Journal of Machine Learning Research 17 (2016) 1-45 Submitted 9/14; Revised 2/16; Published 12/16

Learning Latent Variable Models by Pairwise Cluster Comparison: Part II – Algorithm and Evaluation

Nuaman Asbeh ASBEH@POST.BGU.AC.IL
Department of Industrial Engineering and Management
Ben-Gurion University of the Negev
Beer Sheva, 84105, Israel

Boaz Lerner BOAZ@BGU.AC.IL
Department of Industrial Engineering and Management
Ben-Gurion University of the Negev
Beer Sheva, 84105, Israel

Editors: Isabelle Guyon and Alexander Statnikov

Abstract

It is important for causal discovery to identify any latent variables that govern a problem and the relationships among them, given measurements in the observed world. In Part I of this paper, we were interested in learning a discrete latent variable model (LVM) and introduced the concept of *pairwise cluster comparison* (PCC) to identify causal relationships from clusters of data points and an overview of a two-stage algorithm for *learning PCC* (LPCC). First, LPCC learns exogenous latent variables and latent colliders, as well as their observed descendants, by using pairwise comparisons between data clusters in the measurement space that may explain latent causes. Second, LPCC identifies endogenous latent non-colliders with their observed children. In Part I, we showed that if the true graph has no serial connections, then LPCC returns the true graph, and if the true graph has a serial connection, then LPCC returns a pattern of the true graph. In this paper (Part II), we formally introduce the LPCC algorithm that implements the PCC concept. In addition, we thoroughly evaluate LPCC using simulated and real-world data sets in comparison to state-of-the-art algorithms. Besides using three real-world data sets, which have already been tested in learning an LVM, we also evaluate the algorithms using data sets that represent two original problems. The first problem is identifying young drivers' involvement in road accidents, and the second is identifying cellular subpopulations of the immune system from mass cytometry. The results of our evaluation show that LPCC improves in accuracy with the sample size, can learn large LVMs, and is accurate in learning compared to state-of-the-art algorithms. The code for the LPCC algorithm and data sets used in the experiments reported here are available online.

Keywords: learning latent variable models, graphical models, clustering, pure measurement model

1. Introduction

We began Part I by describing the task of learning a latent variable model (LVM). We dispensed with the linearity assumption (for a child given its parents) and concentrated on the discrete case. In addition, we did not limit our analysis to learning latent-tree models and focused on multiple indicator models (MIMs) that are a very important subclass of structural equation models (SEM) – models that are widely used in applied and social sciences to analyze causal relations. By borrowing ideas from unsupervised learning, we could introduce the notion of pairwise cluster comparison (PCC). PCC compares pairwise clusters of data points representing instantiations of the observed variables to identify those pairs of clusters that exhibit changes in the observed variables due to changes in their ancestor latent variables. Changes in a latent variable that are manifested in changes in its descendant observed variables reveal this latent variable and its causal paths of influence in the domain. Learning PCC (LPCC) was introduced as a tool to transform data clusters into knowledge about latent variables – their number, types, cardinalities, and interrelations among themselves and with the observed variables – that is needed to learn an LVM.

Part I provided preliminaries and the theoretical support of LPCC. Several definitions and theorems that were already introduced also play an important role in Part II. To ease reading Part II, on the one hand, and to supply the necessary theoretical background, on the other hand, we have summarized these definitions, propositions, and theorems from Part I here in Appendix A. Following is a brief summary of the PCC concept and LPCC algorithm; the full details appear in Part I.

First in the LPCC algorithm is clustering of data that are sampled from the observed variables in the unknown model. Clustering in the current implementation is based on the self-organizing map (SOM) algorithm (Kohonen, 1997), although any other clustering algorithm that does not need a preliminary determination of the number of clusters may be suitable.[1] Second, LPCC selects an initial set of major clusters (Section 4.3 of Part I; Definition 12 in Appendix A[2]). Third, LPCC learns an LVM in two stages. In the first stage (Section 4.1 of Part I), LPCC analyzes PCCs[3] (Definition 15) between two major clusters to find maximal sets of observed (**MSO** by Definition 16) variables that always change together. By Theorem 1, variables of a particular **MSO** are children of a particular exogenous latent variable or its latent non-collider descendant or children of a particular latent collider. This stage allows the identification of exogenous latent variables and latent colliders together and their corresponding observed descendants. Then (Section 4.2 of Part I), LPCC distinguishes the latent colliders from the exogenous latent variables using Theorem 2. To complete this stage, LPCC iteratively improves the selection of the major clusters (Section 4.3 of Part I), and the entire stage is repeated until convergence. In the second stage, LPCC identifies endogenous latent non-colliders with their children (Section 4.4 of Part I). Because distinguishing endogenous latent non-colliders from their exogenous ancestors could not be performed using major-major PCCs, in this stage LPCC needs to apply a mechanism to split these two types of latent variables from each other and then direct them using comparison of major clusters to (a special type of) minor clusters (2S-MC; Definition 14) that correspond to 2-order minor effects (Definition 13). For this task, LPCC

[1] See for example Section 3.6, where we replaced SOM with hierarchical clustering.

[2] The definitions and theorems that are mentioned here are borrowed from Part I and are summarized in Appendix A.

[3] PCC is a procedure by which pairs of clusters are compared through a comparison of their centroids, and the result can be represented by a binary vector 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 clusters.

analyzes 2S-PCCs (Definition 18), which are PCCs between major and minor clusters that show two sets (this is the source of "2S" in the name 2S-PCC) of two or more elements in the PCC, and identifies **2S-MSO**s (Definition 19), which are maximal sets of observed variables that always change their values together in all 2S-PCCs. Different **2S-MSO**s due to an exogenous latent variable represent latent non-colliders that are descendants of this exogenous variable; hence, LPCC can distinguish between the two types of variables by analyzing **2S-MSO**s (Theorem 3). To direct the edges between latent non-colliders on a path emerging in an exogenous latent, LPCC checks changes of several 2S-PCCs with respect to changes of the latent non-colliders' exogenous ancestor. Theorem 4 guarantees that LPCC finds all diverging connections and represents all serial connections using a pattern of the true graph, which completes learning the LVM. A flowchart of the LPCC algorithm is given in Figure 1.

A main section of Part II is a formal description of the two-stage LPCC algorithm, which is founded on the PCC concept. Part II also provides an experimental evaluation of LPCC, in comparison to state-of-the-art algorithms, using simulated data sets (Section 3.1) and real-world data sets (Sections 3.2–3.6). The outline of the paper is as follows:

- **Section** 2: **The LPCC algorithm** introduces and formally describes a two-stage algorithm that implements the PCC concept;
- **Section 3: LPCC evaluation** evaluates LPCC, in comparison to state-of-the-art algorithms, using simulated data sets (Section 3.1) and real-world data sets (Sections 3.2–3.6);
- **Section 4**: **Related works** compares LPCC to state-of-the-art LVM learning algorithms;
- **Section 5: Discussion** summarizes the theoretical advantages (from Part I) and the practical benefits (from this part) of using LPCC;
- **Appendix A** provides essential assumptions, definitions, propositions, and theorems from Part I;
- **Appendix B** supplies additional results for the experiments with the simulated data sets (Section 3.1); and
- **Appendix C** provides PCC analysis for two example databases.

2. The LPCC algorithm

We introduced a two-stage algorithm, LPCC, that implements the PCC concept (Part I). The algorithm gets a data set **D** over the observed variables **O** and learns an LVM. In the first stage, LPCC learns the exogenous variables and the latent colliders as well as their descendants using the LEXC algorithm (Section 2.1). In the second stage, LPCC augments the graph learned by LEXC by learning the endogenous latent non-colliders and their children using the LNC algorithm (Section 2.2).

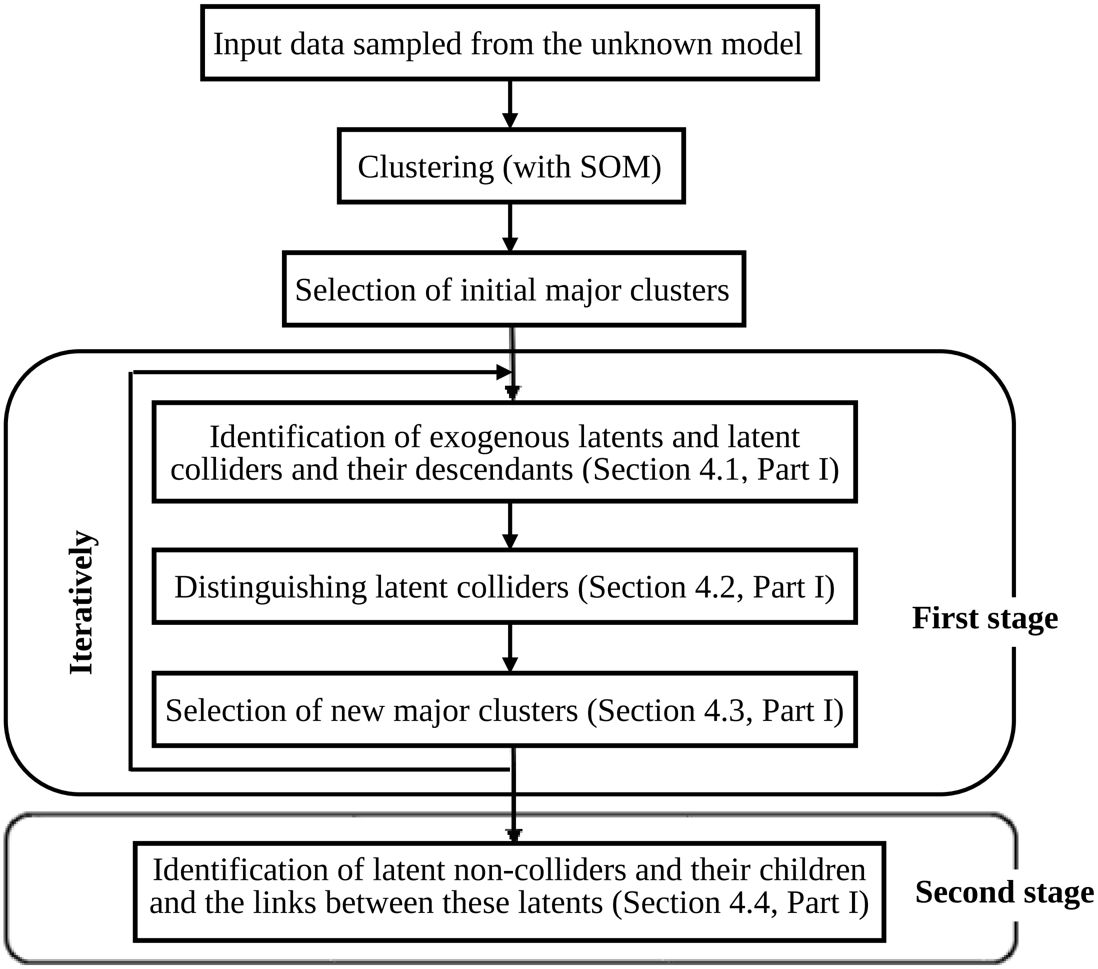
Input data sampled from the unknown model
Clustering (with SOM)
Selection of initial major clusters
Identification of exogenous latents and latent colliders and their descendants (Section 4.1, Part I)
Distinguishing latent colliders (Section 4.2, Part I)
Selection of new major clusters (Section 4.3, Part I)
Iteratively
First stage
Identification of latent non-colliders and their children and the links between these latents (Section 4.4, Part I)
Second stage

Figure 1: An overview of the LPCC algorithm as described in Part I.

2.1 Learning exogenous and latent colliders (LEXC)

LEXC (Algorithm 1) 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 c is represented by its centroid.

In the second step, LEXC performs an initial selection of the major clusters set (Definition 12 in Appendix A), 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 4.3 of Part I). $\mathbf{MC} = \{\mathbf{MC}_i\}_{i=1}^{n}$ is a matrix that holds information about the major clusters, where each matrix row represents a centroid of one of the n major clusters (see, e.g., Table 2 in Part I).

In the third step, LEXC creates a matrix that represents all PCCs (Definition 15), derived from **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 (see, e.g., Table 4 in Part I). The k-th element of $\mathbf{PCC}_{ij}$ represents by "1" a change in value, if one exists, in the observed variable $O_k \in \mathbf{O}$ when comparing $\mathbf{MC}_i$ and $\mathbf{MC}_j$ (for example, Table 4 in Part I shows a change in element 7, corresponding to X_7, of *PCC2,9* between *C2* and *C9*). We use the notation $\mathbf{PCC}_{ij} \rightarrow \delta O_k$ if the value of O_k has been changed and $\mathbf{PCC}_{ij} \rightarrow \neg\delta O_k$ otherwise.

In the fourth step, LEXC identifies exogenous latents and their descendants (Theorem 1) using a matrix **MSOS** that holds all **MSO**s (Definition 16) that always change their corresponding values together in all major–major PCCs in **PCCM**. For each identified $\mathbf{MSO}_i$, LEXC adds a latent L_i to $\mathbf{G}$ and to a latent set $\mathbf{L}$ and also edges from L_i to each observed variable $O \in \mathbf{MSO}_i$. 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, each corresponding to one condition in Theorem 2, the latent variables that are collider nodes in the graph along with their latent ancestors. In the first phase, LEXC considers for each latent variable $L_i \in \mathbf{L}$, a set of potential ancestors from the other latents in $\mathbf{L}$. We call them potential ancestors because another condition should be fulfilled in the second phase to turn them into actual ancestors. To simplify the notation, we represent the latent as an object and the set of potential ancestors as a field of this object, called **PAS** (for potential ancestor set). For example, L_i.**PAS** represents that LEXC identifies a potential ancestor set **PAS** to latent L_i. In addition, we use the notation $\mathbf{PCC}_{fg} \rightarrow \delta L_i$ if all of the variables in $\mathbf{Ch}_i$ change their values in $\mathbf{PCC}_{fg} \in \mathbf{PCCM}$ and $\mathbf{PCC}_{fg} \rightarrow \neg\delta L_i$ otherwise. In the first phase of the fifth step, LEXC checks for each $L_i \in \mathbf{L}$ whether there exists a vector $\mathbf{PCC}_{fg} \in \mathbf{PCCM}$ in which L_i changes value together with $L_j \in \mathbf{L}$, but not with $L_k \in \mathbf{L}, \forall k \neq i, j$, and if so, it adds L_j to L_i.**PAS**. At the end of this phase, the set L_i.**PAS** contains all of the latents in $\mathbf{L}$ that change values with L_i in **PCCM**. Still, this is not enough to decide that L_i is a collider of the variables in L_i.**PAS**. An additional condition must be fulfilled, which is that L_i should never have changed in any $\mathbf{PCC}_{fg} \in \mathbf{PCCM}$ unless at least one of the variables in L_i.**PAS** has also changed in this $\boldsymbol{P}\mathbf{CC}_{fg}$ (Section 4.2 of Part I). The second phase of the fifth step checks this condition, and

Algorithm 1 *LEXC*

```
Input: Data set $\mathbf{D}$ over the observed variables $\mathbf{O}$
Output: Graph $\mathbf{G}$ that includes the exogenous latent variables and the latent colliders and their descendants in LVM
Initialize:
Create an empty graph $\mathbf{G}$ over the observed variables $\mathbf{O}$
$\mathbf{c} = \phi, \mathbf{MC} = 0, \mathbf{PCCM} = 0, \mathbf{L} = \phi, \mathbf{MSOS} = \phi$
% First step: perform clustering
$\mathbf{c} \leftarrow$ perform clustering on $\mathbf{D}$ and represent each cluster by its centroid
% Second step: select an initial set of major clusters
For each $c_i \in \mathbf{c}$
    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 each $\mathbf{MC}_i$, $\mathbf{MC}_j \in \mathbf{MC}$, $j > i$
    $\mathbf{PCCM} \leftarrow$ compute $\mathbf{PCC}_{ij}$
% Fourth step: identify latent variables and their observed children
$\mathbf{MSOS} \leftarrow$ find all possible $\mathbf{MSO}$s using $\mathbf{PCCM}$
For each $\mathbf{MSO}_i \in \mathbf{MSOS}$
    Add a new latent variable $L_i$ to $\mathbf{G}$ and to $\mathbf{L}$
    For each observed variable $O \in \mathbf{MSO}_i$
        Add $O$ and an edge $L_i \rightarrow O$ to $\mathbf{G}$
% Fifth step: identify latent collider variables and their parents
For each $L_i \in \mathbf{L}$
% First phase
    $L_i.\mathbf{PAS} = \phi$
    For each $L_j \in \mathbf{L}$, $j \neq i$
        If $\exists\ \mathbf{PCC}_{fg} \in \mathbf{PCCM}$ s.t ($\mathbf{PCC}_{fg} \rightarrow \delta L_i \wedge\ \mathbf{PCC}_{fg} \rightarrow \delta L_j \wedge \mathbf{PCC}_{fg} \rightarrow \neg \delta L_k, \forall k \neq i, j$), then
            Add $L_j$ to $L_i.\mathbf{PAS}$
% Second phase
    if $\forall\ \mathbf{PCC}_{fg} \in \mathbf{PCCM}$ s.t $\mathbf{PCC}_{fg} \rightarrow \delta L_i$
        $\exists PAS \in L_i.\mathbf{PAS}$ s.t $\mathbf{PCC}_{fg} \rightarrow \delta PAS$
    then $\forall PAS \in L_i.\mathbf{PAS}$, add a new edge $PAS \rightarrow L_i$ to $\mathbf{G}$.
% Sixth step: search for a new set of major clusters
$\mathbf{NMC} = \phi$
Find the cardinality of each $L_i \in \mathbf{L}$, then identify $\mathbf{ex}$s
    For each $\mathbf{ex} \in \mathbf{ex}$s
        Find $c^* = argmax_{c \in \mathbf{c}} P(c \mid \mathbf{ex})$ and add $c^*$ to $\mathbf{NMC}$
    If $\mathbf{NMC} = \mathbf{MC}$
        Return $\mathbf{G}$
    Else
        $\mathbf{MC} \leftarrow \mathbf{NMC}$, $\mathbf{PCCM} = 0$, $\mathbf{L} = \phi$ , $\mathbf{G} \leftarrow$ empty graph over $\mathbf{O}$
        Go to “Third step”
```

if fulfilled, it adds an edge from each variable in L_i.**PAS** to L_i to complete the identification of L_i as a collider.

