Copula PC Algorithm for Causal Discovery from Mixed Data

Ruifei Cui, Perry Groot, and Tom Heskes

{R.Cui,Perry.Groot,T.Heskes}@science.ru.nl

Institute for Computing and Information Sciences – Data Science
Radboud University Nijmegen

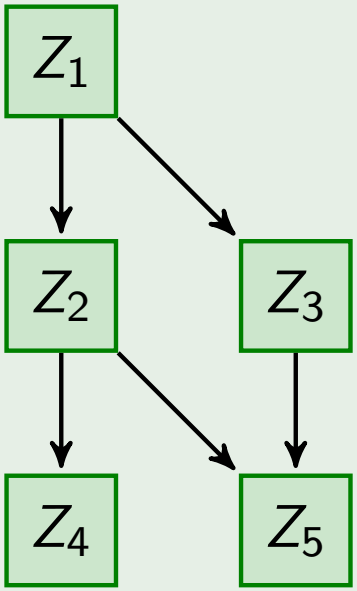
16th September 2016

Outline

Outline

Preliminaries – Causal Discovery

Example

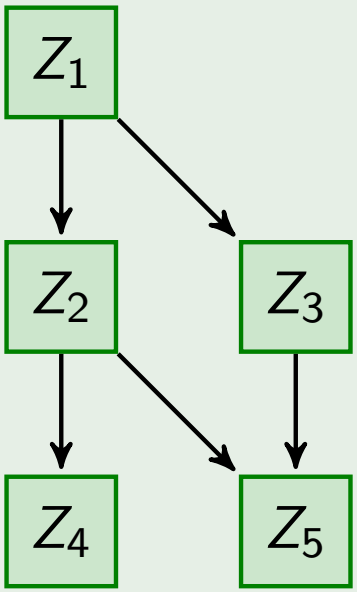


- Vertices set $\boldsymbol{V} = \{Z_1, \ldots, Z_5\}$ represents random variables, and edges set $\boldsymbol{E}$ represents relations between pairs of variables.
- Given the graph, the data are assumed to be generated via

$$
\begin{aligned}
Z_1 &= \epsilon_1; \\
Z_2 &= aZ_1 + \epsilon_2; \\
Z_3 &= bZ_1 + \epsilon_3; \\
Z_4 &= cZ_2 + \epsilon_4; \\
Z_5 &= dZ_2 + eZ_3 + \epsilon_5.
\end{aligned}
$$

Preliminaries – Causal Discovery

Example



- Vertices set $\boldsymbol{V} = \{Z_1, \ldots, Z_5\}$ represents random variables, and edges set $\boldsymbol{E}$ represents relations between pairs of variables.
- Given the graph, the data are assumed to be generated via

$$
\begin{aligned}
Z_1 &= \epsilon_1; \\
Z_2 &= aZ_1 + \epsilon_2; \\
Z_3 &= bZ_1 + \epsilon_3; \\
Z_4 &= cZ_2 + \epsilon_4; \\
Z_5 &= dZ_2 + eZ_3 + \epsilon_5.
\end{aligned}
$$

- Causal discovery aims to infer the (invariant parts of the) underlying graph from observational data.

Preliminaries – the PC Algorithm

The PC algorithm is a reference causal discovery algorithm.

Preliminaries – the PC Algorithm

The PC algorithm is a reference causal discovery algorithm.

The basic procedure

1. start from a fully connected undirected graph;
2. remove edges according to conditional independence tests, resulting in an undirected graph, called skeleton;
3. apply various edge orientation rules, resulting in a partially directed graph, called CPDAG.

