latentcor

Mingze Huang, Irina Gaynanova, Christian L. Müller

2021-07-19

Introduction

R package `latentcor` utilizes the powerful semi-parametric latent Gaussian copula models to estimating latent correlations between mixed data types. The package allows to estimate correlations between any of continuous/binary/ternary/zero-inflated (truncated) variable types. The underlying implementation takes advantage of fast multi-linear interpolation scheme with a clever choice of grid points that give the package a small memory footprint, and allows to use the latent correlations with sub-sampling and bootstrapping.

A Simple Example

In this example, we will generate two variables with different data types. Each variable has 100 observations. First variable will be ternary, second variable will be continuous.

```
sampledata = GenData(n = 100, types = c("ter", "con"))
```

The `sampledata` is a list with several elements:

- `X`: a matrix (100×2), the first column is the ternary variable; the second column is the continuous variable.
- `plotX`: NULL

```
X = sampledata$X
```

`X` is just the input matrix for estimation.

Then we can estimate latent correlation matrix of these 2 variables.

```
estimate = estR(X, types = c("ter", "con"))
```

`estimate` is a list with several elements:

- zratios: a list of zratios. The first element of the list is a (2×1) vector indicates the cumulative proportions for zeros and ones in the ternary variable (e.g. first element in vector is the proportion of zeros, second element in vector is the proportion of zeros and ones.) The second element of the list is NA for continuous variable.
- K: Kendall τ (τ_a) correlation matrix for these 2 variables.
- R: estimated latent correlation matrix of these 2 variable.
- plotR: NULL

Thus, the latent correlation matrix for these 2 variable is `R`.

```
R = estimate$R
```

Another example

Latent Correlation of Latent Gaussian Copula Model

Latent Gaussian Copula Model for Mixed Data

Definition 1 (Continuous model) A random $X \in \mathcal{R}^{\checkmark}$ satisfies the Gaussian copula model if there exist monotonically increasing $f = (f_j)_{j=1}^{p}$ with $Z_j = f_j(X_j)$ satisfying $Z \sim N_p(0, \Sigma)$, $\sigma_{jj} = 1$; $X \sim NPN(0, \Sigma, f)$.

Definition 2 (Binary model) A random $X \in \mathcal{R}^{\checkmark}$ satisfies the binary latent Gaussian copula model if there exists $W \sim NPN(0, \Sigma, f)$ such that $X_j = I(W_j > c_j)$, where $I(\cdot)$ is the indicator function and c_j are constants.

Definition 3 (Truncated model) A random $X \in \mathcal{R}^{\checkmark}$ satisfies the truncated latent Gaussian copula model if there exists $W \sim NPN(0, \Sigma, f)$ such that $X_j = I(W_j > c_j)W_j$, where $I(\cdot)$ is the indicator function and c_j are constants.

Definition 4 (Ternary model) A random $X \in \mathcal{R}^{\checkmark}$ satisfies the binary latent Gaussian copula model if there exists $W \sim NPN(0, \Sigma, f)$ such that $X_j = I(W_j > c_j) + I(W_j > c'_j)$, where $I(\cdot)$ is the indicator function and $c_j < c'_j$ are constants.

Mixed Latent Gaussian Copula Model The mixed latent Gaussian copula model jointly models $W = (W_1, W_2, W_3, W_4) \sim NPN(0, \Sigma, f)$ such that $X_{1j} = W_{1j}$, $X_{2j} = I(W_{2j} > c_{2j})$, $X_{3j} = I(W_{3j} > c_{3j})W_{3j}$ and $X_{4j} = I(W_{4j} > c_{4j}) + I(W_{4j} > c'_{4j})$.

Bridge Functions

Estimation of latent correlations is achieved via the bridge function F such that $E(\hat{\tau}_{jk}) = F(\sigma_{jk})$, where σ_{jk} is the latent correlation between variables j and k, and $\hat{\tau}_{jk}$ is the corresponding sample Kendall's τ. Given observed $\mathbf{x}_j, \mathbf{x}_k \in \mathcal{R}^{\backslash}$,

$$\hat{\tau}_{jk} = \hat{\tau}(\mathbf{x}_j, \mathbf{x}_k) = \frac{2}{n(n-1)} \sum_{1 \le i < i' \le n} sign(x_{ij} - x_{i'j}) sign(x_{ik} - x_{i'k})$$

where n is the sample size. Using F one can construct $\hat{\sigma}_{jk} = F^{-1}(\hat{\tau}_{jk})$ with the corresponding estimator $\hat{\Sigma}$ being consistent for Σ (Fan et al. 2017; Quan, Booth, and Wells 2018; Yoon, Müller, and Gaynanova 2021). The explicit form of F has been derived for all combinations of continuous(C)/binary(B)/truncated(T)/ternary(N) variables (Fan et al. 2017; Yoon, Müller, and Gaynanova 2021).

