Causal Discovery on Dependent Binary Data

Alex Chen
Department of Statistics and Data Science
University of California, Los Angeles

Qing Zhou
Department of Statistics and Data Science
University of California, Los Angeles

Abstract

The assumption of independence between observations (units) in a dataset is prevalent across various methodologies for learning causal graphical models. However, this assumption often finds itself in conflict with real-world data, posing challenges to accurate structure learning. We propose a decorrelation-based approach for causal graph learning on dependent binary data, where the local conditional distribution is defined by a latent utility model with dependent errors across units. We develop a pairwise maximum likelihood method to estimate the covariance matrix for the dependence among the units. Then, leveraging the estimated covariance matrix, we develop an EM-like iterative algorithm to generate and decorrelate samples of the latent utility variables, which serve as decorrelated data. Any standard causal discovery method can be applied on the decorrelated data to learn the underlying causal graph. We demonstrate that the proposed decorrelation approach significantly improves the accuracy in causal graph learning, through numerical experiments on both synthetic and real-world datasets.

1 INTRODUCTION

Causal discovery methods designed for observational data address the challenges in causal inference under non-experimental settings across many applied domains. Directed acyclic graphs (DAGs), where nodes correspond to random variables, are a class of graphical models for modeling causal relationships among a set of variables. Suppose there are p variables and let $PA_j \subset [p] := \{1, \ldots, p\}$ be the parent set of node j. Consider a data matrix $X \in \mathbb{R}^{n \times p}$, where each row $x_i \in \mathbb{R}^p$ corresponds to one of the n units in the data. Let X_j be the jth column of X, which is generated by a structure equation model (SEM) as shown in Equation (1) with an independent error vector ε_j:

$$X_j = f_j(X_{PA_j}, \varepsilon_j), \quad \varepsilon_j = (\varepsilon_{1j}, ..., \varepsilon_{nj}) \overset{\text{iid}}{\sim} \mathbb{P}, \quad (1)$$

for $j \in [p]$, where $X_{PA_j} = \{X_i : i \in PA_j\}$. Note that the error terms ε_{ij}, $i \in [n]$ are independent and identically distributed with some distribution $\mathbb{P}$.

A main assumption of this model is the joint independence among the n units of the data matrix X due to the independence of the errors. More specifically, under the i.i.d assumption, the rows (units) $x_1, \ldots, x_n$ of the data matrix X are independent and thus the $n \times n$ covariance matrix among the n units is diagonal. It is important to distinguish the between-unit covariance matrix from the $p \times p$ covariance matrix among the columns $X_1, \ldots, X_p$, which is typically dense. In this work, the primary motivation is to consider potential dependence among the rows $x_1, \ldots, x_n$ in the learning of the underlying DAG.

Under the i.i.d data assumption, many causal structure estimation methods have been developed. There are three main types of structure learning: constraint-based, score-based, and hybrid methods. One of the primary constraint-based methods is the PC Algorithm (Spirtes et al., 2000), which uses conditional independence tests to remove edges from an initial complete undirected graph and then orients the edges based on the Meek's rules (Meek, 1995). Variants include PC-select by Bühlmann et al. (2010) for causal structure estimation on one variable, PC-stable from Colombo et al. (2014) for order-independence, FCI by Spirtes et al. (2000) for latent confounders, and rankPC by Harris and Drton (2013) for non-normal data. Score-based methods search over a certain graph space to find a graph that optimizes a scoring function such as BIC (E. Schwarz, 1978) or minimum-description length (Roos, 2017). Popular examples include the

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

Greedy Equivalence Search (GES) (Chickering, 2002) and the Greedy Hill Climbing (Gámez et al., 2011). Other variants include Fast Greedy Equivalence Search (FGES) (Ramsey et al., 2017) and regularized likelihood maximization approaches (Aragam and Zhou, 2015; Gu et al., 2019; Fu and Zhou, 2013). Hybrid methods combine the two approaches, constraint and score-based learning, such as MMHC (Max-min Hill climbing) (Tsamardinos et al., 2006) and GFCI (Ogarrio et al., 2016). A more complete overview of existing methods for causal learning is provided by Glymour et al. (2019) and Nogueira et al. (2022).

1.1 Motivation and contributions

Real-world data, however, often deviates from the i.i.d assumption. In social and behavioral sciences, the "non-iidness", also termed as *couplings* (Cao, 2015), manifests in various forms. Individuals often share characteristics within social groups or families. Consequently, data points become interdependent, breaking the i.i.d assumption. In single-cell RNA sequencing (scRNA-seq), we seek to uncover gene regulatory networks (GRNs) which can be conceptualized as causal networks among genes. Cells in scRNA-seq data have inherent dependence due to various spatial and temporal associations among the cells, such as cell differentiation from the same population. The dependence among units in such data clearly violates the i.i.d. assumptions used in traditional causal discovery methods.

Rather than adapting a specific structure learning method to handle data dependence, we propose a general approach to transform dependent data into an independent surrogate, so that many existing structure learning methods can be applied to the independent surrogate data. More specifically, the main contributions of this work are as follows. First, we propose a DAG model for dependent binary data based on a latent utility model, where errors ε_j across units are modeled by an unknown covariance matrix Σ. Then, we develop a pairwise maximum likelihood method to estimate the covariance among the units the data. Lastly, given the estimated Σ, we develop an EM algorithm to generate surrogate independent data, which are samples of the latent utility variables in our model. We demonstrate, with both synthetic and real-world datasets, that structure learning on the surrogate data is much more accurate than on the original dependent data using state-of-the-art methods.

1.2 Related works

In the context of DAG-based causal inference, there are only a few very recent methods that take into account the potential dependence among units. Bhattacharya et al. (2020) developed a method for causal inference under *partial interference* (Hudgens and Halloran, 2008). They assume a known causal DAG among the variables and then estimate causal effects under dependence among the units x_i. In contrast, we do not assume knowledge of a DAG structure and in fact our primary goal is to learn the underlying causal DAG. Li et al. (2021) proposed a linear Gaussian DAG on network data, by introducing dependent exogenous variables $\varepsilon_j = (\varepsilon_{1j}, \ldots, \varepsilon_{nj})$:

$$X_j = \sum_{k \in PA_j} \beta_{kj} X_k + \varepsilon_j, \quad \varepsilon_j \sim \mathcal{N}_n(0, \Sigma), \tag{2}$$

where β_{kj} is the causal effect of X_k on X_j and the covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ is positive definite. They suggest estimating the precision matrix $\Theta = \Sigma^{-1}$ and using the Cholesky factor L of Θ as a means to decorrelate the data matrix X, that is, the decorrelated data $\widetilde{X} = L^\intercal X$. This approach demonstrated promising results in addressing data dependence in DAG learning, however, there are a few intrinsic limitations. First, their decorrelation approach is not applicable to discrete data, since $L^\intercal X$ is in general outside of the discrete data domain and this transformation has no clear interpretation. Second, Li et al. (2021) assume that the support of Θ is restricted to a *known* network (graph) among the n units, which is somewhat limited for many forms of dependent data. The wide use and availability of discrete data prompts a novel decorrelation approach. In this work, we introduce a different data-generating mechanism for dependent discrete data and develop an associated decorrelation method for improving causal graph estimation. We do not assume a known network among the units. Our method generates continuous proxy data in the process of removing cross-units dependence, on which a standard structure learning algorithm can be applied to estimate the underlying DAG. As shown by our numerical results, this approach substantially improves the structure learning accuracy and provides a much better model fit to the data.

