unless both $B - CP_1$ and $D - CP_1$ are into CP_1. However, if both $B - CP_1$ and $D - CP_1$ (i.e., T_4 and T_5) are into CP_1, then $CP_1 - CP_2$ is out of CP_1 and into CP_2, since $T_2 - T_3 - T_5$ is a trek by construction, and therefore cannot contain a collider. Since D is an ancestor of CP_2 and CP_2 is in a trek connecting E and D, then CP_2 is an ancestor of E. All paths $CP_2 \rightarrow \cdots \rightarrow E$ should include CP_3 by definition, which implies that CP_2 is an ancestor of CP_3. B cannot be an ancestor of CP_3, or otherwise CP_3 would have to be an ancestor of CP_1, creating the cycle $CP_3 \rightarrow \ldots CP_1 \rightarrow \cdots \rightarrow CP_2 \rightarrow \cdots \rightarrow CP_3$. CP_3 would have to be an ancestor of B, since $B - CP_3 - CP_1$ is assumed to be a trek into CP_1 and CP_3 is not an ancestor of CP_1 (Figure B.2(c)). If CP_3 is an ancestor of B, then there is a trek $C \leftarrow \cdots \leftarrow CP_2 \rightarrow \ldots CP_3 \rightarrow B$, which does not include CP_1. Therefore, CP_3 is not in T_4.

If CP_3 is in T_3, B and D should both be ancestors of CP_1, or otherwise there will be a trek connecting them that does not include CP_3. Again, this will imply that CP_1 is an ancestor of CP_2. If some trek $E - CP_3$ is not into CP_3, then this creates a trek $D - CP_1 - CP_3 - E$ that does not contain CP_2, contrary to our hypothesis. If every trek $E - CP_3$ is into CP_3, then some other trek $CP_3 - D$ that is out of CP_3 but does not include CP_1 has to exist. But then this creates a trek connecting C and D that does not include CP_1, which contradicts the definition $CP_1 = \{A, C\} \times \{B, D\}$. A similar reasoning forbids the placement of CP_3 in T_2.

Therefore, all treks connecting A and C should include CP_1. We will now show that all treks connecting B and D should also include CP_1. We know that all treks connecting elements in $\{A, C, D\}$ go through CP_1. We also know that all treks between $\{B, E\}$ and $\{C, D\}$ go through CP_3. This is illustrated by Figure B.2(d). A possible trek from CP_3 to D that does not include CP_1 (represented by the dashed edge connecting $CP3$ and D) would still have to include CP_2, since all treks in $\{A, D\} \times \{C, E\}$ go through CP_2. If $CP_1 = CP_2$, then all treks between B and D go through CP_1. If $CP_1 \neq CP_2$, then such $CP_3 - D$ trek without CP_1 but with CP_2 would exist, implying that some trek $C - D$ without both CP_1 and CP_2 would exist, contrary to our hypothesis.

Therefore, we showed that all treks connecting elements in $\{A, B, C, D\}$ go through the same point CP_1. By symmetry between B and E, it is also the case that CP_1 is in all treks connecting elements in $\{A, E, C, D\}$. From this one can verify that $CP_1 = CP_2$. We will show that CP_1 is

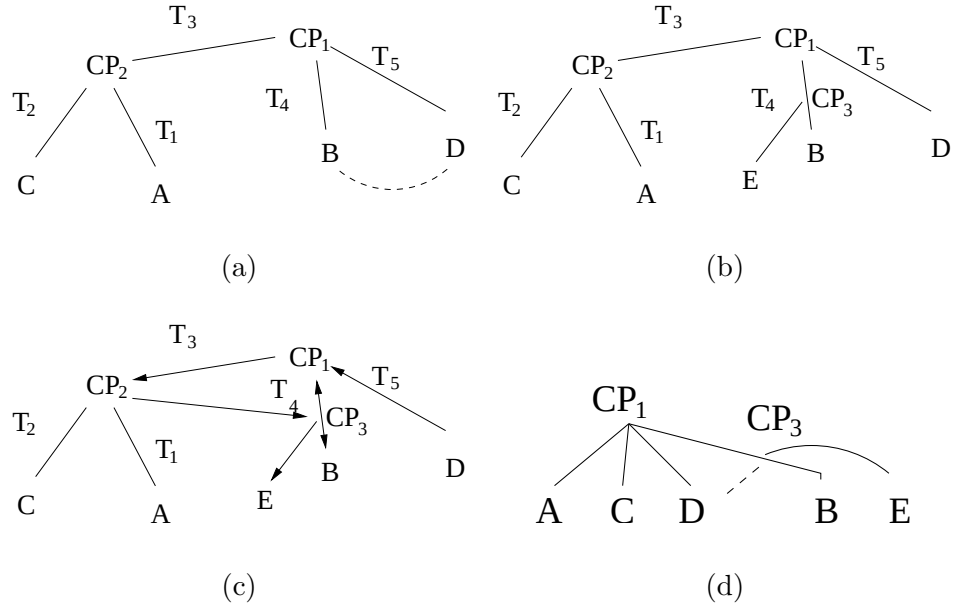


Figure B.2: Several illustrations depicting cases used in the proof of Lemma B.9.

also a choke point for $\{B, E\} \times \{C, D\}$ (althought it might be the case that $CP1 \neq CP_3$). Because $CP_1 = CP_2$, one can verify that choke point CP_3 has to be in a trek connecting B and CP_1. There is a trek connecting B and CP_1 that is into CP_1 if and only if is a trek connecting B and CP_3 that is into CP_3. The same holds for E. Therefore, there is a trek connecting B and CP_1 that is into CP_1 if and only if there is a trek connecting E and CP_1 that is into CP_1. However, if there is a trek connecting B and CP_1 into CP_1, then there is no trek connecting C and CP_1 that is into CP_1 (because of choke point $\{A, C\} \times \{B, D\}$ and Lemma B.8). This also implies there is no trek $E - CP_1$ into CP_1, and because CP_1 is a $\{A, D\} \times \{C, E\}$ choke point, Lemma B.8 will imply that there is no $D - CP_1$ into CP_1. Therefore, all treks connecting pairs $\{B, E\} \times \{C, D\}$ will be either on the $\{B, E\}$ side or $\{C, D\}$ of CP_1. CP_1 is a $\{B, E\} \times \{C, D\}$ choke point.

Because CP_1 is a $\{A, C\} \times \{B, D\}$, $\{A, D\} \times \{C, E\}$ and $\{B, E\} \times \{C, D\}$ choke point, then no pair in $\{A, B, C, D\}$ can be connected to CP_1 by a trek into CP_1. This implies that CP_1 d-separates all elements in $\{A, B, C, D\}$ and therefore CP_1 is a choke point for all tetrads in this set. □

Lemma B.10 *Let $G(\mathbf{O})$ be a linear latent variable graph, and let $\mathbf{O}' = \{A, B, C, D, E\} \subseteq \mathbf{O}$. If all elements in $\mathbf{O}'$ are marginally correlated, and constraints $\sigma_{AB}\sigma_{CD} = \sigma_{AD}\sigma_{BC}$, $\sigma_{AC}\sigma_{DE} = \sigma_{AE}\sigma_{CD}$ and $\sigma_{BE}\sigma_{DC} = \sigma_{BD}\sigma_{CE}$ hold, then all three tetrad constraints hold in the covariance submatrix formed by any foursome in $\{A, B, C, D, E\}$.*

Proof: As in Lemma B.9, let CP_1 be a choke point $\{A, C\} \times \{B, D\}$, and let CP_2 be a choke point $\{A, D\} \times \{C, E\}$. Let CP_3 be choke point $\{B, C\} \times \{D, E\}$.

We first show that all treks between C and A go through CP_1. Assume there is a trek connecting A and C through CP_2 but not CP_1, analogous to Figure B.2(a). Let $T_1, \ldots, T_5$ be defined as in

Lemma B.9. Since all treks between C and D go through CP_3, choke point CP_3 should be either at T_2, T_3 or T_4.

