Causual Discovery for latent attributes

November 26, 2024

1 Lognormal ACDM

- $A = (a_1, \ldots, a_K)$; latent skill profile (a binary K-dimensional vector)
- $p_\alpha = \mathbb{P}(A = \alpha)$
- $Y_i, i = 1, \ldots, J$: $N \times 1$ matrix
- $X = (Y_1, Y_2, \ldots, Y_J)$: observed responses ($N \times J$ matrix)
- Q: $J \times K$ Q-matrix

For the lognormal parameters $(\mu_{j,\alpha}, \sigma^2_{j,\alpha})$, one can choose to model $\mu_{j,\alpha}$ as additive in α_k's and $\sigma^2_{j,\alpha}$ to not depend on α:

$$\mu_{j,\alpha} = \beta_{j,0} + \sum_{k=1}^{K} \beta_{j,k} q_{j,k} \alpha_k, \quad \sigma^2_{j,\alpha} = \gamma_j.$$

$Y_j \mid \mathbf{A}$ can be written as

$$Y_j \mid \mathbf{A} = \alpha \sim \text{lognormal}\left(\beta_{j,0} + \sum_{k=1}^{K} \beta_{j,k} q_{j,k} \alpha_k, \gamma_j\right).$$

Recall that the probability density function of an exponential family distribution can be written as:

$$g(y_j; \eta) = h(y_j) e^{\eta^T T(y_j) - A(\eta)}.$$

Following the convention for the exponential family distributions, η collects the natural parameters, $T(Y_j)$ collects the sufficient statistics, and $A(\eta)$ is the log-partition function.

Remark 1. *Initialization:*

- $Z = \log(X)$
- $\widehat{\beta}_{j,0} = \sum_{i=1}^{N} Z_{ij}/N = \bar{Z}_{:,j}$
- $\hat{\gamma}_j = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} \left(Z_{ij} - \bar{Z}_{:,j}\right)^2}$
- $Z'_{ij} = Z_{ij} - \widehat{\beta}_{j,0}$
- *Apply SVD to* $Z'_{ij} = U \Sigma V^T$
- $\hat{A}_{pre} = \sqrt{N} U_{,1:K}$, $\hat{B}_{pre} = \Sigma_{1:K,1:K} V^T_{,1:K} / \sqrt{N}$
- $\hat{A} = \mathit{Binarization}(\mathit{varimax}(\hat{A}_{pre})) = \mathit{Binarization}(\hat{A}_{pre} O)$, $\hat{B} = O^{-1} \hat{B}_{pre}$
- $\hat{\beta} = \begin{pmatrix} \begin{pmatrix} \widehat{\beta}_{1,0} \\ \widehat{\beta}_{2,0} \\ \cdots \\ \widehat{\beta}_{J,0} \end{pmatrix} & \hat{B}^T \end{pmatrix}$

2 Greedy Equivalence Search

D is a set of N observed data. Two DAGs $\mathcal{G}$ and $\mathcal{G}'$ are equivalent–denoted $\mathcal{G} \approx \mathcal{G}$'–if the independence constraints in the two DAGs are identical.

The GES algorithmis a two-phase greedy search through the space of DAG equivalence classes. The algorithm represents the states of the search using CPDAGs, performing transformation operators to these graphs to move in the space. Each operator corresponds to a DAG edge modification, and is scored using a DAG scoring function that we assume has three properties. First, we assume the scoring function is score equivalent, which means that it assigns the same score to equivalent DAGs. Second, we assume the scoring function is locally consistent, which means that, given enough data, (1) if the current state is not an IMAP (independence map) of $\mathcal{G}$, the score prefers edge additions that remove incorrect independencies, and (2) if the current state is an IMAP of $\mathcal{G}$, the score prefers edge deletions that remove incorrect dependencies. Finally, we assume the scoring function S_c is decomposable, which means we can express it as a sum of node:

$$S_c(\mathcal{G}, D) = \sum_{i=1}^{n} S_c(X_i, \mathbf{Pa}_i^{\mathcal{G}})$$

The first phase of the GES—called forward equivalence search or FES—starts with an empty (i.e., no-edge) CPDAG and greedily applies GES insert operators until no operator has a positive score; these operators correspond precisely to the union of all single-edge additions to all DAG members of the current (equivalence class) state. After FES reaches a local maximum, GES switches to the second phase—called backward equivalence search or BES—and greedily applies GES delete operators until no operator has a positive score; these operators correspond precisely to the union of all single edge deletions from all DAG members of the current state.

Definition 1. *Given a DAG $\mathcal{G}$ and a probability $p(\cdot)$, we say that $\mathcal{G}$ is a perfect map of p if:*

- *every independence constraint in p is implied by the structure $\mathcal{G}$*
- *every independence implied by the structure $\mathcal{G}$ holds in p.*

If there exists some DAG that is a perfect map of a probability distribution $p(\cdot)$, we say that p is DAG-perfect.

Assumption 1. *Each case in the observed data D is an iid sample from DAG-perfect probability distribution $p(\cdot)$.*

Assumption 2. *Conditional distributions are multinomial.*

Theorem 1. *Given the above assumptions, let $\mathcal{C}$ be the CPDAG that results from applying the GES algorithm to D, then $\mathcal{C} \approx \mathcal{G}$ in the limit of large m.*

Remark 2. *The scoring function is assumed to be locally consistent, which can be implied by consistency and decomposability. By [1], the Bayesian scoring criterion is locally consistent.*

Remark 3. *We can only get a PDAG after the forward/backward phase. After that, we need to convert it to the CPDAG.*

3 Theoretical Guarantee

Consider an ExpACDM with true parameters (p_0, β_0, γ_0) where the generic identifiability conditions hold. Now suppose we have observed responses X which are N independent and identically distributed response vectors from the ExpACDM stated before. Then We apply EM algorithm to X ($N \times J$ matrix) to obtain an estimate of p_0, $\hat{p}_N$. Consequently, we can generate $N'(N \leq N' \leq N\log(N))$ latent attribute vectors from the multinomial distribution with parameter $\hat{p}_N$ and then apply GES to these N' latent attribute vectors, say, X_A ($N' \times K$ matrix), to obtain a CPDAG $\mathcal{G}_N$. We will show that as $N \to \infty$, $\mathcal{G}_N$ is a perfect map of p_0.