2 A DAG MODEL FOR DEPENDENT DATA

To generalize the SEM in Equation (2) to binary variables $x_{ij} \in \{0, 1\}$, we use a probit regression model for $[x_{ij} \mid x_{ik}, k \in PA_j]$ under a latent-variable formulation. We introduce a set of auxiliary latent variables z_{ij} for $j \in [p]$ and $i \in [n]$. The binary value of x_{ij} is determined by the latent variable z_{ij}:

$$z_{ij} = \sum_{k \in PA_j} \beta_{kj} x_{ik} + \varepsilon_{ij} = x_i \beta_j + \varepsilon_{ij}, \quad (3)$$

$$x_{ij} = I(z_{ij} > 0), \quad (4)$$

for $i \in [n]$ and $j \in [p]$, where $\beta_j = (\beta_{1j}, \ldots, \beta_{pj}) \in \mathbb{R}^p$ such that $\text{supp}(\beta_j) = PA_j$. Under this formulation, z_{ij} is regarded as the utility for $x_{ij} = 1$, while the utility for $x_{ij} = 0$ is the baseline (zero). This approach allows us to accommodate dependence among discrete units by assuming

$$\varepsilon_j = (\varepsilon_{1j}, \ldots, \varepsilon_{nj}) \sim \mathcal{N}_n(0, \Sigma), \quad \text{for all } j. \quad (5)$$

Due to an identifiability issue in the model defined by Equations (3) and (4), we impose the constraint $\text{diag}(\Sigma) = 1$. Without this constraint, the model becomes over-parameterized, and Σ and β are not identifiable. The dependence between the exogenous variables $\varepsilon_i := (\varepsilon_{ij} : j \in [p])$ and ε_k causes the dependence between the two units x_i and x_k, $i, k \in [n]$. This assumption aligns with the fact that the exogenous variables are the source of randomness in the general SEM (1).

Let $Z = (z_{ij})_{n \times p}$ and $Z_j = (z_{1j}, \ldots, z_{nj})$ be the jth column. Plugging $x_{ik} = I(z_{ik} > 0)$, for all $i = 1, \ldots, n$, in Equation (3), we have

$$Z_j = \sum_{k \in PA_j} \beta_{kj} I(Z_k > 0) + \varepsilon_j, \quad j \in [p]. \quad (6)$$

This defines an SEM for $Z = (Z_1, \ldots, Z_p)$ with a DAG $\mathcal{G}(Z)$.

Lemma 2.1. *Under the latent utility model in Equations* (3) *and* (4)*, the DAG $\mathcal{G}(X)$ among the observed discrete variables X is identical to the DAG $\mathcal{G}(Z)$ among the latent variables Z.*

Thus, we develop a method to impute and decorrelate the latent auxiliary data $Z = (Z_j)$ to work with continuous data rather than discrete. Then, we apply a standard structure learning method on decorrelated data to learn the underlying DAG. This is the high-level idea of our method. The key complexity of our model is the interplay between two types of dependence, one over all variables and one over all units. The causal relations over the variables $X_1, \ldots, X_p$ are modeled through a DAG (3) and (4) and the relationship among different units are modeled through a joint distribution over the exogenous variables $\varepsilon_1, \ldots, \varepsilon_n$ (5).

Remark. Our model does differ from the latent-thresholding model (Spirtes, 1996; Silva, 2005) for discrete variables, where one often assumes a Gaussian DAG for $Z_1, \ldots, Z_p$ and each discrete variable is defined by thresholding, i.e. $X_j = I(Z_j > \tau_j)$.

In our view, both models are reasonable. Using gene regulatory networks as an example, our model postulates a causal network over the activation/suppression (discrete status) of the genes, while the latent-thresholding model assumes a causal network over the exact expression level (continuous measure) of the genes. When the original continuous data is noisy, which is common in many applications, our model could be more robust.

3 METHODS

Our main idea is to learn the latent variables Z_j. Given the underlying latent data $Z_j, j \in [p]$, the causal graph we estimate among Z will be identical to the causal graph among X according to Lemma 2.1. Since Z is continuous, we can apply the Cholesky factor of an estimated precision matrix among the units to remove data dependence. Our methodology begins by estimating the covariance Σ via a pairwise likelihood approach. Given $\widehat{\Sigma}$, our algorithm iterates between approximating the latent data Z_j and estimation of the parameters $\beta_j, j \in [p]$ through an EM-algorithm. The Cholesky factor of the estimated precision matrix can then be used to remove the dependence among the latent data to use for DAG learning by any standard causal structure estimation method.

3.1 Covariance estimation

Under classical i.i.d. settings, covariance estimation among discrete variables is a well-studied topic in statistical modeling. Fan et al. (2017) developed a rank based estimator using Kendall's tau to calculate correlations assuming i.i.d samples. Olsson (1979) uses a bi-variate normal cdf method to estimate the correlation ρ_{ij} between two discrete variables and also assumes i.i.d samples to obtain an accurate estimate of the correlation. Cui et al. (2016) uses Gibbs sampling on rank-based data to sample and average correlation matrices from an inverse-Wishart distribution. However, their method does not utilize a prior that can specify sparsity in the covariance matrix. The challenges posed by our model have spurred the necessity to develop a novel covariance estimation approach for our problem, distinguished by several key departures from existing literature. First, our model introduces dependence into discrete data through latent background variables, ε_j, compared to the aforementioned methods where data is assumed to be fully observed. Additionally, since the distribution of X_j depends on its parent variables, we do not have independence among the data involved in the likelihood of Σ. Furthermore, our method allows for the introduction of sparsity based on potential domain expertise, enhancing practical utility.

Jointly estimating the $n \times n$ covariance matrix Σ in Equation (5) is very challenging because the error variables ε_{ij} are unobserved. Therefore, we propose a pairwise MLE method to estimate Σ. Because $\Sigma_{ii} = 1$ for $i \in [n]$, our pairwise method is applied to estimate each correlation ρ_{ij}, for $i, j \in [n]$.

Without loss of generality, let us consider the estimation of ρ_{12}, the correlation between the first two units. There are four possible outcomes for (x_{1j}, x_{2j}) for any $j \in [p]$, i.e. (0,0), (0,1), (1,0), (1,1). Using Equation (4),

$$(x_{1j}, x_{2j}) = \left(I\left(\varepsilon_{1j} > -x_1\beta_j\right), I\left(\varepsilon_{2j} > -x_2\beta_j\right)\right). \quad (7)$$

According to Equation (5), with $\text{diag}(\Sigma) = 1$, the distribution of the two error variables $(\varepsilon_{1j}, \varepsilon_{2j})$, is a bivariate normal with an unknown parameter ρ_{12},

$$\begin{pmatrix}\varepsilon_{1j}\\ \varepsilon_{2j}\end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix}0\\0\end{pmatrix}, \quad \begin{pmatrix}1 & \rho_{12}\\ \rho_{12} & 1\end{pmatrix}\right] \text{ for all } j \in [p]. \quad (8)$$

Given β_j, we can use Equation (7) to find the probability mass function of (x_{1j}, x_{2j}) through the CDF of a bivariate Gaussian:

$$P(x_{1j}, x_{2j} | D_j^{1,2}, \rho_{12}) = \iint_{D_j^{1,2}} \phi\left(u_1, u_2 | \rho_{12}\right) du_1 du_2,$$

where $D_j^{1,2} \subset \mathbb{R}^2$ is the domain for the integral and ϕ is the pdf for the bivariate normal in Equation (8). For example, if $(x_{1j}, x_{2j}) = (1, 1)$ then the domain $D_j^{1,2} = (-x_2\beta_j, \infty) \times (-x_2\beta_j, \infty)$, where $\text{supp}(\beta_j) = PA_j$. The likelihood of ρ_{12} given all p pairs (x_{1j}, x_{2j}), $j \in [p]$ is

$$L(\rho_{12} | x_1, x_2) = \prod_{j=1}^{p} P(x_{1j}, x_{2j} | D_j^{1,2}, \rho_{12}). \quad (9)$$

For any pair of units $(a, b) \in [n] \times [n]$, our estimate $\widehat{\rho}_{ab}$ is the maximizer of the likelihood function $L(\rho_{ab} | x_a, x_b)$, which can be found easily using a univariate numerical optimization method. Figure 4 in Supplementary Material 7.1 illustrates the accuracy of our estimate compared to the true correlation.

The covariance estimation method relies on an initial estimate of β, necessitating initial parent estimates $\widehat{PA}_j$ for $j \in [p]$. We apply the Max-Min Hill Climbing (MMHC) algorithm (Tsamardinos et al., 2006) to estimate a DAG and then use logistic regression (See Supplementary Material 7.6) to estimate β_j from PA_j. Our covariance estimation is robust to the initial estimates, as suggested by the numerical results in Supplementary Material 7.6.

