Identification of Latent Variables From Their Footprint In Bayesian Network Residuals

Anonymous Author(s)
Affiliation
Address
email

Abstract

Graph-based causal discovery methods aim to capture conditional independencies consistent with the observed data, differentiating causal correlations from indirect or induced ones. Successful construction of graphical models of data depends, among others, on the assumption of observability. For partially observed data, graphical model structures may become arbitrarily incorrect, and effects implied by such models may be wrongly attributed, carry wrong magnitude, or mis-represent direction of correlation. Wide applicability and application of graphical models to increasingly less and less curated "big data" highlights the need for continued attention to the unobserved confounder problem.

We present a novel method that aims to control for the latent structure of the data by deriving proxies for the latent space from the residuals of the inferred graphical model. Under mild assumptions, our method improves structural inference. In addition, when the model is being used to predict outcomes, this method un-confounds the coefficients on the parents of the outcomes and leads to improved predictive performance when out-of-sample regime is very different from the training data. We show that such improvement of the predictive model is intrinsically capped and cannot be improved beyond a certain limit as compared to the confounded model. Furthermore, we propose an algorithm for computing a ceiling for the dimensionality of the latent space which may be useful in future approaches to the problem.

1 Introduction

Construction of graphical models (GMs) at its heart pursues two related objectives: accurate inference of the structure of conditional independencies and construction of predictive models for outcomes of interest with the purpose of estimating average causal effect (ACE) with respect to interventions (**?**, **?**). These two distinct goals mandate that certain identifiability conditions for both types of tasks be met. Of particular interest to this work is the condition of observability: namely, whether all of the relevant predictors have been observed. When this condition is not met, both accurate GMs and correct ACEs are hard to infer.

Hitherto, in the absence of full observability, literature has focused on addressing the subset of problems when ACE could be estimated reliably in the absence of this guarantee (such as when conditional exchangeability holds - e.g., **?**).

We aim to show that there exist circumstances when observability can be asymptotically achieved, and thus exchangeability ensured, even when the causal drivers of outcome are confounded by a number of latent variables. This can be achieved when the confounding is **pleiotropic** - when the latent variable affects a "large enough" number of variables, some driving an outcome of interest

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

Set	Meaning	Indexing
S	samples	$S_i, i \in \{1, \ldots, s\}$
V	observed predictor variables	$V_j, j \in \{1, \ldots, v\}$
U	unobserved predictor variables	$U_l, l \in \{1, \ldots, u\}$
O	outcomes (sinks)	$O_k, k \in \{1, \ldots, o\}$
D	$\{V, O\}$ - observable data	
D_u	$\{V, O, U\}$ - implied data	
θ	parameters	$\theta_i, i \in \{1, \ldots, t\}$
P^N	parents of variable N	$P^N_i, i \in \{1, \ldots, p\}$
C^N	children of variable N	$C^N_q, i \in \{1, \ldots, c\}$
G	graph over D	
G_u	graph over D_u	

Table 1: Notation

and others not (**?**). Notably, this objective cannot be achieved when confounding affects only the variables of interest and their causal parents (**?**).

Intuitively, presence of broad unobserved confounding gives rise to violation of conditional independence among the affected variables downstream from the latent confounder. Likelihood methods for GM construction aim to minimize unexplained variance for all variables in the network by accounting for conditional independencies in the data (**?**, **?**). Lack of observability of a causally important variable will induce dependencies among its descendants in the graph that cannot be fully ascribed to any single "heir" of the latent variable except by chance due to noise. Such unexplained interdependency results in model residuals correlated with the latent variable and not fully explained by any putative graph parents, and thus to inferred connectivity that's "excessive" as compared to the true network and in appearance of (near-)cliques (**?**).