Theorem 1 Let $W_1 \in \mathcal{R}^{\checkmark^{\infty}}$, $W_2 \in \mathcal{R}^{\checkmark^{\in}}$, $W_3 \in \mathcal{R}^{\checkmark^{\ni}}$, $W_4 \in \mathcal{R}^{\checkmark^{\triangle}}$ be such that $W = (W_1, W_2, W_3, W_4) \sim NPN(0, \Sigma, f)$ with $p = p_1 + p_2 + p_3 + p_4$. Let $X = (X_1, X_2, X_3, X_4) \in \mathcal{R}^{\checkmark}$ satisfy $X_j = W_j$ for $j = 1, ..., p_1$, $X_j = I(W_j > c_j)$ for $j = p_1 + 1, ..., p_1 + p_2$, $X_j = I(W_j > c_j)W_j$ for $j = p_1 + p_2 + 1, ..., p_3$ and $X_j = I(W_j > c_j) + I(W_j > c'_j)$ for $j = p_1 + p_2 + p_3 + 1, ..., p$ with $\Delta_j = f(c_j)$. The rank-based estimator of Σ based on the observed n realizations of X is the matrix $\hat{\mathbf{R}}$ with $\hat{r}_{jj} = 1$, $\hat{r}_{jk} = \hat{r}_{kj} = F^{-1}(\hat{\tau}_{jk})$ with block structure

$$\hat{\mathbf{R}} = \begin{pmatrix} F_{CC}^{-1}(\hat{\tau}) & F_{CB}^{-1}(\hat{\tau}) & F_{CT}^{-1}(\hat{\tau}) \\ F_{BC}^{-1}(\hat{\tau}) & F_{BB}^{-1}(\hat{\tau}) & F_{BT}^{-1}(\hat{\tau}) \\ F_{TC}^{-1}(\hat{\tau}) & F_{TB}^{-1}(\hat{\tau}) & F_{TT}^{-1}(\hat{\tau}) \end{pmatrix}$$

$$F_{CC}(r) = \frac{2}{\pi} sin^{-1}(r)$$

$$F_{BC}(r; \Delta_j) = 4\Phi_2(\Delta_j, 0; \frac{r}{\sqrt{2}}) - 2\Phi(\Delta_j)$$

$$F_{BB}(r;\Delta_j,\Delta_k) = 2\{\Phi_2(\Delta_j,\Delta_k;r) - \Phi(\Delta_j)\Phi(\Delta_k)\}$$
$$F_{TC}(r;\Delta_j) = -2\Phi_2(-\Delta_j,0;\frac{1}{\sqrt{2}}) + 4\Phi_3(-\Delta_j,0,0;\Sigma_3(r))$$
$$F_{TB}(r;\Delta_j,\Delta_k) = 2\{1-\Phi(\Delta_j)\}\Phi(\Delta_k) - 2\Phi_3(-\Delta_j,\Delta_k,0;\Sigma_{3a}(r)) - 2\Phi_3(-\Delta_j,\Delta_k,0;\Sigma_{3b}(r))$$
$$F_{TT}(r;\Delta_j,\Delta_k) = -2\Phi_4(-\Delta_j,-\Delta_k,0,0;\Sigma_{4a}(r)) + 2\Phi_4(-\Delta_j,-\Delta_k,0,0;\Sigma_{4b}(r))$$
$$F_{NC}(r;\Delta_j^1,\Delta_j^2) = 4\Phi_2(\Delta_j^2,0;\frac{r}{\sqrt{2}}) - 2\Phi(\Delta_j^2) + 4\Phi_3(\Delta_j^1,\Delta_j^2,0;\Sigma_{3c}(r)) - 2\Phi(\Delta_j^1)\Phi(\Delta_j^2)$$
$$F_{NB}(r;\Delta_j^1,\Delta_j^2,\Delta_k) = 2\Phi_2(\Delta_j^2,\Delta_k,r)(1-\Phi(\Delta_j^1)) - 2\Phi(\Delta_j^2)(\Phi(\Delta_k) - \Phi_2(\Delta_j^1,\Delta_k,r))$$

$$\begin{aligned} F_{NT}(r;\Delta_j^1,\Delta_j^2,\Delta_k) =& -2\Phi(-\Delta_j^1)\Phi(\Delta_j^2) + 2\Phi_3(-\Delta_j^1,\Delta_j^2,\Delta_k;\Sigma_{3e}(r)) \\ &+ 2\Phi_4(-\Delta_j^1,\Delta_j^2,-\Delta_k,0;\Sigma_{4c}(r)) + 2\Phi_4(-\Delta_j^1,\Delta_j^2,-\Delta_k,0;\Sigma_{4d}(r)) \end{aligned} \quad (1)$$

$$\begin{aligned} F_{NN}(r;\Delta_j^1,\Delta_j^2,\Delta_k^1,\Delta_k^2) =& 2\Phi_2(\Delta_j^2,\Delta_k^2;r)\Phi_2(-\Delta_j^1,-\Delta_k^1;r) \\ &- 2[\Phi(\Delta_j^2) - \Phi_2(\Delta_j^2,\Delta_k^1;r)][\Phi(\Delta_k^2 - \Phi_2(\Delta_j^1,\Delta_k^2;r))] \end{aligned} \quad (2)$$

with $\Delta_j = \Phi^{-1}(\pi_{0j})$, $\Delta_k = \Phi^{-1}(\pi_{0k})$, $\Delta_j^1 = \Phi^{-1}(\pi_{0j})$, $\Delta_j^2 = \Phi^{-1}(\pi_{0j}+\pi_{1j})$, $\Delta_k^1 = \Phi^{-1}(\pi_{0k})$, $\Delta_k^2 = \Phi^{-1}(\pi_{0k}+\pi_{1k})$ and