Preliminaries – the PC Algorithm for Gaussian data

How to conduct conditional independence tests

- Based on partial correlations, denoted by $\rho_{uv|S}$: when $\boldsymbol{Z} \sim \mathcal{N}(0, C)$,
  - $Z_u \perp\!\!\!\perp Z_v | \boldsymbol{Z}_S \Leftrightarrow \rho_{uv|S} \to 0 \Rightarrow$ remove edge $Z_u - Z_v$,
  - $\rho_{uv|S}$ can be computed from the correlation matrix C,
  - Using standard hypothesis testing with significance level α, the conditional independence tests boils down to

$$Z_u \perp\!\!\!\perp Z_v | \boldsymbol{Z}_S \Leftrightarrow \sqrt{n - |S| - 3} \left| \frac{1}{2} \log \left(\frac{1+\hat{\rho}_{uv|S}}{1-\hat{\rho}_{uv|S}} \right) \right| \leq \Phi^{-1}(1 - \alpha/2).$$

Preliminaries – Gaussian Copula Model

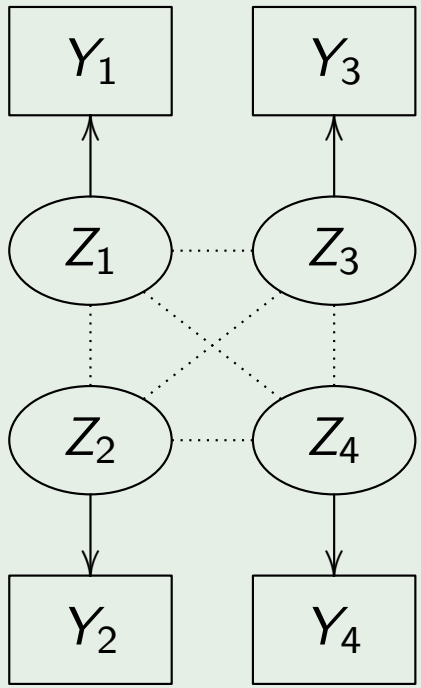
Definition (Gaussian Copula Model)

Consider two random vectors $\boldsymbol{Z} = (Z_1, \ldots, Z_p)$ and $\boldsymbol{Y} = (Y_1, \ldots, Y_p)$, satisfying the conditions

1. $\boldsymbol{Z} \sim \mathcal{N}(0, C)$ (latent)
2. $Y_i = F_i^{-1}(\Phi(Z_i))$ for $i = 1 : p$

where ${F_i}^{-1}(t)$ is the pseudo-inverse of a cumulative distribution function F_i. Then this model is called a *Gaussian copula model* with correlation matrix C and univariate margins F_i.

Example



Gaussian Copula Model.

Problem Analysis

Challenges current PC algorithms face

- Rank PC, a modification of standard PC, uses rank correlation instead of Pearson correlation and works fine when all F_i in the Gaussian copula model are continuous.
- However, the presence of discrete margins brings on two challenges.

Problem Analysis

Challenge 1

Ties make $rank(Y_i)$ different from $rank(Z_i)$, so that

$$Rcorr(\boldsymbol{Y}) \neq Rcorr(\boldsymbol{Z}) \approx C.$$

So, we can no longer use the rank correlations on the observed data as an estimate of the latent correlations.

Problem Analysis

Challenge 1

Ties make $rank(Y_i)$ different from $rank(Z_i)$, so that

$$Rcorr(\boldsymbol{Y}) \neq Rcorr(\boldsymbol{Z}) \approx C.$$

So, we can no longer use the rank correlations on the observed data as an estimate of the latent correlations.

How to estimate the underlying correlation matrix from mixed data?

Problem Analysis

Challenge 2

- Discrete variables incur some information loss.
- This needs to be taken into account when applying conditional independence tests.

Problem Analysis

Challenge 2

- Discrete variables incur some information loss.
- This needs to be taken into account when applying conditional independence tests.

We therefore introduce the notion of an "effective number" of data points, denoted by $\hat{n}(\leq n)$:

Problem Analysis

Challenge 2

- Discrete variables incur some information loss.
- This needs to be taken into account when applying conditional independence tests.

We therefore introduce the notion of an "effective number" of data points, denoted by $\hat{n}(\leq n)$:

$$Z_u \perp\!\!\!\perp Z_v | Z_S \Leftrightarrow \sqrt{\hat{n} - |S| - 3} \left| \frac{1}{2} \log \left(\frac{1 + \hat{\rho}_{uv|S}}{1 - \hat{\rho}_{uv|S}} \right) \right| \leq \Phi^{-1}(1 - \alpha/2)$$

Problem Analysis

Challenge 2

- Discrete variables incur some information loss.
- This needs to be taken into account when applying conditional independence tests.

