Causal Clustering for 2-Factor Measurement Models

Erich Kummerfeld, Joe Ramsey, Renjie Yang, Peter Spirtes, and Richard Scheines[*]

Department of Philosophy, Carnegie Mellon University

Abstract. Many social scientists are interested in inferring causal relations between "latent" variables that they cannot directly measure. One strategy commonly used to make such inferences is to use the values of variables that can be measured directly that are thought to be "indicators" of the latent variables of interest, together with a hypothesized causal graph relating the latent variables to their indicators. To use the data on the indicators to draw inferences about the causal relations between the latent variables (known as the *structural model*), it is necessary to hypothesize causal relations between the indicators and the latents that they are intended to indirectly measure, (known as the *measurement model*). The problem addressed in this paper is how to reliably infer the measurement model given measurements of the indicators, without knowing anything about the structural model, which is ultimately the question of interest. In this paper, we develop the *FindTwoFactorClusters* (*FTFC*) algorithm, a search algorithm that, when compared to existing algorithms based on vanishing tetrad constraints, also works for a more complex class of measurement models, and does not assume that the model describing the causal relations between the latent variables is linear or acyclic.

1 Introduction

Social scientists are interested in inferring causal relations between "latent" variables that they cannot directly measure. For example, Bongjae Lee conducted a study in which the question of interest was the causal relationships between *Stress*, *Depression*, and (religious) *Coping*. One strategy commonly used to make such inferences is to use the values of variables that can be measured directly (e.g. answers to questions in surveys) that are thought to be "indicators" of the latent variables of interest, together with a hypothesized causal graph relating the latent variables to their indicators. A model in which each latent variable of interest is measured by multiple indicators (which may also be caused by other latents of interest as well as by an error variable) is called a *multiple indicator model* [1]. Lee administered a questionnaire to 127 students containing questions

[*] We thank NSF for providing funding through grant 1317428 for this work. We thank anonymous referees for helpful suggestions.

T. Calders et al. (Eds.): ECML PKDD 2014, Part II, LNCS 8725, pp. 34–49, 2014.

whose answers were intended to be indicators of *Stress*, *Depression*, and *Coping*. There were 21 questions relating to *Stress* (such as meeting with faculty, etc.) which students were asked to rate on a seven point scale, and similar questions for the other latents [2].

To use the data on the indicators to draw inferences about the causal relations between the latents (known as the *structural model*), it is necessary to hypothesize causal relations between the indicators and the latents that they are intended to indirectly measure (i.e. the subgraph containing all of the vertices, and all of the edges except for the edges between the latent variables, known as the *measurement model*). Given the measurement model, there are well known algorithms for making inferences about the structural model [2]. The problem addressed in this paper is how to reliably infer the measurement model given sample values of the indicators, without knowing anything about the structural model. In [2], Silva et al. developed an algorithm that reliably finds certain kinds of measurement models without knowing anything about the structural model other than its linearity and acyclicity. Their method first employs a clustering method to identify "pure" measurement sub-models (discussed below). (Note that in this context, *variables* rather than *individuals* are being clustered.) In this paper, we develop the *FindTwoFactorClusters* (*FTFC*) algorithm, an algorithm for reliably generating pure measurement submodels on a much wider class of measurement models, and does not assume that the model describing the causal relations between the latent variables is linear or acyclic.

1.1 Structural Equation Models (SEMs)

We represent causal structures as structural equation models (SEMs). In what follows, random variables are in italics, and sets of random variables are in boldface. Linear structural equation models are described in detail in [3]. In a structural equation model (SEM) the random variables are divided into two disjoint sets, the *substantive variables* (typically the variables of interest) and the *error variables* (summarizing all other variables that have a causal influence on the substantive variables) [3]. Corresponding to each substantive random variable V is a unique error term ϵ_V. A *fixed parameter SEM* S has two parts $\langle \phi, \theta \rangle$, where ϕ is a set of equations in which each substantive random variable V is written as a function of other substantive random variables and a unique error variable, together with θ, the joint distributions over the error variables. Together ϕ and θ determine a joint distribution over the substantive variables in S, which will be referred to as the distribution entailed by S. A *free parameter linear SEM model* replaces some of the real numbers in the equations in ϕ with real-valued variables and a set of possible values for those variables, e.g. $X = a_{X,L}L + \epsilon_X$, where $a_{X,L}$ can take on any real value. In addition, a free parameter SEM can replace the particular distribution over ϵ_X and ϵ_L with a parametric family of distributions, e.g. the bi-variate Gaussian distributions with zero covariance. The free parameter SEM also has two parts $\langle \Phi, \Theta \rangle$, where Φ contains the set of equations with free parameters and the set of values the free parameters are allowed to take, and Θ is a family of distributions over the error variables. In general, we will assume that there is a finite set of free parameters, and all allowed values of the free parameters lead to fixed parameter SEMs that have a reduced form (i.e. each substantive variable X can be expressed as a function of the error variables of X and the error variables of its ancestors), all variances and partial variances among the substantive variables are finite and positive, and there are no deterministic relations among the measured variables.

The *path diagram* (or *causal graph*) of a SEM with is a directed graph, written with the conventions that it contains an edge $B \to A$ if and only if B is a non–trivial argument of the equation for A. The error variables are not included in the path diagram unless they are correlated, in which case they are included and a double-headed arrow is placed between them. A fixed-parameter acyclic structural equation model (without double-headed arrows) is an instance of a Bayesian Network $\langle G, P(V) \rangle$, where the path diagram is G, and $P(V)$ is the joint distribution over the variables in G entailed by the set of equations and the joint distribution over the error variables, which in this case is just the product of the marginal distribution over the error variables [4]. A polynomial equation Q where the variables represent covariances is *entailed* by a free parameter SEM when all values of the free parameters entail covariance matrices that are solutions to Q. For example, a vanishing tetrad difference holds among $\{X, W\}$ and $\{Y, Z\}$, iff $cov(X,Y)cov(Z,W) - cov(X,Z)cov(Y,W) = 0$, and is entailed by a free parameter linear SEM S in which X, Y, Z, and W are all children of just one latent variable L.