$$\Sigma_3(r) = \begin{pmatrix} 1 & \frac{1}{\sqrt{2}} & \frac{r}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 1 & r \\ \frac{r}{\sqrt{2}} & 2 & 1 \end{pmatrix}, \qquad \Sigma_{3a}(r) = \begin{pmatrix} 1 & -r & \frac{1}{\sqrt{2}} \\ -r & 1 & -\frac{r}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{r}{\sqrt{2}} & 1 \end{pmatrix}$$
$$\Sigma_{3b}(r) = \begin{pmatrix} 1 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & -\frac{r}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & -\frac{r}{\sqrt{2}} & 1 \end{pmatrix}, \qquad \Sigma_{3c}(r) = \begin{pmatrix} 1 & 0 & \frac{r}{\sqrt{2}} \\ 0 & 1 & -\frac{r}{\sqrt{2}} \\ \frac{r}{\sqrt{2}} & -\frac{r}{\sqrt{2}} & 1 \end{pmatrix}$$
$$\Sigma_{3d}(r) = \begin{pmatrix} 1 & 0 & -r \\ 0 & 1 & 0 \\ -r & 0 & 1 \end{pmatrix}, \qquad \Sigma_{3e}(r) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & r \\ 0 & r & 1 \end{pmatrix}$$
$$\Sigma_{4a}(r) = \begin{pmatrix} 1 & 0 & \frac{1}{\sqrt{2}} & -\frac{r}{\sqrt{2}} \\ 0 & 1 & -\frac{r}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{r}{\sqrt{2}} & 1 & -r \\ -\frac{r}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -r & 1 \end{pmatrix}, \qquad \Sigma_{4b}(r) = \begin{pmatrix} 1 & r & \frac{1}{\sqrt{2}} & \frac{r}{\sqrt{2}} \\ r & 1 & \frac{r}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{r}{\sqrt{2}} & 1 & r \\ \frac{r}{\sqrt{2}} & \frac{1}{\sqrt{2}} & r & 1 \end{pmatrix}$$
$$\Sigma_{4c}(r) = \begin{pmatrix} 1 & 0 & 0 & \frac{r}{\sqrt{2}} \\ 0 & 1 & -r & \frac{r}{\sqrt{2}} \\ 0 & -r & 1 & -\frac{1}{\sqrt{2}} \\ \frac{r}{\sqrt{2}} & \frac{r}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 1 \end{pmatrix}, \qquad \Sigma_{4d}(r) = \begin{pmatrix} 1 & 0 & r & \frac{r}{\sqrt{2}} \\ 0 & 1 & 0 & \frac{r}{\sqrt{2}} \\ r & 0 & 1 & \frac{1}{\sqrt{2}} \\ \frac{r}{\sqrt{2}} & \frac{r}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1 \end{pmatrix}$$

Algorithm 1 (Original method for latent correlation computation) **Input**: $F(r) = F(r,\mathbf{\Delta})$ - bridge function based on the type of variables j, k

- Step 1. Calculate $\hat{\tau}_{jk}$ using (1).
- Step 2. For binary/truncated variable j, set $\hat{\mathbf{\Delta}}_j = \hat{\Delta}_j = \Phi^{-1}(\pi_{0j})$ with $\pi_{0j} = \sum_{i=1}^n I(x_{ij}=0)/n$. For ternary variable j, set $\hat{\mathbf{\Delta}}_j = (\hat{\Delta}_j^1, \hat{\Delta}_j^2)$ where $\hat{\Delta}_j^1 = \Phi^{-1}(\pi_{0j})$ and $\hat{\Delta}_j^2 = \Phi^{-1}(\pi_0+\pi_1)$ with $\pi_{0j} = \sum_{i=1}^n I(x_{ij}=0)/n$ and $\pi_{1j} = \sum_{i=1}^n I(x_{ij}=1)/n$.
- Compute $F^{-1}(\hat{\tau}_{jk})$ as $\hat{r}_{jk} = argmin\{F(r) - \hat{\tau}_{jk}\}^2$ solved via `optimize` function in R.

Multilinear Interpolation

Inversion via Multilinear Interpolation

The inverse bridge function is an analytic function of at most 5 parameters (see Theorem 1):

- Kendall's τ
- Proportion of zeros in the 1st variable
- (Possibly) proportion of zeros and ones in the 1st variable
- (Possibly) proportion of zeros in the 2nd variable
- (Possibly) proportion of zeros and ones in the 2nd variable

Definition 4 (Bilinear interpolation) Suppose we have 4 neighboring data points $f_{ij} = f(x_i, y_j)$ at (x_i, y_j) for $i, j \in \{0, 1\}$. For $\{(x, y)|x_0 \leq x \leq x_1, y_0 \leq y \leq y_1\}$, the bilinear interpolation at (x, y) is

$$\hat{f}(x, y) = (1-\alpha)(1-\beta)f_{00} + (1-\alpha)\beta f_{01} + \alpha(1-\beta)f_{10} + \alpha\beta f_{11}$$

where $\alpha = \frac{x-x_0}{x_1-x_0}$ and $\beta = \frac{y-y_0}{y_1-y_0}$.