We therefore introduce the notion of an "effective number" of data points, denoted by $\hat{n}(\leq n)$:

$$Z_u \perp\!\!\!\perp Z_v | Z_S \Leftrightarrow \sqrt{\hat{n} - |S| - 3} \left| \frac{1}{2} \log \left(\frac{1 + \hat{\rho}_{uv|S}}{1 - \hat{\rho}_{uv|S}} \right) \right| \leq \Phi^{-1}(1 - \alpha/2)$$

How to estimate the effective number of data points from mixed data?

Outline

A Two-step Approximate Inference Method

A Two-step Approximate Inference Method

Definition (Projected Inverse Wishart Distribution)

If Σ follows an inverse-Wishart $\mathcal{W}^{-1}(\Sigma; \Psi, \nu)$ and C is the corresponding correlation matrix, then C follows a projected inverse-Wishart:

$$P(C) = \mathcal{PW}^{-1}(C; \Psi, \nu)\,.$$

A Two-step Approximate Inference Method

Using Bayesian framework, we usually choose the prior for correlation matrix to be

$$P(C) = \mathcal{PW}^{-1}(C; \Psi_0, \nu_0).$$

A Two-step Approximate Inference Method

Using Bayesian framework, we usually choose the prior for correlation matrix to be

$$P(C) = \mathcal{PW}^{-1}(C; \Psi_0, \nu_0).$$

- In fully Gaussian cases, we have exact inference:

$$P(C|\boldsymbol{Z}) = \mathcal{PW}^{-1}(C; \Psi_0 + \boldsymbol{Z}^T\boldsymbol{Z}, \nu_0 + n)\,.$$

A Two-step Approximate Inference Method

Using Bayesian framework, we usually choose the prior for correlation matrix to be

$$P(C) = \mathcal{PW}^{-1}(C; \Psi_0, \nu_0).$$

- In fully Gaussian cases, we have exact inference:

$$P(C|\boldsymbol{Z}) = \mathcal{PW}^{-1}(C; \Psi_0 + \boldsymbol{Z}^T\boldsymbol{Z}, \nu_0 + n)\,.$$

- In Gaussian copula cases, we cannot get the exact inference, but it is easy to draw samples from the posterior distribution $P(C|\boldsymbol{Y})$ in some way, e.g., Hoff's Gibbs sampler based on the extended rank likelihood.

$$C^{(1)}, \ldots, C^{(m)} \leftarrow \mathit{Gibbs\_sampler}(\boldsymbol{Y})$$

A Two-step Approximate Inference Method

- For a Gaussian copula model, we assume:

$$P(C|\boldsymbol{Y}) \approx \mathcal{PW}^{-1}(C; \hat{C}, \hat{n}) \text{ for some } \hat{C}, \hat{n}\,.$$

- Then, we can estimate the two parameters $\hat{C}$ and $\hat{n}$ from the Gibbs samples $C^{(1)}, \ldots, C^{(m)}$.

A Two-step Approximate Inference Method

- For a Gaussian copula model, we assume:

$$P(C|\boldsymbol{Y}) \approx \mathcal{PW}^{-1}(C; \hat{C}, \hat{n}) \text{ for some } \hat{C}, \hat{n}\,.$$

- Then, we can estimate the two parameters $\hat{C}$ and $\hat{n}$ from the Gibbs samples $C^{(1)}, \ldots, C^{(m)}$.
- How to estimate?

A Two-step Approximate Inference Method

We proved a property of the projected inverse-Wishart Distribution.

Theorem

If the correlation matrix C follows a projected inverse-Wishart distribution with parameters $\Psi(\Psi_{ii} = 1)$ and ν, i.e.,

$$P(C) = \mathcal{PW}^{-1}(C; \Psi, \nu),$$

then, for each off-diagonal element $C_{ij}(i \neq j)$ and large ν, we have

$$\mathrm{E}[C_{ij}] \approx \Psi_{ij} \quad and \quad \mathrm{Var}[C_{ij}] \approx \frac{(1 - (\mathrm{E}[C_{ij}])^2)^2}{\nu}.$$

