JMLR: Workshop and Conference Proceedings 25:1–16, 2012 Asian Conference on Machine Learning

Learning Latent Variable Models by Pairwise Cluster Comparison

Nuaman Asbeh ASBEH@POST.BGU.AC.IL
Boaz Lerner BOAZ@BGU.AC.IL
Dept. of Industrial Engineering and Management, Ben-Gurion University of the Negev, Israel

Editor: Steven C.H. Hoi and Wray Buntine

Abstract

Identification of latent variables that govern a problem and the relationships among them given measurements in the observed world are important for causal discovery. This identification can be made by analyzing constraints imposed by the latents in the measurements. We introduce the concept of *pairwise cluster comparison PCC* to identify causal relationships from clusters and a two-stage algorithm, called LPCC, that learns a latent variable model (LVM) using PCC. First, LPCC learns the exogenous and the collider latents, as well as their observed descendants, by utilizing pairwise comparisons between clusters in the measurement space that may explain latent causes. Second, LPCC learns the non-collider endogenous latents and their children by splitting these latents from their previously learned latent ancestors. LPCC is not limited to linear or latent-tree models and does not make assumptions about the distribution. Using simulated and real-world datasets, we show that LPCC improves accuracy with the sample size, can learn large LVMs, and is accurate in learning compared to state-of-the-art algorithms.

Keywords: learning latent variable models, graphical models, clustering

1. Introduction

Statistical methods, such as factor analysis, are most commonly used to reveal the existence and influence of latent variables. While these methods accomplish effective dimensionality reduction and may fit the data reasonably well, the resulting models might not have any correspondence to real causal mechanisms (Silva et al., 2006). On the other hand, the focus of learning Bayesian networks (BNs) is on relations among observed variables, while the detection of latent variables and their interrelations has received little attention. Learning latent variable models (LVMs) using Inductive Causation* (IC*) (Pearl, 2000) and Fast Causal Inference (FCI) (Spirtes et al., 2000) returns partial ancestral graphs, which indicate for each link whether it is a (potential) manifestation of a hidden common cause for the two linked variables. The structural EM algorithm (Friedman, 1998) learns a structure using a fixed set of previously given latents. By searching for "structural signatures" of latents, substructures that suggest the presence of latents (in the form of dense sub-networks) can be detected (Elidan et al., 2000). Also, the recovery of latent trees of binary and Gaussian variables has been suggested (Pearl, 2000). Hierarchical latent class (HLC) models of discrete variables, where observed variables are mutually independent given the latents, are learned for clustering (Zhang, 2004).

However, for models that are not tree-constrained, e.g., models where multiple latents may have multiple indicators (observed children), i.e., multiple indicator models (MIM), most of these algorithms may lead to unsatisfactory results. MIM are a subclass of structural equation models (SEM) that are widely used in applied and social sciences together with BN to analyze causal relations (Shimizu et al., 2011). An attempt to fill the gap between latent-tree models and MIM has recently been made (Silva et al., 2006), but it was limited to linear models of continuous variables. In this study, we make another attempt in this direction and target the goal of Silva et al. (2006), but concentrate on the discrete case and dispense with the linearity assumption. We propose a concept and an algorithm that combine learning causal graphical models with clustering. The concept and algorithm learn a causal LVM by comparing clusters of data representing the observed variables.

2. Preliminaries

The goal of our study is to reconstruct an LVM from i.i.d. data sampled from the observed variables in the unknown model. To accomplish this, we propose learning pairwise cluster comparison (LPCC) that assumes: 1) The underlying model is $BN = \langle \mathbf{G}, \mathbf{\Theta} \rangle$ encoding a discrete joint probability distribution P for the set of random variables $\mathbf{V} = \mathbf{L} \cup \mathbf{O}$, where $\mathbf{G}$ is a DAG whose nodes correspond to the latents $\mathbf{L}$ and observed variables $\mathbf{O}$. $\mathbf{\Theta}$ is the set of parameters, i.e., conditional probabilities of variables in $\mathbf{V}$ given their latent parents. 2) The underlying model is MIM, in which each latent has at least two observed children and may have latent parents (e.g., G3 in Figure 1). Notice that the model is not limited to a tree as in Zhang (2004), since latent variables may also be colliders (e.g., G2), but latent tree models are a sub-class of MIM. 3) The measurement model (Silva, 2005) of $\mathbf{G}$ is pure, i.e., each observed variable has only one latent parent and no observed parent. Silva (2005) focuses on such models as a principled way of testing conditional independence among latents. If $\mathbf{G}$ is not pure, LPCC learns a pure sub-model of $\mathbf{G}$, if one exist, similar to Silva (2005), but Silva (2005) requires that each latent has at least three indicators, whereas LPCC requires only two. Based on assumptions 2 and 3, the observed variables in $\mathbf{G}$ are d-separated given the latents.

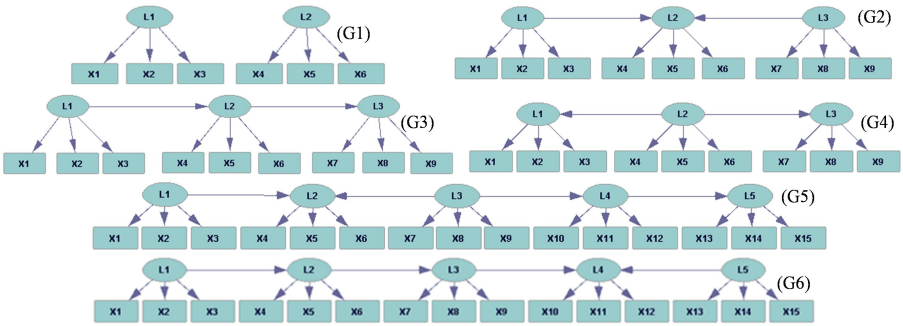

Figure 1: Four basic example LVMs (G1-G4) and two larger graphs combining all types of links between the latents (G5 and G6) that are learned by LPCC.

We distinguish between observed and latent variables and between exogenous (**EX**) and endogenous (**EN**) variables. **EX** have zero in-degree, are autonomous and unaffected by the values of the other variables (e.g., L1 in G3), whereas **EN** are all non-exogenous variables in **G** (e.g., L2/X1 in G3). We identify three types of variables: 1) Exogenous latents, $\mathbf{EX} \subset \mathbf{L}$ (all exogenous variables are latents); 2) Endogenous latents, $\mathbf{EL} \subset (\mathbf{L} \cap \mathbf{EN})$, that are divided into collider latents (e.g., L2 in G2) and non-collider latents (e.g., L2/L3 in G3 and L1/L3 in G4); and 3) Observed variables, $\mathbf{O} \subset \mathbf{EN}$, which are always endogenous and childless. We denote value configurations of **EX**, **EL**, **O** and **EN** (when we do not know whether the endogenous variables are latent or observed) by **ex**, **el**, **o** and **en** respectively.