For practical results, we restrict our attention to covariance matrices with a block-diagonal structure. Block structure often occurs in data with clusters of dependent units. We do not need to assume any sparse structure within each block. We apply the pairwise MLE to each ρ_{ij} between two units in the same block and apply a regularization step to ensure the positive definiteness of $\widehat{\Sigma}$. We obtain a positive semi-definite matrix by truncating the negative eigenvalues λ_i of our estimated covariance matrix, i.e. $\lambda_i' = \max\{0, \lambda_i\}$ and then re-scale the off-diagonal elements by a factor < 1 (e.g. 0.9).

3.2 Latent Data Recovery and Decorrelation

Given the estimated covariance matrix $\widehat{\Sigma}$, we develop an EM algorithm (Dempster et al., 1977) to recover the latent data and update parameter estimates. In the E-step, given β, we impute the latent data, Z, utilizing draws of ε from a Truncated-Normal distribution. In the M-step, decorrelation is applied to the latent data when maximizing the complete data log-likelihood to update β.

With the estimated covariance matrix $\widehat{\Sigma}$ from Section 3.1, the distribution of the error term ε_j is $\mathcal{N}_n(0, \widehat{\Sigma})$. However, the data vector X_j is determined by the relationship between ε_j and $-X\beta_j$ as specified in Equation (7). Thus, given X and the current parameter β, ϵ_j is distributed

$$\varepsilon_j \mid X, \beta_j \sim \mathcal{N}_T(0, \widehat{\Sigma}), \quad (10)$$

which is truncated at $-X\beta_j$. More specifically, the truncation at $-X\beta_j$ introduces a total of n constraints in the truncated normal distribution in the form of

$$\begin{aligned}\varepsilon_{ij} &> -x_i\beta_j \quad \text{if } x_{ij} = 1\\ \varepsilon_{ij} &\leq -x_i\beta_j \quad \text{if } x_{ij} = 0.\end{aligned}$$

We leverage the block structure of the covariance matrix to draw ε_j by simulating a multivariate truncated normal distribution within each block using a Gibbs sampler. The average of the N draws yields an approximate expectation $\mathbb{E}(\varepsilon_j \mid X, \beta_j)$. We can then reconstruct latent auxiliary data Z via Equation (3).

After reconstructing latent data Z_j, $j \in [p]$, we develop a decorrelation method using the estimated covariance matrix $\widehat{\Sigma}$ in the M-step to update each β_j. Let $L^\intercal$ be the Cholesky factor of $\Theta := \Sigma^{-1}$ such that $\Theta = LL^\intercal$. Given $\varepsilon_j \sim \mathcal{N}_n(0, \Sigma)$, applying the Cholesky factor $L^\intercal$ results in $L^\intercal\varepsilon_j \sim \mathcal{N}_n(0, I_n)$, a vector of independent Gaussian variables. Accordingly, for each $j \in [p]$, decorrelation of the latent data Z_j can be performed by applying $\widehat{L}^\intercal$ to Z_j such that

$$\widehat{L}^\intercal Z_j = \widehat{L}^\intercal X\beta_j + \widehat{L}^\intercal\varepsilon_j, \quad (11)$$

where $\widehat{L}^\intercal$ is the Cholesky factor of $\widehat{\Sigma}^{-1}$. This simplifies to a regression of $\widehat{L}^\intercal Z_j$ onto $\widehat{L}^\intercal X$ to estimate β_j,

but restricting the support of β_j to $\widehat{PA}_j$. This process maximizes the expectation of the complete data log-likelihood $\log p(X, Z \mid \beta)$ to update β_j. Ridge regression is used in order to avoid overfit to potentially inaccurate draws of ε.

The iterative process outlined in Algorithm 1 consists of recovering the latent data Z, decorrelating Z, and estimating β. In Supplementary Material 7.4, Figure 7, a single example run of this algorithm is shown. Notably, the β values begin to converge to a stable point after approximately 5 iterations.

Algorithm 1 EM-based decorrelation algorithm

Given an estimated Cholesky factor $\widehat{L}$ from $\widehat{\Sigma}$ and initial estimated graph $\widehat{\mathcal{G}}$, iterate between the following steps until a stop criterion is met:

1. Average draws of $\widehat{\varepsilon}$ according to Equation (10).
2. Reconstruct latent variable data $\widehat{Z} = X\widehat{\beta} + \widehat{\varepsilon}$.
3. Decorrelate latent data $\widehat{L}^\intercal \widehat{Z} = \widehat{L}^\intercal X\widehat{\beta} + \widehat{L}\widehat{\varepsilon}$.
4. Perform p ridge regressions of $\widehat{L}^\intercal \widehat{Z}_j$ on $\widehat{L}^\intercal X_j$ for each $j \in [p]$ to obtain $\widehat{\beta}$ where the support of X_j is the parents of variable j.

3.3 Structure Learning

Lemma 2.1 suggests that we may use the imputed Z for causal discovery of X. Algorithm 1 produces decorrelated data, $\widehat{L}^\intercal Z$, at each iteration, where unit-dependence has been largely removed. With close-to-independent data, conventional structure learning methods can be employed to estimate a DAG among Z. Due to Monte Carlo simulation of ε, the $\widehat{Z}$ is not an exact expectation. We mitigate this inaccuracy by using a consensus or average across multiple imputations of Z in DAG learning.

Since we are using observational data, in general, one can only learn an equivalence class represented by a CPDAG. We employ two approaches to generate the final estimated CPDAG. In the first approach, we run a standard structure learning algorithm with default parameters on $M = 10$ decorrelated datasets imputed at different iterations of Algorithm 1. We accept an edge into our final graph estimate if at least half of the estimated CPDAGs agree. We call this the *consensus approach*. In the second approach, we average all M decorrelated datasets and then run a standard structure learning algorithm on the averaged dataset, which we call the *average approach*. From a computation time perspective, *average approach* necessitates only a single DAG structure estimate while the *consensus approach* requires M estimates. We compare these two approaches to a *baseline approach* that uses a standard structure learning algorithm directly to the discrete dependent data.

4 EXPERIMENTAL RESULTS

For simulations, we use either random DAGs or those from the *bnlearn* repository. Given a DAG structure, edge weights were uniformly sampled from the interval $[-0.9, -0.6] \cup [0.6, 0.9]$. We then sampled noise variables, ε, according to Equation (5) and a specified covariance structure detailed below. We simulate the continuous auxiliary variables Z_j and discretize Z_j using Equation (4) to obtain our observed data X_j. The process is repeated according to a topological ordering until we have discrete data X_j for all $j \in [p]$.

Three covariance structures (of Σ) among the n units were employed based on a block-diagonal structure. Within each cluster (block), the units were correlated following one of the structures where block sizes range from 10 to 15 in simulated data: (i) *Equal* covariance implies fully-connected units in a cluster with $\Sigma_{ij} = \theta$ if $i \neq j$ where $\theta \sim \mathcal{U}(0.4, 0.7)$, (ii) *Toeplitz* covariance means units are connected in a Markov Chain, with $\Sigma_{ij} = \theta^{|i-j|/5}$ where $\theta \sim \mathcal{U}(0.1, 0.25)$, and (iii) *Mixed* covariance means each block is randomly *Toeplitz* or *Equal*. To validate the covariance estimation accuracy, we constructed 10 random DAGs for each pair of parameters $n \in \{100, 500\}$ units and $p \in \{100, 1000\}$ variables and simulated data according to aforementioned data generating process using a mixed covariance structure for Σ. After obtaining the initial estimates of $\widehat{\beta}$, we apply the pairwise MLE to estimate each ρ_{ij} within the known block structure. Then, we calculated the RMSE of $\widehat{\Sigma}$ with respect to the true covariance matrix Σ^* for each random DAG. As reported in Supplementary Material 7.2, our covariance estimates have low RMSE values ≤ 0.15 for $p = 100$ and ≤ 0.06 for $p = 1000$.