If CP_3 is at T_2 or T_3, then treks B and D should collide at CP_1, or otherwise there will be a trek connecting B and D that does not include CP_3. This implies that CP_1 is an ancestor of CP_3. If there is a trek connecting D and CP_3 that intersects T_2 or T_3 not at CP_1, then there will be a trek connecting C and D that does not include CP_1, which would be a contradiction. If there is no such a trek connecting D and CP_3, then CP_3 cannot be a $\{B, C\} \times \{D, E\}$ choke point. If CP_3 is at T_4, a similar case will follow.

Therefore, all treks connecting A and C include CP_1. By symmetry between $\{A, B, E\}$ and $\{C, D\}$, CP_1 is in all treks connecting any pair in $\{A, B, C, D, E\}$. Using the same arguments of Lemma B.9, one can show that CP_1 is a choke point for any foursome in this set. □

Lemma B.11 *Let $G(\mathbf{O})$ be a linear latent variable graph, and let $\mathbf{O}' = \{A, B, C, D, E\} \subseteq \mathbf{O}$. If all elements in $\mathbf{O}'$ are marginally correlated, and constraints $\sigma_{AB}\sigma_{CD} = \sigma_{AD}\sigma_{BC}$, $\sigma_{AC}\sigma_{DE} = \sigma_{AE}\sigma_{CD}$ and $\sigma_{AB}\sigma_{CE} = \sigma_{AC}\sigma_{BE}$ hold, then all three tetrad constraints hold in the covariance matrix of $\{A, C, D, E\}$.*

Proof: As in Lemmas B.9 and B.10, let CP_1 be a choke point $\{A, C\} \times \{B, D\}$, and let CP_2 be a choke point $\{A, D\} \times \{C, E\}$. Let CP_3 be a choke point $\{A, E\} \times \{B, C\}$. We will first show that all treks connecting A and C either go through CP_1 or all treks connecting A and D go through CP_2.

As in Lemma B.9, all treks connecting C and D contains CP_1 and CP_2. Let T be one of these treks. Assuming that A and C are connected by some trek that does not contain CP_1 (but must contain CP_2) implies a family of graphs represented by Figure B.2(a).

Since there is a choke point $CP_3 = \{A, E\} \times \{B, C\}$, the only possible position for CP_3 in Figure B.2(a) is in trek $A - CP_2$. If $CP_2 \neq CP_3$, then no choke point $\{A, D\} \times \{C, E\}$ can exist, since CP_3 is not in T. Therefore, either all treks between A and C contain CP_1, or $CP_2 = CP_3$.

If the first case holds, a similar argument will show that all treks between any element in $\{A, C, D\}$ and node E will have to go through CP_1. If the second case holds, a similar argument will show that all treks between any element in $\{A, C, D\}$ and node E will have to go through CP_2.

Therefore, there is a node CP such that all treks connecting elements in $\{A, C, D, E\}$ go throught some choke point. Similarly to the proof of Lemma B.9, using Lemma B.8, the given tetrad constraints will imply that CP is a choke point for all tetrads in $\{A, C, D, E\}$ for both cases $CP = CP_1$ and $CP = CP_2$. □

Theorem 4.11 *Let $\mathbf{X} \subseteq \mathbf{O}$ be a set of observed variables, $|\mathbf{X}| < 6$. Assume $\rho_{X_1X_2} \neq 0$ for all $\{X_1, X_2\} \subseteq \mathbf{X}$. There is no possible set of tetrad constraints within $\mathbf{X}$ for deciding if two nodes $\{A, B\} \subset \mathbf{X}$ do not have a common parent in a latent variable graph $G(\mathbf{O})$.*

Proof: It will suffice to show the result for linear latent variable models, since they are more constrained than non-linear ones. Moreover, we will be able to make use of the Tetrad Representation Theorem and the equivalence of d-separations and vanishing partial correlations, facilitating the proof.

This is trivial for domains of size 2 and 3, where no tetrad constraint can hold. For domains of size 4, let $\mathbf{X} = \{A, B, C, D\}$ be our four variables. We will show that it does not matter which

tetrad constraints hold among these four variables (excluding logically inconsistent constraints), there exist two linear latent variable graphs with observable variables $\{A, B, C, D\}$, G' and G'', where in the former A and B do not share a parent, while in latter they do have a parent in common. This will be the main technique used during the entire proof. Another technique is showing that some combinations of tetrad constraints will result in contradictory assumptions about existing constraints, and therefore we do not need to create the G' and G'' graphs corresponding to these sets.

By Lemma 4.10, if we do not have any tetrad corresponding to a choke point $\{A, V_1\} \times \{B, V_2\}$, then the result follows immediately. We therefore consider only the cases where the tetrad constraint corresponding to choke point $\{A, C\} \times \{B, D\}$ exists, without loss of generality. This assumption will be used during the entire proof.

Bi-directed edges $X \leftrightarrow Y$ will be used as a shorthand representation for the path $X \leftarrow L \rightarrow Y$, where L is some new latent independent of its non-children.

Suppose first that all possible three tetrad constraints hold in the covariance matrix Σ of $\{A, B, C, D\}$, i.e., $\sigma_{AB}\sigma_{CD} = \sigma_{AC}\sigma_{BD} = \sigma_{AD}\sigma_{BC}$. Let G' have two latent nodes L_1 and L_2, where L_1 is a common parent of A and L_2, and L_2 a parent of B, C and D. Let G'' have a latent node L_1 as the only parent of A, B, C and D, and no other edges, and the result will follow for this case.

Suppose now only one tetrad constraint holds instead of all three, i.e., the one entailed by a choke point between pairs $\{A, C\} \times \{B, D\}$ (the analogous case would be the pairs $\{A, D\} \times \{B, C\}$). Create G' again by using two latents L_1 and L_2, making L_2 a parent of B and D, and making L_1 a parent of L_2, A and C. Create G'' from G', by adding the edge $L_1 \rightarrow B$.

Now suppose our domain $\mathbf{X} = \{A, B, C, D, E\}$ has five variables, where Σ will now denote the covariance matrix of $\mathbf{X}$. Again, we will show how to build graphs G' and G'' in all possible consistent combinations of vanishing and non-vanishing tetrad constraints. This case is more complicated, and we will divide it in several major subcases. Each subcase will have an sub-index, and each sub-index inherits the assumptions of higher-level indices. Some results about entailment of tetrad constraints are stated without explicit detail: they can be derived directly by a couple of algebraic manipulations of tetrad constraints or from Lemmas B.9, B.10 and B.11.

Case 1: **There are choke points $\{A, C\} \times \{B, D\}$ and $\{A, B\} \times \{C, D\}$.** We know from the assumption of existence of a choke point $\{A, C\} \times \{B, D\}$ and results from Chapter 3 that this is equivalent of having a latent variable d-separating all elements in $\{A, B, C, D\}$. Let G_0 be as follows: let L_1 and L_2 be two latent variables, let L_1 be a parent of $\{A, L_2\}$, and let L_2 be a parent of $\{B, C, D, E\}$. We will construct G' and G'' from G_0, considering all possible combinations of choke points of the form $\{V_1, V_2\} \times \{V_3, E\}$.

Case 1.1: there is a choke point $\{A, C\} \times \{D, E\}$.

Case 1.1.1:there is a choke point $\{A, D\} \times \{C, E\}$. As before, this implies a choke point $\{A, E\} \times \{C, D\}$. **We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$ and $\{X_1, X_2\} \times \{B, E\}$.** From the given constraints $\sigma_{BD}\sigma_{AC} = \sigma_{BC}\sigma_{AD}$ (choke point $\{A, B\} \times \{C, D\}$) and $\sigma_{DE}\sigma_{AC} = \sigma_{CE}\sigma_{AD}$ (choke point $\{A, E\} \times \{C, D\}$), we have $\sigma_{BD}\sigma_{CE} = \sigma_{BC}\sigma_{DE}$, a $\{B, E\} \times \{C, D\}$ choke point. Choke points $\{B, E\} \times \{A, C\}$ and $\{B, E\} \times \{A, D\}$ will follow from this conclusion. Finally, if we assume also the existence of some choke point $\{X_1, B\} \times \{X_2, E\}$, then all choke points of this form will exist, and one can let $G' = G_0$. Otherwise, if there is no choke point $\{X_1, B\} \times \{X_2, E\}$, let G' be G_0 with the added edge $B \leftrightarrow E$. Construct G'' by adding edge $L_2 \rightarrow A$ to G'.