The joint probability distribution over **V** that is represented by a BN, which is assumed to encode the distribution is:

$$P(\mathbf{V}) = \prod_{V_i \in \mathbf{V}} P(V_i = |\mathbf{Pa_i}) \tag{1}$$

where $\mathbf{Pa_i}$ are the parents of V_i in **G**. It can be factorized under our assumptions as:

$$P(\mathbf{V}) = P(\mathbf{EX}, \mathbf{EL}, \mathbf{O}) = \prod_{EX_i \in \mathbf{EX}} P(EX_i) \prod_{EL_j \in \mathbf{EL}} P(EL_j|\mathbf{L_j}) \prod_{O_k \in \mathbf{O}} P(O_k|L_k) \tag{2}$$

where $\mathbf{L_j}, L_k \subset (\mathbf{EX} \cup \mathbf{EL})$ and $\mathbf{L_j}$ are the latent parents of the endogenous latent EL_j, and L_k is the latent parent of the observed O_k.

Proposition 1 The joint probability over **V** due to an exogenous value assignment **ex** to **EX** is determined only by this assignment and BN.

Proof: The first product in (2) for assignment **ex** depends on the priors for **EX**, and the other two products depend only on **ex** and the BN probabilities:

$$P(\mathbf{V}|\mathbf{EX} = \mathbf{ex}) = P(\mathbf{EX}, \mathbf{EL}, \mathbf{O}|\mathbf{EX} = \mathbf{ex}) =$$
$$\prod_{EX_i \in \mathbf{EX}} P(EX_i = ex_i) \prod_{EL_j \in \mathbf{EL}} P(EL_j = el_j|\mathbf{L_j} = \mathbf{l_j^{ex}}) \prod_{O_k \in \mathbf{O}} P(O_k = o_k|L_k = l_k^{\mathbf{ex}}) \tag{3}$$

where $\mathbf{l_j^{ex}}$ and $l_k^{\mathbf{ex}}$ are configurations of $\mathbf{L_j}$ and L_k, respectively due to **ex**. ■

Proposition 1 is a keystone in our analysis because it shows a path of hierarchical influence of latents on observed variables - from exogenous latents through endogenous latents to observed variables. Recognition and use of this path of influence guides LPCC in learning LVMs. To formalize our ideas, we introduce several concepts. First, we define local influence on a single EN of its direct parents. Second, we use local influences and the BN Markov property to generalize to the influence of **EX** on **EN**. Analysis of the influence of all configurations **ex**s on all **en**s enables learning the structure and parameters of the model and causal discovery. Finally, we show how these concepts can be analyzed and an LVM be learned from the result of clustering the data.

Definition 1 A *local effect* on an endogenous variable EN is the influence of a configuration of EN's *direct latent parents* on any of EN values. Following, we define a *major local effect* as the largest local effect on EN. A major local effect on EN_i is identified by the maximal

conditional probability of a specific en_i given a configuration $\mathbf{l_i}$ of its latent parents $\mathbf{L_i}$, i.e., $MAE_i(\mathbf{l_i}) = max_{en_i'} P(EN_i = en_i' | \mathbf{L_i} = \mathbf{l_i})$. All probabilities of EN_i's values conditioned on $\mathbf{l_i}$ that are smaller than $MAE_i(\mathbf{l_i})$ identify the minor local effects set, $MIES_i(\mathbf{l_i})$. Similarly, the *major value* is the en_i corresponding to $MAE_i(\mathbf{l_i})$, i.e., the most probable value of EN_i due to $\mathbf{l_i}$, $MAV_i(\mathbf{l_i}) = argmax_{en_i'} P(EN_i = en_i' | \mathbf{L_i} = \mathbf{l_i})$. A *minor value* is an en_i corresponding to a minor local effect, and $MIVS_i(\mathbf{l_i})$ is the set of minor values correspond to $MIES_i(\mathbf{l_i})$. When $EN_i = O_i$ and for a value l_i of its single latent parent L_i, $MAE_i(l_i) = max_{o_i'} P(O_i = o_i' | L_i = l_i)$ and $MAV_i(l_i) = argmax_{o_i'} P(O_i = o_i' | L_i = l_i)$ for the major local effect and value, respectively.

By aggregation over all local influences, we can generalize these concepts, through the BN parameters and Markov property, from local influences on specific endogenous variables to influence on all endogenous variables in the graph.

Definition 2 An *effect* on **EN** is the influence of a configuration **ex** of **EX** on **EN**. Following, we define a *major effect* (MAE) as the largest effect of **ex** on **EN** and a *minor effect* (MIE) as any non-MAE effect of **ex** on **EN**. A *major value configuration* (MAV) is the **en** of **EN** corresponding to MAE, i.e., the most probable **en** due to **ex** and *minor value configuration* is an **en** corresponding to any MIE.

Based on Proposition 1, we can quantify the effect of **ex** on **en**. For example, a major effect of **ex** can be factorized according to the (weighted by the product of priors, $P(EX_i = ex_i)$) product of major local effects on **EN**:

$$MAE(\mathbf{ex}) = \prod_{EX_i \in \mathbf{EX}} P(EX_i = ex_i) \prod_{EL_j \in \mathbf{EL}} MAE_j(\mathbf{l_j^{ex}}) \prod_{O_k \in \mathbf{O}} MAE_k(l_k^{\mathbf{ex}}). \tag{4}$$

Any effect in which at least one EN takes on a minor local effect is minor, and any configuration in which at least one EN takes on a minor value is minor. Consequently, a configuration in which each variable takes on the major value is major, i.e., MAV. We denote the set of all minor effects for **ex** with $MIES(\mathbf{ex})$ and the set of all minor configurations with $MIVS(\mathbf{ex})$.

Practically, we use observational data generated from an LVM and measured over the observed variables, where each configuration of observed variables is a result of an assignment of a configuration **ex** to the exogenous variables **EX**. We define:

Definition 3 An *observed value configuration* and an *observed major value configuration* due to **ex** are the parts in **en** and MAV, respectively, that correspond to the observed variables.

Proposition 2 Only a single observed value configuration due to **ex** is major.

Proof: Due to the probabilistic nature of BN, **ex** creates several observed value configurations, but the maximization operations in (4) ensure that only one of them is major. ■

Due to the probabilistic nature of BN, each observed value configuration due to **ex** is represented by several data patterns. Clustering the data produces several clusters per each **ex**, where each cluster corresponds to another observed value configuration. Based on Proposition 2, only one of the clusters corresponds to an observed major value configura-

tion, whereas the other clusters correspond to observed minor value configurations, and we distinguish them using Definition 4:

Definition 4 For each configuration **ex**, there is only a single cluster that corresponds to the observed major value configuration and thus represents the major effect $MAE(\mathbf{ex})$. This cluster is the *major cluster*, and the clusters that represent the minor effects in $MIES(\mathbf{ex})$ are *minor clusters*.

To resolve between different minor effects/clusters, we make two definitions:

Definition 5 A *k-order minor effect* is a minor effect in which exactly k ENs have minor local effects. We call each **en** corresponding to a k-order minor effect a *k-order minor configuration*.