In the sixth and last step, and to deal with possible false positive and false negative errors (Section 4.3 of Part I), LEXC searches for a new set of major clusters **NMC** based on the already learned graph and all the clusters that initially were identified by SOM. First, LEXC learns for each latent $L_i \in \mathbf{L}$ its cardinality, which is the number of different value configurations of L_i corresponding to all value configurations of $\mathbf{Ch}_i$ in **D**. Each such value configuration of observed children is due to a value l_i of L_i, and we denote it by $l_i \rightarrow \boldsymbol{ch}_i$. Then, LEXC finds the set of all possible **ex**s (all possible configurations of all exogenous latents in **L**, $L_i \in \mathbf{L} \cap \mathbf{EX}$). For each **ex**, LEXC finds the most probable cluster, $c^* = argmax_{c \in \boldsymbol{c}} P(c|\mathbf{ex})$, where the posterior probability $P(c|\mathbf{ex})$ for each $c \in \mathbf{c}$ is approximated by the ratio between c's size and the size of **D**. Thus, the cluster for which the values corresponding to the children of $L_i \in \mathbf{L} \cap \mathbf{EX}$, $l_i \rightarrow \boldsymbol{ch}_i$, are most probable due to l_i in **ex** is selected as the most probable to represent this **ex**. Each such cluster is added to **NMC**. If **NMC**=**MC**, **NMC** cannot improve the graph, and thus LEXC stops and returns the learned graph **G**. Otherwise, LEXC reinitializes **MC** to be **NMC** and relearns a new graph.

2.2 Learning latent non-colliders (LNC)

Using the data set **D**, LNC has to split the set of latent variables **L** in graph **G**, which was learned by LEXC, into exogenous latents and latent non-colliders. First, LNC (Algorithm 2) adds |**L**| elements to the end of each vector in **D** and creates an incomplete data set **IND**. For a vector in **IND** for which values of the observed children for a specific latent $L_i \in \mathbf{L}$ take major values, the value of the latent can be reconstructed exactly, $l_i \rightarrow \boldsymbol{ch}_i$; however, when not all observed children take major values, this value of the latent cannot be reconstructed, and this is the reason why **IND** is incomplete. Second, using the EM algorithm (Lauritzen, 1995; Dempster et al., 1977) and **IND**, LNC learns (Section 4.4 of Part I) **G**'s parameters and uses them to compute a threshold (Appendix B in Part I) on the maximal size of 2-MCs. This threshold is needed to find 1-order minor clusters (1-MCs; Definition 14). Note that after learning the parameters, the graph turns into a model, M0. Third, for each exogenous latent $EX_i \in \mathbf{L} \cap \mathbf{EX}$ in turn, LNC tests if EX_i should be split (Section 4.4 of Part I). For this test, LNC needs first to find the set of 1-MCs for EX_i and compute all the PCCs between these clusters and the major clusters for EX_i. We denote the set of these PCCs by **PCCS**. Then, LNC finds all the PCCs in **PCCS** that are **2S-PCC** (Definition 18); these will be used to identify all possible **2S-MSO**s (Definition 19) and thus all possible latent non-collider descendants that should be split from EX_i (Theorem 3).

After identifying the latent non-colliders' descendants of EX_i and splitting them from EX_i, LNC finds the links between these latents (Section 4.4 of Part I). LNC first finds the set **L'** of all latents whose children change alone in some 2S-PCCs. These are the candidates to be EX_i or its leaves (Proposition 10). Then, for each $L' \in \mathbf{L}'$, LNC finds the 2S-PCCs in **2S-PCC** in which the observed children of L' do not change and are due to comparisons with the same major cluster. This set is denoted by **2S-PCC**'. Then, for every two latent non-collider descendants that were split from EX_i, LNC checks if there is a directed link between them using Theorem 4. Note that, we assume by default that L' is a leaf, so LNC does not need to redirect the links in the diverging connection case. After finding all the possible directed paths, LNC identifies if the connection is serial (in case $|\mathbf{L'}|$ is exactly two) and if so it makes the links on this path undirected; otherwise, the path is directed as part of a diverging connection. Finally, LNC returns a pattern **G**, which represents a Markov equivalence class of the true graph.

Algorithm 2 *LNC*

```
Input: Data set $\mathbf{D}$ over the observed variables $\mathbf{O}$ and the graph $\mathbf{G}$ learned by LEXC
Output: The final learned LVM $\mathbf{G}$
Initialize: $\mathbf{IND}$ =0, $\mathbf{PCCS}$= $\phi$, $\mathbf{2S\text{-}PCC}$= $\phi$, $\mathbf{2S\text{-}MSOS}$= $\phi$, $\mathbf{2S\text{-}PCC}'$= $\phi$
Create $\mathbf{IND}$ (see text)
Learn $\mathbf{G}$'s parameters using the EM algorithm to obtain an LVM, M0
For each latent $EX_i \in \mathbf{L}$
% Identify and split the latent non-collider descendants of $EX_i$
    Find the set of 1-MCs according to M0
    $\mathbf{PCCS} \leftarrow$ compute all PCCs between the 1-MCs and the major clusters for $EX_i$
    $\mathbf{2S\text{-}PCC} \leftarrow$ find all 2S-PCCs in $\mathbf{PCCS}$
    $\mathbf{2S\text{-}MSOS} \leftarrow$ find all possible $\mathbf{2S\text{-}MSO}$s using $\mathbf{2S\text{-}PCC}$
    For each $\mathbf{2S\text{-}MSO}_j \in \mathbf{2S\text{-}MSOS}$
        Add a latent non-collider $NC_j$ to $EX_i$, $\mathbf{L}$, and $\mathbf{G}$
        For each observed variable $O \in \mathbf{2S\text{-}MSO}_j$
            Split $O$ from the children of $EX_i$ and add an edge $NC_j \rightarrow O$ to $\mathbf{G}$
% Identify the links between the new latent non-colliders that were split from $EX_i$
    $\mathbf{L}' \leftarrow$ all latents that were split (including $EX_i$) and whose observed children change alone
    in some 2S-PCC
    For each $L' \in \mathbf{L}'$ % assume by default $L'$ is a leaf and apply Theorem 4
        $\mathbf{2S\text{-}PCC}' \leftarrow$ all 2S-PCCs in $\mathbf{2S\text{-}PCC}$ in which the observed children of $L'$ do not change
        For each two latent non-colliders $NC_j$,$NC_k$, $k \neq j$ that were split from $EX_i$:
            If
                1) the observed children of $NC_k$ always change with those of $NC_j$ in $\mathbf{2S\text{-}PCC}'$; and
                2) the observed children of $NC_j$ change $t$ times and the observed children of $NC_k$
                change $t+1$ times in $\mathbf{2S\text{-}PCC}'$
            Then add a directed edge from $NC_k$ to $NC_j$ to $\mathbf{G}$
% Identify if the connection is serial, and if so make the links in the path undirected
If $|\mathbf{L}'|$=2
    If there are two paths with the same latents but opposite directions, then make the edges
    between the latents undirected.
Return $\mathbf{G}$
```

3. LPCC Evaluation

We implemented the LPCC algorithm in Matlab, except for the SOM algorithm that was implemented using the SOM Toolbox (Vesanto et al., 2000). We evaluated LPCC using simulated data sets (Section 3.1) and five real-world data sets: data from the political action survey (Section 3.2), Holzinger and Swineford's data (Section 3.3), the HIV test data (Section 3.4), data of young driver (YD) involvement in road accidents (Section 3.5), and a mass cytometry data set of the immune system (Section 3.6). In the case of the real-world data sets, we did not have an objective measure for evaluation; thus, we compared the LPCC output to hypothesized, theoretical models from the literature and to the outputs of four state-of-the-art learning algorithms. The first algorithm is FCI (Spirtes et al., 2000), and because we noticed (see below) for the political action survey and Holzinger and Swineford's data sets that FCI is not suitable for learning MIM models, we did not use it for the other data sets. The second algorithm is for learning HLC models (Zhang, 2004), and since the theoretical models for all but the HIV data set are not latent-tree models, we used this algorithm only for the HIV data set. The third algorithm is exploratory factor analysis (EFA). Because the theoretical models for the political action survey and Holzinger and Swineford's data set were already tested by confirmatory factor analysis [(Joreskog, 2004); (Arbuckle, 1997, p. 375); and (Joreskog and Sorbom, 1989, p. 247)], we completed the examination of EFA also to the YD and mass cytometry data sets. The fourth algorithm, which is actually two algorithms, BuildPureClusters (BPC) and BuildSinglePureClusters (BSPC) of Silva (2005), is especially suitable for MIM models. BPC is Silva's (2005) main algorithm; hence, we used it in all the evaluations. 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 the Tetrad IV package, which can take discrete data (as in all the data sets in this evaluation) as input and treat them as continuous.[4] 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; Wishart, 1928), applying three significance levels of 0.01, 0.05 (Tetrad's default), and 0.1. For the simulated data sets, we compared LPCC to EFA and BPC.

3.1 Evaluation using simulated data sets

We used Tetrad IV to construct the graphs G1, G2, G3, and G4 of Figure 2, once with binary and once with ternary variables. The priors on the exogenous latents were always distributed uniformly. We compared performances for three parameterization levels that differ by the conditional probabilities, p_j=0.7, 0.75, and 0.8, between a latent L_k and each of its children EN_i. For all graphs in the binary case, except L2 in G2, $P(EN_i = v \mid L_k = v) = p_j$, $v = 0$ or 1. For all graphs in the ternary case, except L2 in G2, $P(EN_i = v \mid L_k = v) = p_j, P(EN_i \neq v \mid L_k = v) = (1 - p_j)/2$, $v = 0$, 1, or 2. Concerning L2 in G2, $P(L_2 = 0 \mid L_1 L_3 = 00, 01, 10) = P(L_2 = 1 \mid L_1 L_3 = 11) = p_j$ in the binary case and $P(L_2 = v \mid \max\{L_1, L_3\} = v) = p_j$ and $P(L_2 \neq v \mid \max\{L_1, L_3\} = v) = (1 - p_j)/2$ 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, using p_j=0.7 poses a larger challenge to learning than p_j=0.75, which poses a larger challenge 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=0.7, 0.75, and 0.8 are 0.4, 0.5,

[4] Although all our data sets are discrete and BSPC is the suggested algorithm in Silva (2005) for discrete data, BSPC is neither published nor implemented in Tetrad IV, and is only mentioned in a complementary chapter in Silva (2005) as a variant of BPC suitable for discrete data. Since no concrete algorithm is suggested for BSPC, we used BPC as described above. However, for the political action survey, we could use the results for BSPC that are provided in Silva (2005). The Tetrad package is available at http://www.phil.cmu.edu/projects/tetrad.

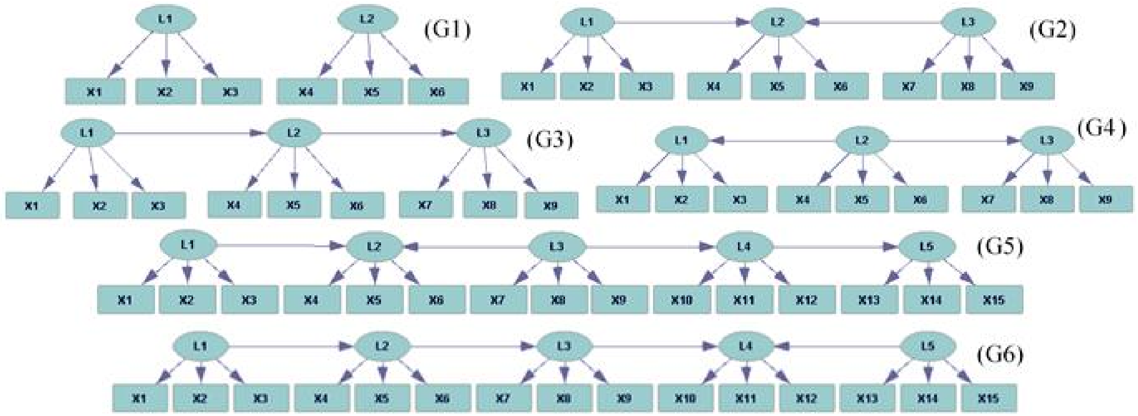

Figure 2: Example LVMs that are all MIMs. Each is based on a pure measurement model and a structural model of different complexity, posing a different challenge to a learning algorithm.

and 0.6, respectively. Note that correlation of 0.4 is relatively low, providing a great challenge to the learning algorithms, and trying to learn an LVM for lower correlation values yields poor results by all algorithms.[5] Tetrad IV was also used to draw data sets of 125, 250, 500, 750, 1000, and 2,000 samples for each test. Overall, we evaluated the LPCC algorithm using 144 synthetic data sets for four graphs (G1–G4), two types of variables, three parameterization levels, and six data set sizes.

In addition, we evaluated LPCC using the two large graphs in Figure 2, G5 and G6, which combine all types of links between the latents, such as serial, converging, and diverging. Each 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 for two parametric settings p_j=0.75 and 0.8. In all cases, we report on the structural hamming distance (SHD) (Tsamardinos et al., 2006) as a performance measure for learning the LVM structure. SHD is a global structural measure that accounts for all the possible learning errors: addition and deletion of an undirected edge, and addition, removal, and reversal of edge orientation.

Figures 3–5 show learning curves for SHD (the lower value is the better one) and increasing sample sizes for LPCC, BPC, and EFA. Figures 3 and 4 show SHD performance in learning G1–G4 with binary variables and ternary variables, respectively, and for two parametric settings, p_j=0.7 and 0.8. Figure 5 shows performance in learning G5 and G6 with binary variables for two parametric settings p_j=0.75 and 0.8 (for p_j=0.7 the algorithms performed poorly and thus their results are excluded here). In addition, in Appendix B, we compare LPCC with BPC (Section B.1) and with EFA (Section B.2) in learning G1–G4 with binary and ternary variables for three parametric settings, p_j=0.7, 0.75, and 0.8. The graphs demonstrate the LPCC sensitivity to the parametric complexity – the

[5] For example, a common practice in EFA is that a correlation (loading) of at least 0.4 is needed in order to add a link between a latent variable and an observed variable.