Case 1.1.2:there is no choke point $\{A, D\} \times \{C, E\}$. Choke point $\{A, E\} \times \{C, D\}$ cannot exist, or this will imply $\{A, D\} \times \{C, E\}$. **We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$ and $\{X_1, X_2\} \times \{B, E\}$.** Choke point $\{A, C\} \times \{B, E\}$ is entailed to exist, since the single choke point that d-separates foursome $\{A, B, C, D\}$ has to be the same choke point for $\{A, C\} \times \{D, E\}$ and therefore a choke point for $\{A, C\} \times \{B, E\}$. No choke point $\{X_1, D\} \times \{X_2, E\}$ can exist, for $X_i \in \{A, B, C\}, i = 1, 2$: otherwise, from the given choke points and $\{X_1, D\} \times \{X_2, E\}$, one can verify that $\{A, D\} \times \{C, E\}$ would be generated using combinations of tetrad constraints. **We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$.** Choke points $\{B, C\} \times \{A, E\}$, $\{B, C\} \times \{D, E\}$, $\{A, B\} \times \{C, E\}$ and $\{A, B\} \times \{D, E\}$ either all exist or none exists. If all exist, let $G' = G_0$ with the extra edge $D \leftrightarrow E$. If none exists, let $G' = G_0$ and add both $B \leftrightarrow E$ and $D \leftrightarrow E$ to G'. Let G'' be G' with the extra edge $L_2 \rightarrow A$.

Case 1.2: there is no choke point $\{A, C\} \times \{D, E\}$.

Case 1.2.1:there is a choke point $\{A, D\} \times \{C, E\}$. This case is analogous to Case 1.1.2 by symmetry within $\{A, B, C, D\}$.

Case 1.2.2:there is no choke point $\{A, D\} \times \{C, E\}$. Assume first there is no choke point $\{A, E\} \times \{C, D\}$. **We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$ and $\{X_1, X_2\} \times \{B, E\}$.** At most one of the choke points $\{X_1, B\} \times \{X_2, E\}$ can exist. Otherwise, any two of them will entail either $\{A, D\} \times \{C, E\}$, $\{A, C\} \times \{D, E\}$ or $\{A, E\} \times \{C, D\}$ by Lemmas B.9, B.10 or B.11. Analogously, no choke point $\{X_1, X_2\} \times \{B, E\}$ can exist.

Without loss of generality, let $\{A, B\} \times \{D, E\}$ be the only possible extra choke point. Create G' by adding edges $C \leftrightarrow E$ and $D \leftrightarrow E$ to G_0. Create G'' by adding edge $L_2 \rightarrow A$ to G'. For the case where no other choke point exists, create G' by adding edges $A \leftrightarrow E$, $B \leftrightarrow E$, $C \leftrightarrow E$ and $D \leftrightarrow E$ to G_0. Create G'' by adding edge $L_2 \rightarrow A$ to G'.

Assume now there is a choke point $\{A, E\} \times \{C, D\}$. **We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$ and $\{X_1, X_2\} \times \{B, E\}$.** No $\{A, B\} \times \{X_1, E\}$ choke point can exist, or by Lemmas B.9, B.10 or B.11 and the given tetrad constraints, some $\{A, X_1\} \times \{E, X_2\}$ choke point will be entailed.

Choke point $\{B, C\} \times \{D, E\}$ exists if and only if $\{B, D\} \times \{C, E\}$ exists. can exist. If both exist, create G' by adding edges $A \leftrightarrow E$ to G_0. Create G'' by adding edge $L_2 \rightarrow A$ to G'. If none exists, create G' by adding edges $A \leftrightarrow E$ and $B \leftrightarrow E$ to G_0. Create G'' by adding edge $L_2 \rightarrow A$ to G'.

Case 2: **There is a choke point $\{A, C\} \times \{B, D\}$, but no choke point $\{A, B\} \times \{C, D\}$.**

Case 2.1: there is a choke point $\{A, C\} \times \{D, E\}$,.

Case 2.1.1: there is a choke point $\{A, D\} \times \{C, E\}$. As before, this implies a choke point $\{A, E\} \times \{C, D\}$. **We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$ and $\{X_1, X_2\} \times \{B, E\}$.** The choke point $\{A, C\} \times \{B, E\}$ is implied. No choke point $\{B, E\} \times \{X_1, D\}$ can exist, or otherwise $\{A, B\} \times \{C, D\}$ will be implied. For the same reason, no choke point $\{B, X_1\} \times \{D, E\}$ can exist. **We only have to consider now subsets of the set of constraints** $\{\{A, B\} \times \{C, E\}, \{C, B\} \times \{A, E\}\}$. The existence of $\{A, B\} \times \{C, E\}$ implies $\{C, B\} \times \{A, E\}$. We only need to consider either both or none.

Suppose none of these two constraints hold. Create G' with two latents L_1, L_2. Let L_1 be a parent of $\{B, L_2\}$, let L_2 be a parent of $\{A, C, D, E\}$. Add the bi-directed edge $B \leftrightarrow E$. Add the bi-directed edge $B \leftrightarrow D$. Create G'' out of G' by adding edge $L_2 \rightarrow B$. Now suppose both

constraints hold. Create G' with two latents L_1, L_2. Let L_1 be a parent of $\{B, L_2\}$, let L_2 be a parent of $\{A, C, D, E\}$. Add the bi-directed edge $B \leftrightarrow D$. Create G'' out of G' by adding edge $L_2 \rightarrow B$.

Case 2.1.2: there is no choke point $\{A, D\} \times \{C, E\}$. Since there is a choke point $\{A, C\} \times \{D, E\}$ by assumption 2.1, there is no choke point $\{A, E\} \times \{C, D\}$ or otherwise we get a contradiction. Analogously, because there is a $\{A, C\} \times \{B, D\}$ choke point but no $\{A, B\} \times \{C, D\}$ (assumption 2), we cannot have a $\{A, D\} \times \{B, C\}$ choke point. This covers all choke points within sets $\{A, B, C, D\}$ and $\{A, C, D, E\}$. **We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$ and $\{X_1, X_2\} \times \{B, E\}$.**

From $\sigma_{AB}\sigma_{CD} = \sigma_{AD}\sigma_{BC}$ (choke point $\{A, C\} \times \{B, D\}$) and $\sigma_{AE}\sigma_{CD} = \sigma_{AD}\sigma_{CE}$ (choke point $\{A, C\} \times \{D, E\}$) one gets $\sigma_{AB}\sigma_{CE} = \sigma_{AE}\sigma_{BC}$, i.e., a $\{B, E\} \times \{A, C\}$ choke point. Choke point $\{B, E\} \times \{A, D\}$ exists if and only if $\{B, E\} \times \{C, D\}$ exists: to see how the former implies the latter, use the tetrad constraint from $\{B, E\} \times \{A, C\}$. Therefore, we have two subcases.

Case 2.1.2.1: there are choke points $\{B, E\} \times \{A, D\}$ and $\{B, E\} \times \{C, D\}$. **We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$.** No choke point $\{B, A\} \times \{C, E\}$ and $\{B, C\} \times \{A, E\}$ can exist (one implies the other, since we have $\{B, E\} \times \{A, C\}$, and all three together with the given choke points will generate $\{A, B\} \times \{C, D\}$, excluded by assumption). Choke points $\{B, C\} \times \{D, E\}$ and $\{B, D\} \times \{C, E\}$ either both exist or both do not exist. The same holds for pair $\{\{B, A\} \times \{D, E\}, \{B, D\} \times \{A, E\}\}$. Let G' be a graph with two latents, L_1, L_2, where L_1 is a parent of $\{L_2, A, C\}$ and L_2 is a parent of $\{B, D, E\}$. Add bi-directed edge $B \leftrightarrow D$ for cases where $\{B, C\} \times \{D, E\}, \{B, D\} \times \{C, E\}$ do not exist. Add bi-directed edge $B \leftrightarrow E$ for cases where $\{B, A\} \times \{D, E\}, \{B, D\} \times \{A, E\}$ do not exist. Let G'' be formed from G' with the addition of $L_1 \rightarrow B$.