Definition 6 Minor clusters that correspond to k-order minor effects are *k-order minor clusters*.

The set of all major clusters reflects the effect of all possible **ex**s, and thus the number of major clusters is expected to be equal to the number of **ex**s. It is easier to identify major clusters than minor clusters because the former reflect the major effects of **EX** on **EN** and therefore are considerably more populated than the latter. Also, we can use the clusters' centroids to represent the clusters.

Finally, to discover causal relationships between the variables in the model using clusters, we introduce the concept of *pairwise cluster comparison* (PCC). PCC measures the differences between clusters, each represents the response of LVM to another **ex**.

Definition 7 A PCC is a comparison between pairs of clusters through a comparison of vectorial representations of the clusters' centroids. The result of PCC is a binary vector in which each element is 1 if there is a difference between the compared centroids, or 0, if there is no difference.

When PCC is between clusters that represent observed major value configurations (i.e., PCC between major clusters), a PCC element of 1 identifies an observed variable that changes its value between the compared clusters due to a change in **ex**. Thus, the 1's in a PCC provide evidence of causal relationships between **EX** and **O**. However, due to the probabilistic nature of BN and the existence of endogenous latents (mediating the connection from **EX** to **O**), some of the clusters are k-order minor clusters (in different orders), representing k-order minor configurations. Thus, when PCC is between a major and a minor clusters, an observed variable in two compared clusters may not necessarily change its value as a result of a change in **ex**. Because the major cluster has zero minor values, a PCC in such a case shows (through the number of 1's) the number of minor values in the centroid of the minor cluster. That is, PCC between major clusters can be the main source to identify causal relationships and that between a major and a minor clusters can account for the secondary impact of **EN** on **O**.

3. Overview of the LPCC Concept

To start our overview of LPCC, we demonstrate through an example the relations between clustering results and learning an LVM. G1 in Figure 1 shows a model having two exogenous variables L1 and L2 that are binary latents, each having three binary children X1, X2, X3 and X4, X5, X6, respectively. L1 and L2 have four possible **ex**s (L1L2= 00, 01, 10, 11). First, we synthetically generated a random data set of 1,000 patterns from G1 over

the six observed variables. We used a uniform distribution over L1 and L2 and set the probabilities of an observed child X_i, $i = 1,,6$, given its latent parent $L_k, k = 1, 2$, to be $P(X_i = v|L_k = v) = 0.8, v = 0, 1$. Second, we clustered the data set and found sixteen clusters, of which four were major (see Section 3.3). This meets our expectation of four major clusters corresponding to the four possible **ex**s. These clusters are presented in Table 1a by their centroids, which are the most prevalent patterns in the clusters, and in Table 1b by their PCCs. For example, *PCC1,2* compares clusters *C1* and *C2*, showing that when moving from *C1* to *C2* only the values corresponding to variables X1, X2, and X3 are changed ($\delta X1 = \delta X2 = \delta X3 = 1$). This requires that the three variables are descendants of the same EX that changed its value between two **exs** represented by *C1* and *C2*. *PCC1,4*, *PCC2,3*, and *PCC3,4* enforce this conclusion. We know from G1 that this EX is the latent L1. A similar conclusion can be deduced about X4, X5, and X6 and L2. LPCC learns LVMs in two phases. In the first phase, LPCC identifies **EX** and their corresponding observed descendants (Section 3.1), identifies collider latents and their corresponding latent parents (Section 3.2), and iteratively updating the selection of the major clusters (Section 3.3). In the second phase, LPCC identifies non-collider endogenous latents and splits them (together with their children) from their previously learned latent ancestors (Section 3.4).

3.1. Identification of latent variables

Table 1b shows that *PCC1,2* provides evidence that X1, X2, and X3 may be descendants of the same exogenous latent (L1, as we know) that has changed its value between the two **ex**s represented by *C1* and *C2*. Relying only on one PCC may be inadequate when concluding that these variables are descendants of the same latent because there may be other latents that have changed their values too. Table 1b shows that *PCC2,3* provides the same evidence about X1, X2, and X3. But, *PCC2,3* also shows that the values corresponding to X4, X5, and X6 have been changed together too, while these values have not been changed in *PCC1,2*. Does it mean that X4, X5, and X6 are also descendants of the same latent parent of X1, X2, and X3? If we combine the two pieces of evidence provided by *PCC1,2* and *PCC2,3* we can answer this question by "no". This is because X4, X5, and X6 have changed their values only in *PCC2,3* but not in *PCC1,2* and thus they cannot be descendants of L1. This insight strengthens the evidence that X1, X2, and X3 are descendants of L1 and they are the only descendants it has. A similar analysis using *PCC1,3* and *PCC2,4* will identify that X4, X5, and X6 are descendants of another latent variable (L2 as we know). Similarly, LPCC considers all PCCs during learning. LPCC hypothesizes that the maximal set of observed variables (**MSO**), which always change their values together in all of the PCCs that show a change between major clusters, are descendants of the same latent variable L. Then, LPCC introduces L to the learned graph and adds **MSO** as its children. Thereby, LPCC identifies latent variables and their observed descendants (all are joined as children of L).

3.2. Identification of collider latent variables

G2 in Figure 1 shows two exogenous latent variables, L1 and L3. For example, both may be binary variables each having two binary observed children X1 and X2 and X5 and X6, respectively. L1 and L3 also collide on a single endogenous latent variable L2 that has two

Centroid	X1	X2	X3	X4	X5	X6
C1	0	0	0	1	1	1
C2	1	1	1	1	1	1
C3	0	0	0	0	0	0
C4	1	1	1	0	0	0

(a)

PCC	$\delta X1$	$\delta X2$	$\delta X3$	$\delta X4$	$\delta X5$	$\delta X6$
PCC1,2	1	1	1	0	0	0
PCC1,3	0	0	0	1	1	1
PCC1,4	1	1	1	1	1	1
PCC2,3	1	1	1	1	1	1
PCC2,4	0	0	0	1	1	1
PCC3,4	1	1	1	0	0	0

(b)

Table 1: (a) Centroids of major clusters for G1. (b) PCCs between the major clusters.

binary children X3 and X4. We expect to find four major clusters in the data generated from G2. Each cluster will correspond to one of the four possible **ex**s (L1L3= 00, 01, 10, 11). In this case, as before, we expect the values of X1 and X2 to be changed together in all the PCCs in which the value of L1 changes, and the values of X5 and X6 to be changed together in all the PCCs in which the value of L3 is changed. However, the values of X3 and X4 will be changed together with those of X1 and X2 in part of the PCCs and together with those of X5 and X6 in the remaining PCCs, but always together in all of the PCCs. This will be evidence that X3 and X4 are descendants of the same latent variable (L2) that is a collider of L1 and L3. To learning that a latent variable L is a collider of a set of other latent parent variables **LP**, LPCC requires that: 1) The values of the descendants of L are changed with the values of the descendants of different latent variables in **LP** (which were already identified in the first phase) in different parts of PCCs between major clusters; and 2) The values of the descendants of L are not changed in any PCC unless the values of the descendants of either of the variables in **LP** are changed too. This insures that L does not change independently of latents in **LP** that are its parents.