1.2 Pure 2-Factor Measurement Models

In *1–factor measurement models* and *2–factor measurement models* each indicator has the specified number of latent parents in addition to its "error" variable. There is often no guarantee, however, that the indicators do not have unwanted additional latent common causes, or that none of the indicators are causally influenced by any other indicators. However, pure measurement models (defined below) have properties described below that make them easy to find, regardless of the structural models, and for that reason the strategy we will adopt in this paper is to search for a subset of variables that form a pure measurement model. In what follows, we will assume that no measured variable (indicator) causes a latent variable.

A set of variables $\mathbf{V}$ is *minimally causally sufficient* when every cause of any two variables in $\mathbf{V}$ is also in $\mathbf{V}$, and no proper subset of $\mathbf{V}$ is causally sufficient. If $\mathbf{O}$ is a set of indicators, and $\mathbf{V}$ is a minimally causally sufficient set of variables containing $\mathbf{O}$, then an *n-factor model* for $\mathbf{V}$ is a model in which there is a partition P of the indicators, and where each element of the partition is a set of indicators, all of which have exactly n latent parents, and that share the same n latent parents; if in addition there are no other edges (either directed, or bidirected representing correlated errors) into or out of any of the indicators the measurement model is said to be *pure*. We will refer to any n-factor model whose measurement model is pure as a *pure n-factor model*. Figure 1 is not a pure 2-factor measurement model. There are three reasons for this: X_1 causes

X_9, X_{15} has three latent direct causes, L_2, L_3, and L_4, and there is a latent cause L_5 of X_8 and L_1. However, note that the sub-model that does not contain the vertices X_1, X_8, X_9 and X_{15} is a 2-pure measurement model, because when those variables are not included, there are no edges out of any indicator, and the only edges into each indicator are from their two latent parents.

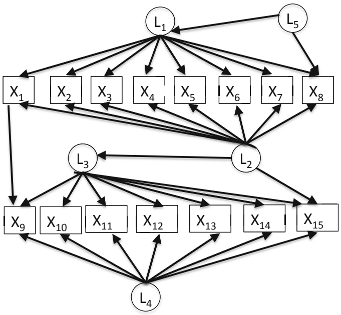

Fig. 1. Impure 2-factor model

Given a measurement model, any subset **S** of **O** for which every member of **S** is a child of the same n latent parents (and has no other parents), is adjacent to no other member of **O**, and has a correlated error with no other member of **V**, is an *n–pure* subset. In Figure 1, $\{X_2, X_3, X_4, X_5, X_6, X_7\}$ is a 2-pure sextet, but $\{X_{10}, X_{11}, X_{12}, X_{13}, X_{14}, X_{15}\}$ and $\{X_2, X_3, X_4, X_{10}, X_{11}, X_{12}\}$ are not.

2 Trek Separation

This section describes the terminology used in this paper. A *simple trek* in directed graph G from i to j is an ordered pair of directed paths $(P_1; P_2)$ where P_1 has sink i, P_2 has sink j, and both P_1 and P_2 have the same source k, and the only common vertex among P_1 and P_2 is the common source k. One or both of P_1 and P_2 may consist of a single vertex, i.e., a path with no edges. There is a trek between a set of variables $\mathbf{V_1}$ and a set of variables $\mathbf{V_2}$ iff there is a trek between any member of $\mathbf{V_1}$ and any member of $\mathbf{V_2}$. Let **A**, **B**, be two disjoint subsets of vertices **V** in G, each with two vertices as members. Let $\mathbf{S(A, B)}$ denote the sets of all simple treks from a member of **A** to a member of **B**.

Let **A**, **B**, $\mathbf{C_A}$, and $\mathbf{C_B}$ be four (not necessarily disjoint) subsets of the set **V** of vertices in G. The pair $(\mathbf{C_A}; \mathbf{C_B})$ *t-separates* **A** from **B** if for every trek $(P_1; P_2)$ from a vertex in **A** to a vertex in **B**, either P_1 contains a vertex in $\mathbf{C_A}$ or P_2 contains a vertex in $\mathbf{C_B}$; $\mathbf{C_A}$ and $\mathbf{C_B}$ are *choke sets* for **A** and **B** [6]. Let $\#\mathbf{C}$ be the number of vertices in **C**. For a choke set $(\mathbf{C_A}; \mathbf{C_B})$, $\#\mathbf{C_A} + \#\mathbf{C_B}$ is the *size of the choke set*. We will say that a vertex X is in a choke set $(\mathbf{C_A}; \mathbf{C_B})$ if $X \in \mathbf{C_A} \cup \mathbf{C_B}$.

The exact definition of linear acyclicity (or LA for short) below a choke set is somewhat complex and is described in detail in [6]; for the purposes of this paper it suffices to note that roughly speaking a directed graphical model is *LA below sets* ($\mathbf{C_A}$; $\mathbf{C_B}$) for $\mathbf{A}$ and $\mathbf{B}$ respectively, if there are no directed cycles between $\mathbf{C_A}$ and $\mathbf{A}$ or $\mathbf{C_B}$ and $\mathbf{B}$, and $\mathbf{A}$ is a linear function with additive noise of $\mathbf{C_A}$, and similarly for $\mathbf{B}$ and $\mathbf{C_B}$.