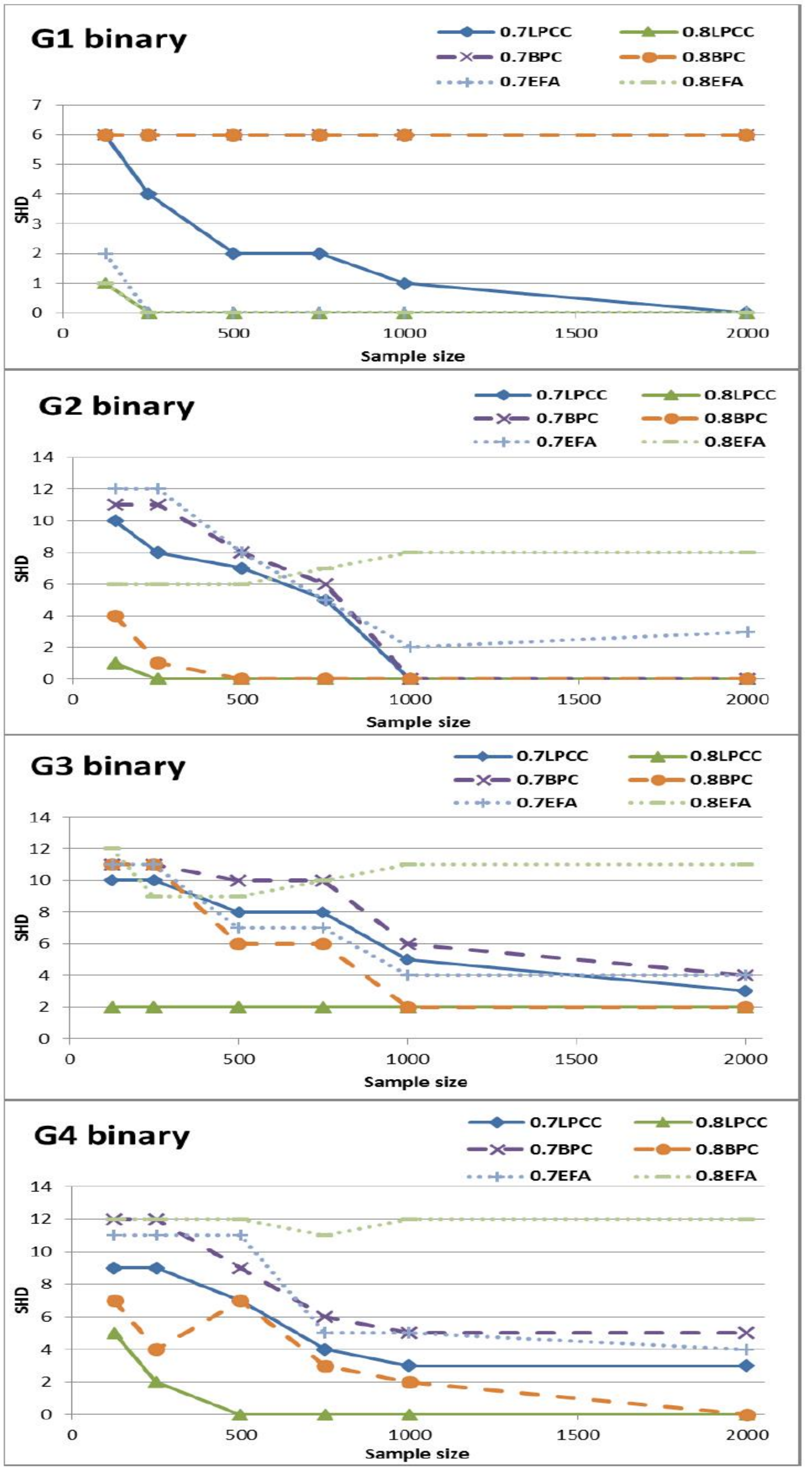

Figure 3: SHD learning curves of LPCC compared to those of BPC and EFA for G1–G4 of Figure 2 with binary variables, two parameterization levels, and increasing sample sizes. The lines of LPCC and EFA for a parametrization of 0.8 coincide for G1.

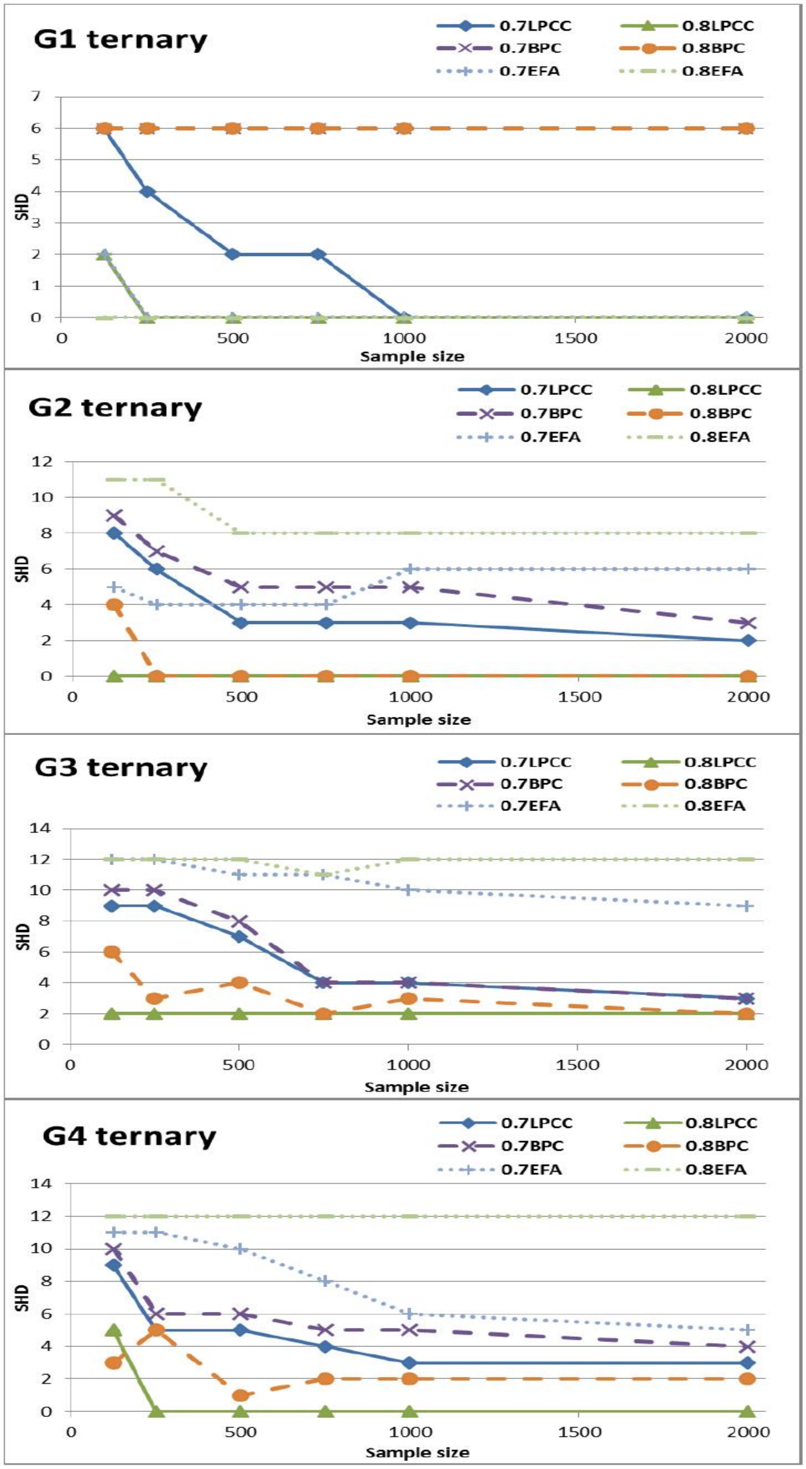

Figure 4: SHD learning curves of LPCC compared with those of BPC and EFA for G1–G4 of Figure 2 with ternary variables, two parameterization levels, and increasing sample sizes. The line of LPCC for a parametrization of 0.8 coincides with that of EFA for a parametrization of 0.7 for G1.

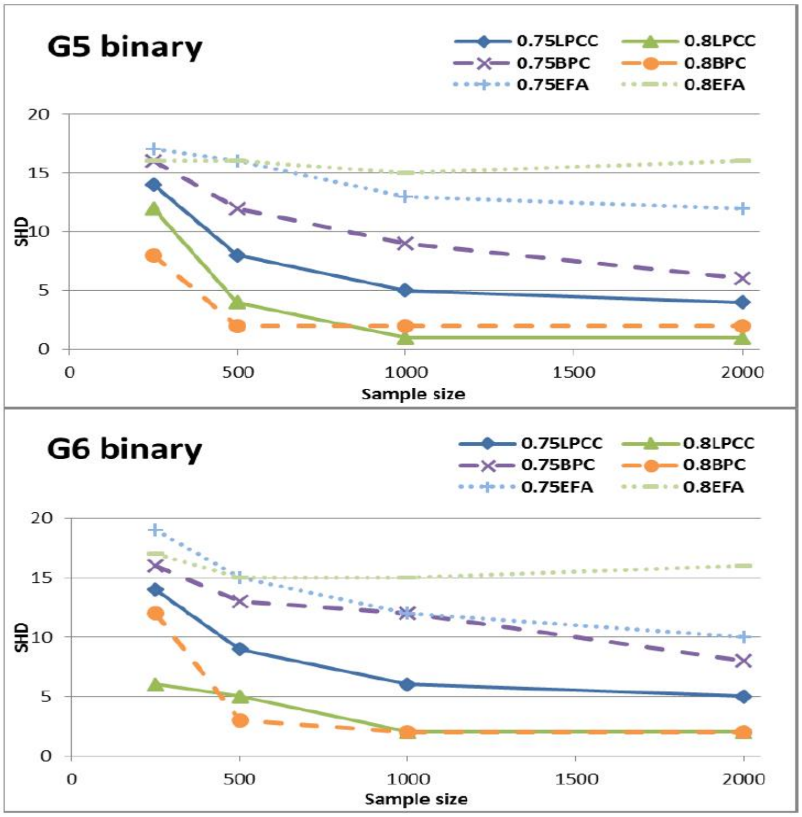

Figure 5: SHD learning curves of LPCC compared to those of BPC and EFA for G5 and G6 of Figure 2 with binary variables, two parameterization levels, and increasing sample sizes.

lower the complexity is, the faster learning is and the sooner the error vanishes – and the LPCC good asymptotic behavior, demonstrating accuracy improvement with the sample size. Generally, Figures 3–5 (and those in Appendix B) show superiority of LPCC over BPC and EFA; LPCC demonstrates higher accuracies (smaller errors) and a better asymptotic behavior than BPC and EFA.

Specifically in regard to EFA, the algorithm – contrary to what is expected from a learning algorithm (see LPCC and BPC) – fails as the experiment conditions improve and the learning task becomes easier (e.g., larger parametrization levels and/or data samples as in the graphs for G4 with binary variables and G2 with ternary variables). Larger parametrization levels increase the chances of EFA to learn links between latent variables and observed variables – some of them are not between a latent and its real child – to compensate for the algorithm's inability to identify links among latents (as EFA assumes latents are uncorrelated). The increase in the sample size helps increase the confidence of EFA in learning these erroneous links (see the graphs for G2, G5, and G6 with binary variables). As Figures 3–5, together with the more detailed Figures 19 and 20 in Appendix B, demonstrate, EFA is inferior to LPCC for all parametrization levels and sample sizes and all graphs but G1. Independent latent variables, as manifested in G1, is the ultimate prerequisite for a successful application of EFA, and indeed, EFA shows competitive (and sometimes, for small sample sizes, even slightly improved) performance to LPCC in learning G1.

Unlike LPCC, BPC is not suitable for learning models such as G1, where the latents are independent and each has fewer than four observed children. This is because BPC requires the variables in a Tetrad constraint to all be 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 (SHD=6 for missing all the six edges in G1). However, it is reasonable to assume that a practitioner would naturally analyze the data before trying BPC, and if they recognize that not all observed variables are correlated (e.g., X1 and X4 for G1), then they will not use BPC. As Figures 3–5, together with the more detailed Figures 17 and 18 in Appendix B, demonstrate, for most graphs, parametrization levels, and sample sizes (except for some cases with small sample sizes), LPCC is superior to BPC.

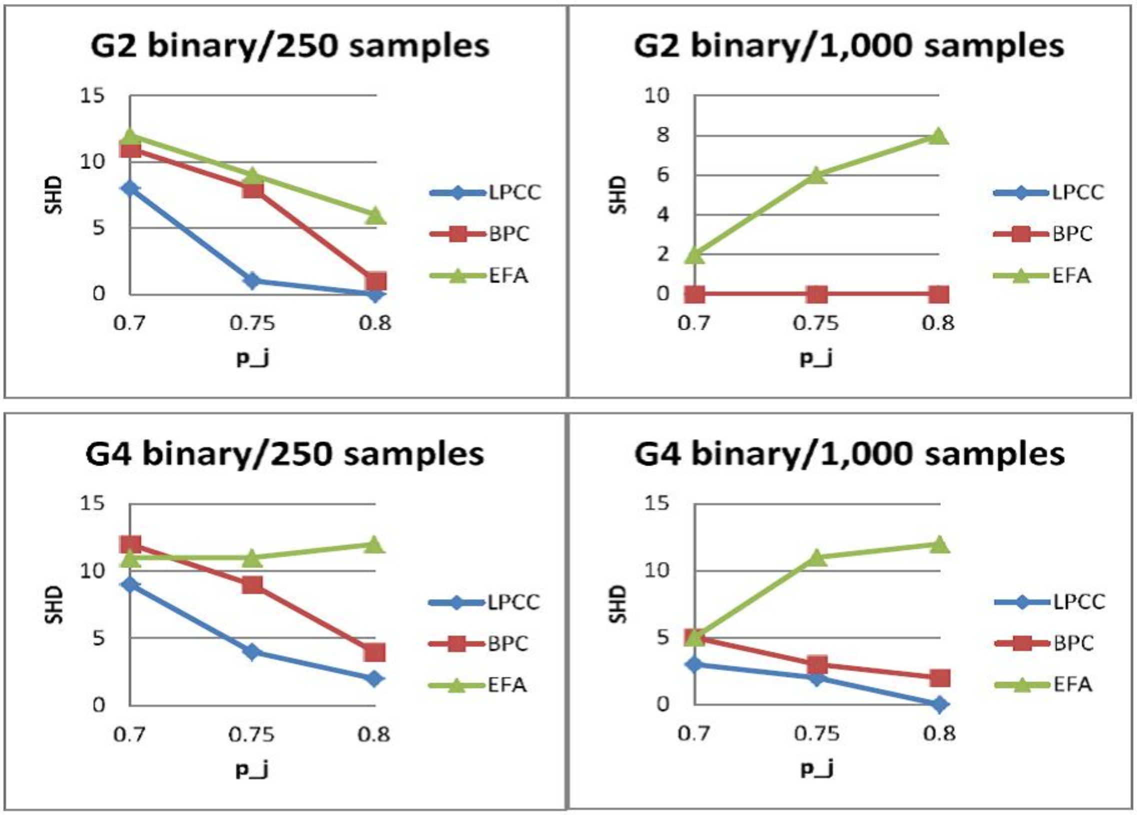

Figure 6: SHD of the LPCC, BPC, and EFA algorithms for increasing parametrization levels for four combinations of learned graphs (G2 and G4) and sample sizes (250 and 1,000 samples). Note that for G2/1,000 samples, both LPCC and BPC learn the structure perfectly for any parametrization level.

Another view of these results is manifested in Figure 6 that shows SHD values for the LPCC, BPC, and EFA algorithms for increasing parametrization levels for four combinations of learned graphs and sample sizes. Figure 6 shows that both 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, for the reasons provided above.

Finally, we expand the evaluation by examining the algorithms when the number of indicators a latent has increases. Figure 7 shows the SHD values of the LPCC, BPC, and EFA algorithms for increasing numbers of binary indicators per latent variable in G2, a parametrization level (p_j) of 0.75, and four sample sizes. The figure exhibits clear superiority of LPCC over BPC and EFA for almost all numbers of indicators and sample sizes. While LPCC hardly worsens its performance with the increase of complexity (number of indicators a latent has), both BPC and EFA are affected by this increase. Also worth mentioning is the difficulty these two latter algorithms have in learning an LVM for which latent variables have exactly two indicators, regardless of the sample size.

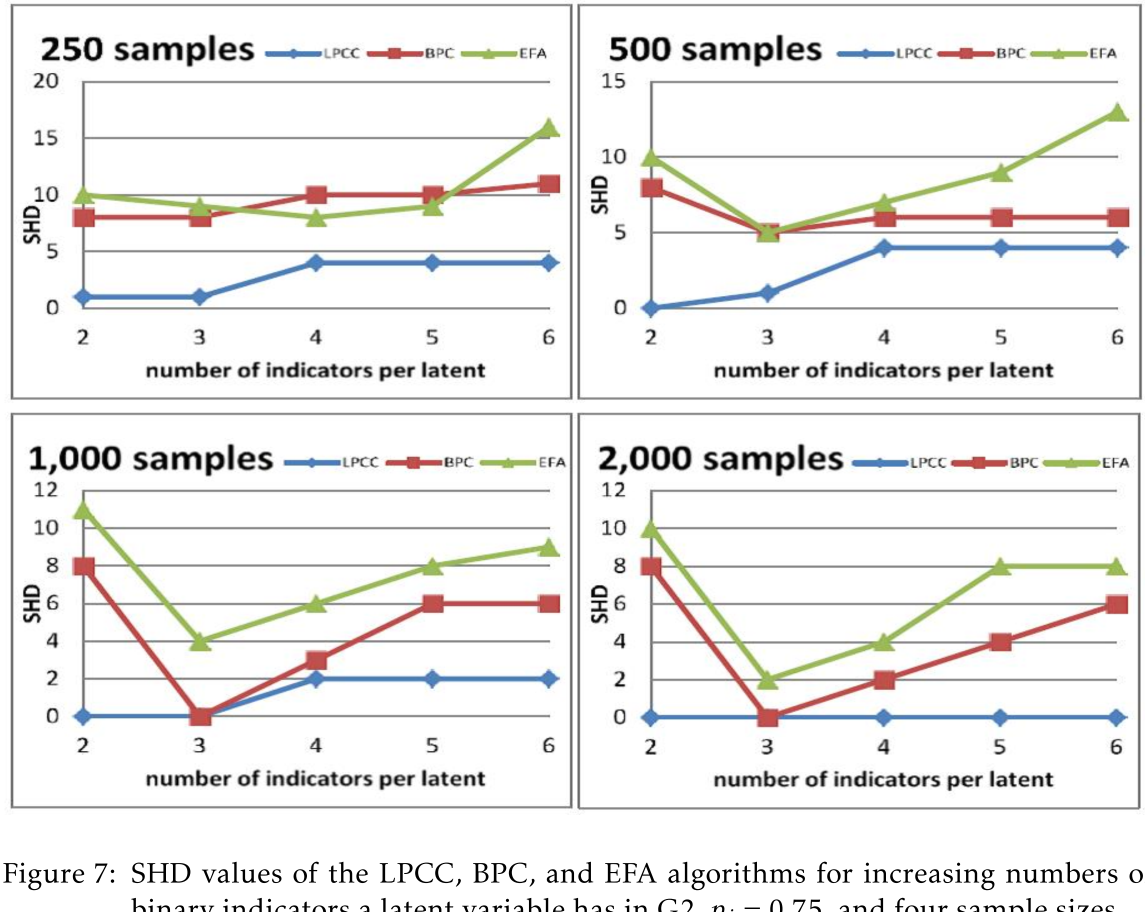

Figure 7: SHD values of the LPCC, BPC, and EFA algorithms for increasing numbers of binary indicators a latent variable has in G2, $p_j = 0.75$, and four sample sizes.