3.3. Strategy for choosing major clusters

In this unsupervised problem of identification the existence of latent variables given only observational data, LPCC has to deal with lack of prior information regarding the distribution over the latent variables. Therefore, in its first iteration, LPCC assumes a uniform distribution over the latents and selects the major clusters based only on the cluster size. Clusters that are larger than the average cluster size are selected as majors. However, this initial selection may generate false negative errors (i.e., deciding a major cluster is minor). This may happen when a latent variable L has a skewed distribution over its values, due to a low probability of L to take on any of its rare values v. The **ex** for which $L = v$ will be represented only by small clusters that could not be chosen as majors, although at least one of them should be major. In addition, the initial selection may perform a false positive error (i.e., deciding a minor cluster is major) as a result of a very weak correlation between L and any of its children X_i. This weak correlation can be represented in the discrete case as almost equal conditional probabilities of an observed child to take on two different values $v_1 \neq v_2$ given the same value of its latent parent v, i.e., $P(X_i = v_1|L = v) \approx P(X_i = v_2|L = v)$. This may lead to splitting a cluster that represents a configuration in which $L = v$ into two clusters with almost the same size, and accepting both as major clusters instead of only one. LPCC adapts an iterative approach to avoiding these possible errors due to inaccurate assumptions. Following learning the initial graph based on cluster sizes, learning the cardinalities of the latent variables and consequently finding all possible **ex**s is made possible (Section 4.1). Then, for each **ex**, we can select the most probable cluster given **ex** and

using the data to be the major cluster that represents this **ex**. That is, the set of major clusters can be updated iteratively and probabilistically and augment LPCC to learn more accurate graphs. This process can be repeated until convergence to a final graph. Since the final graph depends on the initial graph, the iterative approach cannot guarantee finding the optimal model, but to improve the initial graph.

3.4. Identification of non-collider latent variables

G3 in Figure 1 is an example model of three latent variables L1, L2, and L3 in a serial connection. Say, each of the latents is binary having three binary observed children. L1 is the only EX with two possible **ex**s (L1= 0, 1) and L2 and L3 are ENs; L2 is a child of L1 and a parent of L3. We synthetically generated a random data set of 1,000 patterns from G3 over the nine observed variables. We used a uniform distribution over L1 and set the probabilities of an observed child, $X_i, i = 1, , 9$, given its latent parent $L_k, k = 1, 2, 3$, to be $P(X_i = v|L_k = v) = 0.8, v = 0, 1$, and of an endogenous latent $L_j, j = 2, 3$, given its latent parent $L_k, k = 1, 2$, to be $P(L_j = v|L_k = v) = 0.8, v = 0, 1$. Table 2 presents the six largest clusters using their centroids and sizes, from which *C1* and *C2* were selected as major clusters (following Section 3.3). This meets our expectation of two major clusters corresponding to the two possible **ex**s of L1. However, the model learned in the first phase (G0) has only one latent variable (L1), and all of the nine observed descendants are learned as L1's direct children. Thus, in the second phase, LPCC tests the assumption that G0 is true (after learning the model parameters using EM (Dempster et al., 1977); Section 4.2). If the assumption is not true (details to follow), LPCC infers that L1 has non-collider latent children and hence should split L1 to represent these latents. Then, LPCC recursively identifies possible higher order splits of the latents. Following, we demonstrate this procedure to a first order split and generalize it to k-order splits.

Centroid	X1	X2	X3	X4	X5	X6	X7	X8	X9	size	#**MSO**s
C1	1	1	1	1	1	1	1	1	1	49	0
C2	0	0	0	0	0	0	0	0	0	47	0
C3	1	1	1	1	1	1	1	1	0	28	0
C4	0	0	0	0	0	0	0	1	0	24	0
C5	0	1	0	0	0	0	0	0	0	22	0
C6	1	1	1	1	1	1	0	0	0	22	2

Table 2: Largest six clusters represented by their centroids, sizes, and numbers of **MSO**s.

To identify a possible first order split, and thereby reject the assumption about the correctness of G0, LPCC calculates a threshold on the maximal size of 2-order minor clusters (Definition 6). This threshold represents the maximal size of a cluster that has at least two minor values, and it is derived from an approximation of the maximal probability of having two minor values (Appendix A). All clusters with sizes between this threshold and the size of the minimal major cluster, i.e., the smallest cluster having zero minor values (e.g., *C2* in Table 2), are expected to represent 1-order minor clusters. We check if there is a PCC between any of these clusters and any major cluster that shows a group of two or more observed variables that change simultaneously between the clusters (**MSO**; Section 3.1) and thus represents more than the single minor value that is expected. If there is such PCC, we infer that the observed variables are descendants of a latent that is other than L1, and split L1 to express the other latent. *C6* (Table 2) is a cluster that when compared with

C1 or *C2* shows such a pattern and calls LPCC to split L1 to two latents (one for each of the two **MSO**s) making the new latent a parent of X7, X8, and X9, and leaving the remaining variables as L1's children. LPCC recursively performs this procedure to identify higher order splits for each of the new latents until there are no splits. Where in the recursive call of depth k (k order split), LPCC assumes that the latent should not be split, then it selects a threshold for the maximal $k+1$ order clusters and selects the clusters that represent at most k-order clusters. If the PCCs between any selected cluster and the major clusters show more than one **MSO** of the latent's children, each with size of at least k, LPCC splits the latent into new latents that each represent one **MSO**. In our example, both L1 and the new latent (L3) are considered for splitting, and L1 is indeed split again to form L2, which is the parent of X4, X5, and X6. Then, LPCC stops since no higher order splits occurred for any of the new latents. The directions of the links between the latents are determined so latents that have been split in depth k are children of latents that have been split in depth $k+1$. This is also correct for the latent diverging connection (G4 in Figure 1).

4. The LPCC algorithm

We introduce a two-stage algorithm that implements the LPCC concept. The algorithm gets a data set $\mathbf{D}$ over the observed variables $\mathbf{O}$ and learns an LVM. In the first stage, LPCC learns the exogenous and the collider latents (LEXC) as well as their descendants (Algorithm 1). In the second stage, LPCC augments the graph learned by LEXC by learning the non-collider endogenous latents (LNC) and their children.

4.1. Learning exogenous and collider latents (LEXC)

LEXC adapts an iterative approach and learns the initial graph in six steps. The first step is clustering $\mathbf{D}$ using the self-organizing map (SOM) (Kohonen, 1997). We chose SOM because it does not require prior knowledge about the expected number of clusters, which is essential when targeting uncertainty in the number of latent variables in the model, but any other clustering algorithm that preserves this property can replace SOM. The result of the first step is a cluster set $\mathbf{C}$ in which each cluster is represented by its centroid. In the second step, LEXC performs an initial selection of the major clusters set, where a cluster in $\mathbf{C}$ whose size (measured by the number of clustered patterns) is larger than the average cluster size in $\mathbf{C}$ is selected as a major cluster (Section 3.3). $\mathbf{MC} = \{\mathbf{MC}_i\}_{i=1}^{n}$ is a matrix that holds information about the major clusters, where each matrix row represents a centroid of one of the n major clusters.