Case 2.1.2.2: there are no choke points $\{B, E\} \times \{A, D\}$ and $\{B, E\} \times \{C, D\}$. **We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$.** Using the tetrad constraint implied by choke point $\{A, C\} \times \{D, E\}$, one can verify that $\{A, B\} \times \{D, E\}$ holds if and only if $\{B, C\} \times \{D, E\}$ holds (call pair $\{\{A, B\} \times \{D, E\}, \{B, C\} \times \{D, E\}\}$ Pair 1). From the given $\{B, E\} \times \{A, C\}$, we have that $\{A, B\} \times \{C, E\}$ holds if and only if $\{B, C\} \times \{A, E\}$ holds (call it Pair 2). Using the given tetrad constraint corresponding to $\{A, C\} \times \{B, D\}$, one can show that $\{B, D\} \times \{A, E\}$ holds if and only if $\{B, D\} \times \{C, E\}$ (call it Pair 3). We can therefore partition all six possible $\{X_1, B\} \times \{X_2, E\}$ into these three pairs. Moreover, if Pair 1 holds, none of the other two can hold, because Pair 1 and Pair 2 together imply $\{B, E\} \times \{A, D\}$. Pair 1 and Pair 3 together imply $\{B, E\} \times \{C, D\}$.

If neither Pair holds, construct G' as follows. Let G_0 be the latent variable graph containing three latents L_1, L_2, L_3 where L_1 is a parent of $\{A, C, L_2\}$, L_2 is a parent of $\{B, L_3\}$ and L_3 is a parent of $\{D, E\}$. Let G' be G_0 with the added edges $B \leftrightarrow D$ and $B \leftrightarrow E$. If Pair 1 alone holds, let G' be as G_0. In both cases, let G'' be G' with the added edge $L_1 \rightarrow B$.

If Pair 2 holds, but not Pair 3 (nor Pair 1), construct G' as follows. Let G_0 be a latent variable graph with two latents L_1 and L_2, where L_1 is a parent of L_2 and A, and L_2 is a parent of $\{B, C, D, E\}$. Let G' be G_0 augment with edges $B \leftrightarrow D$ and $B \leftrightarrow E$. If Pairs 2 and 3 hold (but nor Pair 1), let G' be G_0 with the extra edge $B \leftrightarrow D$. In both cases, let G'' be G' with the extra edge $L_2 \rightarrow A$. If Pair 3 holds but not Pair 2 (nor Pair 1), let G' have three latents L_1, L_2, L_3, where L_1 is a parent of L_2 and A, L_2 is a parent of L_3 and C, and L_3 is a parent of B, D and E. Let G'' be as G' but with the extra edge $L_3 \rightarrow L_1$.

Case 2.2: there no a choke point $\{A, C\} \times \{D, E\}$.

Case 2.2.1: there is a choke point $\{A, D\} \times \{C, E\}$. Because of the choke points that are assumed not to exist, it follows immediately that choke points $\{A, D\} \times \{B, C\}$, $\{A, E\} \times \{C, D\}$ cannot exist. **We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$ and $\{X_1, X_2\} \times \{B, E\}$.** The choke point $\{A, D\} \times \{B, E\}$ cannot exist, or otherwise when it is combined with choke point $\{A, D\} \times \{C, E\}$, it will generate a contraint corresponding to choke point $\{A, D\} \times \{B, C\}$, which is assumed not to exist. Similarly, $\{A, C\} \times \{B, E\}$ cannot exist because the existence of $\{A, C\} \times \{B, D\}$ will imply $\{A, C\} \times \{D, E\}$. No choke point $\{B, E\} \times \{C, D\}$ can exist either. This follows from choke points $\{A, C\} \times \{B, D\}$, $\{A, D\} \times \{C, E\}$, which with $\{B, E\} \times \{C, D\}$ entail choke point $\{A, B\} \times \{C, D\}$ (Lemma B.9), which is assumed not to exist.

We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$. Choke points $\{B, C\} \times \{D, E\}$ and $\{B, D\} \times \{C, E\}$ are automatically excluded because of $\{A, C\} \times \{B, D\}$, $\{A, D\} \times \{C, E\}$ and Lemma B.10. Combining choke point $\{A, B\} \times \{C, E\}$ with choke point $\{A, D\} \times \{C, E\}$ will generate a choke point $\{B, D\} \times \{C, E\}$, which we just discarded. Therefore, there is no choke point $\{A, B\} \times \{C, E\}$. Combining choke point $\{B, D\} \times \{A, E\}$ with choke point $\{A, C\} \times \{B, D\}$ will generate a choke point $\{B, D\} \times \{C, E\}$, which we just discarded. Therefore, there is no choke point $\{B, D\} \times \{A, E\}$. Combining choke point $\{B, C\} \times \{A, E\}$ with $\{A, C\} \times \{B, D\}$ and $\{A, D\} \times \{C, E\}$ using Lemma B.11 will result in a choke point $\{A, E\} \times \{C, D\}$, which is discarded by hypothesis. Therefore, there is no choke point $\{B, C\} \times \{A, E\}$. Combining choke point $\{A, B\} \times \{D, E\}$ with $\{A, C\} \times \{B, D\}$ and $\{A, D\} \times \{C, E\}$ using Lemma B.11 will result in a choke point $\{A, B\} \times \{C, D\}$, which is discarded by hypothesis. Therefore, there is no choke point $\{A, B\} \times \{D, E\}$.

This means our model can entail only tetrad constraints generated by $\{A, C\} \times \{B, D\}$ and $\{A, D\} \times \{C, E\}$. Let G' have two latent variables L_1 and L_2. Make L_1 the parent of $\{A, C, E, L_2\}$. Let L_2 be the parent of B and D. Add bi-directed edges $B \leftrightarrow E$. Let G'' be G' with the added edge $L_2 \rightarrow A$.

Case 2.2.2: there is no choke point $\{A, D\} \times \{C, E\}$. As before, both $\{A, B\} \times \{C, D\}$ and $\{A, D\} \times \{B, C\}$ are forbidden. We consider two possible scenarios for choke point $\{A, E\} \times \{C, D\}$.

Case 2.2.2.1: there is a choke point $\{A, E\} \times \{C, D\}$. **We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$ and $\{X_1, X_2\} \times \{B, E\}$.** Choke point $\{B, E\} \times \{C, D\}$ does not exist, because this combined with $\{A, E\} \times \{C, D\}$ will result in $\{A, B\} \times \{C, D\}$, excluded by assumption. $\{B, E\} \times \{A, D\}$ cannot exist either: to see this, start from the constraint set $\{\{A, C\} \times \{B, D\}, \{A, E\} \times \{C, D\}, \{B, E\} \times \{A, D\}$. Exchanging the labels of D and E, followed by the exchange of E and C, this is equivalent to $\{\{A, E\} \times \{B, C\}, \{A, D\} \times \{C, E\}, \{B, D\} \times \{A, C\}$. From Lemma B.11, the constraint $\{B, D\} \times \{E, C\}$ is generated. Reverting the substitutions of E and C, and E and D, this is equal to $\{B, E\} \times \{C, D\}$ in the original labeling, which was ruled out at the beginning of this paragraph. A similar reasoning rules out $\{B, E\} \times \{A, C\}$. **We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$.** Choke point $\{A, B\} \times \{D, E\}$ cannot exist. Given the assumed choke point set $\{A, C\} \times \{B, D\}, \{A, E\} \times \{C, D\}, \{A, B\} \times \{D, E\}$, by exchanging labels A and C, one obtains $\{A, C\} \times \{B, D\}, \{A, D\} \times \{C, E\}, \{B, C\} \times \{D, E\}$, which by Lemma B.10 implies choke points among all elements in $\{A, B, C, D, E\}$. A similar reasoning rules out all other choke points of the type $\{X_1, B\} \times \{X_2, E\}$. Construct G' as follows: two latents, L_1 and L_2, where L_1 is a parent of A, C, E and L_2, and L_2 is a parent of B and D. Add the bi-directed edge $B \leftrightarrow E$. Construct G'' by adding edge $L_1 \rightarrow B$ to G'.