For two sets of variables $\mathbf{A}$ and $\mathbf{B}$, and a covariance matrix over a set of variables $\mathbf{V}$ containing $\mathbf{A}$ and $\mathbf{B}$, let $cov(\mathbf{A}, \mathbf{B})$ be the sub-matrix of the covariance matrix that contains the rows in $\mathbf{A}$ and columns in $\mathbf{B}$. In the case where $\mathbf{A}$ and $\mathbf{B}$ both have 3 members, if the rank of the $cov(\mathbf{A}, \mathbf{B})$ is less than or equal to 2, the determinant of $cov(\mathbf{A}, \mathbf{B}) = 0$. In that case the matrix is said to satisfy a *vanishing sextad constraint* since there are six members of $\mathbf{A} \cup \mathbf{B}$ if $\mathbf{A}$ and $\mathbf{B}$ are disjoint. For any given set of six variables, there are 10 different ways of partitioning them into two sets of three; hence for a given sextet of variables there are 10 distinct possible vanishing sextad constraints. The following two theorems [6] (extensions of the theorems in [5]) relate the structure of the causal graph to the rank of the determinant of sub-matrices of the covariance matrix.

Theorem 1. *(Extended Trek Separation Theorem): Suppose G is a directed graph containing* $\mathbf{C_A}$*,* $\boldsymbol{A}$*,* $\mathbf{C_B}$*, and* $\boldsymbol{B}$*, and (*$\mathbf{C_A}$*;*$\mathbf{C_B}$*) t-separates* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *in G. Then for all covariance matrices entailed by a fixed parameter structural equation model S with path diagram G that is LA below the sets* $\boldsymbol{C}_A$ *and* $\boldsymbol{C}_B$ *for* $\boldsymbol{A}$ *and* $\boldsymbol{B}$*, rank*$(cov(\mathbf{A}, \mathbf{B})) \leq \#\mathbf{C_A} + \#\mathbf{C_B}$.

Theorem 2. *For all directed graphs G, if there does not exist a pair of sets* $\mathbf{C'_A}$*,* $\mathbf{C'_B}$*, such that (*$\mathbf{C'_A}$*;* $\mathbf{C'_B}$*) t-separates* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *and* $\#\mathbf{C'_A} + \#\mathbf{C'_B} \leq r$*, then for any* $\mathbf{C_A}$*,* $\mathbf{C_B}$ *there is a fixed parameter structural equation model S with path diagram G that is LA below the choke sets (*$\mathbf{C_A}$*;* $\mathbf{C_B}$*) for* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *that entails rank*$(cov(\mathbf{A}, \mathbf{B})) > r$.

Theorem 1 guarantees that trek separation entails the corresponding vanishing sextad for all values of the free parameters, and Theorem 2 guarantees that if the trek separation does not hold, it is not the case that the corresponding vanishing sextad will hold for all values of the free parameters. It is still possible that if the vanishing sextad does not hold for all values of the free parameters, it will hold for some values of the free parameters. See [6].

3 Algorithm

Before stating the *sample* version of the algorithm (described below), we will motivate the intuitions behind it by an example (Figure 1). Let a *vanishing sextet* be a set of 6 indicators in which all ten sextads among the six variables are entailed to vanish by the Extended Trek Separation Theorem. In general, 2-pure sets of 5 variables (henceforth referred to as *pure pentads*) can be distinguished from non-2-pure sets of 5 variables (henceforth referred to as *mixed pentads*) by the following property: A pentad is 2-pure only if adding each of the other variable in $\mathbf{O}$ to the pentad creates a vanishing sextet. For example, in Figure 1, $\mathbf{S_1} = \{X_3, X_4, X_5, X_6, X_7\}$ is a 2-pure pentad. Adding any other variable to

$\mathbf{S_1}$ creates a sextet of variables which, no matter how they are partitioned, will have one side t-separated from the other side by a choke set $(\{L_1, L_2\} : \emptyset)$. In contrast, $\mathbf{S_2} = \{X_1, X_4, X_5, X_6, X_7\}$ is not pure, and when X_9 is added to $\mathbf{S_2}$, the resulting sextet is not a vanishing sextet, since when X_1 and X_9 are on different sides of a partition, at least 3 variables (including L_1, L_2, and X_1 or X_9) are needed to t-separate the treks between the variables in the two sides of the partition.

The first stage of the algorithm calls *FindPureClusters*, which tests each pentad to see if it has the property that adding any other member of **O** creates a vanishing sextet; if it does have the property it is added to the list *PureList* of 2-pure pentads. *FindPureClusters* tests whether a given sextet of variables is a vanishing sextet by calling *PassesTest*, which takes as input a sextet of variables, a sample covariance matrix, and the search parameter alpha that the user inputs to *FTFC*. *PassesTest* is implemented with an asymptotically distribution-free statistical test of sets of vanishing sextad constraints that is a modification of a test devised by Bollen and Ting [7]. The list of 2-pure pentads at this point of the algorithm is $\{X_{10}, X_{11}, X_{12}, X_{13}, X_{14}\}$ and every subset of X_2 through X_7 of size 5. X_1, X_9, and X_{15} do not appear in any 2-pure pentad. X_8 is also not in any pure sub-cluster, but *FTFC* is unable to detect that it is impure. This is the only kind of impurity *FTFC* cannot detect. See the explanation in Section 4 for why this is the case, and why this kind of mistake is not important. *GrowClusters* then initializes the *ClusterList* to *PureList*.

If any of the 2-pure sets of variables overlap, their union is also 2-pure. So *FTFC* calls *GrowClusters* to see if any of the 2-pure sextets in *PureClusters* can be combined into a larger 2-pure set. Theoretically, *GrowClusters* could simply check whether any two subsets on *PureClusters* overlap, in which case they could be combined into a larger 2-pure set. In practice, however, in order to determine whether a given variable o can be added to a cluster **C** in *ClusterList*, *GrowClusters* checks whether a given fraction (determined by the parameter *GrowParameter*) of the sub-clusters of size 5 containing 4 members of **C** and o are on *PureList*. If they are not, then *GrowClusters* tries another possible expansion of clusters on *ClusterList*; if they are, then *GrowClusters* adds o to **C** in *ClusterList*, and deletes all subsets of the expanded cluster of size 5 from *PureList*. *GrowClusters* continues until it runs out of possible expansions to examine.

