Pairwise Cluster Comparison for Learning Latent Variable Models

Nuaman Asbeh and Boaz Lerner
Department of Industrial Engineering & Management
Ben-Gurion University of the Negev, Israel
asbeh@post.bgu.ac.il; boaz@bgu.ac.il

Abstract

Learning a latent variable model (LVM) exploits values of the measured variables as manifested in the data to causal discovery. Because the challenge in learning an LVM is similar to that faced in unsupervised learning, where the number of clusters and the classes that are represented by these clusters are unknown, we link causal discovery and clustering. We propose the concept of *pairwise cluster comparison* (PCC), by which clusters of data points are compared pairwise to associate changes in the observed variables with changes in their ancestor latent variables and thereby to reveal these latent variables and their causal paths of influence, and the *learning PCC* (LPCC) algorithm that identifies PCCs and uses them to learn an LVM. LPCC is not limited to linear or latent-tree models. It returns a pattern of the true graph or the true graph itself if the graph has serial connections or not, respectively. The complete theoretical foundation to PCC, the LPCC algorithm, and its experimental evaluation are given in [Asbeh and Lerner, 2016a,b], whereas, here, we only introduce and promote them. The LPCC code and evaluation results are available online.

1 LVM LEARNING

Latent variables cannot usually be observed directly from data but only inferred from observed variables (indicators) [Spirtes, 2013]. Concepts such as "life quality," "mental states," and "psychological stress" play a key role in scientific models, and yet are latent [Klee, 1997].

Latent variable models (LVMs) represent latent variables and causal relations among them that are manifested in the observed variables. These models are vital in, e.g., economics, social sciences, natural language processing, and machine learning and have become the focus of many studies. By aggregating observed variables into a few latent variables, each of which represents a "concept" explaining some aspects of the domain, LVMs reduce dimensionality and facilitate interpretation.

Learning an LVM exploits values of the measured variables to infer about causal relationships among latent variables and to predict these variables' values. Although statistical methods, such as factor analysis, effectively reduce dimensionality and may fit the data reasonably well, the resulting models might not have any correspondence to real causal mechanisms [Silva et al., 2006]. Learning Bayesian networks (BNs) focuses on causal relations among observed variables, but the detection of latent variables and their interrelations among themselves and with the observed variables has received little attention. Learning an LVM using Inductive Causation* (IC*) [Pearl, 2000, Pearl and Verma, 1991] 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 FindHidden algorithm [Elidan et al., 2000] detects substructures that suggest the presence of latents in the form of dense subnetworks, but it cannot always find a pure measurement sub-model [Silva et al., 2006]. Also, the recovery of latent trees of binary and Gaussian variables has been suggested [Pearl, 2000]. Hierarchical latent class (HLC) models, which are rooted trees where the leaf nodes are observed while all other nodes are latent, have been proposed [Zhang, 2004]. Two greedy algorithms are suggested [Harmeling and Williams, 2011] to expedite learning of both the structure of a binary HLC and the cardinalities of the latents. Latent-tree models are also used to speed approximate inference in BNs, trading the approximation accuracy with inferential complexity [Wang et al., 2008].

Models in which multiple latents may have multiple in-

dicators (observed children), i.e. multiple indicator models (MIMs) [Bartholomew et al., 2002], are an important subclass of structural equation models (SEM), which are widely used in applied and social sciences to analyze causal relations [Pearl, 2000, Shimizu et al., 2011]. For these models, and others that are not tree-constrained, most of the mentioned algorithms may lead to unsatisfactory results. An algorithm that fills the gap between learning latent-tree models and learning MIMs is BuildPureClusters [BPC; Silva et al., 2006]. It searches for the set of MIMs (an equivalence class) that best matches the set of vanishing tetrad differences [Scheines et al., 1995], but it is limited to linear models [Spirtes, 2013].

We target the goal of Silva et al. [2006], but concentrate on the discrete case. Interestingly, the same difficulty in learning MIMs is also faced in unsupervised learning that confronts questions such as: (1) How many clusters are there in the observed data? and (2) Which classes do the clusters really represent? Due to this similarity, we link learning an LVM and clustering and propose a concept and an algorithm that combine the two disciplines. According to the *pairwise cluster comparison* (PCC) concept, we compare pairwise clusters of data points representing instantiations of the observed variables to identify those pairs of clusters that exhibit major changes in the observed variables due to changes in their ancestor latent variables. Changes in a latent variable that are manifested in changes in the observed variables reveal this latent variable and its causal paths of influence. The *learning PCC* (LPCC) algorithm uses PCCs to learn an LVM by identifying latent variables – exogenous and endogenous (the latter may be either colliders or non-colliders) – and their causal interrelationships as well as their children (latent and observed variables) and causal paths from latent variables to observed variables.

The complete theoretical foundation to PCC is given in Asbeh and Lerner [2016a], and the description of the LPCC algorithm and its experimental evaluation are provided in Asbeh and Lerner [2016b], whereas, here, we only briefly introduce, motivate, and promote them. Following, we give preliminaries to LVM learning. In Section 2, we introduce and motivate PCC, and in Section 3, we provide an overview of the LPCC algorithm. In Section 4, we experimentally compare LPCC to other learning algorithms using synthetic and real-world databases. Finally, in Section 5, we summarize the contribution of LPCC.