Using the estimated covariance matrix $\widehat{\Sigma}$, Algorithm 1 is used to remove the dependence from the data. Figure 6 in Supplementary Material 7.7 compares the correlations in a de-correlated dataset against the correlations in the underlying covariance matrix Σ, showing that the de-correlated dataset has significantly less row-wise correlation than the original data. As mentioned in Section 3.3, we took the approximately decorrelated datasets and used two approaches, *consensus* and *average*. To ensure the results of our algorithm were consistent across different structure learning methods, we chose three different methods, MMHC (Tsamardinos et al., 2006), PC, and Copula-PC (Cui et al., 2016). Copula-PC is an extension of the traditional PC algorithm for non-Gaussian data using copulas. Specifically,

we use the discrete versions of each learning method on the dependent discrete data and the continuous versions on the decorrelated data. Then, we compared the estimated CPDAG against the underlying true CPDAG using an F-1 score.

Eight real DAGs (p variables and s edges) were sourced from the bnlearn repository (Scutari, 2014): **Hailfinder** ($p = 56, s = 66$), **Hepar2** ($p = 70, s = 123$), **Win95pts** ($p = 76, s = 112$), **Munin1** ($p = 186, s = 273$), **Andes** ($p = 223, s = 338$), **Pigs** ($p = 441, s = 592$), **Diabetes** ($p = 413, s = 602$), **Link** ($p = 724, s = 1125$). These DAGs cover various domains, such as severe hail forecasting in northeastern Colorado from **Hailfinder** and the use of an electromyography machine in medical diagnostics in **Munin**. We used the aforementioned parameters for each simulation: $n \in \{100, 500\}$ units, edge weights β were uniformly sampled from $[-0.9, -0.6] \cup [0.6, 0.9]$ and the covariance matrix Σ^* was sampled according to the mixed structure with block sizes ranging from 10 to 15. With these parameters, we simulated data for each DAG, repeating the process 10 times.

Figure 1 reports an F-1 score of estimated CPDAGs to the true CPDAG after using each approach. Red boxplots serve as the *baseline* and the green and blue boxplots correspond to the *average* and *consensus* approaches, respectively. Our Consensus method demonstrates on average 20% improvement to the F-1 score compared to the baseline method, with only three simulations displaying exceptions.

Our real DAG results concur with additional experiments under the same settings as above for random DAGs with $n \in \{100, 500\}$ units, $p \in \{100, 1000\}$ variables and $2p$ edges. Figure 8 in Supplementary Material 7.5 demonstrates between 13-34% improvement for random DAGs with either the *consensus* or *average* approach over the *baseline* method. For *Toeplitz* and *Equal* covariance structures and a more detailed numerical comparison, see Supplementary Material 7.6.

In practical application, robustness to violations of model assumptions is very important. Thus, we performed additional simulations where our model assumptions are violated. We assume $Z_1, \ldots, Z_p$ follow a nonlinear DAG model:

$$z_{ij} = \sum_{k \in PA_j} f_{kj}(z_{ik}) + \varepsilon_{ij},$$

where, for each edge $k \to j$, $f_{kj} = (\beta_{kj} z_{ik})^2$ with probability $\frac{1}{2}$ and $f_{kj}(z_{ik}) = \beta_{kj} z_{ik}$ with probability $\frac{1}{2}$. The error vector ε_j follows the same n-variate joint Gaussian in Equation (5). Then, the observed variables are $x_{ij} = I(z_{ij} > \tau_j)$, where τ_j is a cutoff chosen as the median of Z_j. There are two deviations from the model assumptions specified in Equations (4) and (5). First, the parent set of Z_j are X_{PA_j} in Equation (4), compared to Z_j's parents being Z_{PA_j} in this simulation. Second, we assume nonlinear relations in this simulation. As shown in Figure 2, for most cases the consensus approach was able to improve the accuracy of the three DAG learning methods. This demonstrates that our decorrelation method is quite robust to violations against the model assumptions.

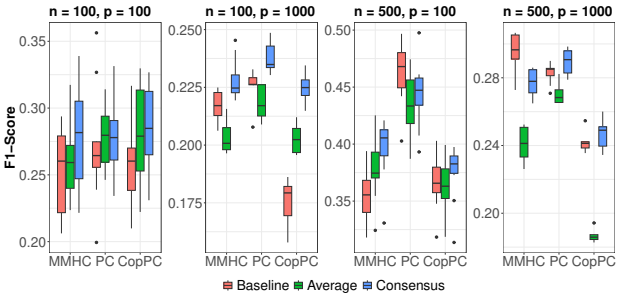

Figure 2: F-1 scores before and after decorrelation across 10 simulations for each setting of (n, p) under deviations from our model assumptions.

5 APPLICATION ON SINGLE-CELL RNA-SEQ DATA

With recent advancements in single-cell RNA-sequencing technology, researchers can measure gene expression for thousands of cells. We seek to learn gene regulatory networks (GRNs) that encode the causal relationships governing gene expression in different biological processes. Each node in a GRN represents a gene and a direct edge from gene X to Y indicates a direct regulatory effect of X on Y. In this study, we use an RNA-seq dataset published in Chu et al. (2016), which consists of gene expression measurements from approximately 20,000 genes across $n = 1018$ cells. This dataset is publicly available at the Gene Expression Omnibus with series accession number GSE75748. The cells in this data include undifferentiated human embryonic stem cells (hESCc), different lineage-specific progenitor cells that can differentiate into specific cell types, and fibroblasts that served as a control sample. Since lineage-specific progenitors were differentiated from the same population of hESCs, dependence among these cells is highly expected.

We processed the data following the methodology outlined by Li and Li (2018) involving imputation of missing values and application of a log transformation. Our study focuses on the estimation of a GRN among a subset of $p = 51$ target genes selected by Chu et al. (2016). The remaining genes are referred to as background