Previously, methods have been proposed for inferring latent variables affecting GMs by the means of EM (as far back as **?**, **?**). However, for a large enough network local gradients do not provide a reliable guide, nor do they address the cardinality of the latent space. Methods for using near-cliques for detection of latent variables in directed acyclic graphs (DAGs) (**?**), including with gaussian graphical models (GGMs), have been proposed (**?**) that address both problems by analyzing near-cliques in DAGs. Most closely to our work, a method had been proposed for calculating latent variables "locally" in linear and "invertible continuous" networks, and relating such estimates to observed data to speed up structure search and enable discovery of hidden variables (**?**).

Here we propose an approach similar to that of **?** that takes advantage of global network residuals to, under some assumptions, asymptotically correctly infer the latent variables and un-confound predictors of outcome-only variables in the graph even when the latent variables confound these predictors. We begin with gaussian graphical models and generalize this approach to homeomorphic relationships, including ordinal data.

2 Background And Notation

We are going to concern ourselves with a factorized joint probability distribution over a set of observed and latent variables (see Table 1 for notation).

Assume that the joint distribution D (D_u) is factorized as a directed acyclic graph, G (G_u). We will consider individual conditional probabilities describing nodes and their parents, $P(V|parents(V), \theta)$, where θ refers to the parameters linking the parents of V to V. $\hat{\theta}$ will refer to an estimate of these parameters. We will furthermore assume that G is constructed subject to regularization and using unbiased estimators for $P(V|parents(V), \hat{\theta})$. We will further assume that D_u plus any given constraints are sufficient to infer the true graph up to markov equivalence. For convenience, we'll focus on the actual true graph's parameters, so that, using unbiased estimators, $E[\hat{\theta_m}|D_u] = \theta_m, \forall m$.

Mirroring D (or D_u), we will define a matrix R (or R_u) of the same dimensions - $s \times (v + o)$ (or $s \times (v + o + u)$)) - that captures the residuals of modeling every variable $N \in \{V, O, (U)\}$ via G (or G_u). In the linear case, these would be regular linear model residuals, but more generally we will consider probability scale residuals (PSR, **?**). That is, we define $R[i, j] = PSR(P(V_j|parents(V_j), \hat{\theta}_j)|D[i, j])$, the residuals of V_j given its graph parents. Notice that the use of probability-scale residuals allows us to define R and R_u for all ordinal variable types, up to rank-equivalence.

3 Algorithm

3.1 Gaussian Graphical Models (GGMs)

Recall that, for some $V_j \in \{V, U, O\}$, $P_k^{V_j}$ denotes the kth parent of V_j. For GGMs, we can write down a fragment of any DAG G as a linear equation:

$$V_j = \beta_{j0} + \beta_{j1} P_1^{V_j} + \cdots + \beta_{jp} P_p^{V_j} + \xi_j, \qquad \xi_j \sim \mathcal{N}(0, \sigma_j). \tag{1}$$

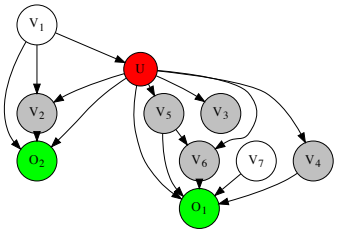

Figure 1: Graph G_u. U influences the outcomes, O, and a number of predictors, V, confounding many of the $V_j \to O_k$ relationships. Gray nodes are affected by U.

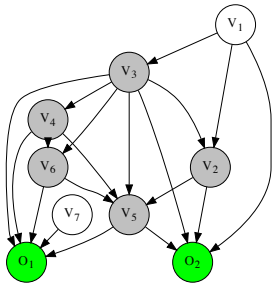

Figure 2: Graph G. With U latent, the graph adjusts, introducing spurious edges.