To understand the differences among the three algorithms in more detail, we analyze the errors they make, for example, when using 1,000 samples (the reference is the bottom-left graph in Figure 7). When the number of indicators a latent has is less than 4, LPCC learns the LVM perfectly, and when this number is greater, LPCC errs twice in missing an edge from a latent to one of its indicators. BPC cannot learn an LVM using two indicators per latent, and thus it misses all eight edges in G2 and returns an empty graph. It successfully learns the LVM when each latent has exactly three indicators, but then fails to direct the edges among the latent variables and misses at least a single edge between a latent and an indicator when the latent variables have more than three indicators each. For two indicators per latent, EFA detects only two factors and fails to connect them. It connects one factor to six indicators and the second factor to five indicators, and thereby errs in learning seven extra edges from latent variables to observed variables, missing two edges from the missing latent variable to two observed variables, and missing the two edges among the latent variables, which accounts for eleven errors in total. For three indicators per latent, EFA detects three indicators for two of the latents and five indicators for the other and misses the edges among the latents, which accounts for four errors in total. For four to six indicators per latent, EFA learns more extra edges between the latent and observed variables, together with missing the edges among the latents. This experiment vividly demonstrates the advantage of LPCC over BPC and EFA in that not only does LPCC detect edges between latent and observed variables more accurately, but it also detects latent-latent connections in all scenarios, which is impressive especially when the sample size is small and/or the number of indicators a latent has is large.

3.2 The political action survey data

We evaluated LPCC using a simplified political action survey data set over the following six variables (Joreskog, 2004):

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

These six variables represent the operational definition of 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 is available as part of the LISREL software for latent variable analysis and contains the responses to these questions from a sample of 1,076 United States respondents. A model consisting of two latents that correspond to a previously established theoretical trait of Efficacy and Responsiveness based on Joreskog (2004) is given in Figure 8a. VOTING is discarded by Joreskog for this particular data based on the argument that the question for VOTING is not clearly phrased.

Similar to the theoretical model, LPCC finds two latents (Figure 8b): One corresponds to NOSAY and VOTING and the other corresponds to NOCARE, TOUCH, and INTEREST (a detailed description of the PCC analysis that led to these results is in Appendix C). 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.

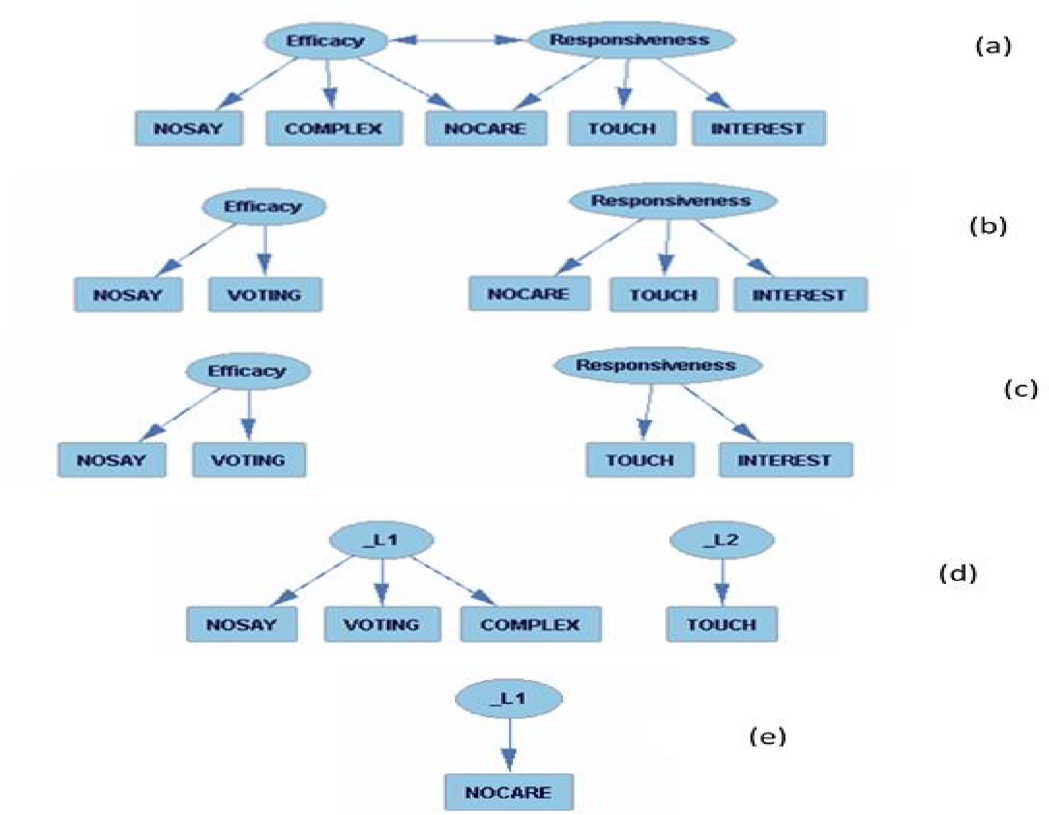

Figure 8: The political action survey: (a) A theoretical model (Joreskog, 2004) and five outputs of (b) LPCC, (c) BSPC, (d) BPC for alpha=0.01 and 0.05, and (e) BPC for alpha=0.1.

BSPC output (Figure 8c) is very similar to LPCC output, except for NOCARE, which was not identified by BSPC as a measure of Responsiveness, making the output obtained by LPCC closer to the theoretical model than that of BSPC. In addition, both algorithms identify VOTING as a child of Efficacy (at the expense of COMPLEX), and thereby challenge the decision made in Joreskog (2004) to discard VOTING from the model. The outputs of the BPC algorithm (Figure 8d) 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 NOSAY, VOTING, and COMPLEX with partial resemblance to the theoretical model (identifying NOSAY and COMPLEX as indicators of this latent) and partial resemblance to the outputs of LPCC and BPC (identifying NOSAY and VOTING as indicators of the latent). However, the second latent found by BPC corresponds only to TOUCH and misses INTEREST (identified in the theoretical model and by LPCC and BSPC as an indicator of Responsiveness) and NOCARE (that is identified in the theoretical model and by LPCC as an indicator of Responsiveness). The output of the BPC algorithm using alpha=0.1 (Figure 8e) gives very little information about the problem as it finds only one latent that corresponds only to NOCARE. These last two figures show the sensitivity of BPC to the significance level, which is a parameter whose value should be determined beforehand. Note that the success of the LPCC and BSPC algorithms emphasizes the importance of such algorithms in learning discrete problems.

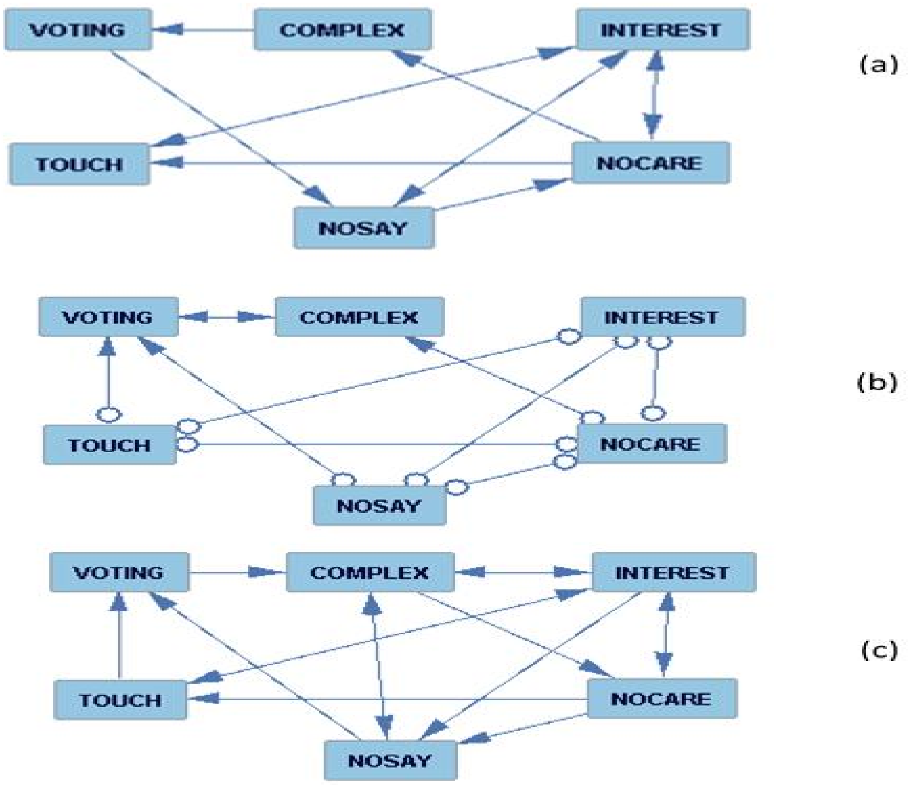

Figure 9: FCI outputs for the political action survey data set and significance levels of (a) 0.01, (b) 0.05, and (c) 0.1.

The outputs of the FCI algorithm using any of the above significance levels are not sufficient (Figure 9). For example, the FCI outputs that were learned using alpha=0.01 (Figure 9a) and 0.05 (Figure 9b) show that NOSAY and INTEREST potentially have a latent common cause. However, these two variables are indicators of different latents in the theoretical model. These results are understandable because unlike LPCC, BPC, and BSPC, FCI is not suitable for learning MIM models such as the political action survey.

3.3 Holzinger and Swineford's data

Holzinger and Swineford (1939) collected data from 26 psychological tests administered to 145 seventh- and eighth-grade children in the Grant-White School in Chicago, Illinois. In this evaluation, we use a subset of this data over only six variables representing the scores in six intelligence tests. The variables are: scores on a visual perception test (VisPerc), scores on a cube test (Cubes), scores on a lozenge test (Lozenges), scores on a paragraph comprehension test (ParComp), scores on a sentence completion test (SenComp), and scores on a word meaning test (WordMean). There are two hypothesized intelligence factors, which are spatial ability and verbal ability factors. The first three variables measure spatial ability and the latter three variables measure verbal ability. A confirmatory factor model that fits this data well was extracted from the Amos manual (Arbuckle 1997, p. 375; Joreskog and Sorbom 1989, p. 247) and is shown in Figure 10a.

We ran LPCC using a dichotomous (binary) presentation of the continuous data. For each variable, scores that were above the average score were recoded as 2, and scores below the average score were recoded as 1. Despite the small size of the data set and the loss of information due to the discretization process, LPCC found two latents (Figure 10b). The first latent corresponds to VisPerc and Lozenges, and the other latent corresponds to ParComp, SenComp, and WordMean (a detailed description of the PCC analysis is in Appendix C). Our model matches the theoretical model, except for missing one link between Spatial and Cubes (and the link between the latents that the model is not supposed to identify).

The outputs of the BPC algorithm using alpha=0.01 and 0.05 (Figure 10c) were not good compared to the theoretical model and LPCC output. In both cases, BPC found only a single latent variable that corresponds to only four of the six indicators, specifically, VisPerc, Lozenges, Cubes, and WordMean. Notice that WordMean and the other three variables belong to two different latent variables in the theoretical model. However, for a significance level of 0.1, BPC output (Figure 10d) is the closest of all models to the theoretical model. These results show the sensitivity of BPC to the significance level, which is a parameter that does not have a predetermined value. LPCC does not have this disadvantage. Note that the superiority of BPC for a significance level of 0.1 for Holzinger and Swineford's data set is in contrast to the model inferiority for other significance levels (Figure 10c) and for the political action survey data set with any significance level.

The output of the FCI algorithm using a significance level of 0.01 or 0.05 (Figure 11a) indicates that ParComp, SenComp, and WordMean potentially have a latent common cause. In addition, Lozenges potentially has a latent common cause with VisPerc and with Cubes, but there is no link between VisPerc and Cubes. For alpha of 0.1, the output of the FCI algorithm (Figure 11b) indicates that ParComp, SenComp, and WordMean potentially

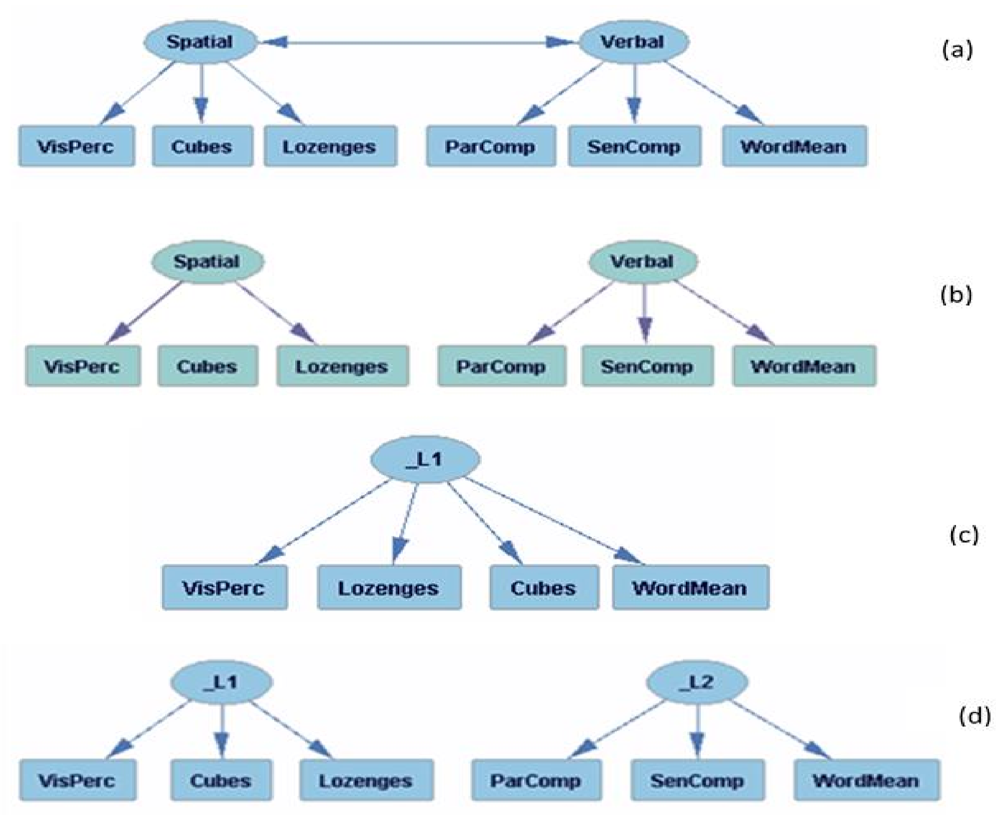

Figure 10: (a) A theoretical model for Holzinger and Swineford's data set based on a confirmatory factor model that fits this data well and the outputs of (b) LPCC, (c) BPC for alpha=0.01 and 0.05, and (d) BPC for alpha=0.1.

have a latent common cause, and Lozenges, VisPerc, and Cubes potentially have another latent common cause. This model matches the theoretical model (Figure 11a) except for the bidirectional edge between the latents.

3.4 The HIV test data

We also evaluated LPCC using the HIV test data (Zhang, 2004). This data set consists of results for 428 subjects of four diagnostic tests for the human immunodeficiency virus (HIV): "radioimmunoassay of antigen ag121" (A); "radioimmunoassay of HIV p24" (B); "radioimmunoassay of HIV gp120" (C); and "enzyme-linked immunosorbent assay" (D). A negative result is represented by 0 and a positive result by 1. LPCC learned a model identical to that in Zhang (2004) (Figure 12), where X1 and X2 are both binary latent variables. However, unlike the algorithm in Zhang (2004) that aims at learning tree-latent models like the one required for the HIV data, LPCC is not limited to latent-tree models. BPC returned an empty model for any conventional alpha.

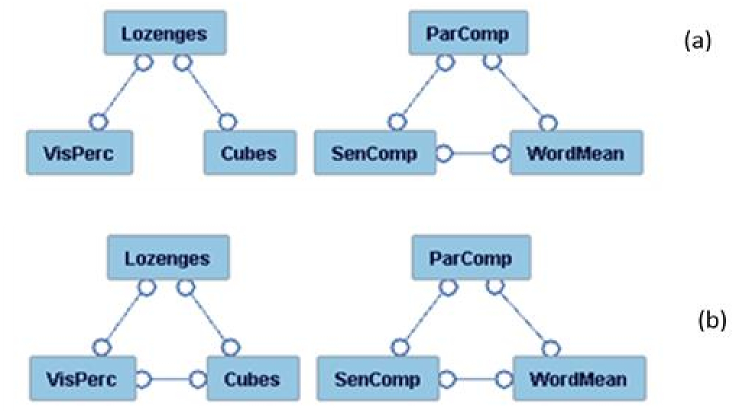

Figure 11: FCI outputs for Holzinger and Swineford's data set and significance levels of (a) 0.01 and 0.05, and (b) 0.1.