First, we present two assumptions that LPCC makes: **A1:** The underlying model is a Bayesian network, BN=$<\mathbf{G},\mathbf{\Theta}>$, encoding a discrete joint probability distribution P for a set of random variables $\mathbf{V}=\mathbf{L}\cup\mathbf{O}$, where $\mathbf{G}=<\mathbf{V},\mathbf{E}>$ is a directed acyclic graph whose nodes $\mathbf{V}$ correspond to latents $\mathbf{L}$ and observed variables $\mathbf{O}$, and $\mathbf{E}$ is the set of edges between nodes in $\mathbf{G}$. $\mathbf{\Theta}$ is the set of parameters; and **A2:** No observed variable in $\mathbf{O}$ is an ancestor of any latent variable in $\mathbf{L}$ (the *measurement assumption* [Spirtes et al., 2000]). We define [following Silva et al., 2006]: **D1:** A model satisfying A1 and A2 is a *latent variable model*; **D2:** Given an LVM $\mathbf{G}$ with a variable set $\mathbf{V}$, the subgraph containing all variables in $\mathbf{V}$ and all and only those edges directed into variables in $\mathbf{O}$ is called the *measurement model* of $\mathbf{G}$; **D3:** Given an LVM $\mathbf{G}$, the subgraph containing all and only $\mathbf{G}$'s latent nodes and their respective edges is called the *structural model* of $\mathbf{G}$; and **D4:** A *pure measurement model* is a measurement model in which each observed variable has only one latent parent and no observed parent. Then, we also assume that: **A3:** The measurement model of $\mathbf{G}$ is pure; and **A4:** Each latent in the true model $\mathbf{G}$ has at least two observed children and may have latent parents.

As Silva et al. [2006] pointed out, factor analysis, principal component analysis, and regression analysis adapted to learning LVMs are well understood, but have not been proven under any general assumptions to learn the true causal LVM, calling for better learning methods. Causal structure discovery – learning the number of latent variables, their interconnections, and connections to the observed variables, as well as the interconnections among the observed variables – is difficult and requires making assumptions about the problem. By assuming that the true model manifests local influence of each latent variable on at least a small number of observed variables, Silva et al. [2006] show that learning the complete Markov equivalence class of MIM is feasible. Similarly, we assume the true model is pure (A3). When it is pure, LPCC will identify it correctly (or find its pattern that represents the equivalence class of the true graph), and when it is not, LPCC will learn a pure submodel of the true model, in both cases using only two indicators per latent (compared to three indicators per latent that are required by BPC [Silva et al., 2006]).

Note the tradeoff between the structural and parametric assumptions that an algorithm for learning an LVM has to make; the fewer parametric assumptions it makes, the more structural assumptions it has to make and vice versa. While BPC needs to make a parametric assumption about the linearity of the model, and the latent-tree algorithms [Zhang, 2004, Harmeling and Williams, 2011, Wang et al., 2008] restrict the learned structure to a tree [1], LPCC assumes that the model structure is pure, and **A5:** A latent collider has no latent descendants.

[1] LPCC is not limited to a tree because it allows latent variables to be colliders

2 PRELIMINARIES TO PCC

Figure 1 sketches a range of pure measurement models, from basic to more complex. G1 is a basic MIM of two unconnected latents, and G2 shows a structural model having a latent collider. Note that such an LVM cannot be learned by latent-tree algorithms such as in Zhang [2004]. G3 and G4 demonstrate serial and diverging structural models, respectively, that together with G2 cover the three basic structural models. G5 and G6 manifest more complex structural models comprising a latent collider and a combination of serial and diverging connections. As the structural model becomes more complicated, the learning task becomes more challenging; hence, G1–G6 present a spectrum of challenges. [2]

Section 2.1 builds the infrastructure to PCC that relies on understanding the influence of the exogenous latent variables on the observed variables. This influence is divided into major and minor effects that are introduced in Section 2.2. Section 2.3 links this structural influence to data clustering and introduces the pairwise cluster comparison concept for learning an LVM.

2.1 INFLUENCE OF EXOGENOUS LATENTS ON OBSERVED VARIABLES

We distinguish between observed ($\mathbf{O}$) and latent ($\mathbf{L}$) variables and between exogenous (**EX**) and endogenous (**EN**) variables. **EX** variables have zero in-degree, and are autonomous and unaffected by the values of the other variables (e.g., L1 in G6 in Figure 1), whereas **EN** are all non-exogenous variables in $\boldsymbol{G}$ (e.g., L2 in G2 and G6, and X1 in all graphs in Figure 1). We identify three types of variables: (1) Exogenous latents, $\mathbf{EX}\subset(\mathbf{L}\cap\mathbf{NC})$ [all exogenous variables are latent non-colliders (**NC**)]; (2) Endogenous latents, $\mathbf{EL}\subset(\mathbf{L}\cap\mathbf{EN})$, which are divided into latent colliders $\mathbf{C}\subset\mathbf{EL}$ (e.g., L2 in G5) and latent non-colliders $\mathbf{S}\subset(\mathbf{EL}\cap\mathbf{NC})$ (e.g., L3 in G6), thus $\mathbf{NC}=(\mathbf{EX}\cup\mathbf{S})$; and (3) Observed variables, $\mathbf{O}\subset\mathbf{EN}$, which are always endogenous and childless, that are divided into children of exogenous latents $\mathbf{OEX}\subset\mathbf{O}$ (e.g., X9 in G2), children of latent colliders $\mathbf{OC}\subset\mathbf{O}$ (e.g., X5 in G2), and children of endogenous latent non-colliders $\mathbf{OS}\subset\mathbf{O}$ (e.g., X4 in G3). We denote value configurations of **EX**, **EN** (when we do not know whether the endogenous variables are latent or observed), **EL**, **C**, **NC** (when we do not know whether the non-collider variables are exogenous or endogenous), **S**, **O**, **OEX**, **OC**, and **OS** by **ex**, **en**, **el**, **c**, **nc**, **s**, **o**, **oex**, **oc**, and **os**, respectively.