Theorem 2. *(Consistency) Let $S(\cdot, \cdot)$ denote the BIC scoring criterion.*

- *If $\mathcal{H}$ contains p_0 and $\mathcal{G}$ does not contain p_0, then $S(\mathcal{H}, X_A) > S(\mathcal{G}, X_A)$ for sufficiently large N.*
- *If $\mathcal{H}$ and $\mathcal{G}$ both contain p_0, and $\mathcal{G}$ contains fewer parameters than $\mathcal{H}$, then $S(\mathcal{G}, X_A) > S(\mathcal{H}, X_A)$ for sufficiently large N.*

The BIC score for a Bayesian network $\mathcal{G}$ given data X_A is:

$$S(\mathcal{G}, X_A) = N' \left(\sum_{i=1}^{K} \mathbf{I}_{\hat{P}}(a_i; \mathrm{Pa}_{a_i}^{\mathcal{G}}) - \mathbf{H}_{\hat{P}}(a_i) \right) - \frac{\log N'}{2} \mathrm{Dim}[\mathcal{G}],$$

where: - $\mathbf{I}_{\hat{P}}(a_i; \mathrm{Pa}_{a_i}^{\mathcal{G}})$ is the mutual information between a_i and its parents under $\hat{P}$. - $\mathbf{H}_{\hat{P}}(a_i)$ is the entropy of a_i under $\hat{P}$. - $\mathrm{Dim}[\mathcal{G}]$ is the number of independent parameters in $\mathcal{G}$.

Proof. By Proposition 18.1 in [4], since X_A is a collection of iid samples, we can decompose the BIC score as:

$$S(\mathcal{G}, X_A) = N'(\sum_{i=1}^{K} \mathbf{I}_{\hat{P}}(a_i; \mathrm{Pa}_{a_i}^{\mathcal{G}}) - \mathbf{H}_{\hat{P}}(a_i)) - \frac{\log N'}{2}\mathrm{Dim}[\mathcal{G}].$$

where $\hat{P}$ is the empirical distribution of $(a_1, \ldots, a_K)$ in X_A and $I_P(X;Y) := H_P(X) + H_P(Y) - H_P(X,Y)$ ($H_P(X) = \sum_{x \in \mathrm{Val}(X)} -P(X=x)\log(P(X=x))$, $H_P(X,Y) = \sum_{x \in \mathrm{Val}(X)} \sum_{y \in \mathrm{Val}(Y)} -P(X=x, Y=y)\log(P(X=x, Y=y))$).

- Since $\mathcal{G}$ does not contain p_0, there does not exist a set of parameter values θ such that the parameterized Bayesian-network model $(\mathcal{G}, \theta)$ represents p_0 exactly. Hence, $\mathcal{G}$ must imply an independence assumption that p_0 does not support. Since $\mathcal{H}$ contains p_0, then it follows that this assumption also does not hold in $\mathcal{H}$.

  By definition, $\mathcal{H}$ has all edges which appear in p_0. Recall that there exists an edge in p_0 which does not appear in $\mathcal{G}$. Therefore, there exists $i \in \{1, \ldots, K\}$ such that $\mathrm{Pa}_{a_i}^{\mathcal{H}} \setminus \mathrm{Pa}_{a_i}^{\mathcal{G}} \supseteq \mathrm{Pa}_{a_i}^{*} \setminus \mathrm{Pa}_{a_i}^{\mathcal{G}} \neq \emptyset$. Note that $\mathrm{Pa}_{a_i}^{\mathcal{H}} \setminus (\mathrm{Pa}_{a_i}^{\mathcal{G}} \cap \mathrm{Pa}_{a_i}^{*}) \not\perp\!\!\!\perp_{p_0} a_i$ given $\mathrm{Pa}_{a_i}^{\mathcal{G}} \cap \mathrm{Pa}_{a_i}^{*}$ (otherwise $\mathrm{Pa}_{a_i}^{\mathcal{G}} \cap \mathrm{Pa}_{a_i}^{*} = \mathrm{Pa}_{a_i}^{*}$, a contradiction!) and $\mathrm{Pa}_{a_i}^{\mathcal{G}} \setminus (\mathrm{Pa}_{a_i}^{\mathcal{G}} \cap \mathrm{Pa}_{a_i}^{*}) \perp\!\!\!\perp_{p_0} a_i$ given $\mathrm{Pa}_{a_i}^{\mathcal{G}} \cap \mathrm{Pa}_{a_i}^{*}$. Using $\mathbf{I}_P(X; Y \cup Z) > \mathbf{I}_P(X;Y)$ if and only if Z is not conditionally independent of X given Y, we have

  $$\sum_{i=1}^{K} \mathbf{I}_{p_0}(a_i; \mathrm{Pa}_{a_i}^{\mathcal{H}}) > \sum_{i=1}^{K} \mathbf{I}_{p_0}(a_i; \mathrm{Pa}_{a_i}^{\mathcal{G}} \cap \mathrm{Pa}_{a_i}^{*}) = \sum_{i=1}^{K} \mathbf{I}_{p_0}(a_i; \mathrm{Pa}_{a_i}^{\mathcal{G}}).$$

  As $N \to \infty$, our empirical distribution $\hat{P}$ will converge to p_0 with probability 1, hence $\mathbf{I}_{\hat{P}}(a_i; \mathrm{Pa}_{a_i}^{\mathcal{H}}) = \mathbf{I}_{p_0}(a_i; \mathrm{Pa}_{a_i}^{\mathcal{H}}) + o_P(1)$ in probability as m increases. Therefore, for large m,

  $$\begin{aligned} S(\mathcal{H}, X_A) - S(\mathcal{G}, X_A) &= \frac{1}{2}(\mathrm{Dim}[\mathcal{G}] - \mathrm{Dim}[\mathcal{H}])\log(N') + N'(\sum_{i=1}^{K} \mathbf{I}_{p_0}(a_i; \mathrm{Pa}_{a_i}^{\mathcal{H}}) - \sum_{i=1}^{K} \mathbf{I}_{p_0}(a_i; \mathrm{Pa}_{a_i}^{\mathcal{G}}) + o_P(1)) \\ &= O(\log(N')) + N'(\sum_{i=1}^{K} \mathbf{I}_{p_0}(a_i; \mathrm{Pa}_{a_i}^{\mathcal{H}}) - \sum_{i=1}^{K} \mathbf{I}_{p_0}(a_i; \mathrm{Pa}_{a_i}^{\mathcal{G}})) + o_P(N') \\ &> 0 \end{aligned}$$

- Since $\hat{p}(A = \alpha) = p_0(A = \alpha) + O_p(\frac{1}{\sqrt{N}})$ for any $\alpha \in \{0,1\}^K$, we know $\hat{p}(A_{sub} = \alpha_{sub}) = p_0(A_{sub} = \alpha_{sub}) + O_P(\frac{1}{\sqrt{N}})$ through simple addition, where A_{sub} and α_{sub} are ordered subsets of A and α, respectively. (For example, we have $\hat{p}(a_i = 1) = p_0(a_i = 1) + O_P(\frac{1}{\sqrt{N}})$)