Case 2.2.2.2: there is no choke point $\{A, E\} \times \{C, D\}$. **We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$ and $\{X_1, X_2\} \times \{B, E\}$.** Choke point $\{A, C\} \times$

$\{B, E\}$ does not exist, because this combined with $\{A, C\} \times \{B, D\}$ generates $\{A, C\} \times \{D, E\}$. Choke points $\{A, D\} \times \{B, E\}$ and $\{C, D\} \times \{B, E\}$ cannot both exist, since they jointly imply choke point $\{A, C\} \times \{B, E\}$.

Assume for now that choke point $\{A, D\} \times \{B, E\}$ exists (but not $\{C, D\} \times \{B, E\}$). We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$. Choke point $\{A, B\} \times \{C, E\}$ cannot exist, since by exchanging A and D, B and C in set $\{\{A, C\} \times \{B, D\}, \{A, D\} \times \{B, E\}, \{A, B\} \times \{C, E\}$ we get $\{\{A, C\} \times \{B, D\}, \{A, D\} \times \{B, E\}, \{B, E\} \times \{C, D\}$, which by Lemma B.9 will imply all tetrad constraints with $\{A, B, C, D\}$.

The same reasoning applies to $\{B, C\} \times \{A, E\}$ (exchanging A and D, B and C in the given tetrad constraints) by using Lemma B.10. The same reasoning applies to $\{B, C\} \times \{D, E\}$ (exchanging A and D, B and C in the given tetrad constraints) by using Lemma B.11.

Because of the assumed $\{A, C\} \times \{B, D\}$, either both choke points $\{A, E\} \times \{B, D\}$, $\{C, E\} \times \{B, D\}$ exist or none exists. Because of the assumed $\{A, D\} \times \{B, E\}$, either both choke points $\{A, E\} \times \{B, D\}$, $\{A, D\} \times \{B, E\}$ exist or none exists. That is, either all choke points $\{\{A, E\} \times \{B, D\}, \{A, D\} \times \{B, E\}, \{C, E\} \times \{B, D\}\}$ exist or none exist. If all exist, create G' as follows: use two latents L_1, L_2, where L_1 is a parent of A, C and L_2, L_2 is a parent of B, D and E, and there is a bi-directed edge $C \leftrightarrow E$. Construct G'' by adding edge $L_2 \rightarrow A$ to G'. If none of the three mentioned choke points exist, do the same but with an extra bi-directed edge $B \leftrightarrow E$.

Assume now that choke point $\{C, D\} \times \{B, E\}$ exists (but not $\{A, D\} \times \{B, E\}$). This is analogous to the previous case by symmetry of A and C.

Assume now that no choke point $\{C, D\} \times \{B, E\}$ or $\{A, D\} \times \{B, E\}$ exists. We only have to consider now choke points of the form $\{X_1, B\} \times \{X_2, E\}$. Let Pair 1 be the set of choke points $\{\{A, B\} \times \{C, E\}, \{A, B\} \times \{D, E\}\}$. Let Pair 2 be the set of choke points $\{\{B, C\} \times \{A, E\}, \{B, C\} \times \{D, E\}\}$. Let Pair 3 be the set of choke points $\{\{B, D\} \times \{A, E\}, \{B, D\} \times \{C, E\}\}$. At most one element of Pair 1 can exist (or otherwise it will entail $\{A, B\} \times \{C, D\}$). For the same reason, at most one element of Pair 2 can exist. Either both elements of Pair 3 exist or none exist.

If both elements of Pair 3 exist, then no element of Pair 1 or Pair 2 can exist. For example, $\{B, D\} \times \{A, E\}$ from Pair 3 and $\{B, C\} \times \{A, E\}$ from Pair 2 together entail $\{C, D\} \times \{A, E\}$, discarded by hypothesis. In the case where both elements of Pair 3 exist, construct G' as follows: let L_1 and L_2 be two latents, where L_1 is a parent of A, C and L_2, and L_2 is a parent of B, D and E. Add bi-directed edges $A \leftrightarrow E$ and $C \leftrightarrow E$. Construct G'' by adding $L_2 \rightarrow A$ to G'.

Choke point $\{B, C\} \times \{D, E\}$ (from Pair 2) cannot co-exist with $\{A, B\} \times \{D, E\}$ (from Pair 1) since this entails $\{A, C\} \times \{D, E\}$. Moreover, $\{B, C\} \times \{D, E\}$ cannot co-exist with $\{A, B\} \times \{C, E\}$ (also from Pair 1), since $\{\{A, C\} \times \{B, D\}, \{A, B\} \times \{C, E\}, \{B, C\} \times \{D, E\}\}$, which by exchanging B with D generates $\{\{A, C\} \times \{B, D\}, \{A, D\} \times \{C, E\}, \{B, E\} \times \{C, D\}\}$. From Lemma B.9, this implies all three tetrads in the covariance of $\{A, B, C, D\}$, a contradiction.

By symmetry between A and C, it follows that no two elements of the union of Pair 1 and Pair 2 can simultaneously exist. Let $\{X_1, B\} \times \{X_2, E\}$ be a choke point in the union of Pair 1 and Pair 2 that is assumed to exist. Construct G' as follows: let L_1 and L_2 be two latents, where L_1 is a parent of A, C and L_2, and L_2 is a parent of B, D. If $X_1 = A$ and $X_2 = C$, or if $X_1 = C$ and $X_2 = A$, let L_1 be the parent of E. Otherwise, let L_2 be the parent of E. Add bi-directed edges between E and every element in $\mathbf{X} \backslash \{B, X_1\}$. Construct G'' by adding $L_2 \rightarrow A$ to G'.

Finally, if no element in Pairs 1, 2 or 3 is assumed to exist, create G' and G'' as above, but connect E to all other elements of $\mathbf{X}$ by bi-directed edges. □

Appendix C

Results from Chapter 6

C.1 Update equations for variational approximation

Following the notation in Chapter 6, the equations below provide the update steps on the optimization of the variational lower bound:

1. Optimizing $q(\pi)$ and a^*:

$$q(\pi) = Dirichlet(\pi|a\mathbf{m}) \tag{C.1}$$

where for each element am_s in $a\mathbf{m}$,

$$am_s = a^* m_s^* + \sum_{i=1}^{n} q(s_i) \tag{C.2}$$

To optimize a^*, one has to appeal for a gradient-based technique, such as Newton-Raphson (Beal and Ghahramani, 2003). The gradient is given by:

$$\frac{\partial \mathcal{F}(G, \mathbf{D})}{\partial a^*} = \Psi(a^*) - \Psi\left(\frac{a^*}{S}\right) - \frac{1}{S}\sum_{s=1}^{S}[\Psi(a) - \Psi(am_s)] \tag{C.3}$$

where $\Psi(x)$ here is the digamma function, the derivative of the logarithm of the gamma function.

2. Optimizing $q(\mathbf{B})$ and $\upsilon_{\mathbf{L}}^*$:

Let $\langle g(\mathbf{V})\rangle_{q(\mathbf{V})}$ denote the expected value of $g(\mathbf{V})$ according to the distribution $q(\mathbf{V})$. Since the prior probability of elements in $\mathbf{B^s}$ is a product of marginals for each element in this set, its posterior distribution for will also factorize over each $L_i^{(k)} \in \mathbf{L}_i$, where $1 \leq i \leq n$ is an index over data points, n being the size of the data set. Let $\{L^{(k1)}, \ldots, L^{(km_k)}\} \subset \mathbf{L}$ be the parents of $L^{(k)}$ in G. Let β_{kjs} be the parameter associated with edge $L^{(kj)} \rightarrow L^{(k)}$ in mixture component s. Then the variational posterior distribution $q(\mathbf{B})$ is given by

$$q(\mathbf{B}) = \prod_{k=1}^{|\mathbf{L}|} q(\mathbf{B}_{L_k}^s) \equiv \prod_{k=1}^{|\mathbf{L}|} N(\mathbf{V}_{L_k}\mathbf{M}_{L_k}, \mathbf{V}_{L_k}^{-1}), \tag{C.4}$$

$$\mathbf{B}^{s'}_{L_k} = [\beta_{k1s} \dots \beta_{km_k s}], \tag{C.5}$$

$$\mathbf{M}_{L_{kj}} = \sum_{i=1}^{n} q(s_i)\zeta_{ks} \left\langle L_i^{(k)} L_i^{(kj)} \right\rangle_{q(\mathbf{L}_i|s_i)} \tag{C.6}$$

$$\mathbf{V}_{L_{kjl}} = \sum_{i=1}^{n} q(s_i)\zeta_{ks} \left\langle L_i^{(kj)} L_i^{(kl)} \right\rangle_{q(\mathbf{L}_i|s_i)} + \mathbf{1}(j = l) \times \upsilon^*_{\mathbf{L}} \tag{C.7}$$

where $1 \leq j \leq m_k, 1 \leq l \leq m_k$ and $\mathbf{1}(T) = 1$ if and only if expression T is true, and 0 otherwise.