Since the underlying model is a BN, the joint probability over **V**, which is represented by the BN, is factored according to the local Markov assumption for $\boldsymbol{G}$ to a product of products of 1) prior probabilities for the exogenous variables (**EX**), 2) probabilities of endogenous (collider and non-collider) latent variables (**EL**) conditioned on their latent parents, and 3) probabilities of observed variables (**O**) conditioned on their latent parents.

To demonstrate the influence of exogenous (latent) variables on observed variables and its relation to learning an LVM, we show that changes in values of the observed variables are due to changes in values of the exogenous variables, and thus the identification of the former indicates the existence of the latter. To do that, we analyze the propagation of influence along directed paths [Pearl, 1988] connecting both variables, first among latents, remembering that the paths may contain latent colliders and latent non-colliders, and then paths ending in their sinks (i.e., the observed variables).

We prove in Asbeh and Lerner [2016a] that a) a latent non-collider has only a single exogenous latent ancestor, and there is only a single directed path between them; and b) a latent colider is connected to a set of exogenous latent ancestors via a set of directed paths. By separating the influence of exogenous variables to distinct paths of influence, we can determine the joint probability over (**V**) due to value assignment **ex** to exogenous set **EX** using this assignment and the BN conditional probabilities.

2.2 MAJOR AND MINOR EFFECTS/VALUES

After analyzing the structural influences (path of hierarchies) of the latents on the observed variables, we complement now this analysis with the parametric influences, which we divide into major and minor effects.

We define a local effect on an endogenous variable EN as the influence of a configuration of EN's direct latent parents on any of EN's values. We distinguish between: 1) a major local effect, which is the largest local effect on EN that is identified by the maximal conditional probability of a specific value en of EN given a configuration $\mathbf{pa}$ of EN's latent parents; and 2) a minor local effect, which is any non-major local effect on EN that is identified by a conditional probability of any other value of EN given $\mathbf{pa}$ that is smaller than that of the major effect. Accordingly, we define: 1) a major local value as the value en corresponding to the major effect; and 2) a minor local value as an en corresponding to any minor local effect. We assume the most probable explanation [Pearl, 1988], which is that for every endogenous variable and every configuration of its parents, there exists a certain value

[2] In Section 4, we compare LPCC with BPC and exploratory factor analysis (EFA) using these six LVMs. Since BPC requires three indicators per latent to identify a latent, we determined from the beginning three indicators per latent for all true models to recover. Nevertheless, in Section 4, we evaluate the learning algorithms for increasing numbers of indicators.

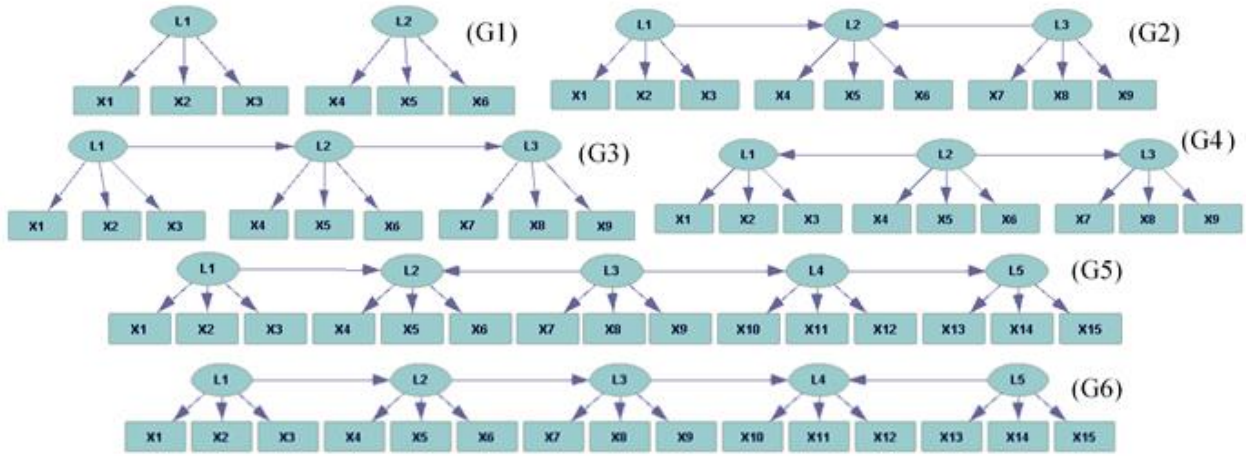


Figure 1: MIMs with Structural Models of Different Complexities Challenging a Learning Algorithm Differently.

that is major, and prove in Asbeh and Lerner [2016a] that this value is unique for every parent configuration.

Second, assuming that for every endogenous variable, different parent values (or parent value configurations for a latent collider) yield different major values, together with using the Markov property and BN parameters, we can aggregate all local influences and generalize local effects on specific endogenous variables to effect on all endogenous variables in the graph. Consequently, a *major effect* (MAE) is the largest effect of **ex** on **EN** and a *minor effect* (MIE) is any non-MAE effect of **ex** on **EN**. Also, a *major value configuration* (MAV) is the configuration **en** of **EN** corresponding to MAE (i.e., the most probable **en** due to **ex**), and a *minor value configuration* is a configuration **en** corresponding to any MIE. In a MAV, each variable in **EN** takes on the major local value and in a MIE, at least one *EN* takes on a minor local value.

Third, we represent the influence on a subset of the endogenous variables of the subset of exogenous variables that impact these endogenous variables. This partial representation of MAE enables LPCC to recover the relationships between exogenous ancestors and only the descendants that are affected by them. We separately analyze the effect of each exogenous variable on each observed variable for which the exogenous is its ancestor and all the latent variables along the path connecting them. We show in Asbeh and Lerner [2016a] the existence and uniqueness of the value a latent non-collider and its observed child get under the influence of an exogenous ancestor; one and only one value of the latent non-collider (observed child)) changes with a change in the value of the exogenous. We also show that a latent collider and its observed child, both descendants of a set of exogenous variables, change their values in any two major configurations only if at least one of the exogenous variables has changed its value in the corresponding two configurations of this exogenous variable.