  By CLT and the definition of $\hat{P}$ (empirical distribution of iid samples from $\hat{p}$), we have

  $$\hat{P}(A = \alpha) = \hat{p}(A = \alpha) + O_P(\frac{1}{\sqrt{N'}}).$$

  By combining the aforementioned results, we conclude that

  $$\hat{P}(A = \alpha) = p_0(A = \alpha) + O_P(\frac{1}{\sqrt{N}}) + O_P(\frac{1}{\sqrt{N'}})$$

  and

  $$\hat{P}(A_{sub} = \alpha_{sub}) = p_0(A_{sub} = \alpha_{sub}) + O_P(\frac{1}{\sqrt{N'}}) + O_P(\frac{1}{\sqrt{N}}).$$

  Recall that $S(\mathcal{G}, X_A) = S_L(\mathcal{G}, X_A) - \frac{\log N'}{2}\mathrm{Dim}[\mathcal{G}]$, where $S_L(\mathcal{G}, X_A) = l(\hat{\theta}_{\mathcal{G}}; X_A)$ ($l(\hat{\theta}_{\mathcal{G}}; X_A)$ is the logarithm of the likelihood function and $\hat{\theta}_{\mathcal{G}}$ are the maximum likelihood parameters for $\mathcal{G}$). Note that $\hat{\theta}_{\mathcal{G}}$ is a collection of parameters relies on the structure of $\mathcal{G}$. However, to calculate $l(\hat{\theta}_{\mathcal{G}}; X_A)$, we don't require all information of $\hat{\theta_{\mathcal{G}}}$. Actually, we only need 2^K parameters: $\hat{p}_{\hat{\theta}_{\mathcal{G}}}(A = \alpha)$ ($\alpha \in \{0,1\}^K$), where $\hat{p}_{\hat{\theta}_{\mathcal{G}}}$ is the point mass function obtained by $\hat{\theta}_{\mathcal{G}}$.

  Since $\mathcal{G}$ and $\mathcal{H}$ both contain p_0, we have

  $$\begin{aligned} |S_L(\mathcal{H}, X_A) - S_L(\mathcal{G}, X_A)| &\leq |S_L(\mathcal{H}, X_A) - l(p_0, X_A)| + |S_L(\mathcal{G}, X_A) - l(p_0, X_A)| \\ &= (S_L(\mathcal{H}, X_A) - l(p_0, X_A)) + (S_L(\mathcal{G}, X_A) - l(p_0, X_A)) \\ &= (l(\hat{p}_{\hat{\theta}_{\mathcal{G}}}, X_A) - l(p_0, X_A)) + (l(\hat{p}_{\hat{\theta}_{\mathcal{H}}}, X_A) - l(p_0, X_A)) \\ &\leq 2(l(\hat{P}, X_A) - l(p_0, X_A)) \end{aligned}$$

, where the last inequality comes from the fact that $\hat{P}$ is the maximum likelihood parameters (for the complete graph/without constraints). We claim that:

$$l(\hat{P}, X_A) - l(p_0, X_A) = O_P(\frac{N'}{N}).$$

Next we write the density of X_A as a form of exponential families $e^{N(T(X_A)^T h(\hat{p}) - b(\hat{p}))}$, where

$$T(X_A) = [\hat{P}(A = \alpha)](\alpha \in \{0,1\}^K \setminus (0,0,0,\ldots,0)),$$

$$h(\hat{p}) = [\log(\frac{\hat{p}(A=\alpha)}{\hat{p}(A=(0,0,0,\ldots,0))})](\alpha \in \{0,1\}^K \setminus (0,0,0,\ldots,0))$$

are both $(2^K - 1)$-dimensional vector and

$$b(\hat{p}) = -\log(\hat{p}(A = (0,0,0,\ldots,0)).$$

By the property of exponential families, we know

$$\frac{d(b(p_0)}{d(h(p))} = [p_0(\alpha)](\alpha \in \{0,1\}^K \setminus (0,0,0,\ldots,0))$$

.

(Let $h(p) = [\log(\frac{p(A=\alpha)}{p(A=(0,0,0,\ldots,0))})] = [x_i]_{1 \le i \le 2^K - 1}$, then $b(p) = -\log(p(A = (0,0,0,\ldots,0)) = \log(1 + \sum_{i=1}^{2^K-1} e^{x_i})$. Then we have $\frac{d(b(p)}{d(h(p))} = [\frac{e^{x_i}}{1+\sum_{i=1}^{2^K-1} e^{x_i}}]_{1 \le i \le 2^K - 1} = [p(\alpha)]((\alpha \in \{0,1\}^K \setminus (0,0,0,\ldots,0))))$

As a result, we have

$$\begin{aligned} l(\hat{P}, X_A) - l(p_0, X_A) &= N'(T(X_A)^T h(\hat{P}) - b(\hat{P}) - (T(X_A)^T h(p_0) - b(p_0)) \\ &= N'(T(X_A)^T (h(\hat{P}) - h(p_0)) + b(p_0) - b(\hat{P})) \\ &= N'(T(X_A)^T (h(\hat{P}) - h(p_0)) - (\frac{d(b(p_0))}{d(h(p))})^T (h(\hat{P}) - h(p_0)) + O_P(\frac{1}{N})) + O_P(\frac{1}{N'}), \end{aligned}$$

where the last identity comes from Taylor expansion and the fact that $\hat{P}(A = \alpha) = p_0(A = \alpha) + O_P(\frac{1}{\sqrt{N}}) + O_P(\frac{1}{\sqrt{N'}})$.