For example, consider O_1 in Figure 1. We can write:

$$O_1 = \beta_0 + \beta_6 V_6 + \beta_5 V_5 + \beta_4 V_4 + \beta_7 V_7 + \beta_u U + \xi_1, \qquad \xi_1 \sim \mathcal{N}(0, \sigma_{O_1}). \tag{2}$$

For any variable N that has parents in G_u, we can group variables in P^N into three subsets: $X_U \in \{P^U, C^U\}$, $X_{\not U} \notin \{P^U, C^U\}$, and the set U itself, and write down the following general form using matrix notation:

$$N = \beta_{N0} + B_U X_U + B_{\not U} X_{\not U} + \beta_U U + \xi_N, \qquad \xi_N \sim \mathcal{N}(0, \sigma_N). \tag{3}$$

Explicit dependence of N on U happens when $\beta_U \neq 0$.

Now consider G - the graph built over the variables $\{V, O\}$ excluding the latent space U. Note that if we deleted U and its edges from G_u without rebuilding the graph, Equation 3 from G_u would read:

$$N = \beta_{N0} + B_U X_U + B_{\not U} X_{\not U} + R_N + \xi_N, \qquad \xi_N \sim \mathcal{N}(0, \sigma_N). \tag{4}$$

The residual term R_N is simply equal to the direct contribution of U to N. The network G would have to adjust to the missingness of U (e.g., Figure 2 vs Figure 1). As a result, R_N will be partially substituted by other variables in $\{P^U, C^U\}$. Still, unless U is completely explained by $\{P^U, C^U\}$ (as described in **?**) and in the absence of regularization (when a high enough number of covariates may lead to such collinearity), R_N will not fully disappear in G. Hence, even after partially explaining the contribution of U to N by some of the parents of N in G,

$$R_N = \beta_0 + \beta_1 U + \xi_N. \tag{5}$$

	Variables			
Samples	V_{11}	V_{12}	$\cdots$	V_{1v}
	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	V_{s1}	V_{s2}	$\cdots$	V_{sv}

	Residuals			
Samples	R_{11}	R_{12}	$\cdots$	R_{1v}
	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	R_{s1}	R_{s2}	$\cdots$	R_{sv}

Table 2: The training data frame (left) implies a matching residual data frame (right) once the joint distribution of all variables is specified via a graph and its parameterization

Therefore, the columns in the residuals table corresponding to G (Table 2) that represent the parents and children of U will contain residuals collinear with U:

$$\begin{aligned} R_i &= \beta_{i0} + \beta_{i1}U + \xi_i \\ R_j &= \beta_{j0} + \beta_{j1}U + \xi_j \\ &\vdots \\ R_k &= \beta_{k0} + \beta_{k1}U + \xi_k. \end{aligned} \tag{6}$$

Rearranging and combining,

$$U = \beta_i^* R_i + \beta_j^* R_j + \cdots + \xi = BR + \xi. \tag{7}$$

Equation 7 tells us that, for graphical gaussian models, components of U are obtainable from linear combinations of residuals, or principal components (PCs) of the residual table R. In other words, U is identifiable by principal component analysis (PCA). Whether the residuals needed for this identification exist depends the *expansion property* as defined in **?**.

Note that this algorithm is the same, in the linear case, as that in **?** (equation 12) except insofar as we show the principal componets to be optimal for discovery of the whole latent space of a given DAG assuming "unconfounded" structure, and we therefore couple structure inference and EM for latent variable discovery as separate rather than interleaved steps (1).

4 Confounding of outcomes

Aside from the inference of the **exact** network - an interesting exercise that is probably futile in any practical application owing to the complexity of the inference problem - the most important utility of causal modeling is to propose suitable predictors, as well as predictors of predictors, for outcomes of interest. These "predictors of predictors" may have practical importance - for instance, when developing a drug, direct predictors of an outcome, say V_6 and O_1 from Figure 1), may not be druggable, but some of the mediators of treatment upstream of direct predictors, such as V_5, may turn out to be promising drug targets. Moreover, in the presence of latent confounding, coefficients of predictors of O_1 may be indeterminate[CITE HERNAN?]. For example, U induces correlation among V_3, V_4, V_5, and V_6 even when these variables are conditionally independent given U.