Definition 5 (Trilinear interpolation) Suppose we have 8 neighoboring data points $f_{ijk} = f(x_i, y_j, z_k)$ at (x_i, y_j, z_k) for $i, j, k \in \{0, 1\}$. For $\{(x, y, z)|x_0 \leq x \leq x_1, y_0 \leq y \leq y_1, z_0 \leq z \leq z_1\}$, the trilinear interpolation at (x, y, z) is

$$\begin{aligned}\hat{f}(x, y, z) =& (1-\alpha)(1-\beta)(1-\gamma)f_{000} + (1-\alpha)(1-\beta)\gamma f_{001} + (1-\alpha)\beta(1-\gamma)f_{010} \\ &+ \alpha(1-\beta)(1-\gamma)f_{100} + (1-\alpha)\beta\gamma f_{011} + \alpha(1-\beta)\gamma f_{101} \\ &+ \alpha\beta(1-\gamma)f_{110} + \alpha\beta\gamma f_{111}\end{aligned}$$

where $\alpha = \frac{x-x_0}{x_1-x_0}$, $\beta = \frac{y-y_0}{y_1-y_0}$ and $\gamma = \frac{z-z_0}{z_1-z_0}$.

In short, d-dimensional multilinear interpolation uses a weighted average of 2^d neighbors to approximate the function values at the points within the d-dimensional cube of the neighbors (Yoon, Müller, and Gaynanova 2021). This can be done by R package `chebpol` (Gaure 2019).

Algorithm 2 (Multilinear interpolation for latent correlation computation) **Input**: Pre-computed values $F^{-1}(g)$ on a fixed grid $g \in \mathcal{G}$ based on the type of variables j and k. For binary/continuous case, $g = (\tau_{jk}, \Delta_j)$; for binary/binary case, $g = (\tau_{jk}, \Delta_j, \Delta_k)$; for truncated/continuous case, $g = (\tau_{jk}, \Delta_j)$; for truncated/truncated case, $g = (\tau_{jk}, \Delta_j, \Delta_k)$; for ternary/continuous case, $g = (\tau_{jk}, \Delta_j^1, \Delta_j^2)$; for ternary/binary case, $g = (\tau_{jk}, \Delta_j^1, \Delta_j^2, \Delta_k)$; for ternary/truncated case, $g = (\tau_{jk}, \Delta_j^1, \Delta_j^2, \Delta_k)$; for ternay/ternary case, $g = (\tau_{jk}, \Delta_j^1, \Delta_j^2, \Delta_k^1, \Delta_k^2)$.

- Step 1 and Step 2 same as Algorithm 1.
- Step 3. Set $\hat{r}_{jk} = \hat{F}^{-1}(\hat{g})$, where $\hat{F}^{-1}$ is the multilinear interpolation of $F^{-1}(\cdot)$ using $\mathcal{G}$.

Approximation via hybrid Scheme

To avoid interpolation in areas with high approximation errors close to the boundary, we use hybrid scheme (Yoon, Müller, and Gaynanova 2021). The derivation of approximate bound for BC, BB, TC, TB, TT cases see (Yoon, Müller, and Gaynanova 2021). The derivation of approximate bound for NC, NB, NN, NT case see Appendix.

$$\begin{aligned}\bar{\tau}_{BC}(\pi_{0j}) &= 2\pi_{0j}(1-\pi_{0j}) \\ \bar{\tau}_{BB}(\pi_{0j}, \pi_{0k}) &= 2\min(\pi_{0j}, \pi_{0k})\{1 - \max(\pi_{0j}, \pi_{0k})\} \\ \bar{\tau}_{TC}(\pi_{0j}) &= 1 - (\pi_{0j})^2\end{aligned}$$

$$\bar{\tau}_{TB}(\pi_{0j}, \pi_{0k}) = 2\max(\pi_{0k}, 1-\pi_{0k})\{1 - \max(\pi_{0k}, 1-\pi_{0k}, \pi_{0j})\}$$
$$\bar{\tau}_{TT}(\pi_{0j}, \pi_{0k}) = 1 - \{\max(\pi_{0j}, \pi_{0k})\}^2$$
$$\bar{\tau}_{NC}(\pi_{0j}, \pi_{1j}) = 2\{\pi_{0j}(1-\pi_{0j}) + \pi_{1j}(1-\pi_{0j}-\pi_{1j})\}$$
$$\bar{\tau}_{NB}(\pi_{0j}, \pi_{1j}, \pi_{0k}) = 2\min(\pi_{0j}(1-\pi_{0j}) + \pi_{1j}(1-\pi_{0j}-\pi_{1j}), \pi_{0k}(1-\pi_{0k}))$$
$$\bar{\tau}_{NT}(\pi_{0j}, \pi_{1j}, \pi_{0k}) = 1 - \{\max(\pi_{0j}, \pi_{1j}, 1-\pi_{0j}-\pi_{1j}, \pi_{0k})\}^2$$
$$\bar{\tau}_{NN}(\pi_{0j}, \pi_{1j}, \pi_{0k}, \pi_{1k}) = 2\min(\pi_{0j}(1-\pi_{0j}) + \pi_{1j}(1-\pi_{0j}-\pi_{1j}), \pi_{0k}(1-\pi_{0k}) + \pi_{1k}(1-\pi_{0k}-\pi_{1k}))$$