To this end, we deduce that

$$l(\hat{P}, X_A) - l(p_0, X_A) = N'((T(X_A) - \frac{d(b(p_0)}{d(h(p))})^T (h(\hat{P}) - h(p_0)) + O_P(\frac{1}{N'})).$$

Since we have $T(X_A) - \frac{d(b(p_0))}{d(h(p))} = [\hat{P}(A = \alpha)] - p_0(\alpha)(\alpha \in \{0,1\}^K \setminus (0,0,0,\ldots,0)) = O_P(\frac{1}{\sqrt{N'}}) + O_P(\frac{1}{\sqrt{N}})$. We conclude that $l(\hat{P}, X_A) - l(p_0, X_A) = O_P(\frac{N'}{N})$. (Assume $\frac{N'}{N} > 1$)

Therefore, asymptotically we have that

$$S(\mathcal{H}, X_A) - S(\mathcal{G}, X_A) = \frac{1}{2}(\mathrm{Dim}[\mathcal{G}] - \mathrm{Dim}[\mathcal{H}]) \log(N') + O_P(\frac{N'}{N}) < 0$$

if $\frac{N'}{N} = o(\log(N'))$

□

Remark 4. *By the proof, we know the condition N and N' need to satisfy is $\frac{N'}{\log(N')} << N$. If we take $N' = N\log(N)$, this condition will always be satisfied.*

Definition 2. *Let D be a set of data consisting of iid records. Let $\mathcal{G}$ be any DAG, and let $\mathcal{G}'$ be the DAG that results from adding the edge $X_i \to X_j$. A scoring criterion $S(\mathcal{G}, D)$ is locally consistent if the following two properties hold:*

- *If $X_j \not\perp\!\!\!\perp_p X_i \mid \mathbf{Pa}_j^{\mathcal{G}}$, then $S(\mathcal{G}', D) > S(\mathcal{G}, D)$.*
- *If $X_j \perp\!\!\!\perp_p X_i \mid \mathbf{Pa}_j^{\mathcal{G}}$, then $S(\mathcal{G}', D) < S(\mathcal{G}, D)$.*

Definition 3. *$X \to Y$ is covered in $\mathcal{G}$ if $\mathbf{Pa}_Y^{\mathcal{G}} = \mathbf{Pa}_X^{\mathcal{G}} \cup X$.*

Definition 4. *An equivalence class $\mathcal{E}'$ is in $\mathcal{E}^+(\mathcal{E})$ if and only if there is some DAG $\mathcal{G} \in \mathcal{E}$ to which we can add a single edge that results in a DAG $\mathcal{G} \in \mathcal{E}'$. The definition of $\mathcal{E}^-(\mathcal{E})$ is completely analogous to that of $\mathcal{E}^+(\mathcal{E})$.*

Theorem 3. *If a scoring method is consistent and decomposable, then it is locally consistent.*

Theorem 4. *Let $\mathcal{E}^*$ denote the equivalence class that is a perfect map of the generative distribution $p_0(\cdot)$. Then in the limit of N, $S(\mathcal{E}^*, X_A) > S(\mathcal{E}, X_A)$ for any $\mathcal{E} \neq \mathcal{E}^*$.*

Lemma 1. *Let $\mathcal{G}$ and $\mathcal{H}$ be any pair of DAGs such that $\mathcal{G} \leq \mathcal{H}$ ($\mathcal{H}$ is an idependence map of $\mathcal{G}$). Let r be the number of edges in $\mathcal{H}$ that have opposite orientation in $\mathcal{G}$, and let m be the number of edges in $\mathcal{H}$ that do not exist in either orientation in $\mathcal{G}$. There exists a sequence of at most $r + 2m$ edge reversals and additions in $\mathcal{G}$ with the following properties:*

1. *Each edge reversed is a covered edge*
2. *After each reversal and addition $\mathcal{G}$ is a DAG and $\mathcal{G} \leq \mathcal{H}$*
3. *After all reversals and additions $\mathcal{G} = \mathcal{H}$*

Theorem 5. *Let $\mathcal{E}'_N$ be the equivalence class that results from the first phase of GES. For sufficiently large N, $\mathcal{E}'_N$ contains p_0.*

Proof. Suppose not, and hence there exists sufficiently large m such that any $G \in \mathcal{E}'_m$, G contains some independence constraint not in p_0.Because the independence constraints of G are characterized by the Markov conditions, there must exist some node X_i in G for which $X_i \not\perp\!\!\!\perp_{p_0} \mathbf{Y}|\mathrm{Pa}_i$, where $\mathbf{Y}$ is the set of non-descendants of X_i. Furthermore, because the composition axiom holds for $p_0(\cdot)$, there must exist at least one singleton non-descendant $Y \in \mathbf{Y}$ for which this dependence holds. By Theorem 3, this implies that the DAG G' that results from adding the edge $Y \rightarrow X_i$ to G (which cannot be cyclic by definition of Y) has a higher score than G. Clearly, $\mathcal{E}(G') \in \mathcal{E}^+(\mathcal{E}_N)$, which contradicts the fact that $\mathcal{E}'_N$ is a local maximum.

We now use Lemma 1 to show that in the second phase, GES will add independence constraints (by "deleting edges") until the equivalence class corresponding to the generative distribution is reached. □

Theorem 6. *Let ϵ_N be the equivalence class that results from GES. For sufficiently large N, ϵ_N is a perfect map of p_0.*

Proof. Given Theorem 2, we know that when the second phase of the algorithm is about to commence, the current state of the search algorithm contains p_0. We are guaranteed that $\mathcal{E}_N$ will continue to contain p_0 throughout the remainder of the algorithm by the following argument. Consider the first move made by GES to a state that does not contain p_0. By definition of $\mathcal{E}^-(\mathcal{E}_N)$, this move corresponds to an edge deletion in some DAG. But it follows immediately from the fact that the score is consistent that any such deletion would decrease the score, contradicting the fact that GES is greedy.

To complete the proof, assume that the algorithm terminates with some sub-optimal equivalence class $\mathcal{E}_N$, and let $\mathcal{E}^*$ be the optimal equivalence class. From Theorem 4, we know that $\mathcal{E}^*$ is a perfect map of p_0, and because $\mathcal{E}_N$ contains p_0, it follows that $\mathcal{E}_m$ must be an independence map of $\mathcal{E}^*$. Let $\mathcal{H}$ be any DAG in $\mathcal{E}_N$, and let $\mathcal{G}$ be any DAG in $\mathcal{E}^*$. Because $\mathcal{G} \leq \mathcal{H}$, we know from Lemma 1 that there exists a sequence of covered edge reversals and edge additions that transforms $\mathcal{G}$ into $\mathcal{H}$. There must be at least one edge addition in the sequence because by assumption $\mathcal{E}_N \neq \mathcal{E}^*$ and hence $\mathcal{G} \neq \mathcal{H}$. Consider the DAG $\mathcal{G}'$ that precedes the last edge addition in the sequence. Clearly $\mathcal{E}(\mathcal{G}') \in \mathcal{E}^-(\mathcal{E}_N)$ and because $\mathcal{G}'$ has fewer parameters than $\mathcal{H}$, we conclude from the consistency of the scoring criterion that $\mathcal{E}_N$ cannot be a local maximum, yielding a contradiction. □

4 Experiments

4.1 Direct causal discovery on latent attributes using existing methods

In practice, we will use BDeu as our scoring criterion.