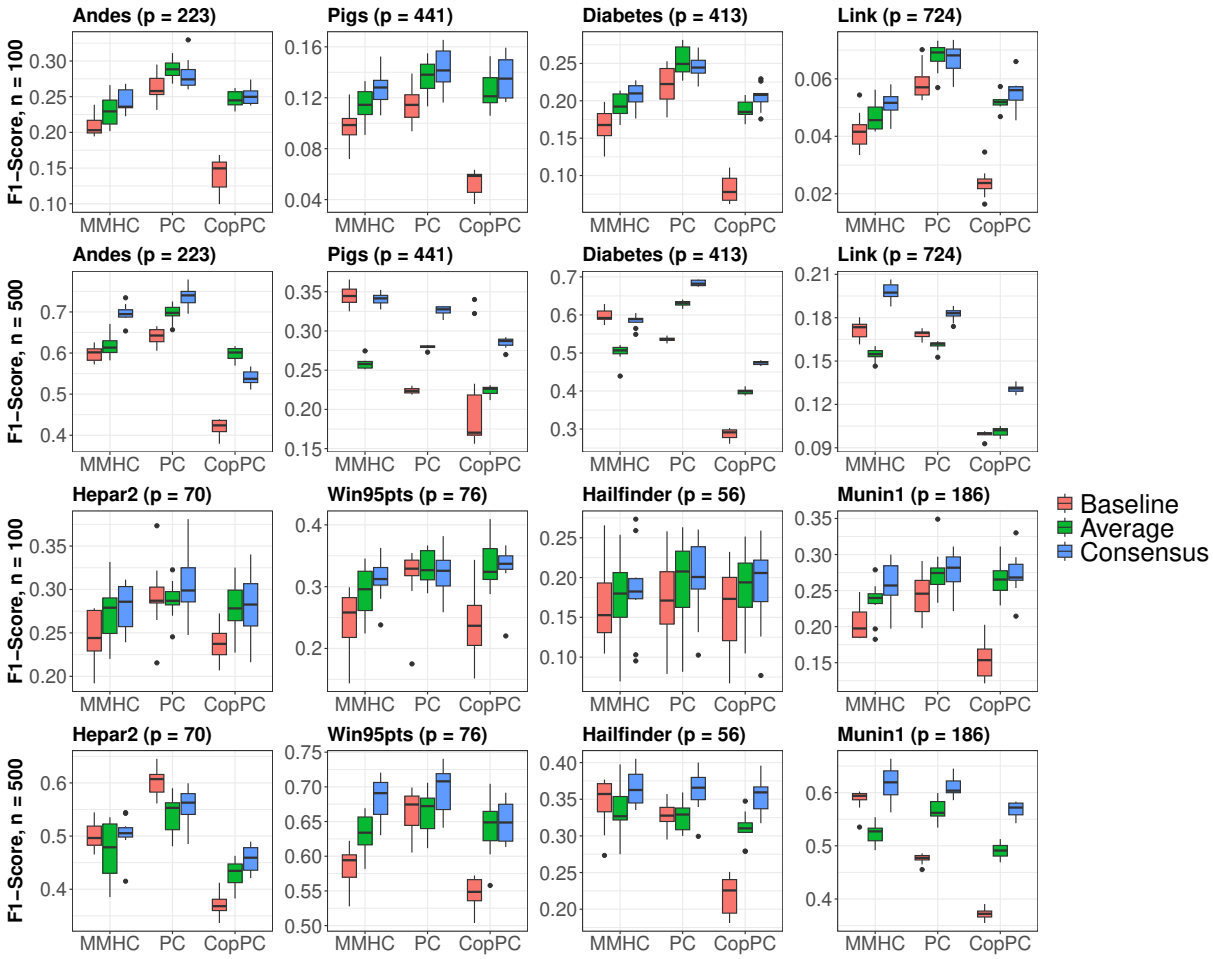

Figure 1: Structure learning accuracy before and after decorrelation for 8 real Bayesian networks.

genes hereafter.

5.1 Pre-estimate of Block Structure

The distinct cell types in the data, by experimental design, suggest that a block structure for the covariance matrix Σ among the cells. However, cells of the same type may not necessarily belong to the same block. Therefore, we applied hierarchical clustering with complete linkage, utilizing 2,000 background genes as the feature vector to partition the cells into clusters. The resulting dendrogram was cut to ensure that a majority of clusters were comprised of at least 15 cells. From each cluster, we randomly sampled 15 to 30 cells and subsequently defined the block structure for Σ by these clusters.

Single-cell RNA-seq data is typically represented as counts, but dropouts, noise, and sparsity can create an unreliable representation of gene expression levels. Discretizing the data into binary values reflects the biological distinction between active and suppressed genes. While this approach entails some loss of information, it is a deliberate trade-off that aligns with the qualitative modeling of gene regulatory networks and may lead to more robust quantification. We used k-means to identify a natural threshold for discretizing the data based on the distribution of expression counts rather than imposing an arbitrary threshold, transforming the data into binary states (0 or 1). After pre-processing, our dataset consisted of $n = 384$ cells, distributed across 14 blocks of 15 to 30 cells, with expression measure for the $p = 51$ target genes.

5.2 Model Evaluation

As the true underlying GRN is unknown, direct evaluation of our estimates is unattainable. Instead, we assessed each method by evaluating the test data likelihood of an estimated graph through cross-validation. To make use of the independence between cell blocks, each CV fold aligned with an estimated block. The primary challenge lies in estimating the covariance matrix

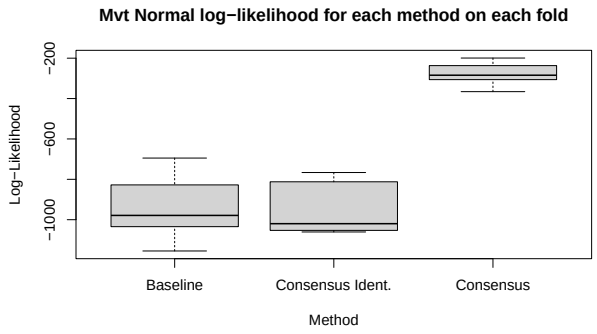

Figure 3: Distribution of the test-data log-likelihood across the 10 folds in the RNA-seq data using three different methods.

Σ for the test data. Using test data would introduce bias in the likelihood evaluation. To address this, we randomly sampled 100 background genes and ran the covariance estimation method using these genes to obtain a pre-estimated $\widetilde{\Sigma}$, which was then used to evaluate the likelihood of each test dataset. Note that no part of the test data was involved in the estimation of $\widetilde{\Sigma}$.

We compare three approaches on this dataset. Each approach estimates a graph $\widehat{\mathcal{G}}$ and the associated parameters from the training data and then evaluates the likelihood of the test data. This process is repeated across 10 folds of CV. The *baseline* method estimates $\widehat{\mathcal{G}}$ using the MMHC method on the discrete data. The second method, *consensus ident.* uses Algorithm 1 with $\widehat{L} = I$, essentially assuming no dependence between the cells. For the third method, we decorrelate latent data and apply the *consensus* approach to estimate $\widehat{\mathcal{G}}$. As shown in Figure 3, the *consensus* approach which considers cell dependence best fits the test data across all CV folds, with a substantial margin from the other two methods. We calculate a normalized likelihood ratio as a comparative tool, which can be defined as the average likelihood ratio of observing a single data point. This is expressed as $\{P(D|\text{Model}_m)/P(D|\text{Model}_b)\}^{1/np}$, where D represents the test data, Model_m is the model estimated from our method, and Model_b is the baseline estimated model. Calculating this metric using the median test data log-likelihoods in Figure 3 results in a ratio of 1.57 between the *consensus* method and the *baseline* method and a ratio of 1.62 between the *consensus* method and the *consensus ident.* method. Experiments with varying cluster sizes and background genes yielded consistent results. The only difference between *consensus ident.* and *consensus* in Figure 3 is the use of the pre-estimated covariance matrix for decorrelation, emphasizing the strong impact of between-cell dependence on fitting a graphical model. This demonstrates that our proposed dependent DAG model fits this RNA-seq dataset much better and confirms the significance of capturing dependence among cells. It shows the dependence can be well-estimated from a random set of genes, supporting our assumption of using the same Σ for all background variables ε_j, $j \in [p]$.

6 DISCUSSION

In this work, we developed the idea of data decorrelation for DAG learning on binary data. The key components include a pairwise MLE for covariance estimation and an iterative algorithm for generating and decorrelating surrogate continuous data.

6.1 Summary

Independence in data is commonly assumed in practical application where it does not hold. Extensive experiments on both synthetic and real data using our method for dependent data showcase significant improvements over existing methods, particularly in cases with $p > n$ and in real RNA-seq data for GRN estimation. We believe our algorithm for causal estimation among dependent units is particularly easy for practical application as users do not need to set additional parameters other than parameters necessary for classical causal discovery methods such as PC or MMHC. The improvement in estimation of causal models is important for applications such as GRNs, providing unexplored avenues for experimentation in drug discovery.

Our experiments reveal that our decorrelation methodology exhibits superior performance in scenarios with a strong underlying correlation structure, while demonstrating comparable performance in other settings. We advocate for using the proposed method in instances where there exists a discernible dependence structure within the dataset or when the sample size is not too large (Due to computation size). To determine the presence of a strong correlation, one practical approach could involve testing if the correlation coefficient ρ_{ab} between rows a and b is significantly different from 0.

6.2 Limitations and future work

We acknowledge a few limitations in our algorithm and experiments. First, our simulations are limited to binary data, which restricts the applicability of our method to real-world data scenarios where multiple categories or mixed data can be more common. Thus, considering a mix of continuous and discrete data could increase the applicability of our method to real data.

Our method can be modified to accommodate multi-

category discrete and mixed data settings. Generalizing to multiple categories is feasible but introduces a few computational challenges. Suppose $x_{ij} \in \{0, \ldots, K\}$. We introduce a set of cutoff values, τ_{jk}, $k = 0, \ldots, K+1$ where $\tau_{j0} = -\infty$ and $\tau_{j,K+1} = \infty$. Let $x_{ij} = k$ if and only if $\tau_{jk} \leq z_{ij} < \tau_{j,k+1}$ in place of Equation (4). Then, we can generalize the covariance estimation method in Section 3.1 by including the cutoffs $\{\tau_{jk}\}$ as unknown parameters. Once Σ and $\{\tau_{jk}\}$ have been estimated, the same decorrelation approach can be applied to simulate the latent continuous variables z_{ij} as described in Section 3.2. The key computational challenges lie in the covariance estimation step, where the number of boundaries/regions increases quadratically with each additional category, complicating the estimation process. Similarly, simulating ε_j from a multivariate truncated normal becomes more complex due to the high-dimensional space constrained by an increasing number of boundaries. For mixed data, if X_j is continuous, we adjust Equation (4) by $x_{ij} = z_{ij}$. Then, we modify the likelihood in Equation (9), where (x_{1j}, x_{2j}) follows a bivariate normal without truncation if they are continuous. Using a similar pairwise MLE we can estimate the covariance matrix Σ, whose Cholesky factor $\widehat{L}^T$ can be used to decorrelate the data. For continuous X_j, decorrelation is achieved simply by $\widehat{L}^T X_j$, while the decorrelated data for binary variables is given by $\widehat{L}^T Z_j$ in Equation (11). Then we may apply a standard DAG learning method to the decorrelated data where all columns are continuous.