Rescale Grid for Interpolation Note that $|\hat{\tau}| \leq \bar{\tau}$, the grid does not need to cover the domain $\tau \in [-1, 1]$. Instead, we rescale them as following: $\tilde{\tau}_{jk} = \frac{\tau_{jk}}{\bar{\tau}_{jk}} \in [-1, 1]$, where $\bar{\tau}_{jk}$ applies the approximation bound function with respect to the data types corresponding to variable j and k. For ternary variable j, we know $\Delta_j^2 > \Delta_j^1$ always holds since $\Delta_j^1 = \Phi^{-1}(\pi_{0j})$ and $\Delta_j^2 = \Phi^{-1}(\pi_{0j} + \pi_{1j})$, the grid should not cover the domain for the areas of $\Delta_j^2 \geq \Delta_j^1$. So that we rescale them as following: $\tilde{\Delta}_j^1 = \frac{\Delta_j^1}{\Delta_j^2} \in [0, 1]$; $\tilde{\Delta}_j^2 = \Delta_j^2 \in [0, 1]$

Algorithm 3 (Multi-linear interpolation with rescaled grid) **Input**: Let $\tilde{g} = h(g)$, pre-computed values $F^{-1}(h^{-1}(\tilde{g}))$ on a fixed grid $\tilde{g} \in \tilde{\mathcal{G}}$ based on the type of variables j and k. For binary/continuous case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j)$; for binary/binary case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j, \tilde{\Delta}_k)$; for truncated/continuous case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j)$; for truncated/truncated case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j, \tilde{\Delta}_k)$; for ternary/continuous case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j^1, \tilde{\Delta}_j^2)$; for ternary/binary case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j^1, \tilde{\Delta}_j^2, \tilde{\Delta}_k)$; for ternary/truncated case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j^1, \tilde{\Delta}_j^2, \tilde{\Delta}_k)$; for ternay/ternary case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j^1, \tilde{\Delta}_j^2, \tilde{\Delta}_k^1, \tilde{\Delta}_k^2)$.

- Step 1 and Step 2 same as Algorithm 1.
- Step 3. Calculate $\tilde{\hat{g}} = h((\hat{g}))$.
- Step 4. Set $\hat{r}_{jk} = (\hat{F} \cdot \hat{h})^{-1}(\tilde{\hat{g}})$, where $(\hat{F} \cdot \hat{h})^{-1}(\cdot)$ is the multilinear interpolation of $F^{-1}(h^{-1}(\cdot))$ using $\tilde{\mathcal{G}}$.

Algorithm 4 (Multi-linear interpolation with rescaled grid and boundary method) **Input**: Let $\tilde{g} = h(g)$, pre-computed values $F^{-1}(h^{-1}(\tilde{g}))$ on a fixed grid $\tilde{g} \in \tilde{\mathcal{G}}$ based on the type of variables j and k. For binary/continuous case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j)$; for binary/binary case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j, \tilde{\Delta}_k)$; for truncated/continuous case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j)$; for truncated/truncated case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j, \tilde{\Delta}_k)$; for ternary/continuous case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j^1, \tilde{\Delta}_j^2)$; for ternary/binary case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j^1, \tilde{\Delta}_j^2, \tilde{\Delta}_k)$; for ternary/truncated case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j^1, \tilde{\Delta}_j^2, \tilde{\Delta}_k)$; for ternay/ternary case, $\tilde{g} = (\tilde{\tau}_{jk}, \tilde{\Delta}_j^1, \tilde{\Delta}_j^2, \tilde{\Delta}_k^1, \tilde{\Delta}_k^2)$.

- Step 1 and Step 2 same as Algorithm 1.
- Step 3. If $|\hat{\tau}_{jk}| \leq 0.9 \times ABD$, apply Algorithm 3; Otherwise apply Algorithm 1.

Appendix

Derivation for bridge function for ternary/truncated case

Without loss of generality, let $j = 1$ and $k = 2$. By the definition of Kendall's τ,

$$\tau_{12} = E(\hat{\tau}_{12}) = E[\frac{2}{n(n-1)} \sum_{1 \leq i \leq i' \leq n} sign\{(X_{i1} - X_{i'1})(X_{i2} - X_{i'2})\}] \tag{3}$$

Since X_1 is ternary,

$$\begin{aligned}
&sign(X_1 - X_1') \\
=&[I(U_1 > C_{11}, U_1' \le C_{11}) + I(U_1 > C_{12}, U_1' \le C_{12}) - I(U_1 > C_{12}, U_1' \le C_{11})] \\
&- [I(U_1 \le C_{11}, U_1' > C_{11}) + I(U_1 \le C_{12}, U_1' > C_{12}) - I(U_1 \le C_{11}, U_1' > C_{12})] \\
=&[I(U_1 > C_{11}) - I(U_1 > C_{11}, U_1' > C_{11}) + I(U_1 > C_{12}) - I(U_1 > C_{12}, U_1' > C_{12}) \\
&- I(U_1 > C_{12}) + I(U_1 > C_{12}, U_1' > C_{11})] \\
&- [I(U_1' > C_{11}) - I(U_1 > C_{11}, U_1' > C_{11}) + I(U_1' > C_{12}) - I(U_1 > C_{12}, U_1' > C_{12}) \\
&- I(U_1' > C_{12}) + I(U_1 > C_{11}, U_1' > C_{12})] \\
=&I(U_1 > C_{11}) + I(U_1 > C_{12}, U_1' > C_{11}) - I(U_1' > C_{11}) - I(U_1 > C_{11}, U_1' > C_{12}) \\
=&I(U_1 > C_{11}, U_1' \le C_{12}) - I(U_1' > C_{11}, U_1 \le C_{12}) \qquad (4)
\end{aligned}$$