Finally, when *GrowClusters* is done, *SelectClusters* goes through *ClusterList*, outputting the largest cluster **C** still on *ClusterList*, and deleting any other clusters on *ClusterList* that intersect **C** (including **C** itself).

Algorithm 1: FindTwoFactor Clusters (FTFC)

```
Data: $Data, V, GrowParameter, \alpha$
Result: $SelectedClusters$
$\langle Purelist, \mathbf{V} \rangle = FindPureClusters(Data, \mathbf{V}, \alpha)$
$Clusterlist = GrowClusters(Purelist, \mathbf{V})$
$SelectedClusters = SelectClusters(Clusterlist)$
```

```
Algorithm 2: FindPureClusters
Data: $\mathbf{V}, \mathbf{Data}, \alpha$
Result: $PureList$
$PureList = \varnothing$
for $\mathbf{S} \subseteq \mathbf{V}, |\mathbf{S}| = 5$ do
    $Impure = FALSE$
    for $v \in \mathbf{V} \setminus \mathbf{S}$ do
        if $PassesTest(\mathbf{S} \cup \{v\}, Data, \alpha) = FALSE$ then
            $Impure = TRUE$
            break
    if $Impure = FALSE$ then
        $PureList = c(\mathbf{S}, PureList)$
$\mathbf{V} = \bigcup_{i \in PureList} i$
return($\langle PureList, \mathbf{V} \rangle$)
```

```
Algorithm 3: GrowClusters
Data: $PureList, \mathbf{V}$
Result: $Clusterlist$
$Clusterlist = PureList$
for $cluster \in Clusterlist$ do
    for $\mathbf{sub} \subset \mathbf{cluster}, |\mathbf{sub}| = 4$ do
        for $o \in \mathbf{V} \setminus \mathbf{cluster}$ do
            $testcluster = \mathbf{sub} \cup \{o\}$
            if $testcluster \in PureList$ then
                $accepted++$
            else
                $rejected++$
        if $accepted \div (rejected + accepted) \geq GrowParameter$ then
            $Clusterlist = c(Clusterlist, \mathbf{cluster} \cup \{o\}$
            for $\mathbf{s} \subset \mathbf{cluster} \cup \{o\}, \mathbf{s} \in Clusterlist$ do
                $Purelist = Purelist \setminus \{\mathbf{s}\}$
```

The complexity of the algorithm is dominated by *FindPureClusters*, which in the worst case requires testing n choose 6 sets of variables, and for each sextet requires testing five of the ten possible vanishing sextad constraints in order to determine if they all vanish. In practice, we have found that it can be easily applied to about 30 measured variables at a time, but not 60 measured variables. http://www.phil.cmu.edu/projects/tetrad_download/launchers/ contains an implementation available by downloading tetrad-5.0.0-15-experimental.jnlp, creating a "Search" box, selecting "BPC" from the list of searches, and then setting "Test" to "TETRAD-DELTA", and "Algorithm" to "FIND_TWO_FACTORS_CLUSTER".

Algorithm 4: SelectClusters

```
Data: $Clusterlist$
Result: $Selectedlist$
$Selectedlist = \varnothing$
while $Clusterlist \neq \varnothing$ do
    Choose a largest $C$
    $Selectedlist = Selectedlist \cup \{C\}$
    for $s \in Clusterlist, s \cap C \neq \varnothing$ do
        $Clusterlist = Clusterlist \setminus \{s\}$
```

4 Correctness of Algorithm

In what follows, we will assume that if there is not a trek between some pair of indicators, or if there are entailed vanishing partial correlations among the observed indicators, or if there are rank constraints of size 1 on the relevant sub-matrices that the relevant variables are removed in a pre-processing phase. We will make the assumption that sextad constraints vanish only when they are entailed to vanish for all values of the free parameters (i.e vanishing sextad constraints that hold in the population are entailed to hold by the structure of the graph, not the particular values of the free parameters). In the linear case and other natural cases, the set of parameters that violates this assumption is Lebesgue measure 0 [6]. This still leaves the question of whether there are common"almost" violations of rank faithfulness that could only be discovered with enormous sample sizes (i.e. the relevant determinants are very close to zero), which we will address through simulation studies.

There is also a *population* version of the *FTFC* algorithm that differs from the sample algorithm described above in two respects. First, in *PassesTest* it takes as input a sextet of variables and a population covariance matrix, and tests whether all ten possible vanishing sextad constraints among a sextet of variables hold exactly. Second, in *GrowClusters* it sets *GrowParameter* to 1 (whereas in the simulation tests *GrowParameter* was set to 0.5.)

In a 2-factor model, two variables *belong to the same cluster* if they share the same two latent parents. A *5×1* sextad contains a sextet of variables, 5 of which belong to one cluster, and 1 of which belongs to a different cluster. For a given variable X, $L_1(X)$ is one of the two latent parents of X, and $L_2(X)$ is a second latent parent of X not equal to $L_1(X)$. An indicator X is *impure* if there is an edge into or out of X other than $L_1(X)$ or $L_2(X)$. Define $\mathbf{L}$ as the set of latent variables L such that $L = L_1(X)$ or $L_2(X)$ for some indicator X. (Latent variables not in $\mathbf{L}$ might be included in the graph if there are more than two common causes of a pair of indicators, or common causes of an indicator or a member of $\mathbf{L}$, e.g. L_5 in Figure 1.

Theorem 3 states that given a measurement model that has a large enough pure sub-model, the output of the *FTFC* algorithm is correct in the sense that the variables in the same output cluster share the same pair of latent parents, and that the only impure indicators X in the output are impure because there is a latent variable not in $\mathbf{L}$ that is a parent of X and $L_1(X)$ or $L_2(X)$ (e.g. L_5 in Figure 1 is a parent of L_1 and X_8). This kind of impurity is not detectible by the algorithm, but is also not important, because it does not affect the estimate of the value of the latent parent from the indicators. In addition, in the output, no single latent parent in $\mathbf{L}$ can be on two treks between latent variables not in $\mathbf{L}$ and an impure indicator; e.g. there cannot be two latent common causes of L_1 and two distinct indicators.