2.3 PCC BY DATA CLUSTERING

Practically, we use observational data that were generated from an unknown LVM and measured over the observed variables. We define an *observed value configuration, observed major value configuration,* and *observed minor value configuration* due to $\mathbf{ex}$ as the parts in $\mathbf{en}$, MAV, and a minor value configuration, respectively, that correspond to the observed variables. We show in Asbeh and Lerner [2016a] that there is only a single observed major value configuration to each exogenous configuration $\mathbf{ex}$ of $\mathbf{EX}$, and there are different observed major value configurations to different exogenous configurations. But, due to the probabilistic nature of BNs, each observed value configuration due to **ex** may be represented by several data points. Clustering these data points may produce several clusters for each **ex**, and each cluster corresponds to another observed value configuration. However, one and only one of the clusters corresponds to each of the observed major value configurations for a specific **ex**, whereas the other clusters correspond to observed minor value configurations.

We define the single cluster that corresponds to the observed major value configuration, and thus also represents the major effect due to configuration $\mathbf{ex}$ of $\mathbf{EX}$, as the *major cluster* for $\mathbf{ex}$, and all the clusters that correspond to the observed minor value configurations due to minor effects as *minor clusters*. However, we distinguish between different types of minor effects/clusters. A *k-order minor effect* is a minor effect in which exactly *k* endogenous variables in $\mathbf{EN}$ correspond to minor local effects. An $\mathbf{en}$ corresponding to a *k*-order minor effect

is a *k-order minor value configuration*. In addition, minor clusters that correspond to *k*-order minor effects are *k-order minor clusters*. The set of all major clusters (corresponding to all observed major value configurations) 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** configurations. Therefore, the identification of all major clusters is a key to the discovery of exogenous variables and their causal interrelations. For this purpose, we introduce the concept of *pairwise cluster comparison* (PCC). PCC measures the differences between two clusters; each represents the response of LVM to another **ex**.

Pairwise cluster comparison is a procedure by which pairs of clusters are compared, e.g., through a comparison of their centroids. The result of PCC between a pair of centroids of dimension $|\mathbf{O}|$, where $\mathbf{O}$ is the set of observed variables, is a binary vector of size $|\mathbf{O}|$ in which each element is 1 or 0 depending, respectively, on whether or not there is a difference between the corresponding elements in the compared centroids. When PCC is between clusters that represent observed major value configurations (i.e., PCC between major clusters), an element of 1 identifies an observed variable that has changed its value between the compared clusters due to a change in **ex**. Thus, the 1s in a major–major PCC provide evidence of causal relationships between **EX** and **O**. Practically, LPCC always identifies all observed variables that are represented by 1s *together* in *all* PCCs as the observed descendants of the same exogenous variable (Section 3.1). However, due to the probabilistic nature of BN and the existence of endogenous latents (mediating the connections from **EX** to **O**), some of the clusters are *k*-order minor clusters (in different orders), representing *k*-order minor configurations/effects. Minor clusters are more difficult to identify than major clusters because the latter reflect the major effects of **EX** on **EN** and, therefore, are considerably more populated by data points than the former. Nevertheless, minor clusters are important in causal discovery by LPCC even though a major–minor PCC cannot tell the effect of **EX** on **EN** because an observed variable in two compared (major and minor) clusters should not necessarily change its value as a result of a change in **ex**. Their importance is because a major cluster cannot indicate (when compared with another major cluster) the existence of minor values. On the contrary, PCC between major and minor clusters shows (through the number of 1s) the number of minor values represented in the minor cluster, and this is exploited by LPCC for identifying the endogenous latents and interrelations among them (Section 3.4). That is, PCC is the source to identify causal relationships in the unknown LVM; major–major PCCs are used for identifying the exogenous variables and their descendants, and major–minor PCCs are used for identifying the endogenous latents, their interrelations, and their observed children.

3 OVERVIEW OF LPCC

LPCC is fed by samples from the observed variables. It clusters the data using the self-organizing map (SOM) algorithm (although any algorithm that does not require a prior setting of the number of clusters is appropriate), and selects an initial set of major clusters. Then LPCC learns an LVM in two stages. First, it identifies (Section 3.1) latent variables (and their observed descendants) without distinguishing exogenous from endogenous latents, before distinguishing latent colliders from exogenous variables (Section 3.2). LPCC iteratively improves the selection of major clusters (Section 3.3), and the entire stage is repeated until convergence. Second, LPCC identifies endogenous latent non-colliders among the previously identified latent variables and splits these two types of latent variables from each other before finding the links between them (Section 3.4).

3.1 IDENTIFICATION OF LATENT VARIABLES

We demonstrate the relations between PCC and learning an LVM by an example. G1 in Figure 1 has two exogenous variables, L1 and L2, each having three children X1, X2, X3 and X4, X5, X6, respectively. [3] For the example, assume all variables are binary, i.e., L1 and L2 have four possible **ex**s (L1L2= 00, 01, 10, 11). We used a uniform distribution over L1 and L2 and set the probabilities of an observed child, X_i, $i = 1, \ldots, 6$, given its latent parent, $L_k, k = 1, 2$ (only if L_k is a direct parent of X_i, e.g., L1 and X1), to be $P(X_i = v \mid L_k = v) = 0.8, v = 0, 1$. First, we generated a synthetic data set of 1,000 patterns from G1. Second, using SOM [Kohonen, 1997], we clustered the data and found 16 clusters, of which four were major (Section 3.3 gives details on how to identify major clusters). 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 their corresponding PCCs are given in Table 1b. For example, *PCC1,2*, comparing clusters *C1* and *C2*, shows that when moving from *C1* to *C2*, only the values corresponding to variables X1–X3 have been changed (i.e., $\delta X_1 = \delta X_2 = \delta X_3 = 1$ in Table 1b). Also *PCC1,4, PCC2,3*, and *PCC3,4* show that X1–X3 always change their values together. This may be evidence that X1–X3 are descendants of the same exogenous la-

[3] We determined three indicators per latent in all true models we learn (Figure 1) because it is required by BPC, making the experimental evaluation in Section 4 fair.