We first describe the true parameter settings used in the simulations. In all simulations, we set the Q-matrix and proportion parameter $\mathbf{p}$ as follows. Consider $K = 5$ latent attributes and $J = 20$ items. The Q-matrix takes the form

$$\mathbf{Q} = \begin{pmatrix} \mathbf{I}_K \\ \mathbf{I}_K \\ \mathbf{I}_K \\ \mathbf{Q}_1 \end{pmatrix}, \text{ where } \mathbf{Q}_1 = \begin{pmatrix} 1 & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & 1 \end{pmatrix}_{K \times K}.$$

The above Q-matrix satisfies our strict identifiability conditions in [2]. α's are generated by the following 3 structures.

Under the Normal-ACDM, we set the coefficients $\beta_{j,k}$ by

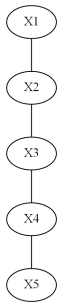
$$\beta_{j,0} = -1;$$
$$\beta_{j,k} = \frac{3}{\sum_{k'=1}^{K} q_{j,k'}} \mathbf{1}(q_{j,k} = 1), \quad \forall j \in [J], k \in [K].$$

The variance parameter $\gamma_j = \sigma_j^2$ is fixed to be 1 for all j.

In each of the independent replicates, we generate data using the above parameter settings and fit EM Algorithm 2 in [2] with a random initialization. After obtaining estimated $\hat{p}_\alpha$, we will generate a new matrix $\mathcal{A}$ based on 3 methods, then we perform GES or PC algorithm on $\mathcal{A}$.

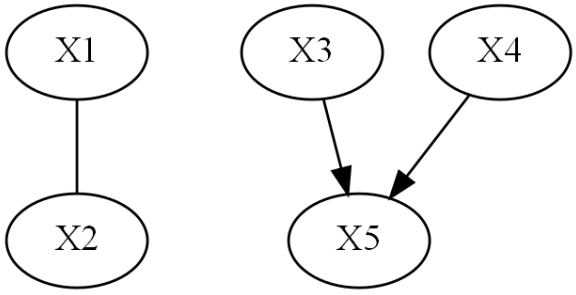
- Method **1**: The counts of each type α in $\mathcal{A}$ $(N' \times K)$ are equal to the corresponding probability in $\hat{p}_\alpha$ multiplied by N'.
- Method **2**: Each row of $\mathcal{A}$ $(N' \times K)$ is generated from a multinomial distribution with parameter $\hat{p}_\alpha$.
- Method **3**: For each row in $X_{i,:}$, we can estimate the latent attribute profile A_i and combine them to obtain $\mathcal{A}$ $(N \times K)$.
- Method **4**: The frequency of each type α in $\mathcal{A}$ $(N' \times K)$ is equal to the corresponding counts in A multiplied by $\frac{N'}{N}$.
- Method **5**: Each row of $\mathcal{A}$ $(N' \times K)$ is generated from a multinomial distribution with parameter p'_α.
- Method **6**: $\mathcal{A}$ is the raw data A $(N \times K)$.

where p'_{alpha} is the vector of frequencies for each type α in the A.



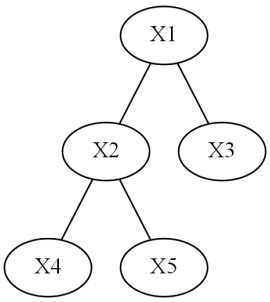
*Structure*1

Figure 1: $P(X_1 = 1) = 0.5$
$P(X_2 = 1 \mid X_1 = 1) = 0.6$
$P(X_2 = 1 \mid X_1 = 0) = 0.4$
$P(X_3 = 1 \mid X_2 = 1) = 0.6$
$P(X_3 = 1 \mid X_2 = 0) = 0.4$
$P(X_4 = 1 \mid X_3 = 1) = 0.4$
$P(X_4 = 1 \mid X_3 = 0) = 0.6$
$P(X_5 = 1 \mid X_4 = 1) = 0.4$
$P(X_5 = 1 \mid X_4 = 0) = 0.6$



*Structure*2

Figure 2: $P(X_1 = 1) = 0.5$
$P(X_3 = 1) = 0.5$
$P(X_4 = 1) = 0.5$
$P(X_2 = 1 \mid X_1 = 1) = 0.6$
$P(X_2 = 1 \mid X_1 - 0) = 0.4$
$P(X_5 = 1 \mid X_3 = 1, X_4 = 1) = 0.8$
$P(X_5 = 1 \mid X_3 = 1, X_4 = 0) = 0.3$
$P(X_5 = 1 \mid X_3 = 0, X_4 = 1) = 0.1$
$P(X_5 = 1 \mid X_3 = 0, X_4 = 0) = 0.9$



*Structure*3

Figure 3: $P(X_1 = 1) = 0.5$
$P(X_2 = 1 \mid X_1 = 1) = 0.6$
$P(X_2 = 1 \mid X_1 = 0) = 0.4$
$P(X_3 = 1 \mid X_1 = 1) = 0.4$
$P(X_3 = 1 \mid X_1 = 0) = 0.6$
$P(X_4 = 1 \mid X_2 = 1) = 0.4$
$P(X_4 = 1 \mid X_2 = 0) = 0.6$
$P(X_5 = 1 \mid X_2 = 1) = 0.6$
$P(X_5 = 1 \mid X_2 = 0) = 0.4$

Remark 5. *If we choose the true values of β, γ and p_α as input, at most 4 iterations are required to satisfy the stopping criterion. (10 replications for Structure 2 and Structure 3)*

After 100 replications, the results for Structure 3 are shown below.

N=2000	BDeu	BIC
Method **6**	100%	46%
Method **3**	95%	46%
Method **1**, $N' = 2000$	93%	34%
Method **2**, $N' = 2000$	82%	34%
Method **1**/**4**, $N' = 10000$	57%/61%	32%/33%
Method **1**/**4**, $N' = 20000$	20%/23%	17%/18%
Method **1**/**4**, $N' = 100000$	1%/0%	4%/5%
Method **2**/**5**, $N' = 10000$	42%/48%	25%/29%
Method **2**/**5**, $N' = 20000$	20%/19%	22%/19%
Method **2**/**5**, $N' = 100000$	1%/0%	3%/6%

N=1000	BDeu
Method **6**	77%
Method **3**	75%
Method **1**, $N' = 1000$	66%
Method **2**, $N' = 1000$	52%
Method **1**, $N' = 5000$	43%
Method **1**, $N' = 10000$	9%
Method **1**, $N' = 50000$	0%
Method **2**, $N' = 5000$	35%
Method **2**, $N' = 10000$	12%
Method **2**, $N' = 50000$	0%

After 200 replications, the results for Structure 2 and Structure 1 are shown below.