Theorem 3 assumes that the relationships between the indicators and their latent parents is linear. This assumption does not in general entail the model is LA below the choke sets for any arbitrary sextad, since in some cases the latent variables that are in a choke set are not the parents of the indicators in the sextad, in which case it is possible that non-linear relationships between the latent variables will lead to a non-linear relationship between the indicators and the latent variables in the choke set. However, for the particular kind of sextads that the *FTFC* algorithm relies on (i.e. 5×1 sextads) all of the choke sets contain parents of the indicators in the sextad. Hence, linear relationships between the indicators and their latent parents do entail the structure is LA below the choke sets for the sextads that the *FTFC* algorithm relies on for determining the structure of the output clustering,

Theorem 3. *If a SEM S is a 2-factor model that has a 2-pure measurement sub-model in which each indicator X is a linear function of $L_1(X)$ and $L_2(X)$, S has at least six indicators, and at least 5 indicators in each cluster, then the population FTFC algorithm outputs a clustering in which any two variables in the same output cluster have the same pair of latent parents. In addition, each output cluster contains no more than two impure indicators X_1 and X_2, one of which is on a trek whose source is a common cause of $L_1(X_1)$ and X_1, and the other of which is on a trek whose source is a common cause of $L_2(X_1)$ and X_2.*

Proof. First we will show that pure clusters of variables in the true causal graph appear clustered together in the output. Suppose $\mathbf{C} = \{X_1, X_2, X_3, X_4, X_5\}$ belong to a single pure cluster with latent variables L_a and L_b. For any sixth variable Y, and any partition of $\{X_1, X_2, X_3, X_4, X_5, Y\}$ into two sets of size 3, $\{X_a, X_b, X_c\}$ and $\{X_d, X_e, Y\}$, $\{X_a, X_b, X_c\}$ is trek-separated from $\{X_d, X_e, Y\}$ by a choke set containing just $\{L_a$ and $L_b\}$ since there are no other edges into or out of $\{X_a, X_b, X_c\}$ except for those from L_a and L_b. Hence $\mathbf{C}$ is correctly added to *PureList*.

Next we show that variables from different pure clusters in the true causal graph are not clustered together in the output. Suppose that two of the variables in $\mathbf{C}$ belong to different clusters. There are two cases. Either every member of $\mathbf{C}$ belongs to a different cluster or some pair of variables in $\mathbf{C}$ belong to the same cluster. Suppose first two members of $\mathbf{C}$, say X_1 and X_2, belong to a single cluster with latents L_a and L_b, and X_3 belongs to a different cluster with latent L_c. In that case, for any sixth variable Y from the same cluster as X_3, the partitions $\{X_1, X_3, X_4\}$ and $\{X_2, X_5, Y\}$ are not trek-separated by any choke set $\mathbf{S}$ of size 2, since L_a, and L_b would both have to be in $\mathbf{S}$ in order for $\mathbf{S}$ to trek-separate X_1 and $\{X_2, X_5, Y\}$, and L_c would have to be in $\mathbf{S}$ in order for $\mathbf{S}$ to trek-separate X_3 and $\{X_2, X_5, Y\}$. Hence $\mathbf{C}$ will correctly not be added to *PureList*. If, on the other hand, every member of $\mathbf{C}$ belongs to a different cluster, then choose a trek T between two variables in $\mathbf{C}$ such that there is no shorter trek between any two members of $\mathbf{C}$ than T. Suppose without loss of generality that these two variables in $\mathbf{C}$ are X_1 and X_2. Because the clusters are pure by assumption, every trek between X_1 and X_2 contains some pair of latent parents L_1 (a parent of X_1) and L_2 (a parent of X_2). The subtrek of T between L_1 and L_2 does not contain any latent parent of any member of $\mathbf{C} \setminus \{X_1, X_2\}$ since otherwise there would be a trek between two members of $\mathbf{C}$ shorter than T. By the assumption that the model has a pure 2-factor model measurement model, there is a third variable X_3 in $\mathbf{C}$ that is not equal to X_1 or X_2, and some other variable Y that belongs to the same cluster as X_3. X_3 and Y have two latent parents, L_{3a} and L_{3b} that do not lie on T. Consider the sextad $\mathrm{cov}(\{X_1, X_3, X_4\}, \{X_2, X_5, Y\})$. Then in order to trek-separate X_1 from the 3 variables in the side of the partition containing X_2, some latent not equal to L_{3a} or L_{3b} is required to be in the choke set. In order to trek-separate X_3 from the side of the partition containing Y, both L_{3a} and L_{3b} are required to be in the choke set. It follows that no choke set of size 2 trek-separates $\{X_1, X_3, X_4\}$ and $\{X_2, X_5, Y\}$, and $\mathbf{C}$ will not be added to *PureList*. Similarly, if two variables are from different impure clusters, they will not both be added to $\mathbf{C}$, since impurities imply the existence of even more treks, and hence choke sets that are at least as large as in the pure case.

Now we will show that only one kind of impure vertex can occur in an output cluster. Suppose that X is in cluster $\mathbf{C}$, but impure. By definition, there is either an edge E into or out of X that is not from $L_1(X)$ or $L_2(X)$. If E is out of X, then by the assumption that none of the measured indicators cause any of the latent variables in G, E is into some indicator Y. If $(\mathbf{S_1} : \mathbf{S_2})$ t-separates X from Y, and $\mathbf{S} = \mathbf{S_1} \cup \mathbf{S_2}$, then $\mathbf{S}$ contains either X or Y. Consider the sextad $cov(\{X, X_a, X_b\}, \{X_c, X_d, Y\})$, where X_a, X_b, X_c, X_d all belong to $\mathbf{C}$. In order to trek-separate X from X_c, $L_1(X)$ and $L_2(X)$ must be in choke set $\mathbf{S}$. Hence in order to separate both sets in the partition from each other, $\mathbf{S}$ must contain at least 3 elements ($L_1(X)$, $L_2(X)$, and X or Y), and there is a 5×1 sextad that is not entailed to vanish, so X is not clustered with the other variables by *FTFC*.