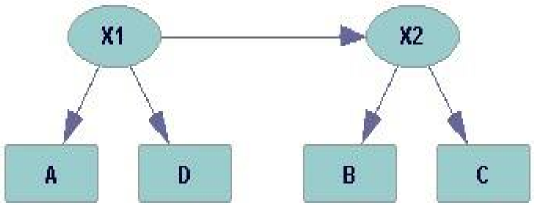

Figure 12: Model learned for HIV using LPCC.

3.5 Explanation of young drivers' involvement in road accidents using LPCC

We produced two databases (DB1 and DB2) from a main database that includes all young drivers (between the ages of 18 and 24 years) who received their private-car driving licenses in Israel between 2002 and 2008. The main database includes more than 600,000 drivers and their parents who were involved in more than 600,000 road accidents and committed more than 2,000,000 traffic offenses in this period. We were interested in explaining young driver (YD) involvement in road accidents and offenses using LPCC and the databases. By "explaining", we mean that we wanted to find the factors among all variables representing a driver, car, accident, offenses, and so forth and the interrelations that could explain YD involvement in road accidents and offenses. These factors could also contribute to prediction of YD involvement in road accidents and offenses with the highest accuracy. We concentrated on the first three months after the accompanied driving phase (ADP), which is a three-month driving phase in which a YD is accompanied by an experienced driver. We concentrated on the three months after ADP because: (1) this is the first solo experience of YDs, and it is when they commit most of their traffic offenses or are involved in most of their road accidents; and (2) we only had detailed monthly records of traffic offenses for the first six months after obtaining a driving license. YDs in the two databases were grouped according to the following classes:

DB1:
Accident and offense: All YDs who had at least one accident and committed at least one offense in the three-month period after ADP (the "period"). There were 345 such drivers; hence, this number defined a group size.
Offense but no accident: 345 drivers who committed at least one offense, but had no accidents in the period.
Accident but no offense: 345 drivers who had at least one accident, but committed no offense in the period.
No accident and no offence: 345 drivers who did not have any accidents or commit any offenses in the period.
In total, there are 1,380 observations (YDs) for DB1.

DB2:
Accident and offense: All YDs who had at least one accident and committed at least one offense in the period (similar to this class in DB1).
No accident and no offense: 345 drivers who did not have any accidents or commit any offenses in the period (similar to this class in DB1).
In total, there are 690 observations (YDs) for DB2.

All observations in both databases are represented by thirteen observed variables that a previous study indicated as relevant to the explanation of YD involvement in road accidents and offenses (Lerner, 2012; Lerner and Meyer, 2012). In addition, we used four observed variables that indicate if YDs or their parents were involved in a road accident or an offense. A detailed description of all seventeen observed variables is given in Table 1.

We ran LPCC on DB1 and DB2 and compared its results to those of EFA and BPC. We ran BPC using a significance level (alpha) of 0.05 (Tetrad's default). Exploratory factor analysis was applied in two phases. First, principal component analysis (PCA) was used for factor extraction, where the Kaiser criterion (Kaiser, 1960) was used for determining the number of factors. Any factor with an associated eigenvalue less than 1.0 was dropped because this value is equal to the information accounted for by an average single observed variable. Second, the factor model was rotated using varimax, which is an orthogonal rotation method of the factor axes to maximize the variance of the squared loadings of a factor on all the variables. Factor loadings, also called component loadings in PCA, are the correlation coefficients between the variables and factors, indicating how strongly the latter influence the former. Analogous to Pearson's correlation coefficient, the squared factor loading is the percent of variance in that indicator variable explained by the factor. The varimax rotation method has the effect of differentiating the original variables by the extracted factors. Each factor tends to have either large or small loadings of any particular variable. A varimax solution yields results that make it as easy as possible to identify each variable with a single factor (with the highest loading on the variable). In confirmatory factor analysis (CFA), loadings should be 0.7 or above to confirm that independent variables identified a priori are represented by a particular factor, using the rationale that the

0.7 level corresponds to about half of the variance in the indicator being explained by the factor. However, the 0.7 standard is high, and real-world data may not meet this criterion, which is why some researchers, including us in this study (particularly for exploratory purposes such as this case), use a lower level, where 0.4 is the common practice (Manly, 1994). In addition, we adapted Occam's razor parsimony principle (to explain the variance with the fewest possible factors) and required the variance explained criterion to be above 50%.

Number	Variable	Variable short name	Variable values
1	Age	gil	1 (17–18), 2 (19–20), 3 (21–22), 4 (23–24)
2	Gender	Min	1 (male), 2 (female)
3	Medical limitations	lim	1 (no), 2 (yes)
4	Father is allowed to drive	MurF	1 (yes), 2 (no)
5	Mother is allowed to drive	MurM	1 (yes), 2 (no)
6	Has a motorcycle license	of	1 (no), 2 (yes)
7	Received "Or Yarok" kit, as part of a graduated driver licensing program [6]	or	1 (didn't receive), 2 (received)
8	Socioeconomic index	GF	1–4 (1–low, 4–high)
9	Ethnic group	KU	1 (Jew), 2 (non-Jew)
10	Father's marital status	FS	1 (single), 2 (married), 3 (divorced), 4 (widowed)
11	Mother's marital status	MS	1 (single), 2 (married), 3 (divorced), 4 (widowed)
12	Father's number of years of education	FED	1–4 (1–low, 4–high)
13	Mother's number of years of education	MED	1–4 (1–low, 4–high)
14	Offenses of YD	OFYD	1 (no), 2 (yes)
15	Accidents of YD	ACYD	1 (no), 2 (yes)
16	Offenses of parents	OFPA	1 (no), 2 (yes)
17	Accidents of parents	ACPA	1 (no), 2 (yes)

Table 1: Seventeen observed variables in DB1 and DB2

Results for DB1:

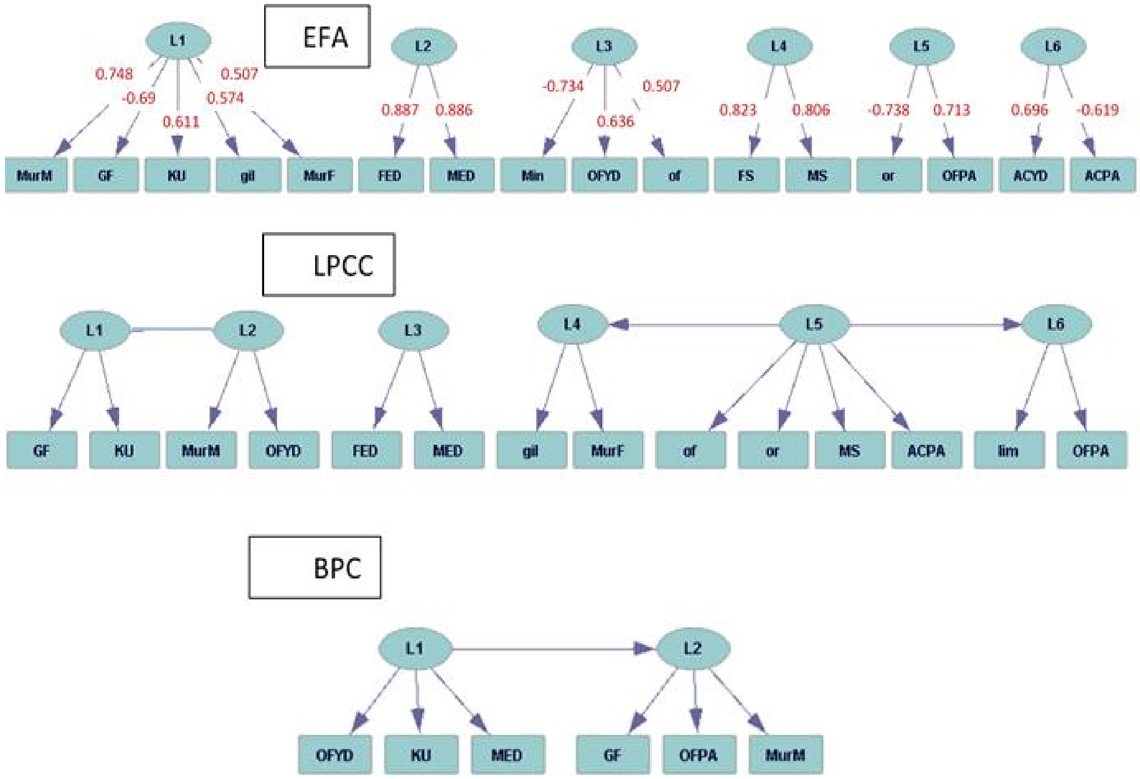

Figure 13: LVMs learned by EFA, LPCC, and BPC for DB1. Numbers for EFA represent the factor loadings, where plus and minus signs indicate positive and negative correlation, respectively.

Both LPCC and EFA found six latent variables (Figure 13). L1 in EFA is a parent of five observed variables, where each describes another aspect of the demographic or socioeconomic state of the YD or his/her family. The variables (see Table 1) *father is allowed to drive* (MurF), *mother is allowed to drive* (MurM), *age* (gil), and *ethnic group* (KU) are positively associated, whereas the variable *socioeconomic index* (GF) is negatively associated with L1. Based on the values of these variables, we can identify groups in the YDs' population, such as a group of YDs that are not Jewish, their parents are not allowed to drive, and their age and their socioeconomic status are low. L1 is not connected by EFA to other latent variables that relate to YD involvement in road accidents or offenses; thus, it does not contribute much to this study. However, L1 and L2 as learned by LPCC relate the socioeconomic state of the YD family (GF and KU are children of L1) with YD offenses (OFYD

[6] The Or Yarok (i.e., "green light" in Hebrew) kit includes documentation and accessories, such as CDs, with instructions, movies, and advice regarding safe driving. Granting the kit before licensure was found (Lerner, 2012) helpful in reducing young drivers' involvement in road accidents.

is a child of L2), thus LPCC links the socioeconomic state of YD with its involvement in traffic offenses. For example, according to LPCC, a YD who is not Jewish, with a low socioeconomic index, and whose mother is not allowed to drive is more likely to be involved in traffic offenses.

Latents L2 in EFA and L3 in LPCC describe the educational status of the YDs' parents and indicate a similar tendency between the parents' educational levels. However, both EFA and LPCC analyses do not link the educational status of the YD family to involvement in road accidents and traffic offenses.

L3 in EFA shows a negative relationship between a YD's gender (Min) or having a motorcycle license (of) and his/her tendency to commit road offenses. Male drivers tend to commit more traffic offenses than female drivers, and drivers who have a motorcycle license tend to commit more traffic offenses than drivers who do not have such a license. L4 in EFA shows the marital status of the YD parents, and not surprisingly, it indicates that the father's and mother's marital status is correlated. L5 in EFA shows that the parents of YDs who did not receive the "Or Yarok kit" tend to commit more traffic offenses. That is, the introduction of the kit in the family seems to also reduce involvement in road offenses of family members other than the YD. L6 in EFA shows a negative relationship between YD involvement in road accidents and the involvement of their parents in accidents. One explanation for this negative relationship could be that in a family in which one member was involved in an accident, other members tend to be more careful and thus decrease their involvement in accidents. Due to the independency assumption (between the factors) in EFA, L3, L4, L5, and L6 are not related, and EFA is not able to holistically represent relations among variables representing demographic and socioeconomic characteristics and road accidents and offenses of YDs and their parents.

LPCC describes involvements in road accidents and offenses in a more comprehensive way using a structure that is based on three latents with a diverging link from L5 to L4 and L6. L5 shows a relationship among the variables *motorcycle license*, *received "Or Yarok kit"*, *mother's marital status*, and *parents' involvement in accidents*. For example, it was found that there is a relation between receiving the "Or Yarok kit" and decreasing values of parents' involvement in accidents. However, we note that the variable parents' involvement in accidents is sparse in DB1, making its relationship with the other variables via L5 quite arguable. An interesting relationship found in the LPCC results is between parents' accidents (child of L5) and parents' offenses (child of L6). This relationship was missed by EFA since each variable in EFA is a child of a different latent that is independent of the other latent (due to EFA's orthogonality assumption).

BPC finds only two latents compared to six latents that are found by LPCC and EFA. A possible explanation for the low number of latents identified by BPC is that BPC requires that a latent have at least three observed children to be identified, whereas LPCC requires only two (see latents L1–L4 and L6 in the LPCC model, each having two observed children). L1 and L2 as learned by BPC relate the demographic-socioeconomic state of the YD family (KU and GF) with YD offenses, as the LPCC model did, but the two latents in the BPC model mix the indicators. It is more reasonable to believe that if L1 is a parent of L2 as BPC identified, then offenses of YDs (OFYD) should be a child of L2 and not of L1, and GF should be a child of L1 and not of L2 since socioeconomic status is expected to affect violent road behavior and not the opposite. BPC also relates OFYD with offenses of their parents (OFPA), a relation that seems reasonable and is not identified by EFA and LPCC. However, the identification of OFYD as L1's child and OFPA as L2's child implies that it is the YD offenses that affect the offenses of their parents and not the opposite, as may be expected. Another relation that is identified only by BPC is between the mother's education level (MED) and both YDs' and their parents' offenses.

Results for DB2:

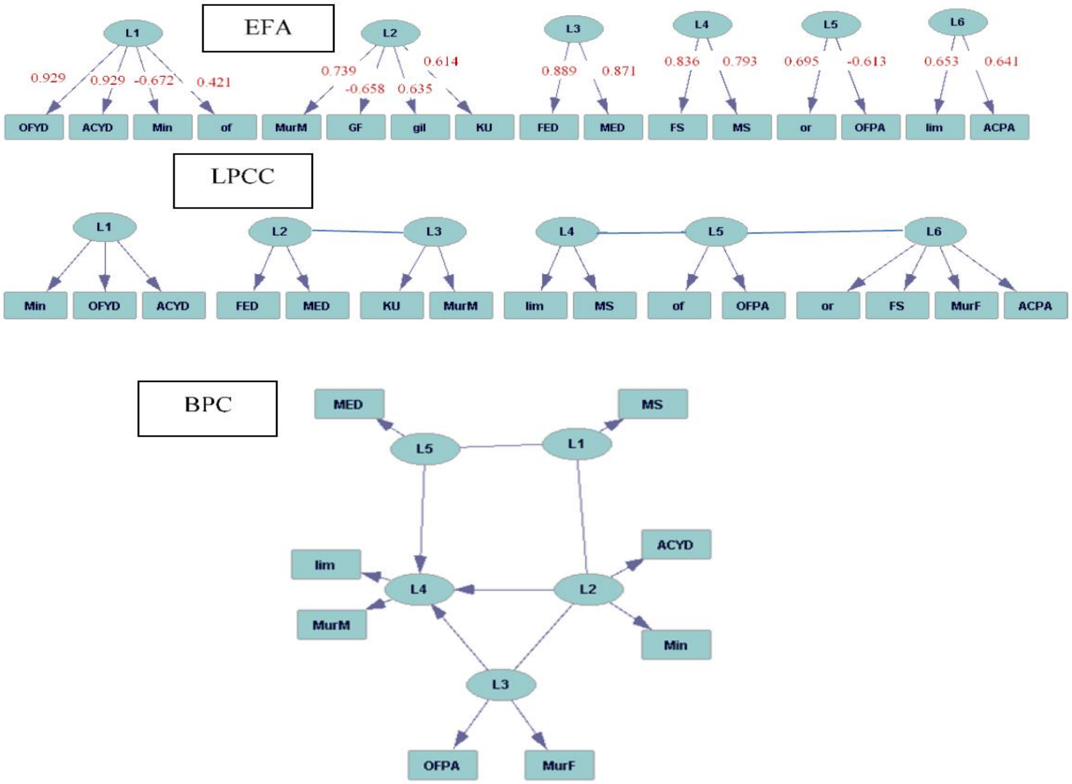

Figure 14: LVMs learned by EFA, LPCC, and BPC for DB2. Numbers for EFA represent the factor loadings, where plus and minus signs indicate positive and negative correlations, respectively.

Again, both the EFA and LPCC algorithms found six latent variables (Figure 14). L1 in EFA is interesting and very important because it links YD involvement in accidents and offenses with gender and motorcycle license. The loading coefficients show that male YDs, especially those who have a motorcycle license, are involved more in both accidents and offenses (in both cases with loadings of 0.929). L1 in LPCC shows a similar relation between gender and a YD's involvement in accidents and offenses, but without a relation to having a motorcycle license. Also in the EFA results, this last relation is quite weak, with a boundary loading value of 0.421, whereas the threshold for connecting an observed variable to a latent factor is 0.4 (which is a common practice in EFA). The relation found between YDs' *accidents* and *offenses* variables is due to the linkage formed in the database by creating it from observations of YDs having both accidents and offenses or neither. Yet, both EFA and LPCC did not manage to find a relation between YDs' accidents and offenses and their parents' accidents and offenses.