Since X_2 is truncated, $C_1 > 0$ and

$$\begin{aligned}
sign(X_2 - X_2') =& - I(X_2 = 0, X_2' > 0) + I(X_2 > 0, X_2' = 0) \\
&+ I(X_2 > 0, X_2' > 0) sign(X_2 - X_2') \\
=& - I(X_2 = 0) + I(X_2' = 0) + I(X_2 > 0, X_2' > 0) sign(X_2 - X_2') \qquad (5)
\end{aligned}$$

Since f is monotonically increasing, $sign(X_2 - X_2') = sign(Z_2 - Z_2')$,

$$\begin{aligned}
\tau_{12} =& E[I(U_1 > C_{11}, U_1' \le C_{12}) sign(X_2 - X_2')] \\
&- E[I(U_1' > C_{11}, U_1 \le C_{12}) sign(X_2 - X_2')] \\
=& - E[I(U_1 > C_{11}, U_1' \le C_{12}) I(X_2 = 0)] \\
&+ E[I(U_1 > C_{11}, U_1' \le C_{12}) I(X_2' = 0)] \\
&+ E[I(U_1 > C_{11}, U_1' \le C_{12}) I(X_2 > 0, X_2' > 0) sign(Z_2 - Z_2')] \\
&+ E[I(U_1' > C_{11}, U_1 \le C_{12}) I(X_2 = 0)] \\
&- E[I(U_1' > C_{11}, U_1 \le C_{12}) I(X_2' = 0)] \\
&- E[I(U_1' > C_{11}, U_1 \le C_{12}) I(X_2 > 0, X_2' > 0) sign(Z_2 - Z_2')] \\
=& - 2E[I(U_1 > C_{11}, U_1' \le C_{12}) I(X_2 = 0)] \\
&+ 2E[I(U_1 > C_{11}, U_1' \le C_{12}) I(X_2' = 0)] \\
&+ E[I(U_1 > C_{11}, U_1' \le C_{12}) I(X_2 > 0, X_2' > 0) sign(Z_2 - Z_2')] \\
&- E[I(U_1' > C_{11}, U_1 \le C_{12}) I(X_2 > 0, X_2' > 0) sign(Z_2 - Z_2')] \qquad (6)
\end{aligned}$$

From the definition of U, let $Z_j = f_j(U_j)$ and $\Delta_j = f_j(C_j)$ for $j = 1, 2$. Using $sign(x) = 2I(x > 0) - 1$, we obtain

$$\begin{aligned}
\tau_{12} =& - 2E[I(Z_1 > \Delta_{11}, Z_1' \le \Delta_{12}, Z_2 \le \Delta_2)] + 2E[I(Z_1 > \Delta_{11}, Z_1' \le \Delta_{12}, Z_2' \le \Delta_2)] \\
&+ 2E[I(Z_1 > \Delta_{11}, Z_1' \le \Delta_{12}) I(Z_2 > \Delta_2, Z_2' > \Delta_2, Z_2 - Z_2' > 0)] \\
&- 2E[I(Z_1' > \Delta_{11}, Z_1 \le \Delta_{12}) I(Z_2 > \Delta_2, Z_2' > \Delta_2, Z_2 - Z_2' > 0)] \\
=& - 2E[I(Z_1 > \Delta_{11}, Z_1' \le \Delta_{12}, Z_2 \le \Delta_2)] + 2E[I(Z_1 > \Delta_{11}, Z_1' \le \Delta_{12}, Z_2' \le \Delta_2)] \\
&+ 2E[I(Z_1 > \Delta_{11}, Z_1' \le \Delta_{12}, Z_2' > \Delta_2, Z_2 > Z_2')] \\
&- 2E[I(Z_1' > \Delta_{11}, Z_1 \le \Delta_{12}, Z_2' > \Delta_2, Z_2 > Z_2')] \qquad (7)
\end{aligned}$$

Since $\{\frac{Z_2'-Z_2}{\sqrt{2}}, -Z1\}$, $\{\frac{Z_2'-Z_2}{\sqrt{2}}, Z1'\}$ and $\{\frac{Z_2'-Z_2}{\sqrt{2}}, -Z2'\}$ are standard bivariate normally distributed variables with correlation $-\frac{1}{\sqrt{2}}$, $r/\sqrt{2}$ and $-\frac{r}{\sqrt{2}}$, respectively, by the definition of $\Phi_3(\cdot,\cdot,\cdot;\cdot)$ and $\Phi_4(\cdot,\cdot,\cdot,\cdot;\cdot)$ we have