N=2000	BDeu
Method **6**	83%
Method **3**	88.5%
Method **1**, $N' = 2000$	83%
Method **2**, $N' = 2000$	75%
Method **1**/**4**, $N' = 10000$	78%/78%
Method **1**/**4**, $N' = 20000$	28%/30%
Method **1**/**4**, $N' = 100000$	0%/0%
Method **2**/**5**, $N' = 10000$	59.5%/65.5%
Method **2**/**5**, $N' = 20000$	22%/22.5%
Method **2**/**5**, $N' = 100000$	0%/0%

N=2000	BDeu
Method **6**	100%
Method **3**	98.5%
Method **1**, $N' = 2000$	98.5%
Method **2**, $N' = 2000$	86.5%
Method **1**/**4**, $N' = 10000$	61.5%/61.5%
Method **1**/**4**, $N' = 20000$	21.5%/24.5%
Method **1**/**4**, $N' = 100000$	0%/0%
Method **2**/**5**, $N' = 10000$	45%/46.5%
Method **2**/**5**, $N' = 20000$	17.5%/19%
Method **2**/**5**, $N' = 100000$	0%/0.5%

The results indicate that the differences between Method 1 and Method 4, as well as Method 2 and Method 5, are insignificant. This suggests that the use of the estimated $\hat{p}$ already yields relatively satisfying outcomes. In the sequel, Methods 4 and 5 will no longer be used.

By the previous proof, we know Method 2 will achieve exact recovery as $N \to \infty$ if $N' = cN$ (c is a constant). We performed 200 replications on Structure 2 to validate the results.

Structure 2	N=2000	N=5000	N=10000	N=50000
Method **6**	87.5%	100%	100%	100%
Method **3**	86.5%	99%	99%	100%
Method **2**, $N' = N$	66%	93.5%	97.5%	100%
Method **2**, $N' = 2N$	79.5%	95%	97%	99%
Method **2**, $N' = 4N$	73%	81%	85.5%	95%
Method **2**, $N' = 7N$	42.5%	49%	58%	68.5%

We have assumed that the Q-matrix has been provided in all the experiments above. Nevertheless, it is practically impossible to know the Q-matrix in advance. Consequently, we may choose a method which can estimate p_α and the Q-matrix simultaneously. We will adopt this method in subsequent experiments.

The three tables below are obtained after performing 200 replications on Structure 1, 2 and 3, respectively, from left to right.

Structure 1	N=2000
Method **6**	100%
Method **2**, $N' = N$	86%

Structure 2	N=2000	N=5000
Method **6**	87.5%	100%
Method **2**, $N' = N$	61%	91%

Structure 3	N=2000
Method **6**	99.5%
Method **2**, $N' = N$	88.5%

For computational efficiency, when K is large, EM algorithm should be replaced by SAEM to estimate p_α. However, this method requires a proper initialization for A, which might be difficult to achieve.

Algorithm 1: EM Algorithm for the General ExpACDM

Data: Responses $Y = (Y_{ij})_{N\times J}$
Result: Parameters β_j, γ_j, p_α's
Initialize parameters β_j, γ_j, p_α's;
regularization parameter λ
while *log-likelihood has not converged* **do**
  In the $(t+1)$**-th iteration,**;
  // **E step**;
  for $j \in [J], \alpha \in \{0,1\}^K$ **do**
    $\eta_{j,\alpha}^{(t)} = h(\beta_{j,0}^{(t)}) + \sum_k \beta_{j,k}^{(t)} \alpha_k, \gamma_j^{(t)}$;
  end
  for $(i,\alpha) \in [N] \times \{0,1\}^K$ **do**
    $\varphi_{i,\alpha}^{(t+1)} = \mathbb{P}(A_i = \alpha \mid Y, \beta_j^{(t)}, \gamma_j^{(t)}, p_\alpha^{(t)}) = \frac{p_\alpha \exp\left(\sum_j \eta_{j,\alpha}^{(t)T} \mathrm{T}(Y_{ij}) - A(\eta_{j,\alpha}^{(t)})\right)}{\sum_{\alpha'} p_{\alpha'} \exp\left(\sum_j \eta_{j,\alpha'}^{(t)T} \mathrm{T}(Y_{ij}) - A(\eta_{j,\alpha'}^{(t)})\right)}$;
  end
  // **M step**;
  for $\alpha \in \{0,1\}$ **do**
    $p_\alpha^{(t+1)} = \frac{\sum_i \varphi_{i,\alpha}^{(t+1)}}{\sum_{i,\alpha'} \varphi_{i,\alpha'}^{(t+1)}}$;
  end
  for $j \in [J]$ **do**
    $(\beta_j, \gamma_j)^{(t+1)} =$
    $\arg\max_{\beta_j,\gamma_j} -\lambda \sum_{k=1}^K |\beta_{j,k}| + \sum_{i,\alpha} \varphi_{i,\alpha}^{(t+1)} \left[h(\beta_{j,0} + \sum_{k=1}^K \beta_{j,k}\alpha_k, \gamma_j)^T \mathrm{T}(Y_{ij}) - A(h(\beta_{j,0} + \sum_{k=1}^K \beta_{j,k}\alpha_k, \gamma_j))\right]$;
  end
end
Output β_j, γ_j, p_α's;

Algorithm 2: SAEM Algorithm for the General ExpACDM

Data: Responses $X = (X_{ij})_{N\times J}$, initial latent variables A_{in}, parameters β_{in}, γ_{in}, regularization parameter λ
Result: Parameters β_j, γ_j, p_α's and estimated A
Initialize β, γ, $p = 0$, $A^{new} = A_{in}$ and $t = 0$;
while *not converged* **do**
  In the $(t+1)$**-th iteration,**;
  $A^{old} \leftarrow A^{new}$;
  // **Stochastic E-step: Update latent variables**;
  foreach $i \in [N]$ **do**
    foreach $k \in [K]$ **do**
      Update conditional distributions of each $A_{i,k}$
      $\mathbb{P}_{i,k} := \mathbb{P}(A_{i,k} = 1 \mid -) = \mathrm{expit}([h(\beta_{j,0} + \sum_{s\neq k} \beta_{j,s} A_{i,s}^{old} + \beta_{j,k}, \gamma_j)^T \mathrm{T}(Y_{ij}) - A(h(\beta_{j,0} + \sum_{s\neq k} \beta_{j,s} A_{i,k}^{old} + \beta_{j,k}, \gamma_j))] - [h(\beta_{j,0} + \sum_{s\neq k} \beta_{j,s} A_{i,s}^{old}, \gamma_j)^T \mathrm{T}(Y_{ij}) - A(h(\beta_{j,0} + \sum_{s\neq k} \beta_{j,s} A_{i,k}^{old}, \gamma_j))])$;
    end
  end
  Sample latent variables A_{sample} from $\mathbb{P}$;
  Update probabilities $p \leftarrow \frac{1}{t+1}\mathbb{P} + \frac{t}{t+1}p$;
  // **M-step: Update parameters**;
  foreach $j \in [J]$ **do**
    Define log-likelihood function f_{loglik} from $X[:,j]$ and A_{sample};
    Update $f_{old}[j] \leftarrow (1 - \frac{1}{t+1}) f_{old}[j] + \frac{1}{t+1} f_{loglik}$;
    $(\beta_j, \gamma_j)^{(t+1)} = \arg\max_{\beta_j,\gamma_j} -\lambda \sum_{k=1}^K |\beta_{j,k}| + f_{old}[j]$
  end
  Compute error from parameter updates in β and γ;
  Update latent variables $A_{new} \leftarrow I(p > 0.5)$;
  $t \leftarrow t + 1$;
end
Update p_α from row counts in A_{new};
Return: p_α, β, γ, A_{new};