In the third step, LEXC creates a matrix that represents all PCCs derived from $\mathbf{MC}$. This matrix is $\mathbf{PCCM} = \{\mathbf{PCC}_{ij}\}_{i=1,j>i}^{n,n}$, where $\mathbf{PCC}_{ij}$ is a Boolean vector representing the result of PCC between major clusters C_i and C_j having centroids $\mathbf{MC}_i$ and $\mathbf{MC}_j$ in $\mathbf{MC}$, respectively. The k^{th} element of $\mathbf{PCC}_{ij}$ represents a change in value, if one exists, in the observed variable $O_k \in \mathbf{O}$ when comparing $\mathbf{MC}_i$ and $\mathbf{MC}_j$. We use the notation $\mathbf{PCC}_{ij} \rightarrow \delta O_k$ if the value has been changed and $\mathbf{PCC}_{ij} \rightarrow \neg\delta O_k$ otherwise.

In the fourth step, LEXC identifies latents and their descendants (Section 3.1) using a matrix **MSOS** that holds all maximal sets of observed (**MSO**) variables that always change their corresponding values together in all of the PCCs in **PCCM**. For each **MSO**, LEXC

Algorithm 1 *LEXC*

{Input: A data set $\mathbf{D}$ over the observed variables $\mathbf{O}$.
Output: An initial leaned graph $\mathbf{G}$ of the exogenous and the collider latents and their descendants in LVM.}
Initialize: Create an empty graph $\mathbf{G}$ over $\mathbf{O}$, $\mathbf{C} = \phi, \mathbf{MC} = \phi, \mathbf{PCCM} = \phi, \mathbf{L} = \phi, \mathbf{OLP} = \phi$
{First step: perform clustering.}
$C \leftarrow$ perform clustering on $\mathbf{D}$ and represent each cluster by its centroid.
{Third step: select initial major clusters set.}
for all $C_i \in \mathbf{C}$: if the size of C_i is larger than the average cluster size in $\mathbf{C}$, then add C_i to $\mathbf{MC}$.
{Third step: create the $\mathbf{PCCM}$ matrix.}
for all $\mathbf{MC}_i \in \mathbf{MC}, \mathbf{MC}_j \in \mathbf{MC}, j > i$: compute $\mathbf{PCC}_{ij}$ and add it to $\mathbf{PCCM}$
{Fourth step: identify latent variables and their observed children}
$\mathbf{MSOS} \leftarrow$ using the $\mathbf{PCC}$ matrix find all possible $\mathbf{MSO}$s
for all $\mathbf{MSO}_i \in \mathbf{MSOS}$:
    add a new latent variable L to $\mathbf{G}$ and to $\mathbf{L}$.
    for all observed variable $O \in \mathbf{MSO}_i$: add a new edge $L \rightarrow O$ to $\mathbf{G}$.
{Fifth step: identify collider latent variables and their parents}
for all $L_i \in \mathbf{L}$
    {first phase}
    $L_i.\mathbf{PPS} = \phi$.
    for all $L_j \in \mathbf{L}, L_j \neq L_i$
        if $\exists \mathbf{PCC} \in \mathbf{PCCM} : (\mathbf{PCC} \rightarrow \delta L_i \wedge \mathbf{PCC} \rightarrow \delta L_j \wedge \mathbf{PCC} \rightarrow \neg\delta L_k, \forall L_k \in L, k \neq i, j)$ **then**
            add L_j to $L_i.\mathbf{PPS}$
    {Second phase}
    if$\forall(\mathbf{PCC} \in \mathbf{PCCM} : \mathbf{PCC} \rightarrow \delta L_i), \exists(PSS \in L_i.\mathbf{PSS} : \mathbf{PCC} \rightarrow \delta PSS)$ **then**
        $\forall(PSS \in L_i.\mathbf{PSS})$, add a new edge $PSS \rightarrow L_i$ to $\mathbf{G}$.
{Sixth step: select a new major clusters set}
$\mathbf{NMC} = \phi$
find the cardinality for each $L_i \in \mathbf{L}$, then create **exs**
for allex $\in$ **exs**
    find the largest cluster that represents **ex**, and add it to $\mathbf{NMC}$
    if $\mathbf{NMC} = \mathbf{MC}$ **then**
        return $\mathbf{G}$
    else
        $\mathbf{MC} = \mathbf{NMC}$, $\mathbf{PCCM} = \phi$, $\mathbf{L} = \phi$, $\mathbf{OLP} = \phi$, $\mathbf{G} \leftarrow$ empty graph over $\mathbf{O}$
        Go to "Third step".

adds a latent L to both $\mathbf{G}$ and latent set $\mathbf{L}$ and edges from L to each observed variable $O \in \mathbf{MSO}$. The observed children of latent $L_i \in \mathbf{L}$ in $\mathbf{G}$ are $\mathbf{Ch}_i$.

In the fifth step, LEXC identifies, in two phases, the latent variables that are collider nodes in the graph along with their latent parents (Section 3.2). In the first phase, LEXC considers for each latent variable $L_i \in \mathbf{L}$ a set of potential parents from the other latents in $\mathbf{L}$. To simplify the notation, we represent the latent as an object and the set of potential parents as a field of this object, called PPS (for potential parent set), e.g., $L_i.\mathbf{PPS}$. In addition, we use the notation $\mathbf{PCC}_{ij} \rightarrow \delta L_i$ if all of the variables in $\mathbf{Ch}_i$ change their values in $\mathbf{PCC}_{ij}$ and $\mathbf{PCC}_{ij} \rightarrow \neg\delta L_i$ otherwise. The algorithm identifies for each latent L_i its $L_i.\mathbf{PPS}$. The first phase checks for each pair $L_i, L_j \in \mathbf{L}$ whether there exists a vector $\mathbf{PCC}_{ij} \in \mathbf{PCCM}$ in which both L_i and L_j have been changed while the other variables in $\mathbf{L}$ have not, and if so it adds L_j to $L_i.\mathbf{PPS}$. In the beginning of the second phase, the set $L_i.\mathbf{PPS}$ contains all of the variables in $\mathbf{L}$ that have been changed with L_i in $\mathbf{PCCM}$. Still, this is not enough to decide that L_i is a collider of the variables in $L_i.\mathbf{PPS}$; therefore, an additional condition must be fulfilled: L_i should have never changed in any $\mathbf{PCC}_{ij} \in \mathbf{PCCM}$ unless at least one of the variables in $L_i.\mathbf{PPS}$ has also changed in this $\mathbf{PCC}_{ij}$ (Section 3.2). The second phase checks this condition, and if fulfilled, it adds an edge from each variable in $L_i.\mathbf{PPS}$ to L_i.