Moreover,

$$(\upsilon^*_{\mathbf{L}})^{(-1)} = \frac{\sum_{s=1}^{S} \sum_{k=1}^{|\mathbf{L}|} \left\langle \mathbf{B}^{s'}_{L_k} \mathbf{B}^{s}_{L_k} \right\rangle_{q(\mathbf{B^s})}}{|\mathbf{B}|} \tag{C.8}$$

where $|\mathbf{B}|$ is the number of elements in $\mathbf{B}$.

3. Optimizing ζ_{ks}, $1 \leq k \leq |\mathbf{L}|$, $1 \leq s \leq S$:

$$\zeta_{ks} = \sum_{i=1}^{n} q(s_i) / \sum_{i=1}^{n} q(s_i) \left\langle (L^{(k)} - \sum_{j=1}^{m_k} \beta_{kjs} L^{(kj)})^2 \right\rangle_{q(\mathbf{L_i}|s_i)q(\mathbf{B^s})} \tag{C.9}$$

4. Optimizing $q(\mathbf{L}_i|s_i)$:

Let $\aleph^s$ be the diagonal matrix such that $\aleph^s_{kk}$ is the corresponding inverse variance ζ_{ks}. Let $\mathbf{B^{s_i}}$ be a matrix of coefficients such that entry $b_{kj} = 0$ if there is no edge $L^{(j)} \to L^{(k)}$ in G. Otherwise, let b_{kj} correspond to the parameter associated with edge $L^{(j)} \to L^{(k)}$ in mixture component s_i.

Let $Ch_X(L^{(k)})$ and $Ch_L(L^{(k)})$ be the children of $L^{(k)}$ in $\mathbf{X}$ and $\mathbf{L}$, respectively. Let $Ch_X(L^{(j)}, L^{(k)}) = Ch_X(L^{(k)}) \cap Ch_X(L^{(j)})$. Let λ_{tks_i} be the parameter associated with edge $L^{(k)} \to X^{(t)}$ in mixture component s_i. Let $Pa_X(X^{(t)})$ be the parents of $X^{(t)}$ in $\mathbf{X}$, and let λ_{tvs_i} be the parameter associated with edge $X^{(v)} \to X^{(t)}$ in mixture component s_i. Finally, let $\mathbf{I}$ be the identity matrix of size $|\mathbf{L}|$.

We optimize the variational posterior $q(\mathbf{L}_i|s_i)$ by:

$$q(\mathbf{L}_i|s_i) = N((\mathbf{V^1} + \mathbf{V^2})\mathbf{M}, (\mathbf{V^1} + \mathbf{V^2})^{-1}), \tag{C.10}$$

$$\mathbf{M}_k = \sum_{X^{(t)} \in Ch_X(L^{(k)})} \Psi_t \left[X_i^{(t)} \left\langle \lambda_{tks_i} \right\rangle_{q(\mathbf{\Lambda^{s_i}})} - \sum_{v \in Pa_X(X^{(t)})} \left\langle \lambda_{tvs_i} \lambda_{tks_i} \right\rangle_{q(\mathbf{\Lambda^{s_i}})} \right], \tag{C.11}$$

$$\mathbf{V^1} = \left\langle (\mathbf{I} - \mathbf{B^{s_i}}) \aleph (\mathbf{I} - \mathbf{B^{s_i}})' \right\rangle_{q(\mathbf{B^{s_i}})} \tag{C.12}$$

$$\mathbf{V^2}_{jk} = \sum_{X^{(t)} \in Ch(L^{(j)}, L^{(k)})} \psi_t \left\langle \lambda_{tjs_i} \lambda_{tks_i} \right\rangle_{q(\mathbf{\Lambda^{s_i}})} \tag{C.13}$$

5. Optimizing $q(\mathbf{\Lambda})$ and $\upsilon^*_{\mathbf{X}}$:

Let $\{Z^{(k1)}, \ldots, Z^{(km_k)}\} \subset \mathbf{L} \cup \mathbf{X} \cup \{1\}$ be the parents of $X^{(k)}$ in G. Let λ_{kjs} be the parameter associated with edge $Z^{(kj)} \rightarrow X^{(k)}$. By convention, let $Z^{(k1)}$ be the intercept term among the parents of $X^{k)}$ (i.e., $Z^{(k1)}$ is constant and set to 1). Then the variational posterior distribution $q(\mathbf{\Lambda})$ is given by

$$q(\mathbf{\Lambda}) = \prod_{s=1}^{S} \prod_{k=1}^{|\mathbf{X}|} q(\mathbf{\Lambda}^{\mathbf{s}}_{X_k}) \equiv \prod_{s=1}^{S} \prod_{k=1}^{|\mathbf{X}|} N(\mathbf{V}_{X_k s}\mathbf{M}_{X_k s}, \mathbf{V}^{-1}_{X_k s}), \tag{C.14}$$

$$\mathbf{\Lambda}^{\mathbf{s}'}_{X_k} = [\lambda_{k1s} \ldots \lambda_{km_k s}], \tag{C.15}$$

$$\mathbf{M}_{X_k j s} = \sum_{i=1}^{n} q(s_i = s)\psi_k \left\langle Z_i^{(k)} Z_i^{(kj)} \right\rangle_{q(\mathbf{L}_i|s)} \tag{C.16}$$

$$\begin{aligned} \mathbf{V}_{X_{kjl}s} \quad = \quad & \textstyle\sum_{i=1}^{n} q(s_i = s)\psi_k \left\langle Z_i^{(kj)} Z_i^{(kl)} \right\rangle_{q(\mathbf{L}_i|s)} \\ & + \mathbf{1}(j = l \cap \{j > 1\}) \times \upsilon^*_{X^{(k)}} \\ & + \mathbf{1}(j = l \cap \{j = 1\}) \times \upsilon^t_{X^{(k)}} \end{aligned}$$

where $1 \leq j \leq m_k, 1 \leq l \leq m_k$, and $\mathbf{1}(T) = 1$ if and only if expression T is true, and 0 otherwise.

Moreover,

$$(\upsilon^*_{X^k})^{(-1)} = \frac{\sum_{s=1}^{S} \sum_{j>1} \left\langle \lambda^2_{kjs} \right\rangle_{q(\mathbf{\Lambda}^{\mathbf{s}})}}{|\mathbf{\Lambda}^{\mathbf{s}'}_{X_k}| - S} \tag{C.17}$$

$$(\upsilon^t_{X^k})^{(-1)} = \frac{\sum_{s=1}^{S} \left\langle \lambda^2_{k1s} \right\rangle_{q(\mathbf{\Lambda}^{\mathbf{s}})}}{S} \tag{C.18}$$

6. Optimizing ψ_k, $1 \leq k \leq |\mathbf{L}|$:

$$\psi_k = \sum_{s=1}^{S} \sum_{i=1}^{n} q(s) / \sum_{s=1}^{S} \sum_{i=1}^{n} q(s) \left\langle (X^{(k)} - \sum_{j=1}^{m_k} \lambda_{kjs} Z^{(kj)})^2 \right\rangle_{q(\mathbf{L}_i|s)q(\mathbf{\Lambda_k})} \tag{C.19}$$

where for each $X^{(k)}$, $\{Z^{(k1)}, \ldots, Z^{(km_k)}\}$ are the parents of $X^{(k)}$ in G.