4.2 Initialization

We may adopt the double-SVD method in [3] to compute a spectral initialization. The rough procedure is as follows:

Algorithm 3: MATLAB Varimax Rotation

Data: Matrix V, number of latent variables K
Result: Rotated matrix R_V
// SVD Decomposition;
$V_{adj} \leftarrow V[:, 1:K]$;
// Varimax Rotation;
$R_V \leftarrow \text{rotatefactors}(V_{adj}, \text{'Method'}, \text{'varimax'})$;
return R_V;

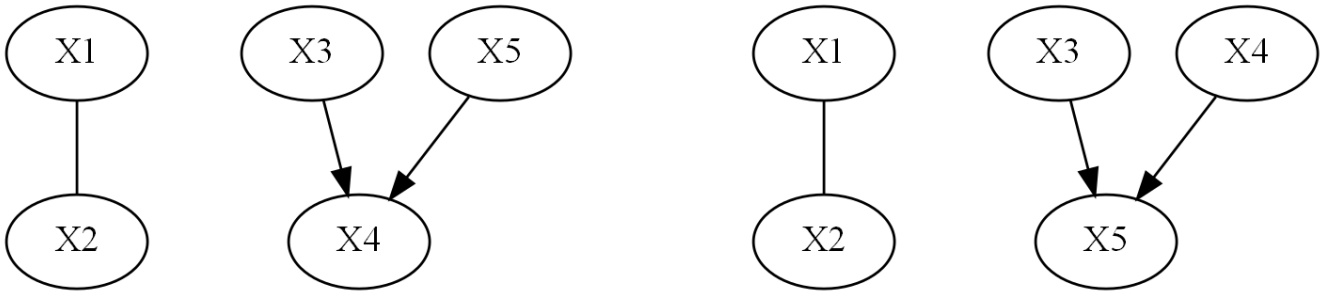
Algorithm 4: Initialization Algorithm

Data: Response matrix $X_{N\times J}$, number of latent variables K
Result: Initial parameters B, γ, A
Generate binary matrix A with all possible combinations of K binary variables;
Initialize $\gamma_{in} \in \mathbb{R}^J$ as zero vectors;
// Preprocessing;
$X_{inv} \leftarrow \log(X)$;
$X_{inv\_adj} \leftarrow X_{inv} - \text{mean}(X_{inv}, \text{axis} = 0)$;
$[U, S, V] \leftarrow \text{svd}(X_{inv\_adj})$;
// Varimax Rotation using MATLAB;
$R_V \leftarrow \text{MATLAB\_Varimax}(V, K)$;
// Threshold and Sign Adjustment;
$B_{est} \leftarrow R_V \cdot \mathbb{I}(|R_V| > threshold)$;
Adjust signs of B_{est} columns based on their mean values;
// Column Permutation;
Permute columns of B_{est} to maximize alignment with block diagonal structure;
$G_{est} \leftarrow \mathbb{I}(B_{est} \neq 0)$;
// Estimate A and B;
$A_{est} \leftarrow X_{inv\_adj} B_{est} (B_{est}^T B_{est})^{-1}$;
$A_{est} \leftarrow \mathbb{I}(A_{est} > 0)$;
$A_{centered} \leftarrow A_{est} - \mathbf{1} \cdot \text{mean}(A_{est}, \text{axis} = 0)$;
$B_{re\_est} \leftarrow ((A_{centered}^T A_{centered})^{-1} A_{centered}^T X_{inv\_adj})^T$;
$B_{re\_est} \leftarrow B_{re\_est} \cdot G_{est} \cdot \mathbb{I}(B_{re\_est} > 0)$;
$b \leftarrow \text{mean}(X_{inv\_adj}, \text{axis} = 0) - B_{re\_est} \cdot \text{mean}(A_{est}, \text{axis} = 0)$;
$B_{ini} \leftarrow [b | B_{re\_est}]$;
// Calculate Final Parameters;
$A_{long} \leftarrow [\mathbf{1} | A_{est}]$;
$Tr \leftarrow X_{inv} - A_{long} B_{ini}^T$;
for $j \in [J]$ **do**
  $\gamma_{in}[j] \leftarrow \frac{1}{N} \sum_{i=1}^N Tr_{ij}^2$;
end
return B_{ini}, γ_{in}, A_{est};

Remark 6. *It appears that GES does not perform well on Structure 3 for some specific parameters (*

- $P(X_1 = 1) = P(X_3 = 1) = P(X_4 = 1) = 0.5$
- $P(X_2 = 1 \mid X_1 = 1) = 0.6, P(X_2 = 1 \mid X_1 - 0) = 0.4$
- $P(X_5 = 1 \mid X_3 = 1, X_4 = 1) = 0.7, P(X_5 = 1 \mid X_3 = 1, X_4 = 0) = 0.3, P(X_5 = 1 \mid X_3 = 0, X_4 = 1) = 0.1, P(X_5 = 1 \mid X_3 = 0, X_4 = 0) = 0.9$

). The primary reason is its inability to distinguish between the following two graphs:



With $N = 1000000$, using Method 6, after 200 replications, 96 resulted in the left graph and 104 in the right graph.

Actually, this distribution contradicts Assumption 1. Note that $P(X_3 = 1, X_5 = 1) = 0.25 = 0.5 \cdot 0.5 = P(X_3 = 1)P(X_5 = 1)$...As a result, $X_3 \perp\!\!\!\perp X_5$ and this distribution is not DAG-perfect. However, if we keep the other parameters fixed while only changing $P(X_5 = 1 \mid X_3 = 1, X_4 = 1) = 0.8$, we can avoid this problem.