The extent of this latter problem can be quantified. Suppose we model O_1 without controlling for U (2):

$$O_1 = \beta_0 + \beta_3 V_3 + \beta_4 V_4 + \beta_5 V_5 + \beta_6 V_6 + \dots.$$

Let's set the coefficient of determination for the model

$$V_3 = \alpha_0 + \alpha_4 V_4 + \alpha_5 V_5 + \alpha_6 V_6 + \dots.$$

equal to ρ_3^2. Then the estimated variance of β_3 in the presence of collinearity can be related to that when collinearity is absent via the following formula (**?**):

$$var(\bar{\beta}_3) = var(\beta_3)\frac{1}{1-\rho_3^2} \propto \frac{1}{1-\rho_3^2}. \tag{8}$$

Formula 8 describes the *variance inflation factor* (VIF) of β_3. Note that $\lim_{\rho\to 1}\frac{1}{1-\rho^2} = \infty$, so even mild collinearity induced by latent variables can severely distort coefficient values and signs, and thus estimation of ACE. The method outlined above will reduce the VIFs of coefficients related to outcomes and thus make all *causal* statements relating to outcomes, such as calculation of ACE, more reliable, since by controlling for $\bar{U}$ - the estimate of U - in the network,

$$\lim_{(U-\bar{U})\to 0} var(\bar{\beta}_i) = var(\beta_i). \tag{9}$$

Can we hope to reach this limit? Consider an output O_j that is also a sink - meaning, it is known to have no children in the network (and therefore selection of predictors for this variable does not depend on the topology of the graph anywhere else) (NEED REFERENCE???). While it is difficult to describe the limit of error on the coefficients of the predictors of O_3, it is straightforward to put a ceiling on the improvement in the likelihood obtainable from modeling U and approximating G_u with $G_{\bar{u}}$. Suppose we eventually model U as a linear combination of a set of variables $X \subset V$, and denote by $X \setminus W$ the set difference: members of X not in W. Then for any outcome O_i predicted by

a set of variables W in the graph G and in truth predicted by the set $Z + U$, we can contrast three expressions (from G and $G_{\bar{U}}$ respectively):

$$\begin{aligned} O_i &= \beta_{i0} + B_W W(a) + \xi_i \qquad (a) \\ O_i &= \beta_{i0} + B_W W + B_{X \setminus W}(X \setminus W)) + \xi_i \qquad (b) \\ O_i^U &= \beta_{i0}^U + B_Z^U Z + B^U U + \xi_i \qquad (c). \end{aligned} \tag{10}$$

Model (a) is the model that was actually accepted, subject to regularization, in G. Model (b) is the "oracular" model of O_i that controls for U non-parsimoniously by controlling for all variables affected by U and not originally in the model. The third model, (c), is the ideal parsimonious model when U is known. We can compare the quality of these models by Bayesian Score, and the full score, in large sample sizes, can be approximated by BIC - the Bayesian Information Criterion (**?**. We assume that the third of these equations would have the lowest BIC (being the best model), and the first being the second highest, since we know that the set of variables $X \setminus W$ didn't make it into the first equation subject to regularization by BIC. Assuming n samples,

$$\begin{aligned} BIC(O_i = \beta_{i0} + B_W W + \xi_i) &= b_a \qquad (a) \\ BIC(O_i = \beta_{i0} + B_W W + B_{X \setminus W}(X \setminus W) + \xi_i) &= b_b = b_c + |X \setminus W| log(n) \qquad (b) \\ BIC(O_i^U = \beta_{i0}^U + B_Z^U Z + B^U U + \xi_i) &= b_c \qquad (c). \end{aligned} \tag{11}$$

The "oracular model" - model (b) - includes all of the true predictors of O_j. Therefore its score will be the same as that of the true model - model (c) - plus the BIC penalty, $log(n)$, for each extra term, minus the cost of having U in the true model (that is, the cardinality of the relevant part of the latent space). We know that the extra information carried by this model was not big enough to improve upon model (a), that is $b_a < b_c + k \log(n)$ for some k. Rearranging:

$$b_c - b_a > -k \log(n). \tag{12}$$

Any improvement in $G_{\bar{U}}$ owing to modeling of $\bar{U}$ cannot, therefore, exceed $k \log(n)$ logs, where $k = |X \setminus W| - |U|$: the information contained in the "oracular" model is smaller than its cost.