Acknowledgements

This work was supported by NSF grant DMS-2305631.

References

Bryon Aragam and Qing Zhou. Concave penalized estimation of sparse gaussian bayesian networks. *The Journal of Machine Learning Research*, 16(1):2273–2328, 2015.

Rohit Bhattacharya, Daniel Malinsky, and Ilya Shpitser. Causal inference under interference and network uncertainty. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 1028–1038. PMLR, 22–25 Jul 2020. URL `https://proceedings.mlr.press/v115/bhattacharya20a.html`.

Peter Bühlmann, Markus Kalisch, and Marloes H Maathuis. Variable selection in high-dimensional linear models: partially faithful distributions and the pc-simple algorithm. *Biometrika*, 97(2):261–278, 2010.

Longbing Cao. Coupling learning of complex interactions. *Information Processing & Management*, 51(2):167–186, 2015.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

Li-Fang Chu, Ning Leng, Jue Zhang, Zhonggang Hou, Daniel Mamott, David T Vereide, Jeea Choi, Christina Kendziorski, Ron Stewart, and James A Thomson. Single-cell rna-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome biology*, 17:1–20, 2016.

Diego Colombo, Marloes H Maathuis, et al. Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.*, 15(1):3741–3782, 2014.

Ruifei Cui, Perry Groot, and Tom Heskes. Copula pc algorithm for causal discovery from mixed data. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part II 16*, pages 377–392. Springer, 2016.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

Gideon E. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6, 03 1978. doi: 10.1214/aos/1176344136.

Jianqing Fan, Han Liu, Yang Ning, and Hui Zou. High dimensional semiparametric latent graphical model for mixed data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(2):405–421, 2017.

Fei Fu and Qing Zhou. Learning sparse causal gaussian networks with experimental intervention: regularization and coordinate descent. *Journal of the American Statistical Association*, 108(501):288–300, 2013.

José A Gámez, Juan L Mateo, and José M Puerta. Learning bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood. *Data Mining and Knowledge Discovery*, 22:106–148, 2011.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Jiaying Gu, Fei Fu, and Qing Zhou. Penalized estimation of directed acyclic graphs from discrete data. *Statistics and Computing*, 29:161–176, 2019.

Naftali Harris and Mathias Drton. Pc algorithm for nonparanormal graphical models. *Journal of Machine Learning Research*, 14(11), 2013.

Michael G Hudgens and M Elizabeth Halloran. Toward causal inference with interference. *Journal of the American Statistical Association*, 103(482):832–842, 2008.

Hangjian Li, Oscar Hernan Madrid Padilla, and Qing Zhou. Learning gaussian dags from network data. *arXiv preprint arXiv:1905.10848*, 2021.

Wei Vivian Li and Jingyi Jessica Li. Modeling and analysis of rna-seq data: a review from a statistical perspective. *Quantitative Biology*, 6:195–209, 2018.

Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, page 403–410, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. 2006.

Ana Rita Nogueira, Andrea Pugnana, Salvatore Ruggieri, Dino Pedreschi, and João Gama. Methods and tools for causal discovery and causal inference. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 12(2):e1449, 2022.

Juan Miguel Ogarrio, Peter Spirtes, and Joe Ramsey. A hybrid causal search algorithm for latent variable models. In *Conference on probabilistic graphical models*, pages 368–379. PMLR, 2016.

Ulf Olsson. Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika*, 44 (4):443–460, 1979.

Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International journal of data science and analytics*, 3:121–129, 2017.

Teemu Roos. *Minimum Description Length Principle*, pages 823–827. Springer US, Boston, MA, 2017. ISBN 978-1-4899-7687-1. doi: 10.1007/978-1-4899-7687-1_894. URL `https://doi.org/10.1007/978-1-4899-7687-1_894`.

Marco Scutari. Bayesian network constraint-based structure learning algorithms: Parallel and optimised implementations in the bnlearn r package. *arXiv preprint arXiv:1406.7648*, 2014.

Ricardo Silva. *Automatic discovery of latent variable models*. Carnegie Mellon University, 2005.

Peter Spirtes. Discovering causal relations among latent variables in directed acyclic graphical models. 1996.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65:31–78, 2006.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

7 ADDITIONAL EXPERIMENTS

7.1 Pairwise Log-Likelihood Correlation Estimate

In Section 3.1, we detail a covariance estimation method that does pairwise correlation estimates between two rows of discrete data. Figure 4 illustrates a single correlation estimation in a random DAG simulation with parameters $n = 500, p = 500$. The true correlation is indicated by the red line and the estimated maximum likelihood estimate of Equation 9 is indicated by the blue line. Our correlation estimate is very close to the true correlation.

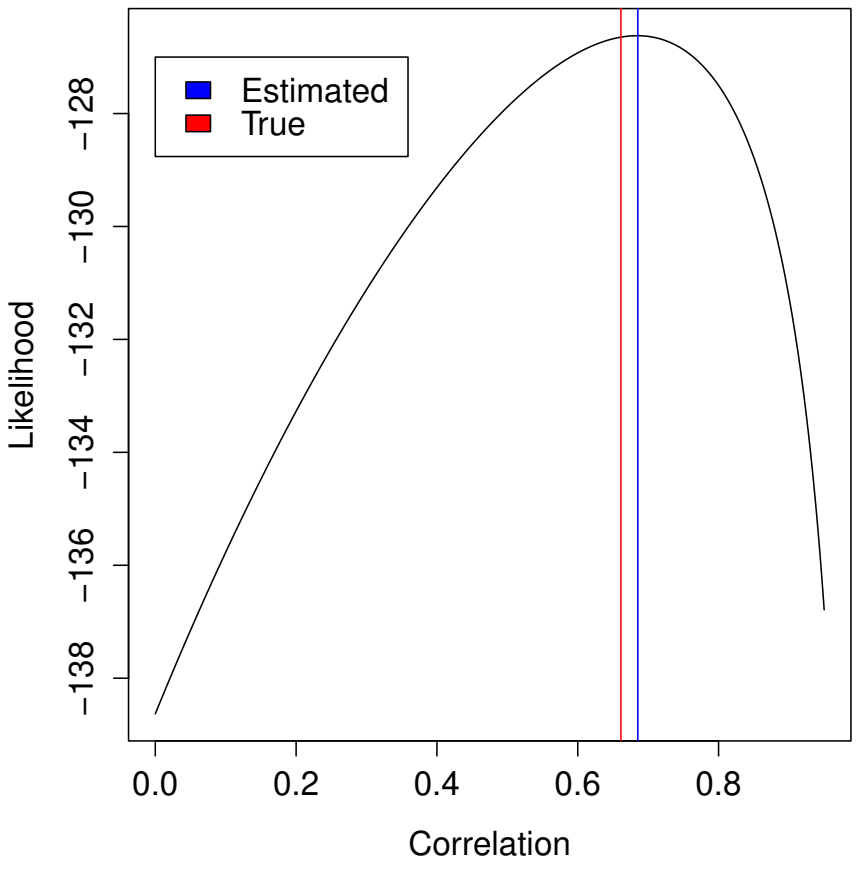

Figure 4: Single simulation of finding the correlation over a pair of units where $n = 500$ and $p = 500$.