Suppose E is into X. If the tail of E is a measured indicator Y, then by the same argument as above, there is a 5×1 sextad that is not entailed to vanish, so X is not clustered with the other variables by *FTFC*. If the tail of E is $L_1(Y)$ or $L_2(Y)$ for some Y that is a measured indicator but not in $\mathbf{C}$, consider the sextad $cov(\{X, X_a, X_b\}, \{X_c, X_d, Y\})$, where X_a, X_b, X_c, X_d all belong to $\mathbf{C}$. In order to trek-separate X from X_c, $L_1(X)$ and $L_2(X)$ must be in choke set $\mathbf{S}$. Hence in order to separate both sets in the partition from each other, $\mathbf{S}$ must contain at least 3 elements ($L_1(X)$, $L_2(X)$, and $L_1(Y)$ or $L_2(Y)$). So there is a 5×1 sextad that is not entailed to vanish, and X is not clustered with the other variables by *FTFC*. If the tail of E is a latent variable L that is not equal to $L_1(Y)$ or $L_2(Y)$ for any Y that is a measured indicator but not in $\mathbf{C}$, then there is a shortest trek T between L and some latent parent $L_1(Y)$ of a measured indicator Y. If T contains a measured indicator, then this reduces to one of the previous cases. If Y is not in $\mathbf{C}$ then any trek-separating set of $\mathbf{S}$ must contain at least 3 elements ($L_1(X)$, $L_2(X)$, and some vertex along T that is not equal to $L_1(X)$ or $L_2(X)$). Hence there is a 5×1 sextad that is not entailed to vanish, and X is not clustered with the other variables by *FTFC*.

Finally, consider the case where there are two indicators X_1 and X_2 in $\mathbf{C}$ such that there is a latent common cause M_1 of X_1 and $L_1(X_1)$ and a latent common cause M_2 of X_2 and $L_1(X_1)$, or there is a latent common causes M_1 of X_1 and $L_2(X_1)$ and a latent common cause M_2 of X_2 and $L_2(X_1)$. Suppose without loss of generality that it is the former. If $M_1 = M_2$, then this reduces to one of the previous cases. Otherwise, there are treks T_1 and T_2 between X_1 and X_2 whose sources are M_1 and M_2 respectively. Because X_1 and X_2 are in the same cluster $\mathbf{C}$, in order to trek-separate X_1 and X_2 with a choke set ($\mathbf{S_1}$:$\mathbf{S_2}$), $\mathbf{S_1}$ or $\mathbf{S_2}$ must contain $L_2(X_1)$. In order to separate T_1 and T_2, $L_1(X_1)$ must be in both $\mathbf{S_1}$ and $\mathbf{S_2}$ since $L_1(X_1)$ occurs on the X_2 side of T_1 and the X_1 side of T_2. It follows that $\mathbf{S_1} \cup \mathbf{S_2}$ contains at least 3 elements. Hence there is a 5×1 sextad that is not entailed to vanish, and X_1 and X_2 are not both clustered with the other variables by *FTFC*.

So after the first stage of the algorithm, *PureList* is correct, and hence *ClusterList* is correct (up to the kinds of impurities just described) before it is subsequently modifed.

Now we will show that each stage of modifying *ClusterList* and *PureList* is correct. For a given cluster $\mathbf{C}$, if a variable o belongs to the same cluster, then for every subset of $\mathbf{C} \cup\{o\}$ of size 5, a choke set that contains $L_a(\mathbf{C})$ and $L_b(\mathbf{C})$ t-separates any two members of $\mathbf{C} \cup\{o\}$. Hence $\mathbf{C} \cup\{o\}$ will have passed the purity test, and be found on *PureList*; hence *GrowClusters* will correctly add o to $\mathbf{C}$, and subsets of $\mathbf{C} \cup\{o\}$ will be correctly deleted from *PureList*. If on the other hand o does not belong to the same cluster as $\mathbf{C}$, then some subsets of $\mathbf{C}$ $\cup\{o\}$ of size 5 are not pure, and will not appear in *PureList*. Hence $\mathbf{C} \cup\{o\}$ will not be added to *ClusterList*. Finally, the same argument showing the kinds of impurities that could occur on *PureList* can be applied to *ClusterList*. □

This theorem entails that if there is a 2-factor model with a 2-pure measurement model with sufficiently many variables and a large enough sample size, then *FTFC* will detect it and output the correct clustering. Unfortunately the converse is not true — there are models that do not contain 2-pure measurement sub-models that entail exactly the same set vanishing sextad differences over the measured variables (i.e. are *sextad-equivalent*)[5]; for those alternative models, *FTFC* will output clusters anyway. However, for linear models, it is possible to perform a chi-squared test of whether the measurement model is 2-pure, using structural equation modeling programs such as *EQS*, or *sem* in *R*, or the tests in *TETRAD IV*. In practice, a pure 2-factor model will be rejected by a chi-squared test given data generated by all of the known models sextad-equivalent to a 2-factor model (because of differences between the models in inequality constraints). For this reason, in ideal circumstances, the *FTFC* algorithm would be one part of a larger generate (with *FTFC*) and test (with structural equation modeling estimating and testing) algorithm. See [6] for details.