Although the available improvement in predictive power is also capped in some way, it is still important to aim for that limit. The reason is, correct inference of causality, especially in the presence of latent variables, is the only way to ensure transportability of models in real-world (heterogeneous-data) applications (see, e.g., **?**).

Up to here, our method is a generalization of work presented in **?**, where the authors show that under some assumptions the latent space can be learned exactly. However, we do not require that the observables be conditionally independent given the latent space and instead *generate* such independence by the use of causal network's residuals, which are, of course, conditionally independent of each other *given the graph and the latent space*. However, since the network among the observables is undefined in the beginning, the structure of the observable network must be learned at the same time as the structure of the latent space, which leads us to the iterative/variational bayes approach presented in 1.

4.1 Gaussian Graphical Models With Interactions

In the presence of interactions among variables in a GGM, equation 5 expressing the deviation of residuals from Gaussian noise may acquire higher-order terms due to interactions among the descendants of the latent space U:

$$R_N = \beta_0 + \beta_1 U + \beta_2 U^2 + \beta_2 U^3 + \ldots. \tag{13}$$

Assuming interactions up to kth power are present in the system being modeled, residuals for each variable may have up to k terms in the model matrix described by equation 13, and if interactions among variables in the latent space U also exist, the cardinality of the principal components of the residuals may far exceed the cardinality of the underlying latent space. Nevertheless, it may be possible to reconstruct a parsimonious basis vector by application of regularization and nonlinear approaches to latent variable modeling, such as autoencoders (**?**) or nonlinear PCA (e.g. using methods from **?**), as will be discussed below.

4.2 Generalization to nonlinear functions

We can show that linear PCA will suffice for a set of transformations broader than GGMs without interactions. In particular, we will focus on nonlinear but homeomorphic functions within the Generalized Additive Model (GAM) family. When talking about multiple inputs, we will require that the relationship of any output variable to any of the covariates in equation 5 is homeomorphic (invertible), and that equation 7 can be marginalized with respect to any right-hand-side variable as well as to the original left-hand side variable. For such class of transformations, mutual information between variables, such as between a single confounder U and some downstream variable N, is invariant (**?**). Therefore, residuals of any variable N will be rank-correlated to $rank(U)$ in a transformation-invariant way. Further, spearman rank-correlation, specifically, is defined as pearson correlation of ranks, and pearson correlation is a special case of mutual information for bivariate normal distribution. Therefore when talking about mutual information between ranks of arbitrarily distributed variables, we can use our results for the GGM case above.

Thus, equation 5 will apply here with some modifications:

$$rank(R_N) = \beta_0 + \beta_1 rank(U) + \xi_N. \tag{14}$$

Since a method has been published recently describing how to capture rank-equivalent residuals (aka probability-scale residuals, or PSR) for any ordinal variable (**?**), we can modify the equation 7 to reconstruct latent space up to rank-equivalence when interactions are absent from the network.

$$rank(U) = \frac{1}{\beta_i} rank(R_i) + \frac{1}{\beta_j} rank(R_j) + \cdots + \xi. \tag{15}$$

When U consists of multiple variables that are independent of each other, the relationship between N and U can be written down using the mutual information chain rule (**?**) and simplified taking advantage of mutual independence of the latent sources:

$$I(N;U) = I(N;U_1,U_2,\ldots,U_U) = \sum_{i=1}^{u} I(N;X_i|X_{i-1},\ldots,X_1) = \sum_{i=1}^{u} I(N;X_i). \tag{16}$$

If interactions among U are present, it may still be possible to approximate the latent space with a suitably regularized nonlinear basis, but we do not, at present, know of specific conditions when this may or may not work. Novel methods for encoding basis sets, such as nonlinear PCA (implemented in the accompanying code), autoencoders, and others, may be brought to bear to collapse the linearly independent basis down to non-linearly independent (i.e. in the mutual information sense) components.