L2 in EFA describes the socioeconomic status of the YD family, similar to L1 in the EFA results for DB1 (but without the *father is allowed to drive* variable). Again, it shows groups in the YD population, such as a group of YDs who are not Jews, whose *mothers are not allowed to drive*, and whose *age* and *socioeconomic* status are low.

L2 and L3 in LPCC also represent the social status of the YD family. L2 explains the family educational level (as L3 in LPCC for DB1), and L3 is a demographic latent that is a parent of the *ethnic group* and the *mother is allowed to drive* variables. Unlike L2 in EFA, which also represents economic status via the variable *socioeconomic index* (GF), L3 in LPCC is only a social variable. Furthermore, L3 in LPCC did not link the age variable to the social representation of YD (as L2 did in EFA). It is also interesting to see the relation that LPCC found between latents L2 and L3, which is between the parents' education levels (L2) and the family demographic status (L3). But, LPCC did not find a relation between L2 and L3 and the *accidents* and *offenses* variables of the YDs or their parents.

L4 in EFA shows the marital status of the YD's parents, similar to the EFA results for DB1. L5 in EFA shows that parents of a YD who did not receive the "Or Yarok kit" tend to commit more road offenses, again similar to the EFA results for DB1. L6 shows a relation between the YD's medical limitations and his/her parents' involvement in accidents; this relation is hard to intuitively explain. Similar to DB1, there are no relations between the latents L4–L6 due to the independency assumption of EFA; hence, there is no relation between the latent that represents the parents' marital status and their involvement in accidents or offenses.

LPCC represents relationships between variables belonging to different latents in an interesting way. L6 represents a relationship between father's marital status, whether they are allowed to drive, their involvement in accidents, and whether their YD received the "Or Yarok kit". To some extent, we should be careful about the reliability of this relationship since the variable ACPA (parents' involvement in accidents) is very sparse in DB2 (there are only three observations that indicate fathers' involvement in accidents, and in all of them the YD did not receive the "Or Yarok kit" and the father is divorced or widowed, and not allowed to drive). Similarly, the representation of L6 in EFA is not highly reliable for the same reason. L5 in LPCC represents a relation between a YD who has a motorcycle license and his/her parents' involvement in offenses, where parents of YDs who have a motorcycle license are more likely to be involved in offenses. An additional interesting relation LPCC found between latents L4, L5, and L6, linking between their observed children, is between parents' involvement in offenses and their children having a motorcycle license, and parents' involvement in accidents and their children receiving an "Or Yarok kit". Furthermore, LPCC shows that given that parents are more involved in offenses or their YD has a motorcycle license (both are children of L5), then the medical limitation of YD or the mother's marital status (children of L4) are irrelevant for predicting the parents' involvement in road accidents (a child of L6). That is, knowing L5's children turns L6 and its children (especially parents' involvement in accidents) independent of L4 and its children. The information about the relationships between the latents that LPCC provides illustrates the added value of using such an algorithm in causal analysis.

Also for DB2, it seems that BPC yields poorer results than LPCC. L2 in BPC is partially equivalent to L1 in LPCC (without OFYD), and some resemblance can be seen between L3–L4 in BPC and L4–L6 in LPCC. L2 and L3 in BPC are parents of L4, which together link medical limitations of the YD and whether the mother is allowed to drive with ACYD and OFPA (although we would expect the former two to be causes of the latter two and not the opposite). However, each of the two remaining latents in the BPC model has only a single observed variable as a child, which gives very little information about the problem. L5 in BPC has a single child, MED, without linking it to FED, although these two variables are highly correlated, as in both the EFA and LPCC models. The same can be said about L1 and its child MS that is not linked to FS, although both are correlated. These two are directly connected in the EFA model and indirectly connected in the LPCC model.

3.6 Identification of cellular subpopulations of the immune system from mass cytometry data sets using LPCC

Definition of immune cell subsets is usually based on flow cytometry data. However, this approach suffers from severe limitations in the number of cellular markers that can be measured simultaneously. Currently, flow cytometry permits the concurrent measurement of only 12–17 cellular markers. A recent technological development in single cell measurement is mass cytometry or cytometry by time of flight (CyTOF). This method allows for quantification of hundreds of thousands of single cells at high dimension (currently up to 40 cellular markers can be measured) in a sample. CyTOF yields phenotypically rich datasets that enable a more accurate identification of cellular subpopulations. Clustering and visualization methodologies have been developed to identify meaningful cell subsets in CyTOF data. However, none provide a systematic and automatic method for identifying the cellular subpopulations represented by these clusters.

We evaluated LPCC's ability to automatically identify such cellular subpopulations in an unsupervised manner and in comparison to BPC and EFA (using the same settings as in Section 3.5). We used a CyTOF-generated dataset of mouse splenocyte samples, collected from 20 mice and stained with a panel containing 37 metal-labelled antibodies. We randomly selected 40,000 single cell measurements from each sample to have 800,000 observations. Cellular markers measured by CyTOF have continuous multimodal distributions. Since LPCC works on discrete observed variables, we needed to perform discretization first. To this end, we randomly selected 40,000 observations for each cellular marker and used this sample to learn a mixture of Gaussians, approximating each marker's distribution [with the number of components selected from K=3-10 using the Bayesian Information Criterion (BIC)]. Next, the learned Gaussian components were sorted by their means, and the ordered means determined the K discrete values of the marker. Thus, K also represents the cardinality of the marker after the discretization. Finally, for each marker, each observation in the data set was assigned the closest discrete value.

The task of identifying cellular subpopulations of the immune system is very challenging due to the high level of mutual feedbacks that exist between the different players (cellular subpopulations) of the immune system and the high level of shared cellular markers between these cellular subpopulations. This observation is demonstrated clearly in the results obtained by BPC, EFA, and LPCC in Figures 15a, 15b, and 16a, respectively. LPCC and EFA managed to learn models with eight latents each and BPC a model with one latent, but none of the three latent variable models seems to be biologically meaningful, as judged by biological experts.

However, LPCC has an advantage over the other two algorithms because to advance learning an LVM, LPCC clusters the input data in its first stage. We exploit this advantage by improving clustering of the input to consider the domain specific properties. This act of clustering – that is natural to LPCC – coincides with the conventional clustering-based analysis of CyTOF data, which is mandatory to this domain because the data shows a hierarchical structure (as outlined below). Therefore, instead of using SOM, we initialized LPCC based on the clustering results obtained by Citrus (Bruggner et al., 2014). Citrus applies hierarchical clustering to the cell events; however, instead of cutting the hierarchy at an arbitrary height to identify the clusters, it uses a minimum cluster size threshold (we used a 1% threshold of the observations, i.e., 8,000 observations), for which only clusters larger than this threshold are selected. By selecting automatically and based on the data only large enough clusters, we preserve the requirement of LPCC of not determining the number of clusters arbitrarily and facilitate the avoidance of noisy clusters. In addition, we performed another purification procedure by selecting only clusters for which the ratio of the cluster marker entropy to the distribution marker entropy is smaller than a threshold.

Following this cluster purification procedure, LPCC found a five-latent variable model that is represented in Figure 16b. L1 is a parent of five markers that represent T cells (except for CD45[7] that may be expressed also by other leukocytes, and CD62L that is an activation marker that also can be expressed by T cells). L2 partially represents monocytes by having the markers CD11b and CD86 as its children; however, this representation is not perfect since it wrongly connects CD34, which is a phenotypic marker of stem cells. Thus, L2 may be representing monocytes that are antigen-representing cells excluding CD34. L3 and L4 are linked by a directed edge from L3 into L4 and together represent macrophage cells. Although both latents represent the same population of cells, LPCC correctly splits it into two latent variables since the children of L3 are expressed only by macrophages, but the children of L4 may also be expressed by monocytes, which are the macrophages' precursor. L5 is a parent of three markers that represent B cells together with another marker, IA-IE, which is an activation marker that can also be expressed by B cells. Still, the model learned by LPCC represents only sixteen of the 37 markers in this experiment. This may be explained by the high level of overlap and number of shared markers among the different subpopulations. Despite that, these results are encouraging and demonstrate a significant improvement compared to previous results obtained by BPC, EFA, and LPCC before cluster purification.

Analysis of cell sub-populations in the immune system naturally lends itself to the use of clustering methodologies, as immunologists traditionally resort to the classification of cells as belonging to cellular subpopulations. Cell subpopulations are usually defined by the stable expression of markers on said cells. Yet, not all markers capture protein expression that is stably expressed, and the expression of some proteins may be noisy or plastic,

[7] The cellular markers we use are well known in immunology (Janeway et al., 2001), and thus their description is avoided here for clarity of the demonstration.

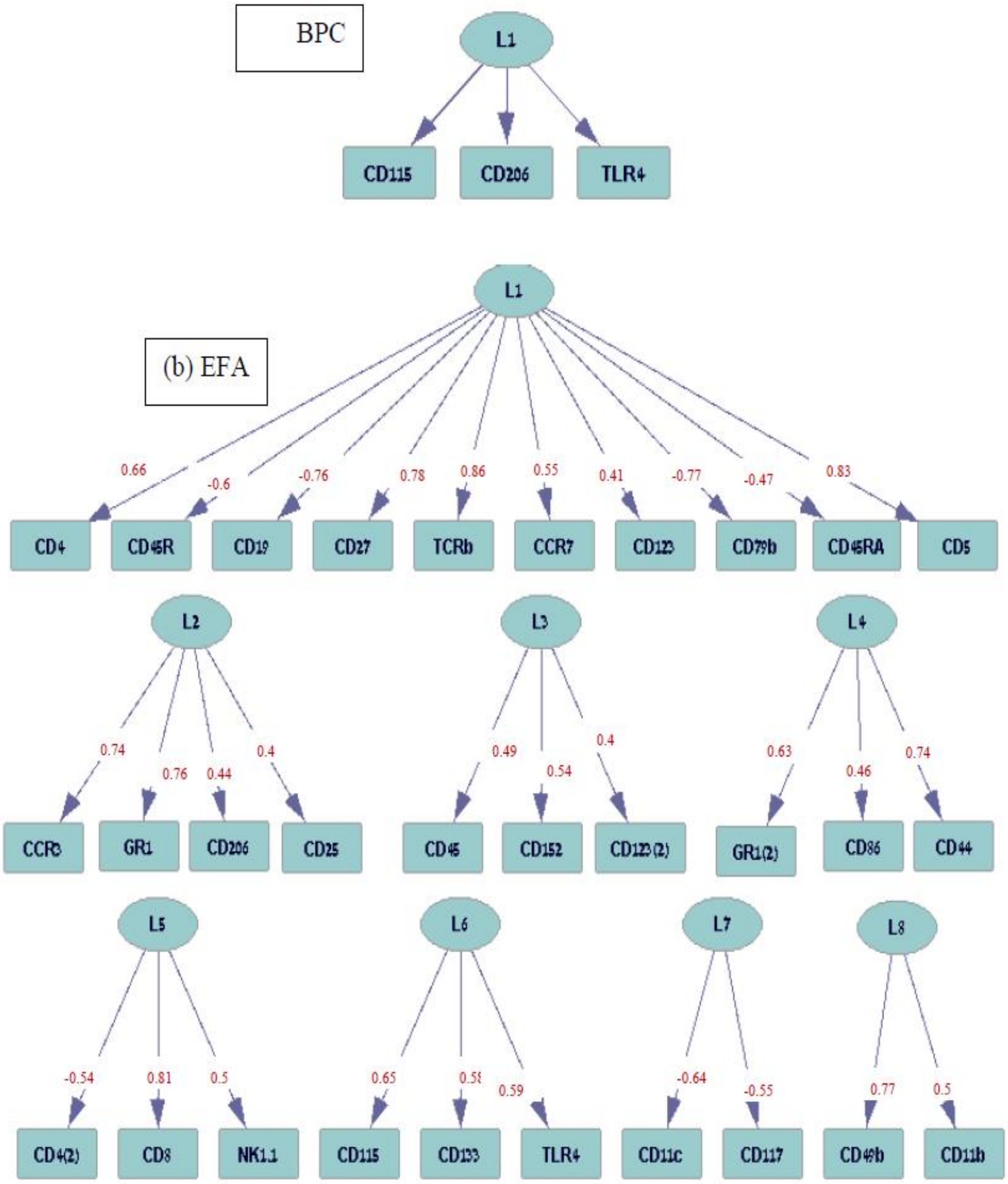

Figure 15: LVMs learned by (a) BPC and (b) EFA. Numbers for EFA represent the factor loadings, where plus and minus signs indicate positive and negative correlation, respectively.

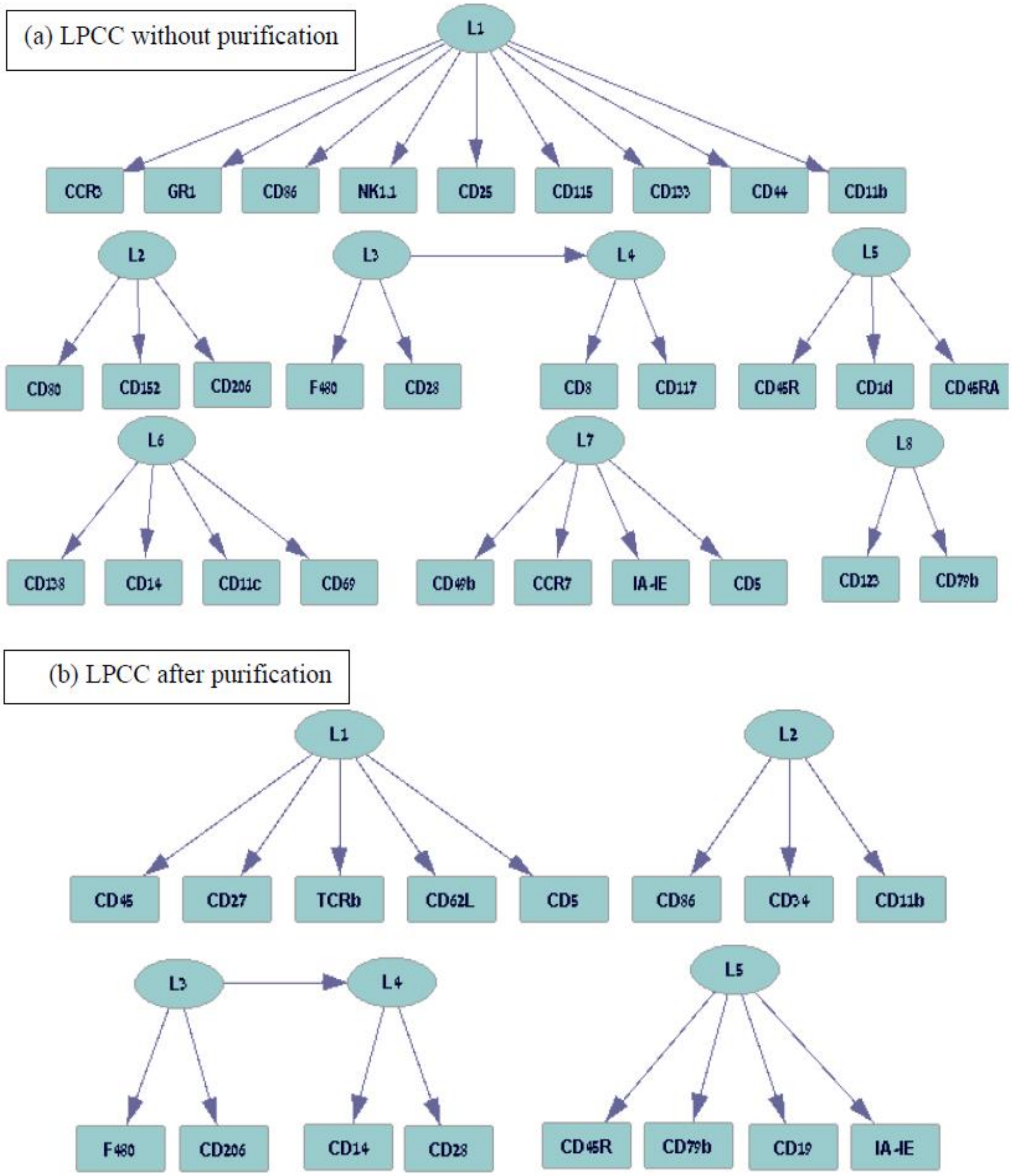

Figure 16: LVMs learned by LPCC (a) before and (b) after a purification procedure.

varying over time and conditions. With this in mind, clustering and noise filtration – two pre-processing steps of LPCC – provide great benefit in this context, yielding improved results. Specifically, the pre-processing step of clustering the data prior to LPCC provides an advantage as it compartmentalizes marker relationships to the context in which they matter, whereas the noise filtering step focuses the analysis to those markers whose relationship with one another may be meaningful.

4. Related Works

The traditional framework for discovering latent variables is factor analysis and its variants (e.g., see Bartholomew et al., 2002). This is, by far, the most common method used in several applied sciences (Glymour, 2002). However, a limitation of factor analysis is its level of subjectivity stemming from the many methodological decisions a researcher must make to complete an analysis, where the results of this analysis largely depend on the quality of these decisions (Henson and Roberts, 2006). Moreover, factor analysis and its variants provide only a limited ability in causal explanation (see Silva, 2005, and our evaluation section). Therefore, in this section, we will focus on related work in the framework of learning causal graphical models beyond the variants of factor analysis.