$$\begin{aligned}
F_{NT}(r;\Delta_j^1,\Delta_j^2,\Delta_k) =& -2\Phi_3\left\{-\Delta_j^1,\Delta_j^2,\Delta_k;\begin{pmatrix}1&0&-r\\0&1&0\\-r&0&1\end{pmatrix}\right\}\\
&+2\Phi_3\left\{-\Delta_j^1,\Delta_j^2,\Delta_k;\begin{pmatrix}1&0&0\\0&1&r\\0&r&1\end{pmatrix}\right\}\\
&+2\Phi_4\left\{-\Delta_j^1,\Delta_j^2,-\Delta_k,0;\begin{pmatrix}1&0&0&\frac{r}{\sqrt{2}}\\0&1&-r&\frac{r}{\sqrt{2}}\\0&-r&1&-\frac{1}{\sqrt{2}}\\\frac{r}{\sqrt{2}}&\frac{r}{\sqrt{2}}&-\frac{1}{\sqrt{2}}&1\end{pmatrix}\right\}\\
&-2\Phi_4\left\{-\Delta_j^1,\Delta_j^2,-\Delta_k,0;\begin{pmatrix}1&0&r&-\frac{r}{\sqrt{2}}\\0&1&0&-\frac{r}{\sqrt{2}}\\r&0&1&-\frac{1}{\sqrt{2}}\\-\frac{r}{\sqrt{2}}&-\frac{r}{\sqrt{2}}&-\frac{1}{\sqrt{2}}&1\end{pmatrix}\right\}
\end{aligned} \tag{8}$$

Using the facts that

$$\begin{aligned}
&\Phi_4\left\{-\Delta_j^1,\Delta_j^2,-\Delta_k,0;\begin{pmatrix}1&0&r&-\frac{r}{\sqrt{2}}\\0&1&0&-\frac{r}{\sqrt{2}}\\r&0&1&-\frac{1}{\sqrt{2}}\\-\frac{r}{\sqrt{2}}&-\frac{r}{\sqrt{2}}&-\frac{1}{\sqrt{2}}&1\end{pmatrix}\right\}\\
&+\Phi_4\left\{-\Delta_j^1,\Delta_j^2,-\Delta_k,0;\begin{pmatrix}1&0&r&\frac{r}{\sqrt{2}}\\0&1&0&\frac{r}{\sqrt{2}}\\r&0&1&\frac{1}{\sqrt{2}}\\\frac{r}{\sqrt{2}}&\frac{r}{\sqrt{2}}&\frac{1}{\sqrt{2}}&1\end{pmatrix}\right\}\\
=&\Phi_3\left\{-\Delta_j^1,\Delta_j^2,-\Delta_k;\begin{pmatrix}1&0&0\\0&1&r\\0&r&1\end{pmatrix}\right\}
\end{aligned} \tag{9}$$

and

$$\begin{aligned}
&\Phi_3\left\{-\Delta_j^1,\Delta_j^2,-\Delta_k;\begin{pmatrix}1&0&0\\0&1&r\\0&r&1\end{pmatrix}\right\}+\Phi_3\left\{-\Delta_j^1,\Delta_j^2,\Delta_k;\begin{pmatrix}1&0&-r\\0&1&0\\-r&0&1\end{pmatrix}\right\}\\
=&\Phi_2(-\Delta_j^1,\Delta_j^2;0)=\Phi(-\Delta_j^1)\Phi(\Delta_j^2)
\end{aligned} \tag{10}$$

So that,

$$
\begin{aligned}
F_{NT}(r;\Delta_j^1,\Delta_j^2,\Delta_k) = &-2\Phi(-\Delta_j^1)\Phi(\Delta_j^2) \\
&+2\Phi_3\left\{-\Delta_j^1,\Delta_j^2,\Delta_k;\begin{pmatrix}1&0&0\\0&1&r\\0&r&1\end{pmatrix}\right\} \\
&+2\Phi_4\left\{-\Delta_j^1,\Delta_j^2,-\Delta_k,0;\begin{pmatrix}1&0&0&\frac{r}{\sqrt{2}}\\0&1&-r&\frac{r}{\sqrt{2}}\\0&-r&1&-\frac{1}{\sqrt{2}}\\\frac{r}{\sqrt{2}}&\frac{r}{\sqrt{2}}&-\frac{1}{\sqrt{2}}&1\end{pmatrix}\right\} \\
&+2\Phi_4\left\{-\Delta_j^1,\Delta_j^2,-\Delta_k,0;\begin{pmatrix}1&0&r&\frac{r}{\sqrt{2}}\\0&1&0&\frac{r}{\sqrt{2}}\\r&0&1&\frac{1}{\sqrt{2}}\\\frac{r}{\sqrt{2}}&\frac{r}{\sqrt{2}}&\frac{1}{\sqrt{2}}&1\end{pmatrix}\right\}
\end{aligned} \tag{11}
$$

Derivation for approximate bound for ternary/continuous case

Let $n_{0x}=\sum_{i=1}^{n_x} I(x_i=0)$, $n_{2x}=\sum_{i=1}^{n_x} I(x_i=2)$, $\pi_{0x}=\frac{n_{0x}}{n_x}$ and $\pi_{2x}=\frac{n_{2x}}{n_x}$, then

$$
\begin{aligned}
|\tau(\mathbf{x})| \leq & \frac{n_{0x}(n-n_{0x})+n_{2x}(n-n_{0x}-n_{2x})}{\binom{n}{2}} \\
=&2\{\frac{n_{0x}}{n-1}-(\frac{n_{0x}}{n})(\frac{n_{0x}}{n-1})+\frac{n_{2x}}{n-1}-(\frac{n_{2x}}{n})(\frac{n_{0x}}{n-1})-(\frac{n_{2x}}{n})(\frac{n_{2x}}{n-1})\} \\
\approx&2\{\frac{n_{0x}}{n}-(\frac{n_{0x}}{n})^2+\frac{n_{2x}}{n}-(\frac{n_{2x}}{n})(\frac{n_{0x}}{n})-(\frac{n_{2x}}{n})^2\} \\
=&2\{\pi_{0x}(1-\pi_{0x})+\pi_{2x}(1-\pi_{0x}-\pi_{2x})\}
\end{aligned} \tag{12}
$$

Approximate bound for ternary/binary case and ternary/ternary case

Combine NC and BC case, we get NB case. So does NN case.