While approximate inference of latent variables for GMs built over invertible functions had been noted in **?**, the above method gives a direct rank-linear approach leveraging the recently-proposed PSRs.

4.3 Generalization to categorical variables

In principle, PSRs can be extended to the case of non-ordinal categorical variables by modeling binary in/out of class label, deviance being correct/false. These models would lack the smooth gradient allowed by ranks and would probably converge far worse and offer more local minima for EM to get stuck in.

5 Implementation

Algorithm 1 below describes our approach to learning the latent space and can be viewed as a type of an expectation-maximization algorithm, possibly nested, if EM is used to learn the DAG at each step.

Algorithm 1: Learning $\bar{U}$ from structure residuals via EM

Data: The set of observed variables $\{V, O\}$
Result: Graph $G_{\bar{U}}(V, O, \bar{U})$
Construct $G = G(V, O)$
Compute $S_0 = BIC_G$
Estimate $\bar{U} = f(R)$
Construct $G_{\bar{U}} = G(V, O, \bar{U})$
Compute $S_{\bar{U}} = BIC(G_{\bar{U}})$
while $S_{\bar{U}} - S_0 > \epsilon$ **do**
  Set $S_0 = S_{\bar{U}}$
  Calculate $R_{\bar{U}}$:
  Set $\bar{U}$ to arbitrary constant values
  foreach *child node* $C \in G_{\bar{U}}, C \notin \bar{U}$ **do**
    Set parents to training data
    $\bar{C} = C|parents(C)$
    Set $R_C = PSR(\bar{C}, C)$
  end
  Estimate $\bar{U} = f(R_{\bar{U}})$
  Construct $G_{\bar{U}} = G(V, O, \bar{U})$
  Compute $S_{\bar{U}} = BIC(G_{\bar{U}})$
end

Algorithm 2: Inferring linearly optimal $\bar{U}$ and assessing its cardinality by permutations

Data: The set of residuals $R_{\bar{U}}$ from modeling D with $G_{\bar{U}}$
Result: Linear approximation to $\bar{U}$
Set significance threshold α (e.g. $\alpha = 0.05$)
Learn $\bar{U} = PCA(R_{\bar{U}})$
Calculate column-wise variance explained V_E^0 for $\bar{U}^*$
Set $V_E^s = 0 \times rank(R_{\bar{U}})$, matrix of variances explained by shuffling
while $se(\bar{U}) > \epsilon$ **do**
  $R_{\bar{U}}^* = shuffle(R_{\bar{U}})$ (column-wise)
  Calculate $\bar{U}^* = PCA(R_{\bar{U}}^*)$
  Calculate V_E^s for $\bar{U}^*$
  Concatenate row-wise: $V_E^s = \{V_E^s; V_E^s\}$
  Fit $B(i)$ beta distributions to each column i of V_E
  For each column i of $\bar{U}$, calculate:

$$P(V_E^0(i)|V_E^s(i)) = \lim_{|V_E^s(i)|\to\infty} \frac{|V_E^s(i) > V_E^0(i)|}{|V_E^s(i)|} \approx 1 - \int_{-\infty}^{V_E^0(i)} PDF(B_i)$$

end
$P(V_E^0(i)|V_E^s(i)) = P(V_E^0(i) \sim V_E^s(i)) \times rank(V_E)$
Drop $V_E^0(i)$ for which $P(V_E^0(i) \sim V_E^s(i)) > \alpha$