7. Optimizing $q(s_i)$:

$$\begin{aligned} q(s_i) \quad = \quad & \tfrac{1}{\mathcal{Z}} \exp[\psi(\alpha m_{s_i}) - \psi(\alpha) + \langle \ln p(\mathbf{L}_i|s_i) \rangle_{q(\mathbf{L_i}|s_i)q(\mathbf{B^{s_i}})} + \tfrac{1}{2} \ln |\Sigma^{s_i}| \\ & - \tfrac{1}{2} tr \left[\Psi \left\langle (\mathbf{X}_i - \mathbf{\Lambda^{s_i}} \mathbf{Z_i}) (\mathbf{X}_i - \mathbf{\Lambda^{s_i}} \mathbf{Z_i})' \right\rangle_{q(\mathbf{L_i}|s_i)q(\mathbf{\Lambda^{s_i}})} \right]] \end{aligned}$$

where Σ^{s_i} is the covariance of $\mathbf{L}$ given $s = s_i$, and $\mathcal{Z}$ is a normalizing constant to ensure $\sum_{s_i=1}^{S} q(s_i) = 1$.

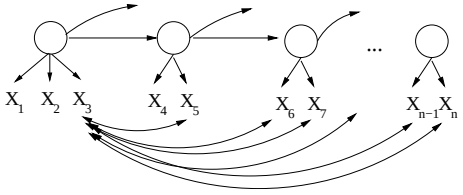


Figure C.1: Let F be a pure one-factor model consisting of indicators $X_1, \ldots, X_n$, and let the true graph among such variables be given as above, where all latent variables are connected and X_3 is connected by bi-directed edges to all other variables not in $\{X_1, X_2, X_4\}$. Variable X_3 will be the one present in the highest number of tetrad constraints that are entailed by F that do not hold in the population.

C.2 Problems with WASHDOWN

The intuition behind WASHDOWN is that nodes that participate in the highest number of invalid tetrad constraints will be the first ones to be eliminated. This should be seen as a heuristic since typical score functions, such as the one suggested in Chapter 6, also take into account quantitative characteristics of the invalid tetrad constraints (i.e., how much they deviate from zero in reality, and not only if the constraint is entailed or not).

However, even if the given score function perfectly ranks models according to the number of invalid tetrad constraints (i.e., where the models with the least number of false constraints will be the ones that will achieve the highest score), this is still not enough to guarantee that WASHDOWN will find a pure measurement model if one exists, as formalized by the next theorem.

Theorem C.1 *Let* **O** *be the set of variables in the dataset given as input to* WASHDOWN. *Assume the score function is the negative number of invalid tetrad constraints that are entailed by the model (so that the best ranked models will be the ones with the least number of invalid entailed tetrad constraints). Then, even if there is some sequence of node deletions (from the one-factor model given at the start of* WASHDOWN*) that creates a pure model with at least three nodes per latent,* WASHDOWN *might not follow any of such sequences.*

Proof: A counter-example can be easily constructed by having an unique set of four indicators that can form a pure one-factor model, and making one of such variables belong to many entailed tetrad constraints that are violated in the population. Figure C.1 illustrates such a case. The only possible one-factor model that can be formed contains variables $X_1 - X_4$. Variable X_3 is present in the highest number of invalid tetrad constraints (by having the number of latents much higher than 4), and will be removed from F in the next step if the score function satisfies the assumptions of the theorem. No other subset of four variables can form a one-factor model, and in the end the empty graph G_0 will have a higher score (assuming consistency of the score function) over whatever set is selected by WASHDOWN. □

C.3 Implementation details

In our implementation of WASHDOWN, we used STRUCTURAL EM (Friedman, 1998) to speed-up the choice of node to be removed. Given a model of n indicators, we calculate the n possible submodels by fixing the distribution of the latents given the data, and then estimating the other parameters of the model before scoring it. Once a node is chosen to be removed, we estimate the full model again and compare it to the current score. This way, the number of full score evaluations is never higher than the number of observable variables for each new cluster that is introduced. For larger sample sizes, one might want to re-estimate the full model only when a local maxima is achieved in order to achieve a much higher speed-up. We did not perform any empirical study on how this might hurt the accuracy of the algorithm.

We actually did not apply the variational score in most of the implementation of WASHDOWN used in the experiments on causal discovery. The reason was the sensitivity of the score function to the initial choice of parameter values: many different local maxima could be generated. Doing multiple re-starts from a large number of initial parameter values slows down the method considerably. Therefore, instead of using the variational score for choosing the node to be removed, we used BIC. The likelihood function is not as sensitive (since there are no hyperparameters to be fit). We still needed to do multiple re-starts (five, in our implementation), but the variance on the score per trial was not as high, and therefore we do not need as many as we would need with the variational function.

However, one can verify in synthetic experiments that the BIC score is considerably less precise than the variational one, underfitting the data much more easily. This is partially due to the difficulty of the problem: in Gaussian models, for instance, a χ^2 test would frequently accept with high significance (> 0.20) a false one-factor model that in reality would contain several nodes from different clusters.

To minimize this problem, we added an extra step to our implementation: suppose X_i is the best choice of node to be removed, but the model where all other indicators are parents of X_i (as in Figure 6.4) still scores less than the pure model with no extra edges. Instead of stopping removing nodes, we do a greedy search that tries to add some edge $X_j \rightarrow X_i$ to the current pure model if that increases the score. If after this search we have some edge $X_j \rightarrow X_i$, we remove X_i and proceed to the next iteration of node removal. This modification is essential for making WASHDOWN work reasonably well with the BIC score function.

A less elegant modification was added on top of that at the end of each cycle of WASHDOWN, before we perform a GRAPHCOMPARISON. We again do a greedy search to add edges between indicators but now without restring which nodes can be at the endpoints, unlike the procedure given in the previous paragraph. If some edge $A \rightarrow B$ is added, we remove node B. A new search for the next edge is done, and we stop when no edge can increase the score of the model.

The variational score function was still used in GRAPHCOMPARISON and MIMBUILD. In our experiments with density estimation, we did not use the BIC score at all, and consequently none of the modifications above, since they slow down the procedure (we did not increase the number of score function evaluations per trial). It would be interesting to compare in a future work if these modifications would result in a better probabilistic model for the given datasets.

Another heuristic that we adopted was requiring a minimum number of indicators per latent. In the case of the regular WASHDOWN, we forced the algorithm to keep at least three indicators per latent all the time (or four indicators, if there is only one latent). If the absence of some node would imply a model without three indicators per latent, then this node would not be considered

for removal. In simulations, this seems to help to increase the accuracy of the model, avoiding unnecessary fragmentation of clusters. The number 3 was chosen since this is the minimum number of indicators to make a single latent factor identifiable (if there is more than one latent, a fourth descendant is avaliable as the child of another latent - otherwise, we require 4 indicators for an one-factor model to be testable). Notice that the original WASHDOWN algorithm of Silva (2002) does not impose this restriction.

In the case of K-LATENTCLUSTERING, which allows multiple latent parents per indicator, we applied a generalized version of this heuristic. Instead of requiring at least 3 indicators per cluster as in WASHDOWN (where each cluster has only one latent parent), we require at least p indicators for a cluster of k latents, where p is the minimum integer such that $p(p+1)/2 \geq kp + p$. That is, the minimum number of indicators such that the number of unique entries in the observed covariance matrix ($p(p+1)/2$) is at least as large as the number of covariance parameters (per mixture component) in the measurement model of the cluster ($kp + p$).

Concerning the bi-directed that are used in the description of FULLLATENTCLUSTERING, we chose not to parameterize them as covariances among the residuals as it is done, e.g., in the Gaussian mixed ancestral graph representation of Richardson and Spirtes (2002), mostly due to the difficulty of defining priors over such graphs and performing parameter fitting under the STRUCTURAL EM framework, as explained in the next paragraphs. Instead, each "bi-directed" edge is just a shorthand representation of a new independent hidden common cause of two children.

That is, each bi-directed edge $X_1 \leftrightarrow X_2$ represents a new independent latent $X_1 \leftarrow L \rightarrow X_2$. The goal is to free the covariance $\sigma_{X_1 X_2}$ across every component of the mixture model, increasing the rank of the covariance matrix only on subsets of the observed variables that include X_1 and X_2, while leaving all other covariances untouched[1].