5 Tests

We tested the *FTFC* algorithm on simulated and real data sets. We did not directly compare it to other algorithms for the non-linear cases, since to our knowledge there are no other algorithms that can handle non-linearities and/or cyclic relations among the latent variables, impurities in the measurement model, and multiple factors for each cluster. Factor analysis has been used to cluster variables, but has not proved successful even in cases where each cluster has a single latent common cause but impurities [2]. The BuildPureClusters Algorithm uses vanishing tetrad constraints, instead of vanishing sextad constraints to cluster variables, but assumes that each cluster has at most one latent common cause [2]. In the linear, acyclic case, we did compare *FTFC* to a semi-automated search for a special case of two-factor linear acyclic models, as described in the section on Linear Acyclic models.

5.1 Simulations

The first directed graph we used to generate data has 3 clusters of 10 measured variables each, with each cluster having two latent variables as causes of each measured variable in the cluster, and one of each pair of latent variables for the second cluster causing one of each pair of latent variables in the first cluster, and one of each pair of latent variables in the third cluster. The second directed acyclic graph we used to generate data in addition contained 7 impurities: X_1 is a parent of X_2 and X_3, X_2 is a parent of X_3, L_1 is a parent of X_{11} and X_{21}, X_{20} is a parent of X_{21}, and L_4 is a parent of X_{30}.

For each graph, we generated data at three different sample sizes, $n = 100$, 500, and 1000. The *FTFC* algorithm was run with significance level (for the vanishing sextad tests) of 0.1 for sample sizes 500 and 1000, and 0.4 for sample size 100. Theoretically, non-linearity among the latent-latent connections should not negatively affect the performance of the algorithm, as long as the sample size is large enough that the asymptotic normality assumed by the sextad test that we employed is a good approximation. Theoretically, non-linearity among the latent-observed connections should negatively affect the performance of the algorithm, since if there are non-linear latent-observed connections, the Extended Trek Separation Theorem generally does not apply. For each graph and each sample size we generated four kinds of models, with each possible combination of linear or non-linear latent-latent connections and linear or non-linear latent-observed connections. In all cases, the non-linearities replace linear relationships with a convex combination of linear and cubic relationships. For example, in the pure model with non-linear latent-latent connections + non-linear latent-measured connections, each variable X was set to the sum over the parents of $0.5 * c_1 * P + 0.5 * d_1 * (.5 * P)^3$ plus an error, where P is one of the parents of X,

c_1 and d_1 were chosen randomly from a Uniform(.35,1.35) distribution, and each error variable was a Gaussian with mean zero, and a variance chosen randomly from a Uniform(2,3) distribution. We tested a few of the simulated data sets with a White test in R for non-linearity, and they rejected the null hypothesis of linearity quite strongly.

In many applications of multiple indicator models, the indicators are deliberately chosen so that the correlations are fairly large (greater than 0.1 in most cases), and all positive; in addition, there are relatively few correlations greater than 0.9. In order to produce correlation matrices with these properties, we had to adjust some of the parameters of the various models we considered according to the type of model (i.e. whether the latent-latent connections were linear or not, whether the latent-measured connections were linear or not, and whether the model was pure of not). We did not however, adjust the model parameters according to the results of the algorithm.

We calculated the precision for each cluster output, and the sensitivity for each cluster output. We then evaluated the output of the algorithm by the number of clusters found, and for each run, the average of the sensitivities and the average of the precisions over the clusters.

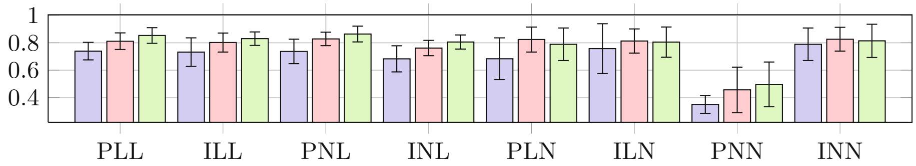

Fig. 2. Average Precision of The Output

The correct number of clusters in each case is 3, and the average number of clusters output ranged between 2.7 and 3.1 for each kind of model and sample size, except for PNN. As expected, non-linearities for the latent-observed connections degraded the performance, and the number of clusters for PNN at sample sizes 100, 500, and 1000 were1.05, 1.38, and 1.54 respectively.

Figure 2 shows the mean (over 50 runs) of the average precision of the clustering output for each simulation case. The error bars shows the standard deviation of the average precision. Figure 3 shows the mean (over 50 runs) average sensitivity of the clustering output for each simulation case. The error bars shows the standard deviation of the average sensitivity. The blue, red and green bars represent cases with 100, 500, and 1000 sample size respectively. In the three-letter lable for every group of three bars, the first letter refers to the purity of the generative model, with “P” being ”Pure” and “I” being “Impure”. The second letter refers to the linearity of the latent-latent connection, with “L” representing linear connections and “N” representing non-linear connections. The third letter refers to the linearity of the latent-measured connection, the letter “L” and “N” have the same meaning as the case of the second letter. For example, “PNL” represent the case in which the generative model is pure, with non-linear latent-latent connections, and linear latent-observed connections. We generated 50 models of each kind, except that due to time limitations, the sample size 100 PNN case has 40 runs, and the sample size 500 PLN case has 10 runs. The run times varied between 44 and 1328 seconds.

In general, as expected, the result is better as the sample size increases, and is worse when there are impurities adding to the graphical model. The non-linear latent-latent connections does not have an obvious effect upon the clustering output. However, as expected, when the non-linear latent-observed connections are added to the generative model, the mean value of the purity is lower than the corresponding linear cases, and the standard deviation of the two measures starts to increase. Most notably, in the case of "PNN", the interaction of the two kinds of non-linearities renders most of the clustering result being very large clusters (as indicated by the small number of clusters output). That is why the average precision becomes very small while the average sensitivity is relatively large.