7.2 RMSE Covariance Estimate

In Section 4, we assess the accuracy of the covariance matrix estimate from our pairwise estimation method. In each of the four settings, we run ten simulations and run the covariance estimation method. Given that each pairwise correlation entails a maximum likelihood estimate with p samples as in (9), the accuracy of covariance estimates in simulations with larger p values resulted in lower RMSEs.

To assess the accuracy of our covariance estimate, the root mean-squared error (RMSE) of the estimated covariance matrix $\widehat{\Sigma}$ was calculated relative to the true covariance Σ^*:

$$\text{RMSE}(\widehat{\Sigma}, \Sigma^*) = \left\{ \frac{1}{|H|} \sum_{(i,j)\in H} (\widehat{\Sigma}_{ij} - \Sigma^*_{ij})^2 \right\}^{1/2},$$

where H is the set of non-diagonal, non-zero elements of Σ^*. The RMSE is calculated among non-zero elements according to the block diagonal structure of Σ^* mentioned in Section 3.1. Accordingly, we only estimate correlations between rows in the same cluster. Note that $\text{diag}(\Sigma) = I$ due to the identifiability issue discussed in Section 2. This metric ensures a focused evaluation of the accuracy with respect to relevant elements of the covariance matrix. Supplementary Material 7.2 shows the distribution of the RMSE values for various parameter sets of n and p where consistently lower RMSE values are observed for larger variable size p.

Let us consider the time complexity of this pairwise covariance estimation approach. Suppose the sample size (number of units) is n_b for block b and there are a total of B blocks. Then it is easy to see that the time complexity of our method is on the order of $\mathcal{O}(\sum_{b=1}^{B} n_b^2)$. For well-balanced block sizes, where n is the total sample size, the time complexity is $\mathcal{O}(n^2/B)$.

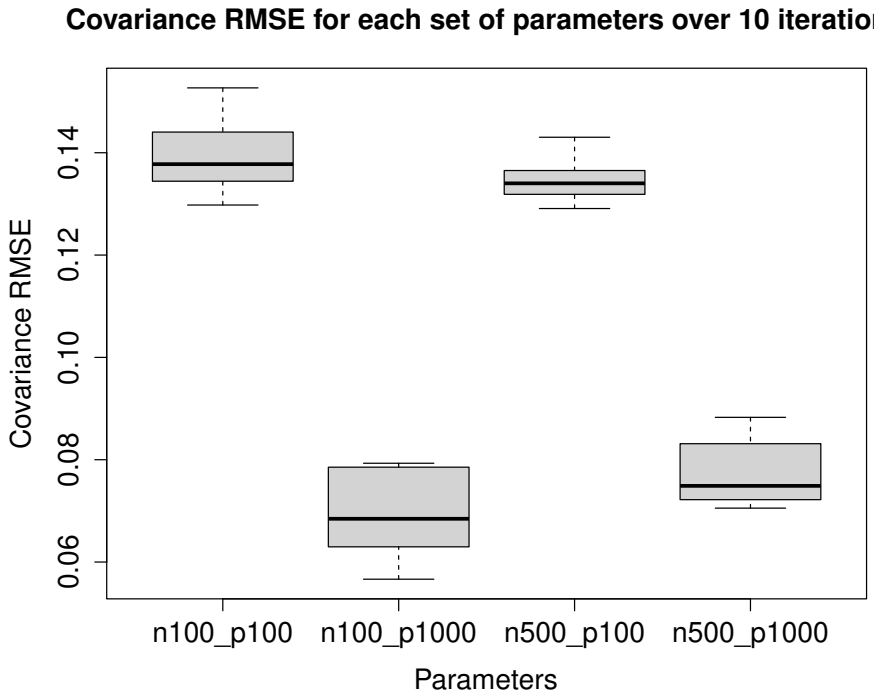

Figure 5: RMSE of estimated $\widehat{\Sigma}$. There are 10 different simulations done corresponding to each box-plot. Simulations used a mixed covariance structure under block sizes ranging from 10 to 15 under a random DAG setting.

7.3 Covariance before and after De-correlation

In Section 3.2, we use decorrelation of the dependencies among the units to obtain approximately independent data. To observe whether our de-correlated data consists of independent data, we use a Pearson correlation among the rows of our data compared to the original data-generating covariance matrix. Figure 6 shows that the correlation between rows gets much smaller through our method.

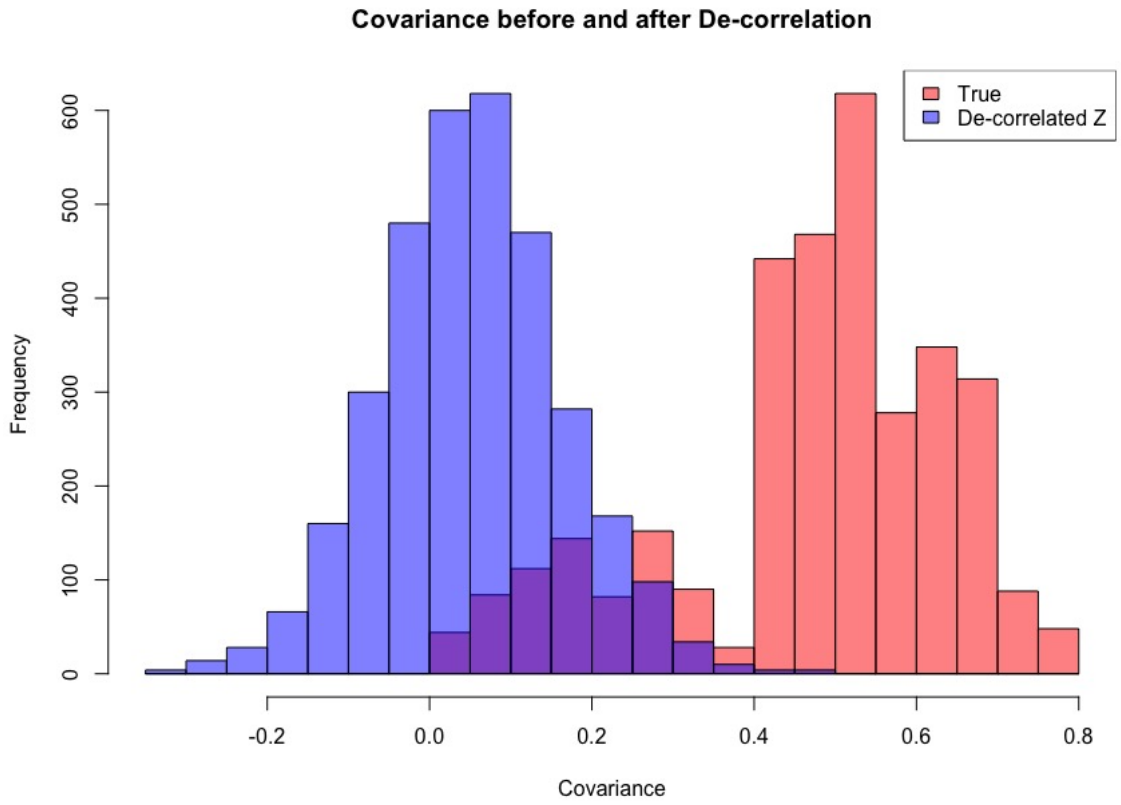

Figure 6: One simulation showing the row-wise sample covariance of the de-correlated Z compared to the true covariance in the original data under the $n = 500$, $p = 1000$ data setting. Clearly, the correlations became much smaller (between -0.1 and 0.1 for 60% of the unit pairs and between -0.2 and 0.2 for 90% of the unit pairs) than the original magnitudes.

7.4 Convergence of Algorithm 1

Section 3.2 describes the latent data recovery from discrete data and decorrelation of the dependencies among the units. We describe the iterative Algorithm 1 that aims to obtain better estimates of β to improve recovery of latent data and thus, more accurate estimates of the decorrelated latent data. Figure 7 shows the difference in β between subsequent iterations is converging.