Table 1: (a) Centroids of Major Clusters for G1 and (b) PCCs between These Major Clusters

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	δX1	δX2	δX3	δX4	δX5	δX6
1-2	1	1	1	0	0	0
1-3	0	0	0	1	1	1
1-4	1	1	1	1	1	1
2-3	1	1	1	1	1	1
2-4	0	0	0	1	1	1
3-4	1	1	1	0	0	0

(b)

tent, which, as we know from the true graph G1, is L1. Note, however that *PCC1,4* and *PCC2,3* show that the values corresponding to X4–X6 have changed together too, whereas these values did not change in *PCC1,2* and *PCC3,4*. Because X4–X6 changed their values only in *PCC1,4* and *PCC2,3* but not in *PCC1,2* and *PCC3,4*, they cannot be descendants of L1. This strengthens the evidence that X1–X3 are the only descendants of L1. A similar analysis using *PCC1,3* and *PCC2,4* will identify that X4–X6 are descendants of another latent variable (L2, as we know).

Therefore, we define a maximal set of observed (**MSO**) variables as the set of variables that always changes its values together in each major–major PCC in which at least one of the variables changes value. For example, X1 (Table 1b) changes its value in *PCC1,2*, *PCC1,4*, *PCC2,3*, and *PCC3,4* and always together with X2 and X3 (and vice versa). Thus, {X1, X2, X3} (and similarly {X4, X5, X6}) is an **MSO**. Each **MSO** includes descendants of the same latent variable *L*, and once LPCC identifies this **MSO**, it introduces *L* to the learned graph as a new latent variable, together with all the observed variables that are included in this **MSO** as *L*'s children. We prove in Asbeh and Lerner [2016a] that every observed variable belongs to one and only one **MSO**, i.e., **MSO**s corresponding to the learned latents are disjoint, which means that LPCC learns a pure measurement model. We also prove that variables of a particular **MSO** are children of a particular exogenous latent variable *EX* or its latent non-collider descendant or children of a particular latent collider *C*. This guarantees that each of multiple latent variables (either an exogenous or any of its non-collider descendants or a collider) is identified by its own MSO.

LPCC cannot yet distinguish between exogenous latents and latent colliders because the main goal at this stage was to identify latent variables and their relations with the observed variables. In Section 3.2, we distinguish these two types of variables, whereas in Section 3.4, we use major–minor PCCs rather that major–major PCCs to distinguish latent non-colliders from exogenous latents.

3.2 DISTINGUISHING LATENT COLLIDERS

To demonstrate distinguishing latent colliders from exogenous variables, we use graph G2 in Figure 1. It shows two exogenous latent variables, L1 and L3, that collide in an endogenous latent variable, L2, each having three observed children X1–X3 (L1), X4–X6 (L2), and X7–X9 (L3). We assume for the example that all variables are binary. Having two exogenous variables, we expect to find four major clusters in the data; each will correspond to one of the four possible **ex**s (L1L3= 00, 01, 10, 11). As for G1 (Section 3.1), we expect the values of X1–X3 to change together in all PCCs following a change in the value of L1, and the values of X7–X9 to change together in all the PCCs following a change in the value of L3. However, the values of X4–X6 will change together with those of X1–X3 in part of the PCCs and together with those of X7–X9 in the remaining PCCs, but always together in all of the PCCs. This will be evidence that X4–X6 are descendants of the same latent collider (L2, as we know). That is, to learn that an already learned latent variable *L* is a collider for a set of other already learned (exogenous) latent ancestor variables $\mathbf{LA} \subset \mathbf{EX}$, LPCC requires that: (1) The values of the children of *L* will change with the values of descendants of different latent variables in **LA** in different parts of major–major PCCs; and (2) The values of the children of *L* will not change in any PCC unless the values of descendants of at least one of the variables in **LA** change. This insures that *L* does not change independently of latents in **LA** that are *L*'s ancestors.

3.3 CHOOSING MAJOR CLUSTERS

Due to a lack of prior information regarding the distribution of latent variables, LPCC, first, assumes a uniform distribution over a latent and selects the major clusters based on their size, i.e., the number of patterns clustered. Clusters that are larger than the average cluster size are selected as majors. However, this initial selection may generate: 1) false negative errors (i.e., deciding a major cluster is minor), when a latent variable *L* has a skewed distribution over its values, and then a rare value can be represented only by small clusters that could not be cho-

sen as majors; and 2) false positive errors (i.e., deciding a minor cluster is major), when due to similar probabilities of different values of an observed child given its parent *L* and a large sample size, a cluster that is supposed to be minor becomes too large and is selected as major.

To avoid such errors, LPCC learns iteratively. Following the selection of major clusters based on their sizes and learning a graph, it becomes possible to find the cardinalities of the latent variables and all possible **ex**s. For each **ex**, we can use the most probable cluster given the data as the major cluster, and using an EM-style procedure [Dempster et al., 1977], update the set of major clusters iteratively and probabilistically to augment LPCC to learn more accurate graphs. The final graph may not be optimal since it depends on the initial graph, but it is an improved version of the initial graph.

