Imputing Missing Data with the Gaussian Copula

Madeleine Udell

Operations Research and Information Engineering
Cornell University

Based on joint work with Yuxuan Zhao, Eric Landgrebe, and Eliot Shekhtman

Fields Institute mini-symposium on low-rank models
June 2021

Data table

age	gender	state	income	education	$\cdots$
29	F	CT	\$53,000	college	$\cdots$
57	?	NY	\$19,000	high school	$\cdots$
?	M	CA	\$102,000	masters	$\cdots$
41	F	NV	\$23,000	?	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$		

Data table

age	gender	state	income	education	$\cdots$
29	F	CT	\$53,000	college	$\cdots$
57	?	NY	\$19,000	high school	$\cdots$
?	M	CA	\$102,000	masters	$\cdots$
41	F	NV	\$23,000	?	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$		

goals:

- impute missing entries?
- estimate confidence for imputations?
- quantify impact of imputation on downstream analyses?
- identify related features?

Data table

n examples (patients, respondents, assets)
p features (tests, questions, performance indicators)

$$\begin{bmatrix} & X & \end{bmatrix} = \begin{bmatrix} X_{11} & \cdots & X_{1p} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{np} \end{bmatrix}$$

- x^i, ith row of X, is feature vector for ith example
- x_j, jth column of X, gives values for jth feature across all examples

Classic approach: low rank matrix completion

- (+) efficient algorithms
- (+) theoretical guarantees
- (+) works well when
  - true rank r is small compared to min(n,p)
  - or when true matrix is approximately low rank
- (-) works poorly for ordinal or mixed data
  - requires additional parameters (e.g., a loss function) to map from low rank parameter to ordinal observations
  - (-) fit with cross-validation $\implies$ slow!

Why low rank?

squareish data matrices $n \sim p$ for large n, p

- $(+)$ often approximately low rank
- $(+)$ Udell and Townsend 2019: approximate rank grows as $\log(n + p)$ under rather general assumptions

Why low rank?

squareish data matrices $n \sim p$ for large n, p

- $(+)$ often approximately low rank
- $(+)$ Udell and Townsend 2019: approximate rank grows as $\log(n + p)$ under rather general assumptions

skinny data matrices

- (-) are unlikely to be (very) low rank $\ll \min(n, p)$
- $(+)$ have lots of observations per column, so can estimate more complex model per column

Imputation challenge depends on the data

- size: skinny (small p) versus wide (large p)
- type: continuous, ordinal, binary, categorical, or mixed
- access: offline data vs streaming data

Imputation challenge depends on the data

- size: skinny (small p) versus wide (large p)
- type: continuous, ordinal, binary, categorical, or mixed
- access: offline data vs streaming data

Goals for imputation:

- high accuracy
- quantified uncertainty (e.g., confidence intervals)
- estimate impact on downstream analyses
- understand correlations between features

Outline

Gaussian copula model

Imputation

Approximate EM

Performance

Low-rank Gaussian copula

Gaussian copula model

We say $x \sim \mathrm{GC}(\Sigma, f) \in \mathbf{R}^p$ follows the *Gaussian copula model* with parameters Σ and $f : \mathbf{R}^p \to \mathbf{R}^p$ if

- *copula:* $z \sim \mathcal{N}(0, \Sigma)$
- *marginals:* $x = f(z)$ for $f = (f_1, \ldots, f_n)$ entrywise monotonic, i.e.,

$$x_j = f_j(z_j), \qquad j = 1, \ldots, p$$

Gaussian copula model

We say $x \sim \mathrm{GC}(\Sigma, f) \in \mathbf{R}^p$ follows the *Gaussian copula model* with parameters Σ and $f : \mathbf{R}^p \to \mathbf{R}^p$ if

- *copula:* $z \sim \mathcal{N}(0, \Sigma)$
- *marginals:* $x = f(z)$ for $f = (f_1, \ldots, f_n)$ entrywise monotonic, i.e.,

$$x_j = f_j(z_j), \qquad j = 1, \ldots, p$$

why use a copula?

- models *nonlinear rulers*
- separates interations from marginals

Gaussian copula model

We say $x \sim \mathrm{GC}(\Sigma, f) \in \mathbf{R}^p$ follows the *Gaussian copula model* with parameters Σ and $f : \mathbf{R}^p \to \mathbf{R}^p$ if