Derivation for approximate bound for ternary/truncated case

Derivation for approximate bound for ternary truncated case: Let $\mathbf{x}\in\mathcal{R}^n$ and $\mathbf{y}\in\mathcal{R}^n$ be the observed n realizations of ternary and truncated variables, respectively. Let $n_{0x}=\sum_{i=0}^{n} I(x_i=0)$, $\pi_{0x}=\frac{n_{0x}}{n}$, $n_{1x}=\sum_{i=0}^{n} I(x_i=1)$, $\pi_{1x}=\frac{n_{1x}}{n}$, $n_{2x}=\sum_{i=0}^{n} I(x_i=2)$, $\pi_{2x}=\frac{n_{2x}}{n}$, $n_{0y}=\sum_{i=0}^{n} I(y_i=0)$, $\pi_{0y}=\frac{n_{0y}}{n}$, $n_{0x0y}=\sum_{i=0}^{n} I(x_i=0\ \&\ y_i=0)$, $n_{1x0y}=\sum_{i=0}^{n} I(x_i=1\ \&\ y_i=0)$ and $n_{2x0y}=\sum_{i=0}^{n} I(x_i=2\ \&\ y_i=0)$ then

$$
|\tau(\mathbf{x},\mathbf{y})| \leq \frac{\binom{n}{2}-\binom{n_{0x}}{2}-\binom{n_{1x}}{2}-\binom{n_{2x}}{2}-\binom{n_{0y}}{2}+\binom{n_{0x0y}}{2}+\binom{n_{1x0y}}{2}+\binom{n_{2x0y}}{2}}{\binom{n}{2}}
$$

Since $n_{0x0y} \leq \min(n_{0x}, n_{0y})$, $n_{1x0y} \leq \min(n_{1x}, n_{0y})$ and $n_{2x0y} \leq \min(n_{2x}, n_{0y})$ we obtain

$$
\begin{aligned}
|\tau(\mathbf{x}, \mathbf{y})| \leq & \frac{\binom{n}{2} - \binom{n_{0x}}{2} - \binom{n_{1x}}{2} - \binom{n_{2x}}{2} - \binom{n_{0y}}{2}}{\binom{n}{2}} \\
& + \frac{\binom{\min(n_{0x}, n_{0y})}{2} + \binom{\min(n_{1x}, n_{0y})}{2} + \binom{\min(n_{2x}, n_{0y})}{2}}{\binom{n}{2}} \\
\leq & \frac{\binom{n}{2} - \binom{\max(n_{0x}, n_{1x}, n_{2x}, n_{0y})}{2}}{\binom{n}{2}} \\
\leq & 1 - \frac{\max(n_{0x}, n_{1x}, n_{2x}, n_{0y})(\max(n_{0x}, n_{1x}, n_{2x}, n_{0y}) - 1)}{n(n-1)} \\
\approx & 1 - (\frac{\max(n_{0x}, n_{1x}, n_{2x}, n_{0y})}{n})^2 \\
= & 1 - \{\max(\pi_{0x}, \pi_{1x}, \pi_{2x}, \pi_{0y})\}^2 \\
= & 1 - \{\max(\pi_{0x}, (1 - \pi_{0x} - \pi_{2x}), \pi_{2x}, \pi_{0y})\}^2
\end{aligned} \tag{13}
$$

References

Croux, Christophe, Peter Filzmoser, and Heinrich Fritz. 2013. "Robust Sparse Principal Component Analysis." *Technometrics* 55 (2): 202–14.

Fan, Jianqing, Han Liu, Yang Ning, and Hui Zou. 2017. "High Dimensional Semiparametric Latent Graphical Model for Mixed Data." *Journal of the Royal Statistical Society. Series B: Statistical Methodology* 79 (2): 405–21.

Filzmoser, Peter, Heinrich Fritz, and Klaudius Kalcher. 2021. *pcaPP: Robust PCA by Projection Pursuit.* https://CRAN.R-project.org/package=pcaPP.

Fox, John. 2019. *Polycor: Polychoric and Polyserial Correlations.* https://CRAN.R-project.org/package=polycor.

Gaure, Simen. 2019. *Chebpol: Multivariate Interpolation.* https://github.com/sgaure/chebpol.

Liu, Han, John Lafferty, and Larry Wasserman. 2009. "The Nonparanormal: Semiparametric Estimation of High Dimensional Undirected Graphs." *Journal of Machine Learning Research* 10 (10).

Quan, Xiaoyun, James G Booth, and Martin T Wells. 2018. "Rank-Based Approach for Estimating Correlations in Mixed Ordinal Data." *arXiv Preprint arXiv:1809.06255.*

Yoon, Grace, Christian L Müller, and Irina Gaynanova. 2021. "Fast Computation of Latent Correlations." *Journal of Computational and Graphical Statistics*, 1–8.