3.4 IDENTIFICATION OF LATENT NON-COLLIDER VARIABLES

So far, the latent non-colliders that are descendants of an exogenous variable *EX* were temporarily combined with it, and all their observed children were temporarily combined with the direct children of *EX*. To exemplify that, check G3 in Figure 1, showing a serial connection of L1, L2, and L3. Assume each latent has three observed children, and all are binary. L1 is the only *EX* with two possible ***ex***s (L1= 0, 1), and L2 and L3 are *NC*s; L2 is a child of L1 and a parent of L3. We set the probabilities of: 1) L1 uniformly; 2) an observed child X_i, i= 1,. . ., 9, given its latent parent L_k, k= 1, 2, 3 (if this is a direct parent), as $P\left(X_i{=}v \mid L_k{=}v\right) = 0.8, v{=}\ 0, 1$; and 3) an endogenous latent L_j, j= 2, 3, given its latent parent L_k, k= 1, 2 (if this is a direct parent), as $P\left(L_j{=}v \mid L_k{=}v\right){=}\ 0.8, v{=}\ 0, 1$. We generated 1,000 patterns from G3 over the nine observed variables. Table 2 presents ten of the seventeen largest clusters by their centroids and sizes, from which *C1* and *C2* were selected as major. This meets our expectation of two major clusters corresponding to the two possible **ex**s of L1. However, because all the elements in *PCC1,2* are 1s (compare *C1* and *C2* in Table 2), the nine observed variables establish a single **MSO** and thus are considered descendants of the same exogenous variable. That is, the model learned in the first stage of LPCC has only one exogenous latent variable (i.e., L1) with direct children that are the nine observed descendants; contrary to G3. Since L2 and L3, which are latent non-colliders that are descendants of L1, were combined with L1, LPCC should now split them from L1 along with their observed children.

Such a split is based on major–minor (rather than major–major) PCCs. First, we define a first-order minor cluster

Table 2: Ten of the Seventeen Largest Clusters for G3

	X1	*X2*	*X3*	*X4*	*X5*	*X6*	*X7*	*X8*	*X9*	size
C1	1	1	1	1	1	1	1	1	1	49
C2	0	0	0	0	0	0	0	0	0	47
C3	1	1	1	1	1	1	1	1	0	28
C4	0	0	0	0	0	0	0	1	0	24
C5	0	1	0	0	0	0	0	0	0	22
C6	1	1	1	1	1	1	0	0	0	22
C7	0	0	1	0	0	0	0	0	0	21
C8	0	0	0	1	1	1	1	1	1	19
C9	0	0	0	0	0	0	1	1	1	18
C10	1	1	1	0	0	0	0	0	0	16

(1-MC) as a cluster that corresponds to a 1-order minor value configuration, which exists when exactly one endogenous variable in **EN** (either latent or observed) has a minor local value as a response to a value that *EX* has obtained[4]. To reveal the existence of latent non-colliders that were previously combined with *EX* and splits them from *EX*, we analyze for each *EX*, PCCs between 1-MCs and the major clusters that identified *EX*.

Table 3: All 2S-PCCs for G3

PCC	δX1	δX2	δX3	δX4	δX5	δX6	δX7	δX8	δX9
1-6	0	0	0	0	0	0	1	1	1
2-6	1	1	1	1	1	1	0	0	0
1-8	1	1	1	0	0	0	0	0	0
2-8	0	0	0	1	1	1	1	1	1
1-9	1	1	1	1	1	1	0	0	0
2-9	0	0	0	0	0	0	1	1	1
1-10	0	0	0	1	1	1	1	1	1
2-10	1	1	1	0	0	0	0	0	0

Table 4: PCCs for *C3* with *C1* and *C2* in Learning G3

PCC	δX1	δX2	δX3	δX4	δX5	δX6	δX7	δX8	δX9
1-3	0	0	0	0	0	0	0	0	1
2-3	1	1	1	1	1	1	1	1	0

Thus, second, we define PCCs between 1-MCs and major clusters that show two sets of two or more observed variables having the same value, different than that of the other set, as 2S-PCC (i.e., PCC of "two sets") and the corresponding 1-MC as 2S-MC. To identify a latent non-collider that was combined to an exogenous latent *EX*, we consider only 2S-PCCs; these PCCs are the result of comparing all the 2S-MCs among the 1-MCs for *EX* with the major clusters that revealed *EX*. For example, Table 3 presents all 2S-PCCs for G3. While a PCC that is not a 2S-PCC identifies a minor value of an observed descendant of *EX* (e.g., *PCC1,3*, where *C1* is major and *C3* is 1-MC, identifies a minor value of X9; see Table 4), a 2S-PCC identifies a minor value in a latent non-collider descendant of *EX* and thereby the existence of this latent. The latter is seen, e.g., in Table 3, in *PCC1,6* showing the change between *C1* (major cluster) and *C6* (1-MC) in the

[4]Based on the identification of a 1-MC [Asbeh and Lerner, 2016a], we find, e.g., in learning G3 that *C2* is the minimal major cluster, and all other clusters (Table 2) are 1-MCs.

values of X7–X9 and in *PCC2,6* showing the change between *C2* (major cluster) and *C6* in the values of X1–X6, thus identifying the existence of a latent non-collider (L3 as we know). This identification yields a split of exogenous L1 into two latents: one (L3) is a parent of X7-X9, and the other is a parent of X1–X6 (which later will split again to L1 and L2, each with its own three children).