A Two-step Approximate Inference Method

- According to the property, we have
  - $\hat{C} \approx \frac{1}{m} \sum_{k=1}^{m} C^{(k)}$
  - $\hat{n} \approx \frac{1}{p(p-1)} \sum_{i \neq j} \nu_{ij}$, where $\nu_{ij} \approx \frac{(1-(\mathrm{E}[C_{ij}])^2)^2}{\mathrm{Var}[C_{ij}]}$

A Two-step Approximate Inference Method

- According to the property, we have
  - $\hat{C} \approx \frac{1}{m} \sum_{k=1}^{m} C^{(k)}$
  - $\hat{n} \approx \frac{1}{p(p-1)} \sum_{i \neq j} \nu_{ij}$, where $\nu_{ij} \approx \frac{(1-(\mathrm{E}[C_{ij}])^2)^2}{\mathrm{Var}[C_{ij}]}$
- Take the two parameters estimated above as the two inputs of the standard PC algorithm, resulting in the Copula PC algorithm, simply denoted by

$$CoPC = pc(\hat{C}, \hat{n}) .$$

A Two-step Approximate Inference Method

- According to the property, we have
  - $\hat{C} \approx \frac{1}{m} \sum_{k=1}^{m} C^{(k)}$
  - $\hat{n} \approx \frac{1}{p(p-1)} \sum_{i \neq j} \nu_{ij}$, where $\nu_{ij} \approx \frac{(1-(\mathrm{E}[C_{ij}])^2)^2}{\mathrm{Var}[C_{ij}]}$
- Take the two parameters estimated above as the two inputs of the standard PC algorithm, resulting in the Copula PC algorithm, simply denoted by

$$CoPC = pc(\hat{C}, \hat{n}) .$$

- We can also use these samples $C^{(1)}, \ldots, C^{(m)}$ to do causal discovery in another strategy: Stable Copula PC algorithm.

Stable Copula PC Algorithm

Stable Copula PC Algorithm

- Choose $l (l < m)$ instances from $C^{(1)}, \ldots, C^{(m)}$;
- Compute and store $\widetilde{G}_i \longleftarrow pc(C^i, \hat{n})$ for $i = 1 : l$, resulting in $\{\widetilde{G}_1, \ldots, \widetilde{G}_l\}$;
- $\widetilde{G}_s \longleftarrow$ Only keep the edges that occur more frequent than the pre-defined threshold among $\{\widetilde{G}_1, \ldots, \widetilde{G}_l\}$.

Outline

Causal Discovery on Simulations

We compare the Rank PC, Copula PC, and Stable Copula PC on simulated data following general Gaussian Copula distribution.

Causal Discovery on Simulations

We compare the Rank PC, Copula PC, and Stable Copula PC on simulated data following general Gaussian Copula distribution.

Simulated Data

$$G \longrightarrow C \longrightarrow \boldsymbol{Z} \sim \mathcal{N}(0, C) \longrightarrow \boldsymbol{Y}$$

- Given a DAG G, it implies a correlation matrix C;
- Draw n data points of $\boldsymbol{Z}$ (with p variables) from $\mathcal{N}(0, C)$;
- $\frac{1}{4}$ of p are discretized into binary variables, another $\frac{1}{4}$ into ordinal with 5 levels, the remaining half are still continuous, resulting in the simulations of $\boldsymbol{Y}$;
- The resulting data follows a general Gaussian Copula distribution with both discrete and continuous margins.

Causal Discovery on Simulations

We compare the Rank PC, Copula PC, and Stable Copula PC on simulated data following general Gaussian Copula distribution.

Measures for Testing the Performance

- True Positive Rate (TPR), percentage of correct edges in the resulting skeleton.
- False Positive Rate (FPR), percentage of spurious edges in the resulting skeleton.
- Structural Hamming Distance (SHD), counting the number of edge insertions, deletions, and flips in order to transfer the estimated CPDAG into the correct CPDAG.
- TPR and FPR are for the skeleton while SHD is for the CPDAG (small value means good performance).

Causal Discovery on Simulations

We compare the Rank PC, Copula PC, and Stable Copula PC on simulated data following general Gaussian Copula distribution.