The main goal of heuristic methods such as those of Elidan et al. (2000) is the reduction of the number of parameters in a BN. The idea is to reduce the variance of the resulting density estimator, achieving better probabilistic predictions. For probabilistic modeling, the results described by Elidan et al. are a convincing demonstration of the suitability of their approach, which is intuitively sound. However, such heuristic methods provide neither formal interpretation of what the resulting structure is, nor explicit assumptions on how such latents should interact with the observed variables. Further, such heuristic methods do not provide an analysis of possible equivalence classes, and consequently, there is no search algorithm that can account for equivalence classes. Therefore, for a causality discovery under the assumption that multiple observed variables have hidden common causes, such as in MIM that is widely used in applied sciences, the results described by Elidan et al. are unsatisfying.

Unlike other algorithms (Pearl, 2000; Zhang, 2004; Harmeling and Williams, 2011; Wang et al., 2008), LPCC is suitable for learning MIM models and not just latent-tree models. This LPCC quality is shared by BPC (Silva et al., 2006). Both LPCC and BIN-A (Harmeling and Williams, 2011) apply clustering as a preprocessing step to learn latent models. But, LPCC applies clustering to the data points, whereas BIN-A clusters the variables using agglomerative hierarchical clustering, which is suitable to learn HLC models, as in Zhang (2004). LPCC provides a consistent and substantive analysis of data-point clustering using the PCC concept and can learn all types of links between the latents; thus, unlike BIN-A, it is not limited to binary latent trees.

FCI (Spirtes et al., 2000) is not comparable to LPCC in learning MIM models as illustrated for the political action survey and the Holzinger and Swineford databases (Sections 3.2 and 3.3). Compared to BPC and FCI, LPCC does not rely on statistical tests and presetting of a significance level for learning LVM.

Contrary to BPC, LPCC concentrates on the discrete case and dispenses with the linearity assumption. However, LPCC assumes that the measurement model is pure; still a weaker assumption than the one latent-tree models make. Unlike LPCC, BPC is not suitable for learning models such as G1 in Figure 2, where the latents are independent and each has fewer than four observed children. This is because BPC requires the variables in a Tetrad constraint to all be 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 (Section 3.1). In addition, BPC is not suitable for learning models such as the HIV model (Section 3.4), where each latent has only two indicators and BPC requires three indicators for a latent to be identified. This also explains the poor results of BPC on the YD databases compared to the LPCC results (Section 3.5).

When the attributes are categorical, cluster analysis is sometimes called latent class analysis (LCA) (Lazarsfeld and Henry, 1968; Goodman, 1974; Bartholomew and Knott, 1999), where data are assumed to be generated by a latent class model (LCM). An LCM consists of a class variable (latent) that represents the clusters to be identified and a number of other variables that represent attributes (observed variables) of objects.[8] LCMs assume local independence; in other words, the observed variables are conditionally independent given the latent variable. A serious problem with the use of LCA, known as local dependence, is that the local independence assumption is often violated. To relax this strong assumption, Zhang (2004) proposed a richer, tree-structured latent variable model, specifically, the HLC model. The network structure is a rooted tree, and the leaves of the tree are the observed variables. HLC models were chosen for two reasons. First, the class of HLC models is significantly larger than the class of LCMs and can accommodate local dependence. Second, inference in an HLC model takes time that is linear in the model size (because it is a tree), which makes it computationally feasible to run EM. However, MIM models learned by LPCC are richer than HLC models that are only a subset of MIMs. Thus, LPCC may contribute to clustering analysis of data generated by richer models, while keeping the advantage of accommodating local dependence.

5. Discussion

In Part I, we introduced the PCC concept and LPCC algorithm for learning LVMs. We showed that LPCC: 1) Is not limited to latent-tree models, and does not make a linearity assumption about the distribution; 2) Learns MIMs; 3) Learns a MIM with no assumptions about the number of latent variables and their interrelations (except the assumption that a latent collider does not have any latent descendants; Assumption 5) and which observed variables are the children of which latents; and 4) Learns an equivalence class of the structural model of the true graph.

In Part II, we formally introduced the LPCC two-stage algorithm. First, LPCC learns the exogenous latents and the latent colliders, as well as their observed descendants, by utilizing pairwise comparisons between data clusters in the measurement space that may explain latent causes. Second, LPCC learns the endogenous latent non-colliders and their children by splitting these latents from their previously learned latent ancestors.

Using simulated and real-world data sets, we showed in Part II that LPCC improves accuracy with the sample size, can learn large LVMs, and has consistently good results

[8] This model has the same graphical structure as the naive-Bayes classifier, but because it is trained in an unsupervised manner (clustering), we refer to it as an LCM.

compared to models that are expert-based or learned by state-of-the-art algorithms. Using LPCC to identify possible causes of young drivers' involvement in road accidents, we found interesting relations among latent and observed variables and can provide illuminating insights into this important problem. Using LPCC to identify cell subpopulations in the immune system, we offer an LVM that makes sense to expert biologists in describing this challenging system. A criticism of LPCC may be its reliance on performing preliminary clustering to the data. Changes in the data used for clustering may affect the LPCC output. Yet, our experience shows that even if the clustering results change for different data samples drawn from the distribution, the same major and 1-order minor clusters are usually identified. In addition, as the biological example (Section 3.6) illustrates, when a structure is inherent to the data, clustering of the data first yields high benefit in learning an LVM later and improves results. Structured real-life problems are prevalent in many disciplines (Vazquez et al., 2004); hence, being a clustering-based LVM learning mechanism gives LPCC an advantage more than a disadvantage.

Finally, a number of open problems that invite further research were provided in the discussion of Part I.

Acknowledgments

Special thanks to the *Israel National Road Safety Authority* and *Ran Naor Foundation for the Advancement of Road Safety Research* for supporting the research to explain young drivers' involvement in road accidents (Section 3.5) and to Shai Shen-Orr, Elina Starosvetsky, and Tania Dubovik from the Department of Immunology at the Technion for providing the mass-cytometry data set and helping in the evaluation of the results of its analysis (Section 3.6). The authors also thank the two anonymous reviewers for their comments and suggestions that helped improve the paper and the special issue editors: Isabelle Guyon and Alexander Statnikov. Nuaman Asbeh thanks the *Planning and Budgeting Committee* (PBC) of the *Israel Council for Higher Education* for its support through a scholarship for distinguished Ph.D. students.

Appendix A. Important assumptions, definitions, propositions, and theorems from Part I (numbers are taken from Part I)

Assumption 5 *A latent collider does not have any latent descendants (and thus cannot be a parent of another latent collider).*

Definition 12 *The single cluster that corresponds to the observed major value configuration, and thus also represents the major effect* $MAE(\mathbf{ex})$ *due to configuration* **ex** *of* **EX**, *is the major cluster for* **ex**, *and all the clusters that correspond to the observed minor value configurations due to minor effects in* $MIES(\mathbf{ex})$ *are minor clusters.*

Definition 13 *A k-order minor effect is a minor effect in which exactly k endogenous variables in* **EN** *correspond to minor local effects. An* **en** *corresponding to a k-order minor effect is a k-order minor value configuration.*

Definition 14 *Minor clusters that correspond to k-order minor effects are k-order minor clusters.*

Definition 15 *Pairwise cluster comparison is a procedure by which pairs of clusters are compared, for example through a comparison of their centroids. The result of PCC between a pair of cluster centroids of dimension* |**O**|*, where* **O** *is the set of observed variables, can be represented by a binary vector of size* |**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.*

Definition 16 *A maximal set of observed* ($\boldsymbol{MSO}$) *variables is 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.*

Definition 18 *2S-PCC is PCC between 1-MC and a major cluster that shows two sets of two or more elements corresponding to the observed variables. Elements in each set have the same value, which is different than that of the other set. Accordingly, this 1-MC is defined as 2S-MC.*

Definition 19 *A **2S-MSO** is the maximal set of observed variables that always change their values together in all 2S-PCCs.*

Proposition 10 *In 2S-PCCs in which only the observed children of a single latent change, the latent is*

1. *EX or its leaf latent non-collider descendant, if the connection is serial; or*
2. *EX's leaf latent non-collider descendant, if the connection is diverging.*

Theorem 1 *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.*

Theorem 2 *A latent variable L is a collider of a set of latent ancestors* **LA**⊂**EX** *only if:*

1. *The values of the children of L change in different parts of some major–major PCCs each time with the values of descendants of another latent ancestor in **LA**; and*
2. *The values of the children of L do not change in any PCC unless the values of descendants of at least one of the variables in **LA** change too.*

Theorem 3 *Variables of a particular **2S-MSO** are children of an exogenous latent variable EX or any of its descendant latent non-colliders NC.*

Theorem 4 *A latent non-collider NC1 is a direct child of another latent non-collider NC2 (both on the same path emerging in EX) only if:*

- *In all 2S-PCCs for which EX does not change, the observed children of NC1 always change with those of NC2 and also in a single 2S-PCC without the children of NC2; and*
- *In all 2S-PCCs for which a latent non-collider leaf descendant of EX does not change, the observed children of NC2 always change with those of NC1 and also in a single 2S-PCC without the children of NC1.*

Appendix B. Additional results for the simulated data (Section 3.1)

B.1 LPCC compared to BPC

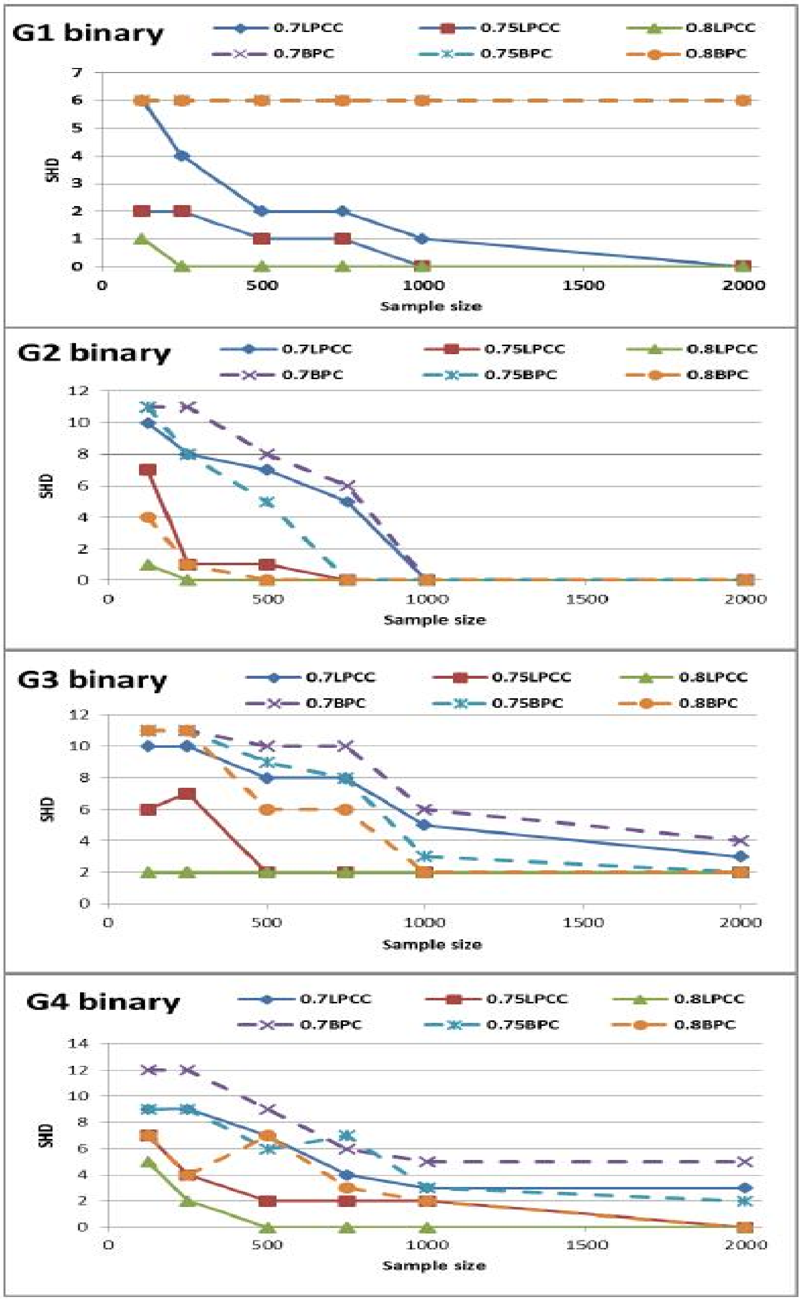

Figure 17: SHD learning curves of LPCC compared with those of BPC for G1–G4 of Figure 2 with binary variables, three parameterization levels, and increasing sample sizes.

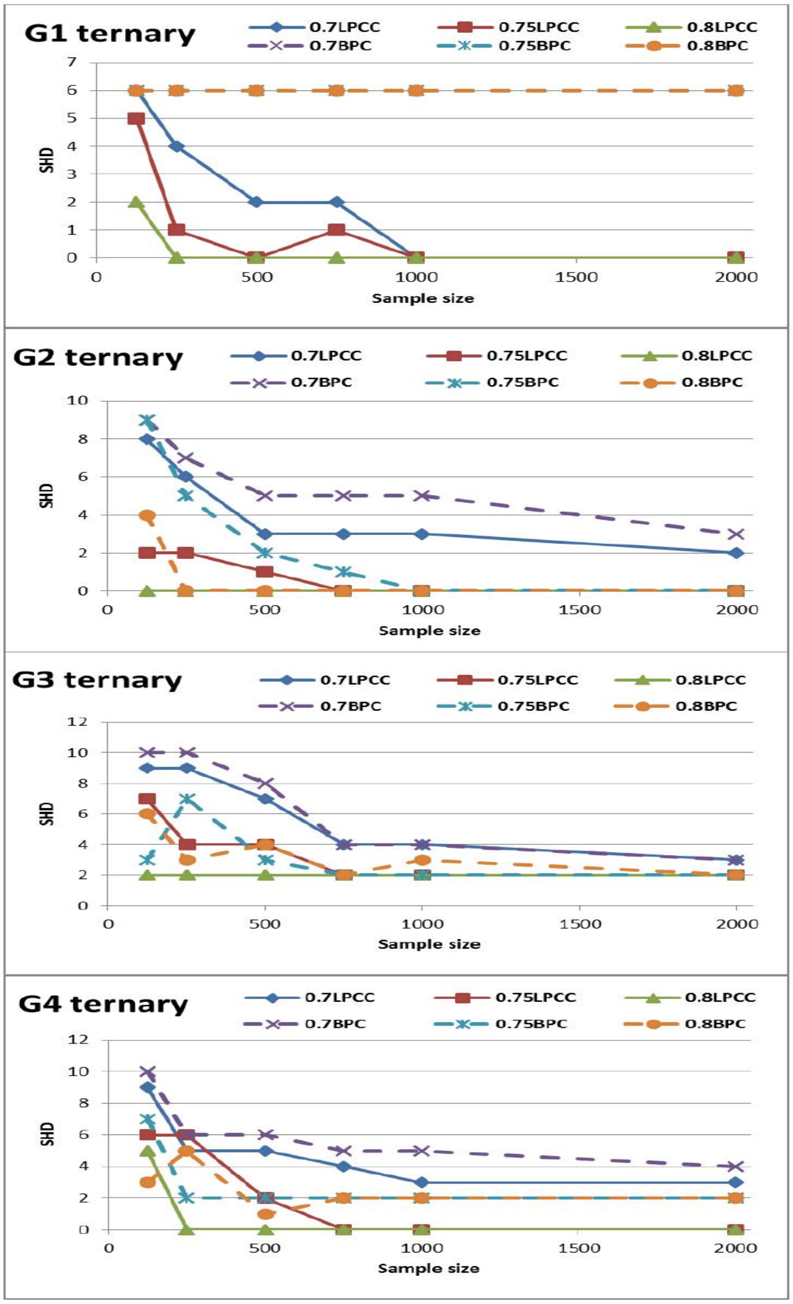

Figure 18: SHD learning curves of LPCC compared with those of BPC for G1–G4 of Figure 2 with ternary variables, three parameterization levels, and increasing sample sizes.