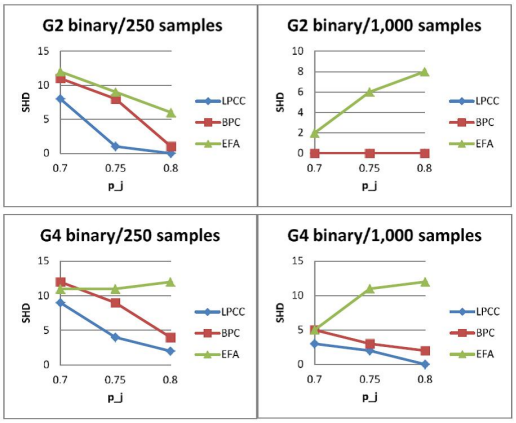
However, relying only on part of the 2S-PCCs may be inadequate to conclude on all possible splits (e.g., *PCC1,8* and *PCC2,8* show that X1–X3 and X4–X9 are children of different latents, but do not suggest the split of X7–X9 as *PCC1,6* and *PCC2,6* do). Thus, it is necessary to introduce for 2S-PCCs a *maximal set of observed variables* (**2S-MSO**) that always change their values together in all 2S-PCCs. For example, X1 in Table 3 changes its value in *PCC2,6*, *PCC1,8*, *PCC1,9*, and *PCC2,10* and always together with X2 and X3 (and the other way around). Thus, {X1, X2, X3} and similarly {X4, X5, X6} and {X7, X8, X9} are **2S-MSO**s. Each **2S-MSO** includes children of the same latent non-collider, which is a descendant of *EX*, or *EX* itself. LPCC detects **2S-MSO**s for each *EX* and thereby identifies its possible splits.

Finally, we show in Asbeh and Lerner [2016a] how to identify links between split latents. For a serial connection, LPCC finds the source (*EX*) and latent sink on the path between them, but not who is who, and thus it can only learn this connection as undirected. For a diverging connection, LPCC learns all directed links among the latents. That is, LPCC learns a pattern over the structural model of $\boldsymbol{G}$, which represents a Markov equivalence class of models among the latents, where in the special case in which $\boldsymbol{G}$ has no serial connection, LPCC learns the true graph.

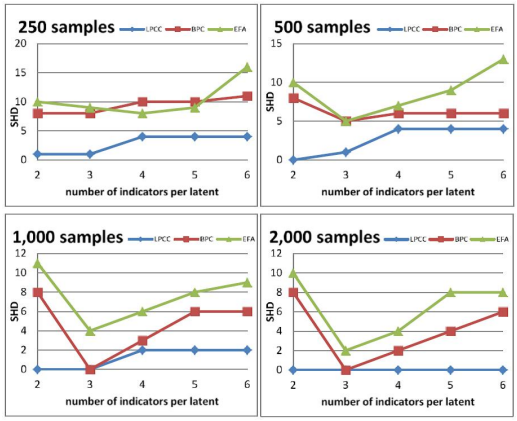
Based on the concepts outlined in Section 3, we formally introduce the LPCC algorithm in Asbeh and Lerner [2016b] and thoroughly evaluate it experimentally using synthetic and real-world data compared with other algorithms. The complete results are given in Asbeh and Lerner [2016b], and a snapshot of them is provided in Section 4.

4 EXPERIMENTAL EVALUATION

Simulated data: We used Tetrad IV to construct graphs G2 and G4 of Figure 1 for three parameterization levels that differ by the conditional probabilities p_j=0.7, 0.75, and 0.8 between a latent and each of its children. Each such level imposes a different complexity on the model and thereby affects the task of learning the latent model and the causal relations (i.e., p_j=0.7 poses a larger challenge to learning than p_j=0.75) [full details about the types of variables and parametrization schemes are omitted; see Asbeh and Lerner [2016b]]. We drew data sets of between 250 and 2,000 samples and report on the structural hamming distance (SHD) [Tsamardinos et al., 2006] as a performance measure for learning the LVM structure. SHD is a structural measure that accounts for all the possible learning errors (the lower value is the better one): addition and deletion of an undirected edge, and addition, removal, and reversal of edge orientation.



(a) SHD of LPCC/BPC/EFA vs. parametrization level. For G2/1,000 samples, LPCC/BPC learn perfectly.



(b) SHD of LPCC/BPC/EFA vs. number of indicators per latent in G2, $p_j = 0.75$, and four sample sizes.

Figure 2: Structural Correctness of Learning Algorithms.

Figure 2a shows SHD values for the LPCC, BPC, and EFA algorithms for increasing parametrization levels for four combinations of learned graphs and sample sizes. It shows that LPCC and BPC improve performance, as expected, with increased levels of latent-observed variable correlation (p_j). LPCC never falls behind BPC, and its advantage over BPC is especially vivid for a small sample size. EFA, besides falling behind LPCC and BPC, also demonstrates worsening of performance with increasing the parametrization level, especially for large sample sizes. Larger parametrization levels increase the

chances of an EFA to learn links between latent variables and observed variables – some of which are not between a latent and its real child – to compensate for the algorithm's inability to identify links among latents (EFA assumes latents are uncorrelated). EFA is inferior to LPCC for all parametrization levels, sample sizes, and graphs.

Figure 2b shows SHD values of the LPCC, BPC, and EFA algorithms for increasing numbers of binary indicators per latent variable in G2, a parametrization level of 0.75, and four sample sizes. The figure exhibits superiority of LPCC over BPC and EFA for all scenarios. While LPCC hardly worsens its performance with the increase of complexity, both BPC and EFA are affected by this increase. They also have a difficulty in learning an LVM for which latent variables have exactly two indicators.

Real-world data – The political action survey (PAS): We evaluated LPCC using a simplified PAS data set over six variables (Joreskog, 2004): NOSAY, VOTING, COMPLEX, NOCARE, TOUCH, and INTEREST. These variables represent political efficacy and correspond to questions to which the respondents have to give their degree of agreement on a discrete ordinal scale of four values. This data set contains the responses from a sample of 1,076 US respondents. A model consisting of two latents that correspond to a previously established theoretical trait of Efficacy and Responsiveness (Figure 3a) discards VOTING Joreskog [2004] based on the argument that the question for VOTING is not clearly phrased.