In the last step, LEXC selects a new set of major clusters $\mathbf{NMC}$ based on the learned graph (Section 3.3). First, it learns the cardinality of each latent $L_i \in \mathbf{L}$, which is the number of different value configurations corresponding to $\mathbf{Ch}_i$ in $\mathbf{D}$, as checked by PCC. Each represents a different value l_i of L_i, that we denote the by $l_i \rightarrow ch_i$. Then, LEXC finds **exs**- the set of all possible **ex**, and for each one it finds the most probable cluster given **ex**. The centroid of this cluster is $c^* = argmax_{C_i \in \mathbf{C}} P(C_i|\mathbf{ex})$, and is added to $\mathbf{NMC}$ (Definition

4). Practically, we approximate $P(C_i|\mathbf{ex})$ by the ratio between the size of this cluster and the size of $\mathbf{D}$. Thus, the largest cluster, where in its centroid, the values corresponding to the children of each $L_i \in \mathbf{EX}$ are according to the value of L_i in $\mathbf{ex}$, is selected to represent this $\mathbf{ex}$. Then if $\mathbf{NMC}$ is equal to the current $\mathbf{MC}$, using $\mathbf{NMC}$ will not produce a different graph, thus LEXC stops and returns the learned graph $\mathbf{G}$. Otherwise LEXC initializes $\mathbf{MC}$ to hold the new set $\mathbf{NMC}$ and performs the next iteration to learn a new graph.

4.2. Learning non-collider latents (LNC)

Using the graph $\mathbf{G}$ learned by LEXC and the data set $\mathbf{D}$, LNC creates an incomplete data set $\mathbf{IND}$ by adding $|L|$ elements to the end of each vector in $\mathbf{D}$. The value of the added element that corresponds to $L_i \in \mathbf{L}$ in each vector in $\mathbf{IND}$ is determined based on the value configuration of $\mathbf{Ch_i}$ in this vector, if this value configuration is equal to any $l_i \rightarrow ch_i$, otherwise it is a missing value. Second, LNC assumes that $\mathbf{G}$ is true, and based on this assumption it learns the parameters of the LVM using EM (Dempster et al., 1977) and $\mathbf{IND}$. We denote this learned model by G0 (Section 3.4). For each latent $L_i \in \mathbf{L}$, LNC assumes that L_i should not be split and tests this assumption recursively (Section 3.4). First, it marginalizes the effects over all other latents $L_j \in \mathbf{L}, j \neq i$, by selecting only the clusters in which all $\mathbf{Ch_j}$ have major values, i.e., the value configuration corresponding to $\mathbf{Ch_j}$ is equal to any $l_j \rightarrow ch_j$ for any value l_j of L_j. Denote this sub-set of the clusters by $\mathbf{C_i}$. This marginalization is essential to ensure that all $\mathbf{MSO}$s to be identified will be found only in among the children of L_i. Then, LNC calculates a threshold MT_k on the maximal size of k-order minor clusters (Appendix A) to identify a possible k order split of L_i. We denote the clusters in $\mathbf{C_i}$ having sizes that are greater than MT_k by $\mathbf{C_{ik}}$. LNC checks if the PCCs between any cluster in $\mathbf{C_{ik}}$ and the major clusters demonstrate at least two $\mathbf{MSO}$s, where each has a size of at least k. If so, LNC splits L_i into two latents. Then, LNC recursively tries to find further splits to the new latents of order $k+1$. If no such split exists, it stops and returns the new latents. After identifying all possible splits, the directions of the links between the latents are determined so latents split in order k are children of latents split in order $k+1$.

Algorithm 2 *LNC*

{A data set D over the observed variables O and the resulting graph G from LEXC.
Output: The final learned LVM G.}
Initialize: $\mathbf{L2} = \phi$ {will hold the new set of latents after the splits.}
Create $\mathbf{IND}$ and learn the parameters of $M0$ using $EM(G, \mathbf{IND})$
for all $L_i \in \mathbf{L}$ in $M0$ **do**
    find $\mathbf{C}_i$
    $\mathbf{L2} = \mathbf{L2} \cup RecSplit(\mathbf{C}_i, 1)$
end for
Direct the links between the new latents, set $\mathbf{L2}$ and return the final graph G.
{sub Procedure $RecSplit(\mathbf{C}_i, k)$}
Find MT_k and $\mathbf{C}_{ik}$
if all PCCs between any cluster in $\mathbf{C}_{ik}$ and MC don't show at least two $\mathbf{MSO}$s **then**
    return L_i
else
    Split L_i: foreach $\mathbf{MSO}$ found create new latent as a parent of the observeds in $\mathbf{MSO}$ and add it to $\mathbf{L'}$
    $\mathbf{R} = \phi$ {will hold the new set of latents after the split of L_i.}
    for all $L_{i'} \in \mathbf{L'}$: find $\mathbf{C}_{i'}$, $\mathbf{R} = \mathbf{R} \cup RecSplit(\mathbf{C}_{i'}, k+1)$
    return $\mathbf{R}$
end if

5. Evaluation

We evaluated LPCC using simulated data sets (Section 5.1) and two real-world data sets: the political action survey (PAS) and HIV (Section 5.2). In the latter case, we did not have an objective measure for evaluation. Therefore, we compared the LPCC outputs to hypothesized, theoretical models from the literature and to the outputs of four state-of-the-art learning algorithms: FCI (Spirtes et al., 2000) (only for PAS), Zhang (2004) (only for HIV), and BuildPureClusters (BPC) and BuildSinglePureClusters (BSPC) of Silva (2005), which are especially suitable for MIM models, similar to LPCC. BPC assumes that the observed variables are continuous and normally distributed, whereas BSPC is a variant of BPC for discrete observed variables. We ran BPC using its implementation in TETRAD IV [1]. BPC learns LVM by testing tetrad constraints at a given significance level (alpha). We used Wishart's tetrad test (Silva, 2005; Spirtes et al., 2000) and three levels of 0.01, 0.05 (TETRAD's default), and 0.1. BSPC is not implemented in TETRAD IV, so we used the results described in Silva (2005) for this algorithm and PAS.