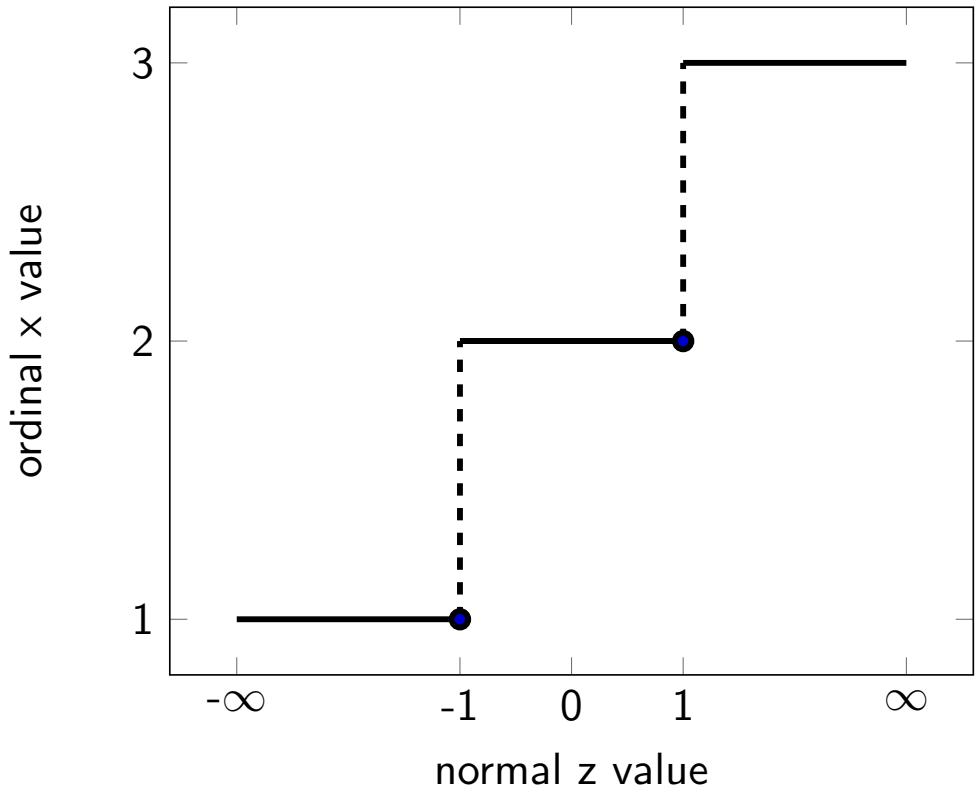
- *copula:* $z \sim \mathcal{N}(0, \Sigma)$
- *marginals:* $x = f(z)$ for $f = (f_1, \ldots, f_n)$ entrywise monotonic, i.e.,

$$x_j = f_j(z_j), \qquad j = 1, \ldots, p$$

why use a copula?

- models *nonlinear rulers*
- separates interations from marginals
- Sklar's theorem: any multivariate joint distribution can be written in terms of univariate marginal distribution functions and a copula which describes the dependence structure

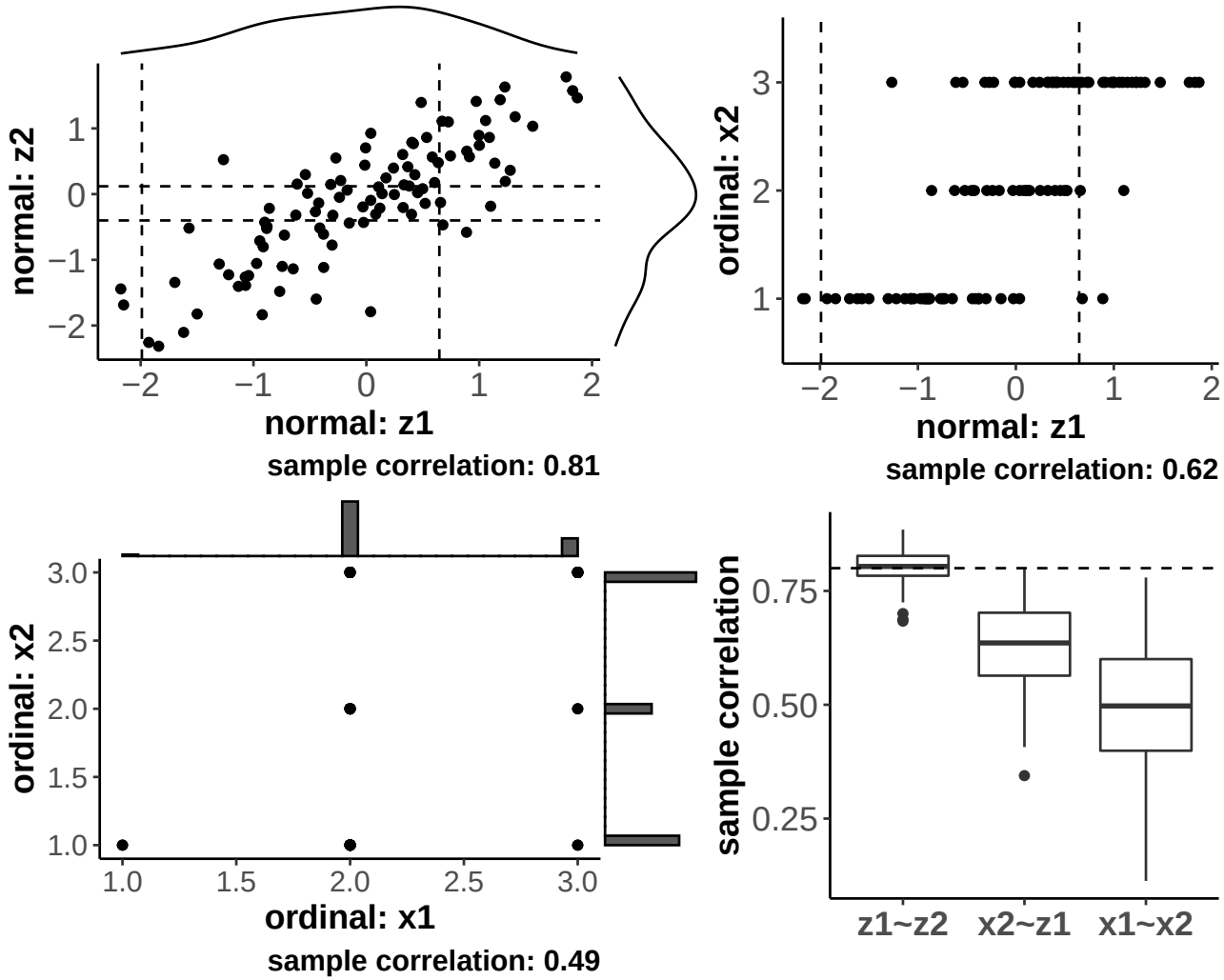
Example: cutoff function f generates ordinal data