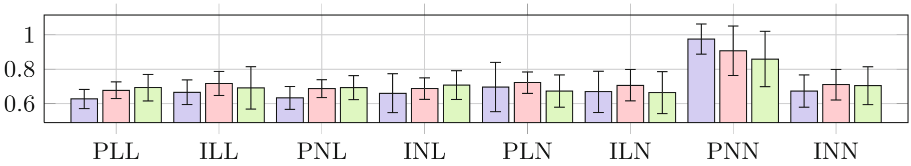

Fig. 3. Average Sensitivity of The Output

5.2 Real Data

We applied *FTFC* to six data sets in *R* for which there are published bifactor models (see the "Bechthold" help page in R). We ran *FTFC* at 5 significance levels 0.05, 0.1, 0.2, 0.3, and 0.4 and chose the best model. In some cases where there were multiple clusters which together did not pass a chi-squared test, we chose the best individual cluster. In Table 1, p is the number of variables, n is the sample size, $indicators$ is the number of indicators in the output, $clusters$ is the number of clusters in the output, and $p-value$ is the p-value of the best model. Because we did not have access to the original raw data (just the correlation matrices), we could not divide the data into a training set and a test set, leading to somewhat higher p-values than we would expect if we calculated the p-value on a separate test set.

Table 1. Results of Application of *FTFC* to *R* data sets

Data Set	p	n	$indicators$	$clusters$	$p - value$
Thurstone	9	213	6	1	0.96
Thurstone.33	9	417	5	1	0.52
Holzinger	14	355	7	1	0.23
Holzinger.9	9	145	6	1	0.82
Bechtholdt.1	17	212	8	1	0.59
Reise	16	1000	13	2	0.32

We also applied *FTFC* to the depression data. Lee's model fails a chi-square test: $p = 0$. Although the depression data set contained too many variables to test how well *FTFC* performed overall, we did use it to test whether it could remove impure variables from given clusters (formed from background knowledge), leading to a model that would pass a chi-squared test. Using the output of *FTFC* at several different significance levels, the best model that we found contained a cluster of 9 coping variables, 8 stress variables, and 8 depression variables (all latent variables directly connected) with a p-value of 0.28.

5.3 The Linear Acyclic Case

A bifactor model is a model in which there is a single general factor that is a cause of all of the indicators, and a set of "specific" factors that are causes of some of the indicators. It is a special case of a two-factor model. The *schmid* function in *R* takes as input a correlation matrix and (at least 3 specific) factors, and outputs a bifactor model; it first performs an ordinary factor analysis and then transforms the output into a bifactor model (which is a proper supermodel of one-factor models). We compare *FTFC* algorithm to a *FTFC-schmid* algorithm hybrid on real and simulated data.

We turned the two-factor model described in the previous set of simulations into a linear bifactor model by collapsing three of the latent variables from different clusters into a single variable. We did not find any functions for reliably automatically estimating the number of factors in a bifactor model, so we compared the *FTFC* algorithm to a *FTFC-schmid* hybrid, in which *FTFC* provided the number of factors input to *schmid*. The hybrid *FTFC-schmid* algorithm removed 1.6% of the intra-cluster impurities (e.g. X_1, X_2, X_3), and 48% of the inter-cluster impurities (e.g. X_{11}, X_{20}, X_{21}, X_{30}) while removing 8% of the pure variables . In contrast, *FTFC* removed 61% of the intra-cluster impurities, and 58% of inter-cluster impurities, while also removing 30% of the pure variables. While *FTFC* incorrectly removed many more pure variables than the hybrid *FTFC-schmid*, for the purposes of finding submodels that pass chi-squared tests, this is far less important than its superiority in removing far more of the impure variables

We then compared *schmid* to *FTFC* on the Reise data. The published bifactor model [8], the output of the *schmid* function in *R* with 5 specific factors (as in the published model), and versions of both of these models that removed the same variables that *FTFC* algorithm did, all failed chi-squared tests and had p-values of 0. The output of *FTFC* removed 3 of the 16 variables, and combined the five specific factors into two specific factors (with the exception of one variable.) We turned the resulting two-factor graph into a bifactor graph, and It passed a chi-squared test with a p-value of 0.32.

6 Future Research

Further research into making the output of *FTFC* more reliable and more stable is needed. It would also be useful to automate the use of chi-squared tests of the output models and to combine the strengths of the *schmid* and *FTFC* algorithms. The ultimate goal of the clustering is to find causal relations among the latent variables; when clusters have multiple latent common causes, some edges become unidentifiable (i.e. the parameters associated with the edge are not a function of the covariance matrix among the measured variables.) Computationally feasible necessary and sufficient conditions for identifiability of linear models are not known, and the possibility that the relations among the latents are non-linear complicates these issues further.

References

[1] Bartholomew, D.J., Steele, F., Moustaki, I., Galbraith, J.I.: The Analysis and Interpretation of Multivariate Data for Social Scientists. Texts in Statistical Science Series. Chapman & Hall/CRC (2002)

[2] Silva, R., Glymour, C., Scheines, R., Spirtes, P.: Learning the structure of latent linear structure models. Journal of Machine Learning Research 7, 191–246 (2006)

[3] Bollen, K.A.: Structural Equations with Latent Variables. Wiley-Interscience (1989)

[4] Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search, 2nd edn. Adaptive Computation and Machine Learning. The MIT Press (2001)

[5] Sullivant, S., Talaska, K., Draisma, J.: Trek Separation for Gaussian Graphical Models. Ann. Stat. 38(3), 1665–1685 (2010)

[6] Spirtes, P.: Calculation of Entailed Rank Constraints in Partially Non-Linear and Cyclic Models. In: Proceedings of the Twenty-Ninth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI 2013), pp. 606–615 (2013)

[7] Bollen, K., Ting, K.: Confirmatory tetrad analysis. Sociological Methodology 23, 147–175 (1993)

[8] Reise, S., Morizot, J., Hays, R.: The role of the bifactor model in resolving dimensionality issues in health outcomes measures. Quality of Life Research 16, 19–31 (2007)