5 Modified PC Algorithm

We already know that PC algorithm itself does not depend on a specific type of conditional independence (CI) test. As long as the CI tests used are accurate, they should not affect the subsequent steps of the PC algorithm. This motivates us to adopt the general PC framework, with modifications to the CI test step tailored to our specific case.

The most commonly used CI tests are typically based on observed data, which will become infeasible in our case. However, we can emulate the form of the conditional mutual information test and calculate the following measure, "pseudo (conditional) mutual information".

$$\widetilde{I}(X;Y) = \sum_{x\in \mathrm{Val}(X)} \sum_{y\in \mathrm{Val}(Y)} \hat{p}(x,y) \log\left(\frac{\hat{p}(x,y)}{\hat{p}(x)\hat{p}(y)}\right),$$

$$\widetilde{I}(X;Y \mid Z) = \sum_{z\in \mathrm{Val}(Z)} \sum_{x\in \mathrm{Val}(X)} \sum_{y\in \mathrm{Val}(Y)} \hat{p}(x,y,z) \log\left(\frac{\hat{p}(x,y \mid z)}{\hat{p}(x \mid z)\hat{p}(y \mid z)}\right)$$

Returning to the main discussion, consider an ExpACDM with true parameters (p_0, β_0, γ_0) where the generic identifiability conditions hold. Now suppose we have observed responses X which are N independent and identically distributed response vectors from the ExpACDM stated before. Then we apply EM algorithm to X ($N \times J$ matrix) to obtain an estimate of p_0, $\hat{p}_N$, which is sufficient for implementing modified PC algorithm.

Recall that in previous analysis, we already know that $\hat{p}_N(A = \alpha) = p_0(A = \alpha) + O_p(\frac{1}{\sqrt{N}})$ for any $\alpha \in \{0,1\}^K$. It follows directly that $\hat{p}_N(A_{sub} = \alpha_{sub}) = p_0(A_{sub} = \alpha_{sub}) + O_P(\frac{1}{\sqrt{N}})$ through simple addition due to the fact that $K << N$, where A_{sub} and α_{sub} are ordered subsets of A and α, respectively. (For example, we have $\hat{p}_N(a_i = 1) = p_0(a_i = 1) + O_P(\frac{1}{\sqrt{N}})$ for $i = 1, \ldots, K$)

By definition, it is easy to see

$$\begin{aligned}
\widetilde{I}_N(a_i, a_j) &= \sum_{x=0}^{1}\sum_{y=0}^{1} \hat{p}_N(a_i = x, a_j = y) \log(\frac{\hat{p}_N(a_i = x, a_j = y)}{\hat{p}_N(a_i = x)\hat{p}_N(a_j = y)}) \\
&= \sum_{x=0}^{1}\sum_{y=0}^{1} (p_0(a_i = x, a_j = y) + O_P(\frac{1}{\sqrt{N}})) \log(\frac{p_0(a_i = x, a_j = y) + O_P(\frac{1}{\sqrt{N}}}{(p_0(a_i = x) + O_P(\frac{1}{\sqrt{N}})(p_0(a_j = y) + O_P(\frac{1}{\sqrt{N}})}) \\
&= I(a_i; a_j) + O_P(\frac{1}{\sqrt{N}})
\end{aligned}$$

Similarly, we have $\widetilde{I}_N(a_i, a_j \mid S) = I(a_i; a_j \mid S) + O_P(\frac{1}{\sqrt{N}})$ where S is a subset of $\{a_1, \ldots, a_K\}$. Consequently, if we select a cut-off τ_N of order greater than $\frac{1}{\sqrt{N}}$, then we can easily obtain the following algorithm. Specifically, we can choose $\tau_N = \frac{1}{N^{2.01}}$.

Algorithm 5: PC Algorithm with Pseudo Mutual Information Test for Skeleton Learning

Input: $a_1, \ldots, a_K, \hat{p}_N, \tau_N$
Output: A CPDAG
Step 1: Initialize complete graph ;
Initialize a complete undirected graph G with nodes $a_1, \ldots, a_K$;
Step 2: Skeleton Identification using Pseudo Conditional Mutual Information ;
foreach *pair of variables* (a_i, a_j) **do**
  if $\widetilde{I}(a_i; a_j) < \tau_N$ **then**
    Remove edge between a_i and a_j in G
  end
end
for $k = 1$ **to** $\max($*number of neighbors*$)$ **do**
  foreach *pair of variables* (a_i, a_j) *still connected* **do**
    foreach *subset* $S \subseteq adj(a_i) \cup adj(a_j)$*, with* $|S| = k$ **do**
      if $\widetilde{I}(a_i; a_j \mid S) < \tau_N$ **then**
        Remove edge between a_i and a_j in G
      end
    end
  end
end
Step 3: Edge Orientation ;
foreach *triple* (a_i, a_j, a_k) *where* a_i *and* a_k *are not adjacent, but both are adjacent to* a_j **do**
  if $a_i \perp a_k \mid \emptyset$ *and* $X_i \not\perp a_k \mid a_j$ **then**
    Orient edges as $a_i \rightarrow a_j \leftarrow a_k$
  end
end
Apply Meek's rules to further orient edges ;
return *The resulting CPDAG*

Nevertheless, this method seems impractical in practice. Even when we input the "true" p_0(frequencies of each type of α in unobserved A) for Structure 2, the output remains incorrect until $N = 2000000$ as we continue to increase N.

Remark 7. *In practice, the computed mutual information* $I(X;Y)$ *is scaled to a chi-square statistic:*

$$\chi^2 = 2N \cdot I(X;Y)$$

where N *is the sample size. This statistic is then compared to the chi-square distribution with degrees of freedom* $(|X| - 1)(|Y| - 1)$*, where* $|X|$ *and* $|Y|$ *are the number of possible values for* X *and* Y*. Independence is rejected if:*

$$\chi^2 > \chi^2_{\alpha,(|X|-1)(|Y|-1)}$$

For the mutual information $I(X;Y)$ *to exceed this threshold, it typically must scale as* $I(X;Y) \sim \frac{1}{N}$*, making it inversely proportional to the sample size.*

References

[1] Optimal Structure Identification With Greedy Search, David Maxwell Chickering, JMLR

[2] New Paradigm of Identifiable General-response Cognitive Diagnostic Models: Beyond Categorical Data Seunghyun Lee and Yuqi Gu, Psychometrika

[3] A note on Exploratory Item Factor Analysis by Singular Value Decomposition, Haoran Zhang, Yunxiao Chen and Xiaoou Li

[4] Probabilistic Graphical Models: Principles and Techniques, Koller and Friedman