5.1. Simulated data sets

We used Tetrad IV to construct the Graphs G1,G2,G3 and G4 of Figure 1, once with binary and once with ternary variables. The priors of the exogenous latents were always distributed uniformly. We compared performances for two parameterization schemes that differ by the conditional probabilities between a latent L_k and each of its children EN_i, i.e., $p_j = 0.75$ and 0.8. For all graphs in the binary case, except L2 in G2, $P(EN_i = v|L_k = v) = p_j, v = 0 or 1$. For all graphs in the ternary case, except L2 in G2, $P(EN_i = v|L_k = v) = p_j, P(EN_i \neq v|L_k = v) = (1 - p_j)/2, v = 0, 1 or 2$. Concerning L2 in G2 , $P(L_2 = 0|L_1L_3 = 00, 01, 10) = P(L_2 = 1|L_1L_3 = 11) = p_j$ in the binary case and $P(L_2 = v|\max(L_1, L_3) = v) = p_j$ and $P(L_2 \neq v|\max(L_1, L_3) = v) = (1 - p_j)/2$ in the ternary case. Each such scheme imposes a different "parametric complexity" on the model and thereby affects the task of learning the latent model and the causal relations. That is, $p_j = 0.75$ undermines learning more than $p_j = 0.8$. For example for G3 and the binary case, the correlations between any latent and any of its children for the parametric settings $p_j = 0.75$ and 0.8 are 0.5 and 0.6, respectively. Tetrad IV was also used to draw data sets of 125, 250, 500, 750, and 1,000 samples for each test. Overall, we evaluated the LPCC algorithm using 80 synthetic data sets for 4 graphs (G1-G4), 2 types of variables, two parameterization schemes, and 5 data set sizes). In addition, we evaluated LPCC using two larger graphs: G5 and G6 of Figure 1 that combine all types of links between the latents, i.e., collider, serial, and diverging. Each such graph has five latents with three observed children each. Tetrad IV was used to draw data sets of 250, 500, 1000, and 2,000 samples, where all variables are binary and the parametric setting is $p_j = 0.8$. In Figure 2, we report on the structural hamming distance (SHD) (Tsamardinos et al., 2006) as a performance measure of learning LVM for increasing sample sizes. SHD is a global structural measure that accounts for all the possible learning errors. Figure 2 shows learning curves for SHD and increasing sample sizes for LPCC as compared to BPC of Silva (2005). The graphs demonstrate LPCC sensitivity to the parametric complexity; the lower is the complexity, the faster is learning and the sooner the error vanishes. In addition,

1. Available at http://www.phil.cmu.edu/projects/tetrad

all plots show improvement in LPCC accuracy with the sample size and better results for LPCC than for BPC. Especially, LPCC demonstrates a better asymptotic behavior. For example, in G3-G6, BPC missed for the largest data sets the correct directions of the serial links between the latents. We also note that unlike LPCC, BPC is not suitable for learning models such as G1, where the latents are independent and each has less than four observed children. This is because BPC requires the variables in a tetrad constraint to be all mutually dependent, where in the case of G1, there are at most three mutually dependent variables, so no tetrad constraint can be tested and no graph is learned.

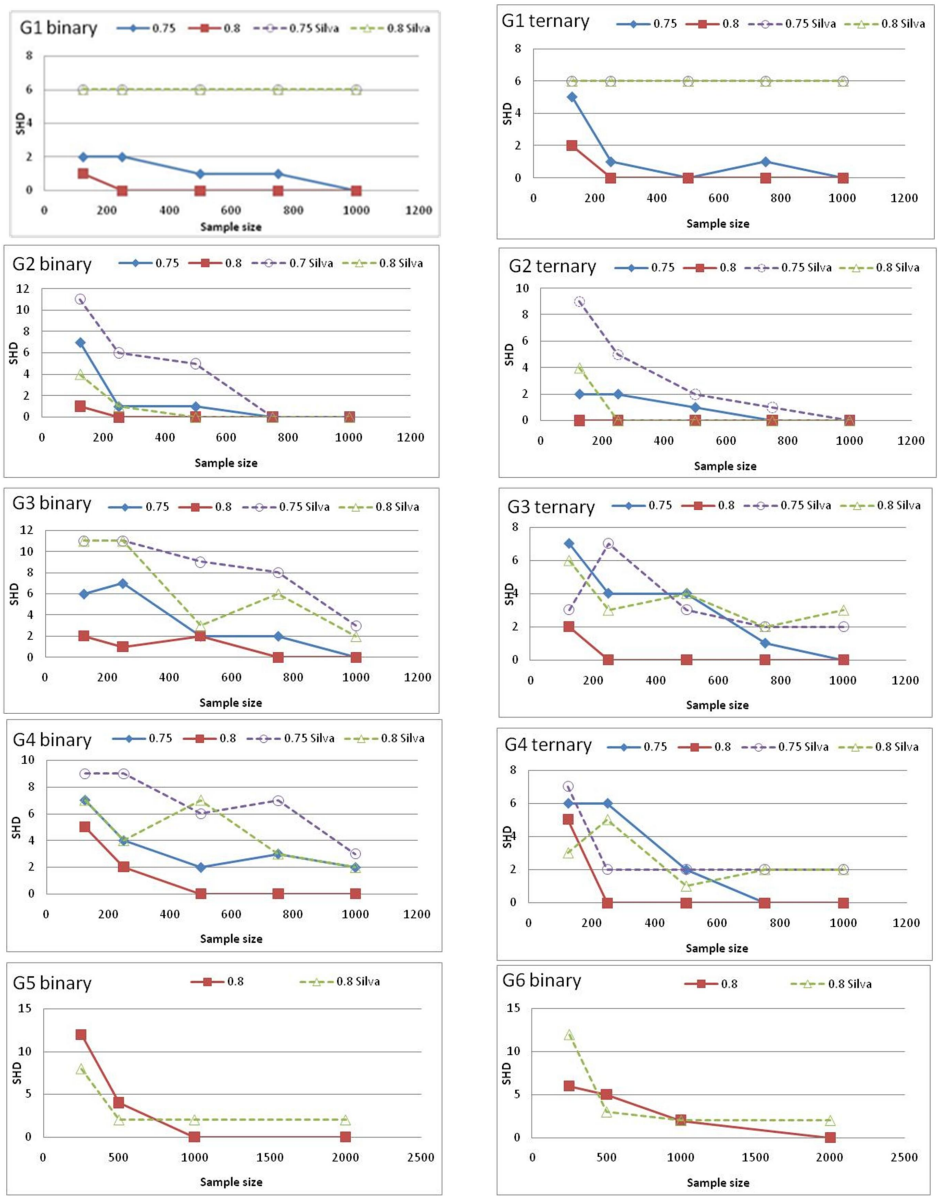

Figure 2: SHD for LPCC and BPC for all graphs of Figure 1 for increasing sample sizes.