How do we learn $\bar{U} = f(R_{\bar{U}})$? In the linear case, we can use PCA, as described above, and in the non-linear case, we can use non-linear PCA, autoencoders, or other methods, as alluded to above as well. However, the linear case provides a useful constraint on dimensionality, and this constraint can be derived quickly. A useful notion of the ceiling constraint on the linear latent space dimensionality can be found in **?**. From a practical standpoint, the dimensionality can be even tighter, and we propose a permutation-test-based method for inferring $ceiling(|U|)$ in Algorithm 2.

The integral should converge faster than the count of times variance explained by U^* on true residuals exceeds that obtained from shuffled residuals, but the permutation test approach of PCA cardinality is also workable, albeit with more iterations. Note that it is necessary to correct for the number of tests performed, and that we use Bonferroni correction as a simple and conservative stand-in. Alternatively, networks built using structural priors of the form proposed in **?** may not need to perform this step.

We observe that our algorithm 2 is very different from but similar in spirit to that proposed in **?**. If we consider $Y = f(X)$, where Y are all DAG outputs and X are all inputs, while f is the suitably parameterized DAG, PCA over residuals (linear or not) normalized by PCA over shuffled residuals provides a measure of "compressibility" of residual space. In other words, while the specifics of the implementaiton are different, in practice we propose a minimum description length algorithm for detecting the latent variables so that the residual space is no longer compressible.

6 Numerical Demonstration

To illustrate the algorithms described in the previous sections we generated synthetic data from the network shown in Figure 3 where two variables V_1 and V_2 drive an outcome Z. Two confounders U_1 and U_2 affect both the drivers and the outcome as well as many additional variables that do not affect the outcome Z. The coefficient values in the network were chosen making sure that faithfullness is fullfilled and that the structure and coefficients are approximately recovered when all variables are observed.

The underlying network inference needed for the algorithm was implemented by bootstrapping the data and running the R package bnlearn **?** on each bootstrap. The resulting ensemble of networks can be combined to obtain a consensus network where only the most confident edges are kept. Similarly, the estimated coefficients can be obtained by averaging the coefficients over bootstraps.

For this example, the consensus network created with edges with confidence larger than 40% recover the true structure and the root mean square error (RMSE) in the coefficient estimates was 0.06 (not shown). This represents a lower bound on the error that we can expect to obtain under perfect reconstruction of the latent space.

When the confounders are unobserved the reconstruction of the network introduces many false edges and results in a RMSE of over four times large. Figure 4 shows the reconstructed network in bnlearn where the red edges are the true edges between V_1 and V_2 and the outcome Z.

We ran algorithms 1 and 2 for 10 iterations using PCA to reconstruction the latent space form the residuals and assuming the latent variables are source nodes. We then tracked the latent variable reconstruction as well as the coefficient's errors. Figure 6 shows the adjusted R^2 between each of the true latent variables and the prediction obtained from the estimated latent space as a function

of the iterations of the algorithm. The lines and error bands are calculated from a local polynomial regression model. The estimated latent space is predictive of both latent variables and the iterative procedure improve the R^2 with respect to U_1 from 0.57 to 0.61 converging in about 5 iterations.

Figure 7 shows the total error in the coefficients between all variables in the networks and the outcome Z (RMSE) as well as the error in the coefficients of the true drivers of Z V_1 and V_2. Both errors converge after the first iteration to an error level of the same magniture as the error when all variables are observed (dashed lines).

Figure 5 shows that final inferred network at iteration 10. The number of edges arriving to the outcome was reduced considerable with respect to the false edges without inferring latent variables (Figure 4). In addition, the coefficients connecting V1 and V2 to the outcome is now closer coefficients. This represent an improvent in ACE estimation.