B.2 LPCC compared to EFA

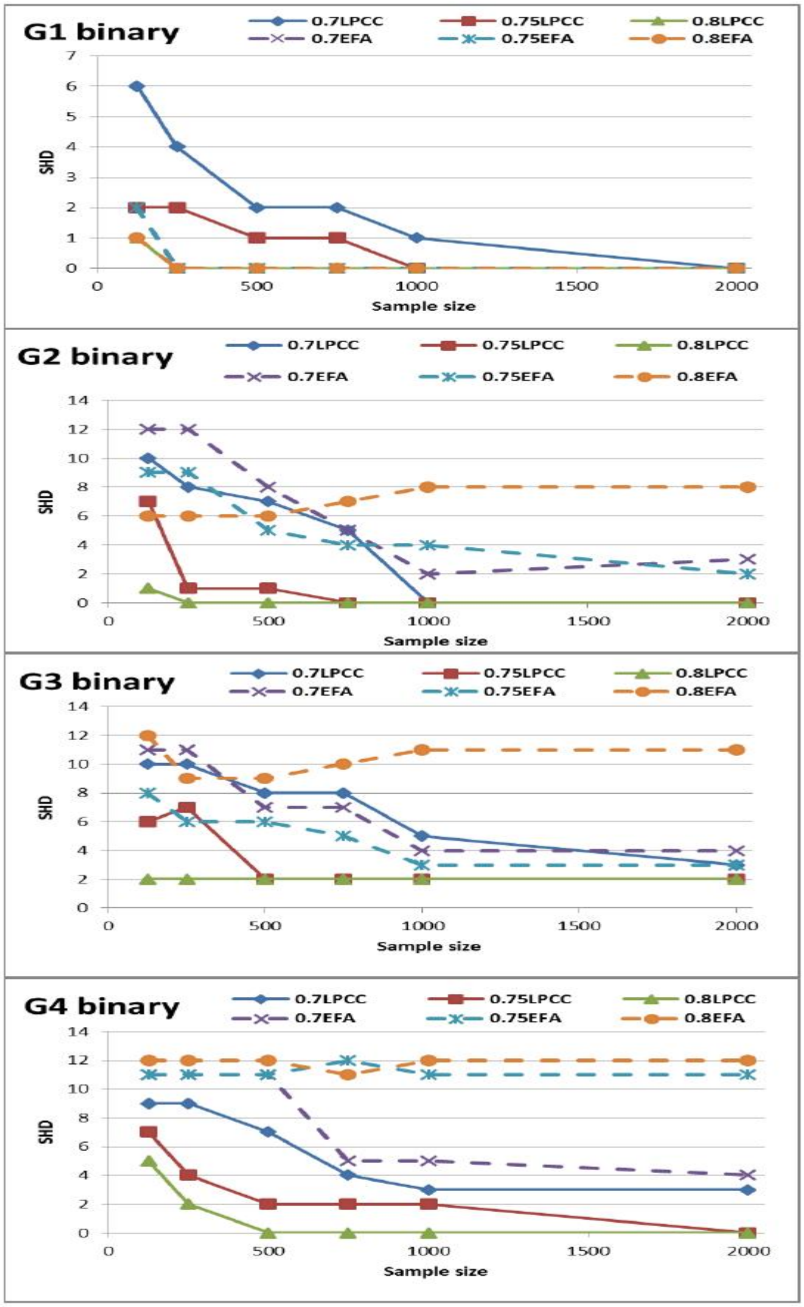

Figure 19: SHD learning curves of LPCC compared with those of EFA for G1–G4 of Figure 2 with binary variables, three parameterization levels, and increasing sample sizes. The lines of LPCC and EFA for a parametrization of 0.8 coincide for G1.

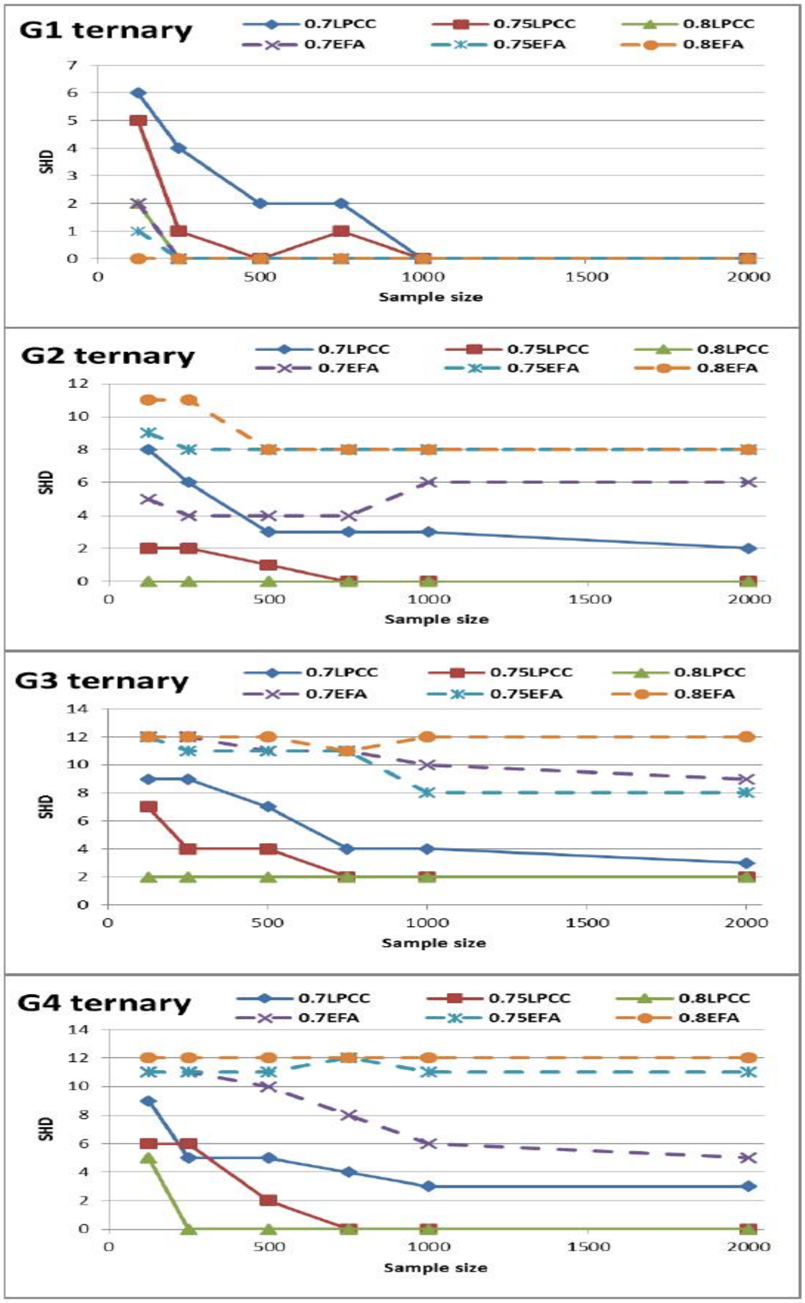

Figure 20: SHD learning curves of LPCC compared with those of EFA for G1–G4 of Figure 2 with ternary variables, three parameterization levels, and increasing sample sizes. The line of LPCC for a parametrization of 0.8 coincides with that of EFA for a parametrization of 0.7 for G1.

Appendix C. PCC analysis for two example databases

C.1 Results for the political action survey data (Section 3.2)

We applied clustering analysis to the political action survey data using SOM having 250 unit map size (similar results were obtained for SOMs having 125 and 500 unit map sizes). U-matrix visualization [9] of the SOM result is given in Figure 21. As presented in Table 2, nine clusters were found, and since four clusters are larger than the average cluster size of 45, only four of the nine clusters are major. Table 3 shows PCCs between these four major clusters. Note that NOSAY and VOTING always change together in all PCCs in which either of them changes, and this is also the case for NOCARE, TOUCH, and INTEREST. Therefore, LPCC found two latents (Figure 8 b): One (Efficacy) corresponds to NOSAY and VOTING and the other (Responsiveness) corresponds to NOCARE, TOUCH, and INTEREST.

Centroid	NOSAY	VOTING	COMPLEX	NOCARE	TOUCH	INTEREST
C1(86)	3	3	2	3	3	3
C2(60)	2	2	2	2	2	2
C3(57)	3	3	2	2	2	2
C4(49)	3	3	3	3	3	3
C5(39)	3	2	2	2	2	2
C6(31)	3	3	2	3	2	2
C7(31)	3	2	3	3	3	3
C8(28)	1	1	1	1	1	1
C9(27)	3	2	2	3	3	3

Table 2: Nine clusters are represented by their centroids for the political action survey data. Cluster sizes are in parentheses. The first four clusters are major.

PCC	$\delta NOSAY$	$\delta VOTING$	$\delta COMPLEX$	$\delta NOCARE$	$\delta TOUCH$	$\delta INTEREST$
PCC1,2	1	1	0	1	1	1
PCC1,3	0	0	0	1	1	1
PCC1,4	0	0	1	0	0	0
PCC2,3	1	1	0	0	0	0
PCC2,4	1	1	1	1	1	1
PCC3,4	0	0	1	1	1	1

Table 3: PCCs between the four major clusters for the political action survey data.

[9] The U-matrix is a widely used visualization of SOM. It computes (for each unit in the SOM) the mean of the distance measures between neighbors. By plotting this data on a 2D map using a color scheme, we can visualize a landscape with walls (red areas) and valleys (blue areas). The walls separate different clusters; they represent extreme distances between neighboring units, whereas patterns mapped to units in the same valley are similar and belong to the same cluster (Ultsch et al., 1993).

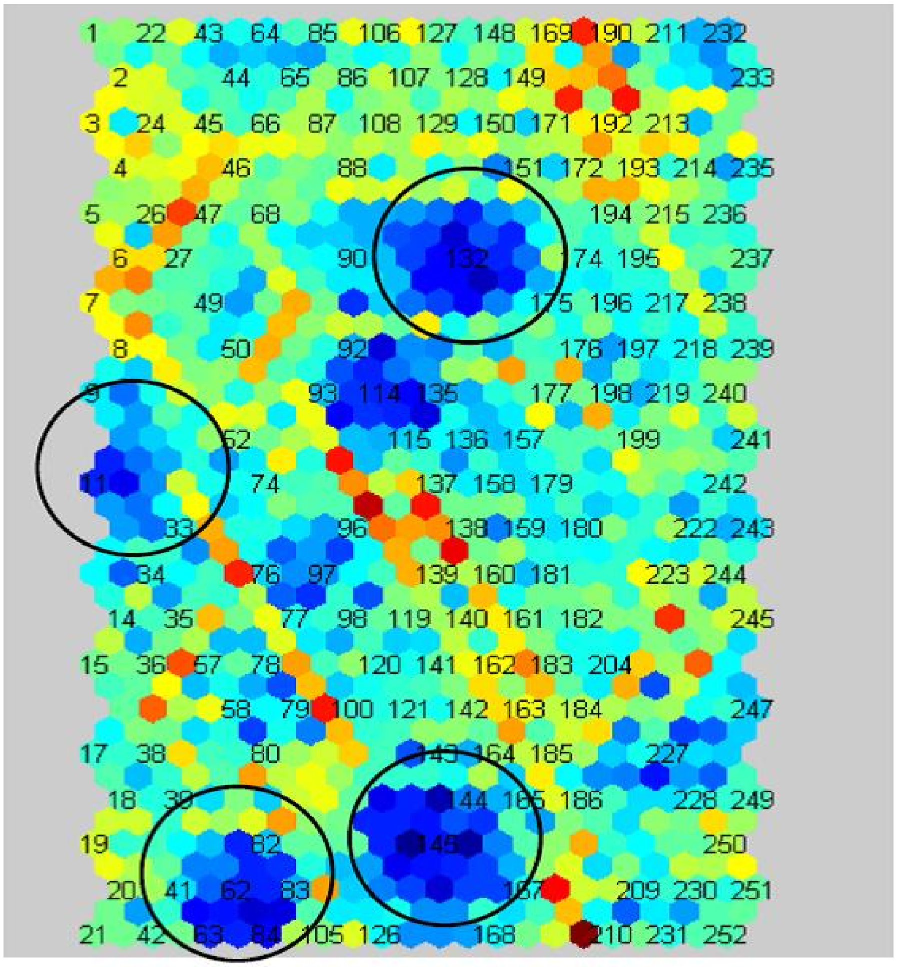

Figure 21: U-matrix visualization of a 250 unit map size SOM (numbers on the map represent the SOM units) obtained for the political action survey data. Vectors that were mapped to the same map unit belong to the same valley (blue area) and the same cluster. Of the nine clusters that were found (Table 2), four are major (circled) and the remaining are minor.

C.2 Results for Holzinger and Swineford's data (Section 3.3)

We applied clustering analysis to the European Values Survey (Holzinger and Swineford's) data set using SOM having a 100 map size. Fifteen clusters were found (Table 4), and since the average cluster size is 7.4, four clusters are major. Based on major-major PCCs (Table 5), one learned latent (Spatial) corresponds to VisPerc and Lozenges (that always change together in all PCCs in which either of them changes) and the other latent (Verbal) corresponds to ParComp, SenComp, and WordMean (Figure 10). It is interesting to see that in half of the PCCs in which VisPerc and Lozenges change together, Cubes also changes. If Cubes would have changed in all PCCs in which VisPerc and Lozenges change together, then it would be found as Spatial's child, and the learned model would be exactly the theoretical model (Figure 10a). This did not happen, probably because the PCCs in which Cubes did not change with VisPerc and Lozenges relate to relatively small clusters.

Centroid	**VisPerc**	**Cubes**	**Lozenges**	**ParComp**	**SenComp**	**WordMean**
C1(19)	1	1	1	1	1	1
C2(18)	2	2	2	2	2	2
C3(9)	1	2	1	1	1	1
C4(8)	2	2	2	1	1	1
C5(7)	1	1	1	1	1	2
C6(7)	2	1	2	1	1	1
C7(7)	1	1	2	1	1	1
C8(6)	2	1	2	2	2	2
C9(6)	2	1	1	1	1	1
C10(5)	2	1	1	2	2	2
C11(5)	1	2	2	2	2	2
C12(4)	1	2	2	2	1	2
C13(4)	2	2	1	2	2	2
C14(3)	1	1	1	2	2	2
C15(3)	2	1	1	1	2	2

Table 4: Fifteen clusters are represented by their centroids for Holzinger and Swineford's Data. Cluster sizes are in parentheses. The first four clusters are major.

PCC	$\delta VisPerc$	$\delta Cubes$	$\delta Lozenges$	$\delta ParComp$	$\delta SenComp$	$\delta WordMean$
PCC1,2	1	1	1	1	1	1
PCC1,3	0	1	0	0	0	0
PCC1,4	1	1	1	0	0	0
PCC2,3	1	0	1	1	1	1
PCC2,4	0	0	0	1	1	1
PCC3,4	1	0	1	0	0	0

Table 5: PCCs between the four major clusters for Holzinger and Swineford's data.

References

J. L. Arbuckle. *Amos Users' Guide Version 3.6*. Small Waters Corporation, Chicago, IL, 1997.

D. J. Bartholomew and M. Knott. *Latent Variable Models and Factor Analysis*. Kendall's Library of Statistics 7. Arnold Press, London, United Kingdom, 2nd edition, 1999.

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

R. V. Bruggner, B. Bodenmiller, D. L. Dill, R. J. Tibshirani, and G. P. Nolan. Automated identification of stratifying signatures in cellular subpopulations. In *Proceedings of the National Academy of Sciences of the United States of America*, pages 111(26):E2770–E2777, 2014.

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.

C. Glymour. *The Mind's Arrow: Bayes Nets and Graphical Causal Models in Psychology*. MIT Press, Cambridge, Massachusetts, 2002.

L. A. Goodman. Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*, 61:215–231, 1974.

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.

R. K. Henson and J. K. Roberts. Use of exploratory factor analysis in published research. Common errors and some comment on improved practice. *Educational and Psychological Measurement*, 66:393–416, 2006.

K. J. Holzinger and F. Swineford. A study in factor analysis: The stability of a bifactor solution. Technical Report 48, Supplementary Educational Monographs, University of Chicago Press, Illinois, 1939.

C.A. Janeway, P. Travers, M. Walport, and M. Schlomchick. *Immunobiology*. Garland Publishing, New York, 5th edition, 2001.

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

K. G. Joreskog and D. Sorbom. *Lisrel 7: A Guide to the Program and Applications*. SPSS, Chicago, 1989.

H. F. Kaiser. The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, 20:141–151, 1960.

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

S. L. Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19:191–201, 1995.

P. F. Lazarsfeld and N. W. Henry. *Latent Structure Analysis*. Houghton Mifflin, Boston, Massachusetts, 1968.

B. Lerner. Young drivers' crash involvement, involvement prediction, and evaluation of the impact of Or Yarok kit on the involvement using machine learning. Technical report, Ran Naor Institute for the Advancement of Road Safety Research, 2012. http://www.rannaorf.org.il/Young_Novice_Drivers_Researches.

B. Lerner and J. Meyer. Identification of factors that account for young drivers' crash involvement and involvement prediction using machine learning. Technical report, Israel National Road Safety Authority, 2012. http://www.rsa.gov.il/English/ResearchAndSurveys/Pages/YoungDrivers.aspx.

B. F. J. Manly. *Multivariate Statistical Methods. A Primer*. Chapman & Hall, London, United Kingdom, 1994.

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

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, 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.

A. Ultsch, G. Guimarães, D. Korus, and H. Li. Knowledge extraction from artificial neural networks and applications. In *Proceedings of Transputer-Anwender-Treffen/World-Transputer-Congress*, pages 194–203, Aachen, Germany, 1993.

A. Vazquez, R. Dobrin, D. Sergi, J.-P. Eckmann, Z.N. Oltvai, and A.-L Barabasi. The topological relationship between the large-scale attributes and local interaction patterns of complex networks. In *Proceedings of the National Academy of Sciences of the United States of America*, pages 101(52):17940–17945, 2004.

J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas. SOM toolbox for Matlab 5. Technical Report A57, Helsinki University of Technology, Helsinki, Finland, 2000.

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.

J. Wishart. Sampling errors in the theory of two factors. *British Journal of Psychology*, 19: 180–187, 1928.

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