5.2. Real-world data sets

We evaluated LPCC using PAS data over the six variables: NOSAY (NS), VOTING (V), COMPLEX (C), NOCARE (NC), TOUCH (T), and INTEREST (I) that are described by Joreskog (2004). These six variables correspond to questions to which the respondent has to give their degree of agreement on a discrete ordinal scale of four values. This data set includes a sample of 1,076 United States respondents. A model consisting of two latents that correspond to a previously established theoretical trait of Efficacy (E) and Responsiveness (R) based on Joreskog (2004) is given (Figure 3a). V is discarded by Joreskog for this particular data set based on the argument that the question for V is not clearly phrased. Similar to the theoretical model (Figure 3a), LPCC finds two latents (Figure 3b): One

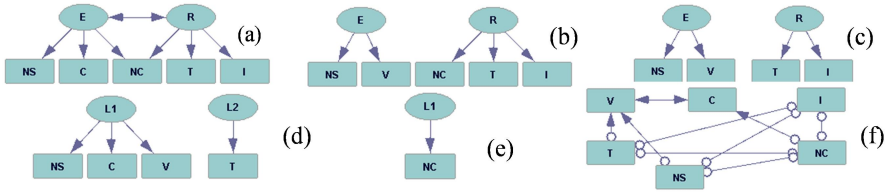

Figure 3: The political action survey: (a) A theoretical model and five outputs of (b) LPCC; (c) BSPC; (d) BPC for alpha=0.01 or 0.05; (e) BPC for alpha=0.1; and (f) FCI for alpha=0.05.

corresponds to NS and V and the other corresponds to NC, T, and I. Compared with the theoretical model, LPCC misses the edge between E and NC (and the edge between the latents, which is not identifiable by the current implementation of LPCC). Nevertheless, LPCC makes no use of prior knowledge. BSPC output (Figure 3c) is very similar to LPCC output, except for NC, which was not identified by BSPC as a measure of R, making the output obtained by LPCC closer to the theoretical model than that of BSPC. In addition, both algorithms identify V as a child of E, and thereby challenge the decision made in Joreskog (2004) to discard V from the model. The outputs of the BPC algorithm (Figure 3d) for both alpha=0.01 and alpha=0.05 are poorer than those of LPCC and BSPC. BPC finds two latents. The first latent corresponds to NS, V and C with partial resemblance to the theoretical model and to the outputs of LPCC and BSPC. However, the second latent found by BPC corresponds only to T and misses I (identified in the theoretical model and by LPCC and BSPC as an indicator of R) and NC (identified in the theoretical model and by LPCC as an indicator of R). The output of the BPC algorithm using alpha=0.1 (Figure 3e) gives very little information about the problem as it finds only one latent that corresponds only to NC. These two last figures show the sensitivity of BPC to the significance level, which is a parameter whose value should be determined beforehand. Moreover, the success of the LPCC and BSPC algorithms emphasizes the importance of such algorithms that can learn discrete data. The outputs of the FCI algorithm using any of the significance levels were not sufficient. For example, the FCI output for alpha=0.05 (Figure 3f) shows that NS and I potentially have a latent common cause. However, these two variables are indicators of different latents in the theoretical model. We also evaluated LPCC using the HIV test data (Zhang, 2004). This data set consists of results on 428 subjects of four diagnostic tests for the

human HIV virus: "radioimmu-noassay of antigen ag121" (X1); "radioimmunoassay of HIV p24" (X2); "radioimmu-noassay of HIV gp120" (X3); and "enzyme-linked immunosorbent assay" (X4). A negative result is represented by 0 and a positive result by 1. LPCC learned a similar model to that in Zhang (2004) (Figure 4), where L1 and L2 are both binary latent variables, but unlike the algorithm in Zhang (2004), LPCC is not limited to tree latent models. BPC returned an empty model for any conventional alpha.

Figure 4: Model learned for HIV using LPCC.

6. Discussion and related work

We introduce the LPCC concept and algorithm for learning LVMs. Using simulated and real-world data sets, we show that LPCC improves accuracy with the sample size, can learn large LVMs, and has consistently good results compared to models that are expert-based or learned by state-of-the-art algorithms. Contrary to other algorithms (Pearl, 2000; Zhang, 2004), LPCC is suitable for learning MIM models and not just latent-tree models. This LPCC quality is shared by BPC (Silva, 2005). Compared to BPC and FCI (Spirtes et al., 2000), LPCC does not rely on statistical tests and pre-setting of a significance level for learning LVM. FCI is not comparable to LPCC in learning MIM models. Unlike BPC, LPCC concentrates on the discrete case and dispenses with the linearity assumption. Further research will focus on: 1) extending LPCC to identify observed variables that are effects of other observed variables; 2) providing a formal analysis for the model identification conditions and its sensitivity to parameterization; and 3) extending the LPCC evaluation using more complex simulated and real-world data sets.

References

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, B 39:1–39, 1977.

G. Elidan, N. Lotner, N. Friedman, and D. Koller. Discovering hidden variables: A structure-based approach. In *Advances in Neural Information Processing Systems*, pages 479–485, 2000.

N. Friedman. The Bayesian structural EM algorithm. In *14th Conference on Uncertainty in Artificial Intelligence*, pages 129–138, 1998.

K. Joreskog. Structural equation modeling with ordinal variables using LISREL. Technical report, Scientific Software International Inc, 2004.

T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, 1997.

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

S. Shimizu, T. Inazumi, Y. Sogawa, A. A. Hyvarinen, Y. Kawahara, T. Washiok, P. Hoyer, and K. Bolle. DirectedLiNGAM: A direct method for learning a linear non-Gaussian structural equation model. *Journal of Machine Learning Research*, 12:1225–1248, 2011.

R. Silva. *Automatic Discovery of Latent Variable Models*. PhD thesis, Carnegie Mellon University, 2005.

R. Silva, R. Scheines, C. Glymour, and P. Spirtes. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7:191–246, 2006.

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

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

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

Appendix A. Setting a threshold for Section 4.2

In this appendix, we provide a detailed description of the calculation of the threshold MT_k on the maximal size of k-order minor clusters from Section 4.2. First we define:

Definition 8 A *maximal major local effect* on an observed child O_t of a latent parent L_i is the maximal major effect on O_t over all values l_i' of L_i, i.e., $MaxMAE_t = max_{l_i'} MAE_t(l_i')$. Similarly, a maximal minor local effect is the maximal minor effect over all values l_i' of L_i, i.e., $MaxMIE_t = max_{l_i'} MIE_i(l_i')$.

In the recursive call of order k, LNC finds an approximation of the threshold for a maximal cluster size in $\mathbf{C_i}$ of $k+1$ minor values corresponding to $\mathbf{Ch_i}$. First, it finds $MaxMAEV_i$ and $MaxMIEV_i$, which are the sorted vectors of $MaxMAE_t$ and $MaxMIE_t$ (Defintion 8) of all $O_t \in \mathbf{Ch_i}$, respectively. Second, it approximates the effect of the other marginalized latents ($margLat$) by computing the product of all maximal major local effects on their observed children, $margLat = \prod_{L_j \in \mathbf{L}, j \neq i} \prod_{o_t \in \mathbf{Ch_j}} MaxMAE_t$. Then, LNC approximates the existence of exactly k minor values of the children of L_i by taking k MaxMIEs and $|Ch_i| - k$ MaxMAEs, $kMin = \prod_{t=1}^{k} MaxMIEV_i(t) \prod_{t=1}^{|Ch_i|-k)} MaxMAEV_i(t)$. It uses the approximations along with the maximal priors of the exogenous latents and the data size N to compute the k^{th} threshold, $MT_k = margLat \cdot kMin \cdot N \cdot \prod_{EX_i \in \mathbf{EX}} \max_{ex_i'} P(EX_i = ex_i')$.