Test Four Situations

- We choose $\mathrm{E}[N] \in \{2\ (\textit{Sparse}), 5\ (\textit{Dense})\}$.
  $\mathrm{E}[N]$ is the average neighborhood size which represents the sparseness of a graph.
- Choose $p \in \{10\ (\textit{small}), 50\ (\textit{large})\}$.
- The different combination of sparseness and dimensionality results in four situations.

Causal Discovery on Simulations – Results on 10 nodes

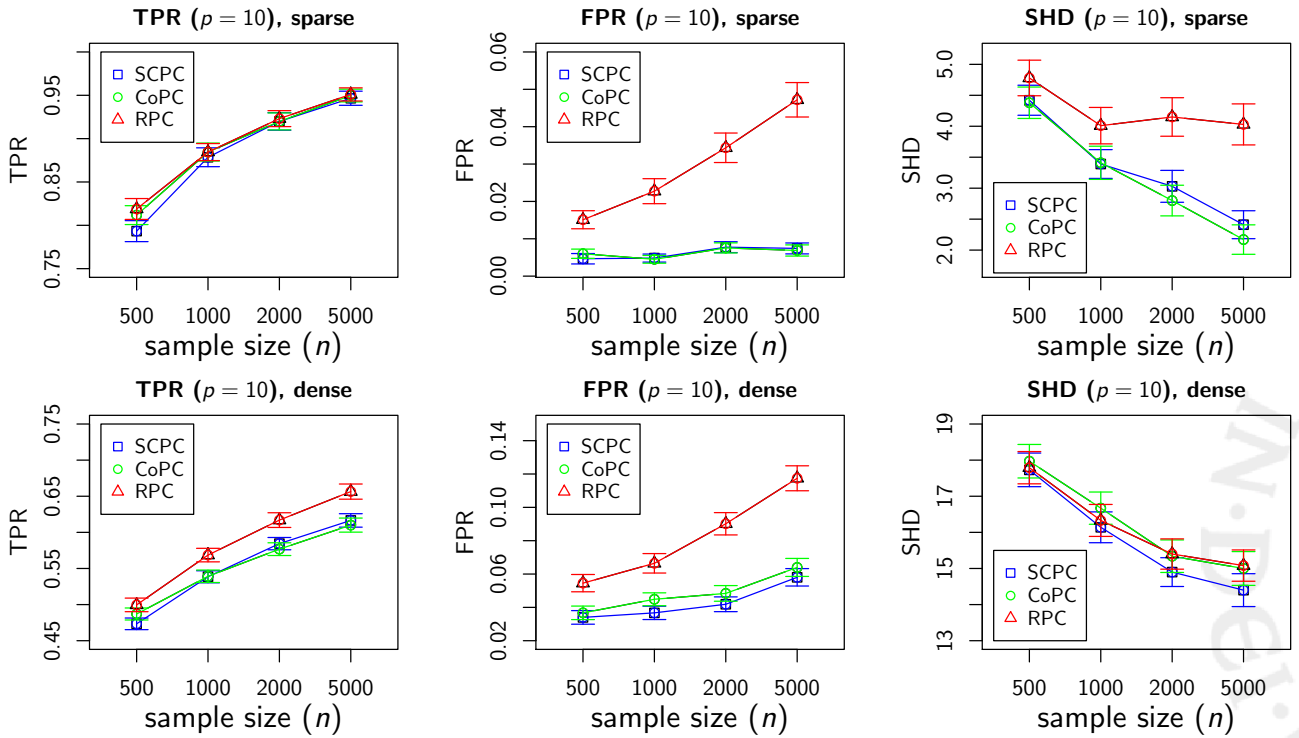


Figure: Performance of Rank PC, Copula PC, and Stable Copula PC for 10 nodes, showing the mean of TPR, FPR, and SHD over 100 experiments together with 95% confidence intervals.