Concerning bi-directed edges and STRUCTURAL EM, we introduce yet another approximation. Let L_{new} be the new hidden variable associated with the "bi-directed" edge $X_1 \leftrightarrow X_2$, and let $\mathbf{L}$ be the current set of latents. We introduce the variational approximation $q(\mathbf{L} \cup \{L_{new}\}) \sim q(\mathbf{L})q(L_{new})$, fixing $q(\mathbf{L})$ and updating only $q(L_{new})$. This still requires fitting a latent variable model for each evaluation, however a model with only one latent and only the edges into X_1 and X_2, which is relatively efficient. Notice this is still a lower bound on the true function. After deciding which bi-directed edge increases the score most (if any), we introduce it into the graph and evaluate the full log-posterior score function.

[1]Those are still not completely free to vary, since the full covariance matrix is constrained to be positive definite.

Bibliography

R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, 1994.

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

F. Bach and M. Jordan. Learning graphical models with Mercer kernels. *Neural Information Processing Systems*, 2002.

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

D. Bartholomew. *Measuring Intelligence: Facts and Falacies.* Cambridge University Press, 2004.

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

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

A. Basilevsky. *Statistical Factor Analysis and Related Methods.* Wiley, 1994.

M. Beal and Z. Ghahramani. The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian Statistics*, 7, 2003.

M. Beal, Z. Ghahramani, and C. Rasmussen. The infinite hidden markov model. *Advances in Neural Information Processing Systems*, 14, 2001.

J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29:213–244, 1997.

C. Bishop. Latent variable models. *Learning in Graphical Models*, 1998.

C. L. Blake and C. J. Merz. UCI repository of machine learning databases, http://www.ics.uci.edu/∼mlearn/mlrepository.html, 1998.

D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, pages 993–1022, 2003.

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

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

K. Bollen. Modeling strategies: in search of the holy grail. *Structural Equation Modeling*, 7:74–81, 2000.

K. Bollen and P. Paxton. Interactions of latent variables in structural equation models. *Structural Equation Modeling*, 5:267–293, 1998.

C. Borgelt and R. Kruse. Induction of association rules: Apriori implementation. *15th Conference on Computational Statistics (Compstat 2002, Berlin, Germany)*, 2002.

W. Buntine and A. Jakulin. Applying discrete PCA in data analysis. *Proceedings of 20th Conference on Uncertainty in Artificial Intelligence*, 2004.

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

M. Carreira-Perpinan. *Continuous Latent Variable Models for Dimensionality Reduction and Sequential Data Reconstruction*. PhD Thesis, University of Sheffield, UK, 2001.

R. Carroll, D. Ruppert, C. Crainiceanu, T. Tosteson, and M. Karagas. Nonlinear and nonparametric regression and instrumental variables. *Journal of the American Statistical Association*, 99:736–750, 2004.

D. Chakrabarti, S. Papadimitriou, D. Modha, and C. Faloutsos. Fully automatic cross-associations. *Proceedings of Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 79–88, 2004.

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

W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Technical Report, University College London*, 2004.

G. Cooper. A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Mining and Knowledge Discovery*, 2, 1997.

G. Cooper. An overview of the representation and discovery of causal relationships using Bayesian networks. *Computation, Causation and Discovery*, pages 3–62, 1999.

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

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

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

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

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

Z. Ghahramani and M. Beal. Variational inference for Bayesian mixture of factor analysers. *Advances in Neural Information Processing Systems*, 12, 1999.

Z. Ghahramani and G. Hinton. The EM algorithm for the mixture of factor analyzers. *Technical Report CRG-TR-96-1. Department of Computer Science, University of Toronto.*, 1996.

J. Gibson. *Freedom and Tolerance in the United States.* Chicago, IL: University of Chicago, National Opinion Research Center [producer], 1987. Ann Arbor, MI: Inter-university Consortium for Political and Social Research [distributor], 1991.

C. Glymour. *The Mind's Arrow: Bayes Nets and Graphical Causal Models in Psychology.* MIT Press, 2002.

C. Glymour and G. Cooper. *Computation, Causation and Discovery.* MIT Press, 1999.

C. Glymour, Richard Scheines, Peter Spirtes, and Kevin Kelly. *Discovering Causal Structure: Artificial Intelligence, Philosophy of Science, and Statistical Modeling.* Academic Press, 1987.

M. Grzebyk, P. Wild, and D. Chouaniere. On identification of multi-factor models with correlated residuals. *Biometrika*, 91:141–151, 2004.

B. Habing. Nonparametric regression and the parametric bootstrap for local dependence assessment. *Applied Psychological Measurement*, 25:221–233, 2001.

H. Harman. *Modern Factor Analysis.* University of Chicago Press, 1967.

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

D. Heckerman. A bayesian approach to learning causal networks. *Proceedings of 11th Conference on Uncertainty in Artificial Intelligence*, pages 285–295, 1995.

D. Heckerman. A tutorial on learning with Bayesian networks. *Learning in Graphical Models*, pages 301–354, 1998.

A. Hyvarinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

A. Jackson and R. Scheines. Single mother's self-efficacy, parenting in the home environment and children's development in a two-wave study. *Submitted to Social Work Research*, 2005.

R. Johnson and D. Wichern. *Applied Multivariate Statistical Analysis.* Prentice Hall, 2002.

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

K. Joreskog. Structural Equation Modeling with Ordinal Variables using LISREL. *Technical Report, Scientific Software International Inc.*, 2004.

B. Junker and K. Sijtsma. Nonparametric item response theory in action: An overview of the special issue. *Applied Psychological Measurement*, 25:211–220, 2001.

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

Y. Kano and S. Shimizu. Causal inference using nonnormality. *Proceedings of the International Symposium of Science of Modeling - The 30th anniversary of the Information Criterion (AIC)*, pages 261–270, 2003.

R. Klee. *Introduction to the Philosophy of Science: Cutting Nature at its Seams*. Oxford University Press, 1996.

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

E. Malinowski. *Factor Analysis in Chemistry*. John Wiley & Sons, 2002.

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

T. Minka. Automatic choice of dimensionality for pca. *Advances in Neural Information Processing Systems*, 13:598–604, 2000.

T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

J. Pan, C. Faloutsos, M. Hamamoto, and H. Kitagawa. Autosplit: Fast and scalable discovery of hidden variables in stream and multimedia databases. *PAKDD*, 2004.

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

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

T. Richardson and P. Spirtes. Ancestral graph Markov models. *Annals of Statistics*, 30:962–1030, 2002.

K. Roeder and L. Wasserman. Practical bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association*, pages 894–902, 1997.

P. Rosebaum. *Observational studies*. Springer-Verlag, 2002.

B. Scholkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

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

R. Silva. The structure of the unobserved. *MSc. Thesis, Center for Automated Learning and Discovery. Technical Report CMU-CALD-02-102, School of Computer Science, Carnegie Mellon University*, 2002.

R. Silva and R. Scheines. Generalized measurement models. *Technical Report CMU-CALD-04-101, Carnegie Mellon University*, 2004.

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

C. Silverstein, S. Brin, R. Motwani, and J. Ullman. Scalable techniques for mining causal structures. *Data Mining and Knowledge Discovery*, 2000.

C. Spearman. "general intelligence," objectively determined and measured. *American Journal of Psychology*, 15:210–293, 1904.

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

E. Stanghellini and N. Wermuth. On the identification of path analysis models with one hidden variable. *Biometrika*, 92:To appear, 2005.

W. Stout. A new item response theory modeling approach with applications to unidimensionality assessment and ability estimation. *Psychometrika*, 55:293–325, 1990.

M. Wall and Y. Amemiya. Estimation of polynomial structural equation models. *Journal of the American Statistical Association*, 95:929–940, 2000.

M. Wedel and W. Kamakura. Factor analysis with (mixed) observed and latent variables in the exponential family. *Psychometrika*, 66:515–530, 2001.

J. Wegelin, A. Packer, and T. Richardson. Latent models for cross-covariance. *Journal of Multivariate Analysis*, page in press, 2005.

J. Wishart. Sampling errors in the theory of two factors. *British Journal of Psychology*, 19:180–187, 1928.

I. Yalcin and Y. Amemiya. Nonlinear factor analysis as a statistical method. *Statistical Science*, 16:275–294, 2001.

M. Zaki. Mining non-redundant association rules. *Data Mining and Knowledge Discovery*, 19: 223–248, 2004.

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