Similar to the theoretical model, LPCC finds two latents (Figure 3b): One corresponds to NOSAY and VOTING and the other corresponds to NOCARE, TOUCH, and INTEREST. Compared with the theoretical model, LPCC misses the edge between Efficacy and NOCARE and the bidirectional edge between the latents. Both edges are not supposed to be discovered by LPCC or BSPC/BPC; the former because the algorithms learn a pure measurement model in which each observed variable has only one latent parent and the latter because no cycles are assumed. Nevertheless, compared with the theoretical model, LPCC makes no use of prior knowledge. BSPC output (Figure 3c) is very similar to LPCC output, except for NOCARE, which was not identified by BSPC. Both algorithms identify VOTING as a child of Efficacy (at the expense of COMPLEX). The outputs of the BPC algorithm (Figures 3d,e) are poorer than those of LPCC and BSPC and are sensitive to the significance level. The output of the FCI algorithm (not shown here) using any significance level is not sufficient (showing, e.g., that NOSAY and INTEREST potentially have a latent common cause where these two variables are indicators of different latents in the theoretical model).

Using simulated and real-world data, we show in Asbeh and Lerner [2016b] that LPCC improves accuracy with sample size, it can learn large LVMs, and it has consistently good results compared to models that are expert-based or learned by state-of-the-art algorithms. Applied to two original domains, LPCC helped identify possible causes of young drivers' involvement in road accidents and cell subpopulations in the immune system.

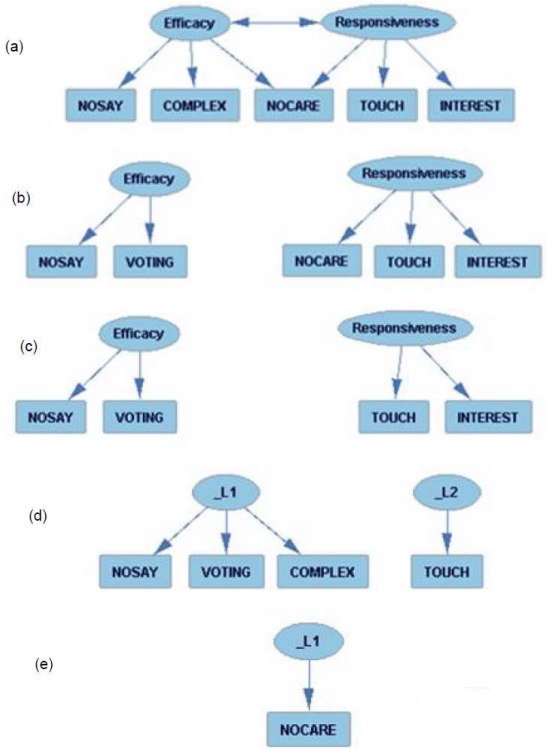


Figure 3: PAS: outputs of (a) a true model, (b) LPCC, (c) BSPC, and BPC with (d) α =0.01, 0.05, and (e) α =0.1.

5 SUMMARY AND CONCLUSIONS

In this work, we introduced the PCC concept and LPCC algorithm for learning discrete LVMs: 1) the PCC concept to learn an LVM ties graphical models with data clustering; 2) LPCC learns MIMs; 3) LPCC is not limited to latent-tree models and does not assume linearity; 4) LPCC assumes that the measurement model of the true graph is pure, but, if the true graph is not, it learns a pure sub-model of the true model, if one exists. LPCC also assumes that a latent collider does not have any latent descendants; 5) LPCC is a two-stage algorithm that exploits PCC. First, it learns the exogenous latents and latent colliders, as well as their observed descendants, and second, it learns the endogenous latent non-colliders and their children by splitting these latents from their previously learned latent ancestors; and 6) LPCC learns an equivalence class of the structural model of the true graph.

References

N. Asbeh and B. Lerner. Learning latent variable models by pairwise cluster comparison: Part I – Theory and overview. *Journal of Machine Learning Research*, 17 (224):1–52, 2016a.

N. Asbeh and B. Lerner. Learning latent variable models by pairwise cluster comparison: Part II – Algorithm and evaluation. *Journal of Machine Learning Research*, 17(233):1–45, 2016b.

D. J. Bartholomew, F. Steele, I. Moustaki, and J. I. Galbraith. *The Analysis and Interpretation of Multivariate Data for Social Scientists (Texts in Statistical Science Series)*. Chapman & Hall/CRC Press, Boca Raton, Florida, USA, 2002.

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 13:479–485, 2000.

S. Harmeling and C. K. I. Williams. Greedy learning of binary latent trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1087–1097, 2011.

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

R. Klee. *Introduction to the Philosophy of Science: Cutting Nature at its Seams*. Oxford University Press, New York, New York, 1997.

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

J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann Press, San Mateo, California, 1988.

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

J. Pearl and T. Verma. A theory of inferred causation. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 441–452, Cambridge, MA, 1991.

R. Scheines, P. Spirtes, C. Glymour, C. Meek, and T. Richardson. The tetrad project: Constraint based aids to causal model specification. Technical report, Department of Philosophy, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1995.

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

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. Calculation of entailed rank constraints in partially non-linear and cyclic models. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, pages 606–615, Bellevue, Washington, 2013.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and Search*. MIT Press, New York, New York, 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.

Y. Wang, N. L. Zhang, and T. Chen. Latent-tree models and approximate inference in Bayesian networks. *Journal of Artificial Intelligence Research*, 32:879–900, 2008.

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