Causal Discovery on Simulations – Results on 50 nodes

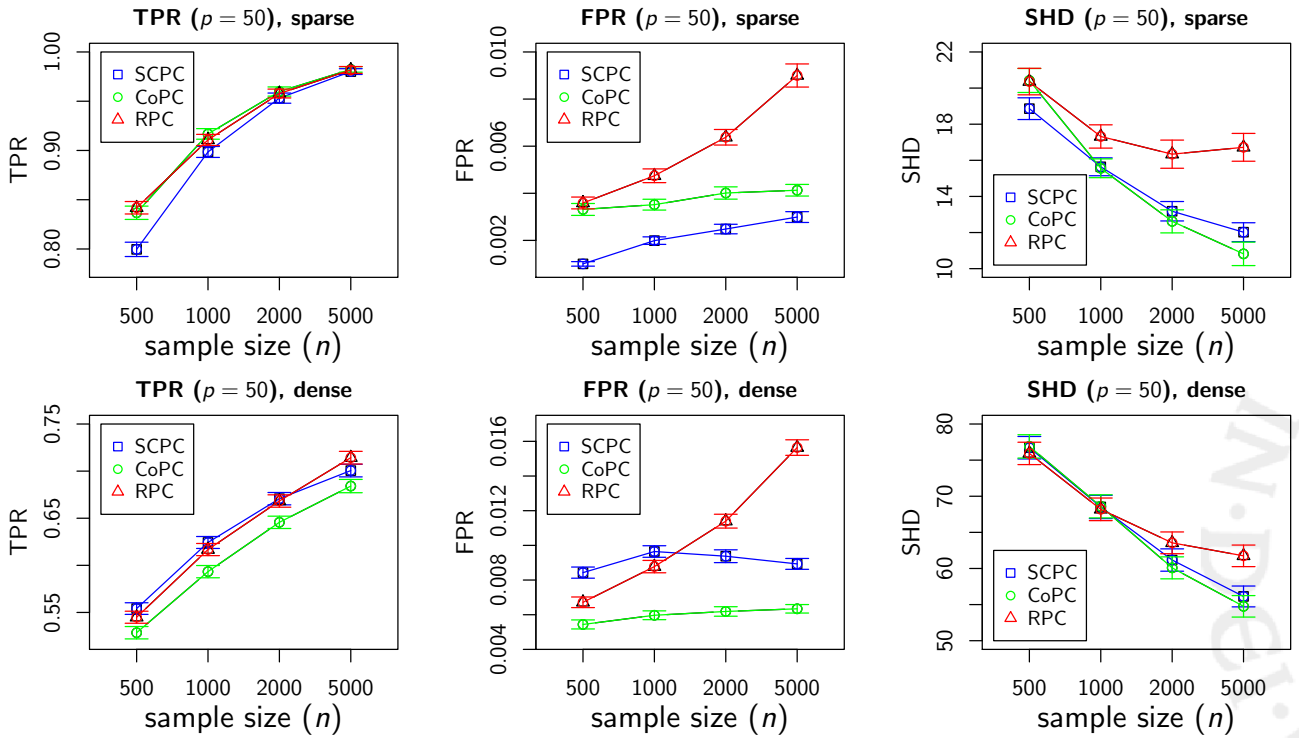


Figure: Performance of Rank PC, Copula PC, and Stable Copula PC for 50 nodes, showing the mean of TPR, FPR, and SHD over 100 experiments together with 95% confidence intervals.

Experimental Conclusions

- For sparse graphs, Copula PC and Stable Copula PC show a large advantage over Rank PC, which becomes more prominent with increasing sample size.

Experimental Conclusions

- For sparse graphs, Copula PC and Stable Copula PC show a large advantage over Rank PC, which becomes more prominent with increasing sample size.
- For dense graphs, the advantage still exists, although smaller than sparse graphs.

Experimental Conclusions

- For sparse graphs, Copula PC and Stable Copula PC show a large advantage over Rank PC, which becomes more prominent with increasing sample size.
- For dense graphs, the advantage still exists, although smaller than sparse graphs.
- Overall, Copula PC and Stable Copula PC outperform Rank PC, especially in the sparse cases with larger sample sizes.

Outline

Summary

- We introduced a novel two-step approach for estimating the causal structure underlying a Gaussian copula model on mixed data.
  1. draw samples on correlation matrix from $P(C|\boldsymbol{Y})$ via Gibbs sampler based on extended rank likelihood;
  2. estimate the underlying correlation matrix and the effective number of data points, assuming these samples follow a projected inverse-Wishart distribution.
- The experimental results show that Copula PC algorithm and stable Copula PC algorithm outperform the current Rank PC algorithm in the presence of discrete margins.

Thanks for your attention!
Any questions?