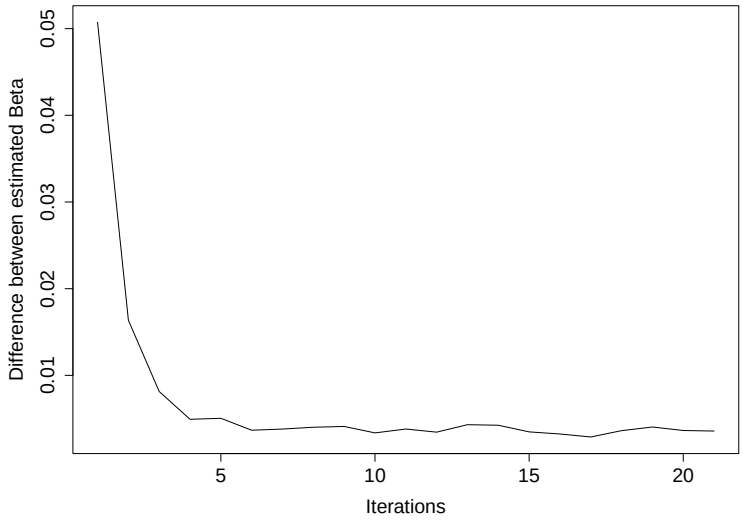

Figure 7: The difference $\|\beta^{(t+1)} - \beta^{(t)}\|$ between betas for every iteration in a simulation with $n = 100$ and $p = 100$.

7.5 Experiments on simulated data under random DAGs

For simulations, we use $n \in \{100, 500\}$ units and $p \in \{100, 1000\}$ variables. Random DAGs with p nodes were fixed to $2p$ edges and edge weights were uniformly sampled from the interval $[-0.9, -0.6] \cup [0.6, 0.9]$.

For visual comparison, we report the box-plots of the F-1 score for four combinations of n and p in Figure 8, which includes scenarios of both $p > n$ and $p < n$. For each parameter set, we conduct 10 simulations employing the MMHC method as both the initial and final DAG learning approach, and 10 simulations using the PC, and Copula-PC method in the same manner, across three approaches (Baseline, Average, Consensus).

Computation time for Algorithm 1 and structure learning is dependent on the number of units, variables, and method. For the largest case ($n = 500, p = 1000$), using the MMHC method, a single experiment takes approximately 12 hours using an internal cluster over two nodes with 32gb of memory. For the smallest case ($n = 100, p = 100$), our method takes approximately 1 or 2 minutes under the same computational settings.

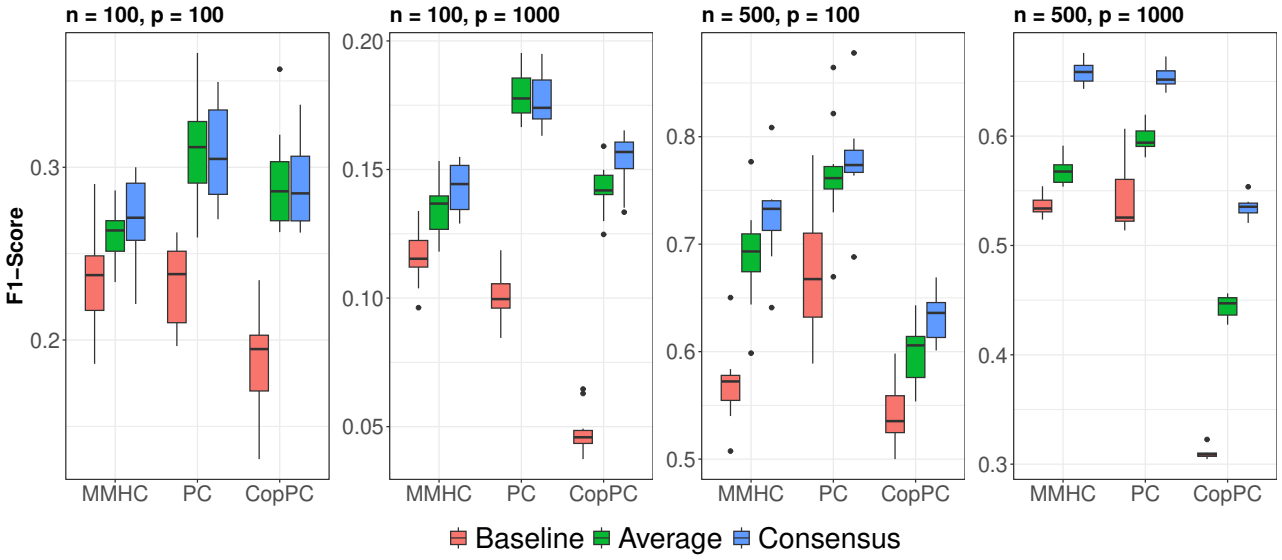

Figure 8: F-1 scores across 10 different simulations under each setting of (n, p) using either MMHC, PC, or Copula-PC as the DAG learning method.

7.6 Covariance estimate robustness to initial DAG learning method

Table 1 details the RMSE for covariance estimates to the true for different covariance structures (Σ-structures) in different settings of sample and variable size (n, p). The covariance estimation method relies on an initial estimate of β. Thus, we try two methods(Neighborhood Lasso and MMHC) for the initial estimate of the dependent discrete data. Neighborhood Lasso is the Lasso regression of $X_j \sim X_{-j}$ for all $j \in [p]$. Neighborhood Lasso does not estimate a DAG but rather a Markov Blanket (Meinshausen and Bühlmann, 2006). Because there is no significant difference in RMSE between the two methods, we opted to use MMHC as it estimates a DAG which is consistent with our resulting output.

After obtaining the parent estimates $\widehat{PA_j}$, the initial estimate of β for the covariance estimation method is done through Algorithm 1 with $\widehat{\Sigma} = I_n$. After the initial estimate of β, we then run the covariance estimation method followed by Algorithm 1 using the estimated covariance $\widehat{\Sigma}$.

Table 1: RMSE of Covariance Estimate between Neighborhood Lasso and MMHC

Σ-Structure	(n,p)	Neighborhood Lasso RMSE	MMHC RMSE
Toeplitz	(100,100)	0.122	0.130
	(100,100)	0.056	0.060
	(500,100)	0.112	0.109
	(500,1000)	0.055	0.053
Equal	(100,100)	0.115	0.108
	(100,1000)	0.039	0.039
	(500,100)	0.108	0.106
	(500,1000)	0.041	0.041

7.7 Details on random DAG results

Table 2 gives some additional results to Figure 8. We include three covariance structures (Mixed, Toepltiz, Equal) described in Section 4 with numerous sets of sample and variable size ((n, p)). The *baseline* approach estimates the DAG based on the dependent discrete data that assumes i.i.d data. The *average* and *consensus* approaches use the decorrelated latent data to estimate the final DAG. The bolded numerical results indicate the best method for the set of covariance structure and (n, p) and the '% Increase' is the increase in F-1 score compared to the baseline approach. In this table, we use MMHC as the structure learning method on both the dependent binary data and continuous decorrelated data.

Table 2: Simulated random DAG F-1 scores of three different methods: Baseline, Average, and Consensus. The respective % increase over the baseline for the best performing method (in bold) is given. Each number refers to the average over 10 different simulations.

Σ-Structure	**(n,p)**	**Baseline**	**Average**	**Consensus**	**% Increase**
Mixed	(100,100)	0.235	0.261	**0.271**	15.3
	(100,500)	0.144	0.174	**0.181**	25.7
	(100,1000)	0.116	0.135	**0.143**	23.3
	(500,100)	0.570	0.690	**0.725**	27.2
	(500,500)	0.548	0.605	**0.658**	23.5
	(500,1000)	0.537	0.605	**0.677**	22.5
Toeplitz	(100,100)	0.231	0.148	**0.260**	12.6
	(100,500)	0.144	**0.255**	0.175	21.5
	(100,1000)	0.115	**0.160**	0.140	21.7
	(500,100)	0.630	0.664	**0.727**	25.1
	(500,500)	0.580	0.619	**0.696**	22.1
	(500,1000)	0.529	0.569	**0.602**	20.6
Equal	(100,100)	0.168	0.278	**0.283**	28.6
	(100,500)	0.139	0.167	**0.182**	33.8
	(100,1000)	0.086	0.144	**0.153**	33
	(500,100)	0.449	0.660	**0.712**	24.9
	(500,500)	0.372	0.550	**0.613**	15
	(500,1000)	0.327	0.564	**0.649**	25.3