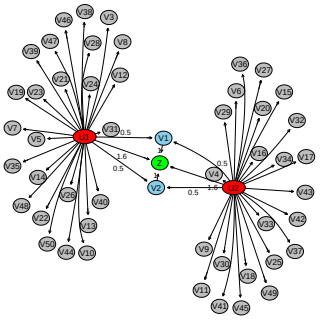

Figure 3: True network.

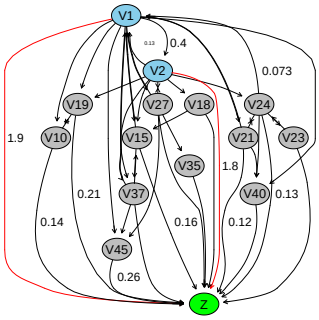

Figure 4: Estimated network when U_1 and U_2 are unobserved.

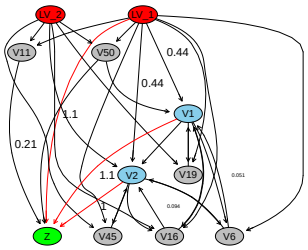

Figure 5: Estimated network at the last iteration of algorithm 1.

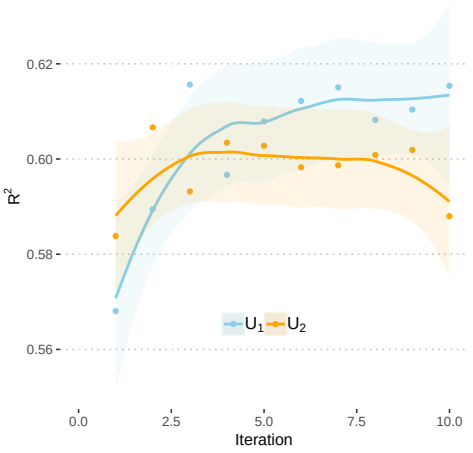

Figure 6: R^2 in the prediction of the latent variable from the selected principal components.

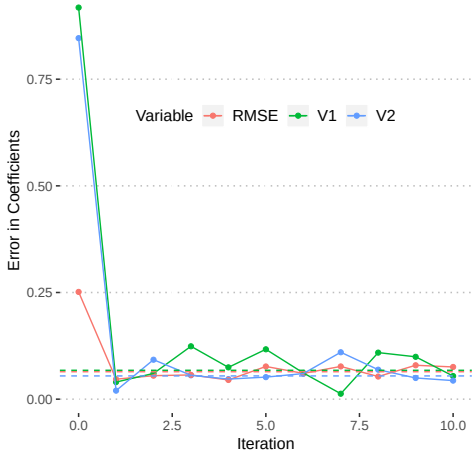

Figure 7: Error in coefficients as a function of the iterations.

7 Conclusions and Future Directions

In this work we present a method for describing the latent variable space that is optimal under linearity, up to rank-linearity. When we cannot provide such guarantees, the method will still identify the terms of the model matrix of the latent space, including any interactions, for models in the GGM family, making it possible to attempt to infer the original (compact) latent space by non-linear modeling and regularization. The method does not place *a priori* constraints on the number of latent variables, and will infer the upper bound on this dimensionality automatically. This method is a generalization

of prior work, both in terms of the global treatment of the residual space, and in terms of stronger statements of applicability in cases when probability-scale residuals are applicable.

In the future, we hope to assess the compressibility of the latent space with deep-learning models, using linear PCA to set the ceiling for the cardinality of the latent space, and to explore the applicability of the resulting hybrid "deep causal" model to epidemiological and biological problems in which it is highly desirable to retain original data features for explainability but is equally necessary to introduce latent variables to account for unobserved pleiotropic confounding. As noted above, calculation of ACE in epidemiological applications is one specific example of this type of an application. Applications of such "deep causal networks" outside these domains - such as in computer vision coupled with sensor data - which should arise in the nascent internet-of-things paradigm - are also possible **?**.