- when x has a continuous distribution, f is invertible
- when x has a discrete distribution, f is not invertible
- we write f^{-1} for the set-valued mapping

$$z \in f^{-1}(x) \quad \Longleftrightarrow \quad x = f(z)$$

Marginals distort correlations



Related work

- Hoff et al. 2007: Gaussian copula for ordinal data, fit with MCMC
- Han and Liu 2014: Gaussian copula for real valued data
- Ganti et al. 2015: low rank model with nonlinear marginals, not probabilistic
- Anderson-Bergman et al. 2018: motivated by copula model; method fits low rank mean but assumes diagonal covariance

Related work

- Hoff et al. 2007: Gaussian copula for ordinal data, fit with MCMC
- Han and Liu 2014: Gaussian copula for real valued data
- Ganti et al. 2015: low rank model with nonlinear marginals, not probabilistic
- Anderson-Bergman et al. 2018: motivated by copula model; method fits low rank mean but assumes diagonal covariance

our method:

- handles real, boolean, ordinal, and missing data
- fully probabilistic ($\implies$ confidence intervals for estimates)
- no hyperparameters (rank, etc)
- fit with fast approximate EM

Outline

Given parameters, imputation is easy

- invertible marginals $f = (f_1, \ldots, f_p)$ (or estimates)
- copula Σ
- observed entries x_{obs} of new row $x \in \mathbf{R}^p$, obs $\subset \{1, \ldots, p\}$
- missing entries mis $= \{1, \ldots, p\} \setminus$ obs

Given parameters, imputation is easy

- invertible marginals $f = (f_1, \ldots, f_p)$ (or estimates)
- copula Σ
- observed entries x_{obs} of new row $x \in \mathbf{R}^p$, obs $\subset \{1, \ldots, p\}$
- missing entries mis $= \{1, \ldots, p\} \setminus$ obs

impute missing entries using normality of z_{mis}:

Given parameters, imputation is easy

- invertible marginals $f = (f_1, \ldots, f_p)$ (or estimates)
- copula Σ
- observed entries x_{obs} of new row $x \in \mathbf{R}^p$, obs $\subset \{1, \ldots, p\}$
- missing entries mis $= \{1, \ldots, p\} \setminus$ obs

impute missing entries using normality of z_{mis}:

- map observations to latent $z_{\text{obs}} = \{f_j^{-1}(x_j) : j \in \text{obs}\}$

Given parameters, imputation is easy

- invertible marginals $f = (f_1, \ldots, f_p)$ (or estimates)
- copula Σ
- observed entries x_{obs} of new row $x \in \mathbf{R}^p$, obs $\subset \{1, \ldots, p\}$
- missing entries mis $= \{1, \ldots, p\} \setminus$ obs

impute missing entries using normality of z_{mis}:

- map observations to latent $z_{\text{obs}} = \{f_j^{-1}(x_j) : j \in \text{obs}\}$
- latent missing z_{mis} are normal given z_{obs}:

$$z_{\text{mis}} | z_{\text{obs}} \sim \mathcal{N}(\Sigma_{\text{mis,obs}} \Sigma_{\text{obs,obs}}^{-1} z_{\text{obs}}, \Sigma_{\text{mis,mis}} - \Sigma_{\text{mis,obs}} \Sigma_{\text{obs,obs}}^{-1} \Sigma_{\text{obs,mis}})$$

Given parameters, imputation is easy

- invertible marginals $f = (f_1, \ldots, f_p)$ (or estimates)
- copula Σ
- observed entries x_{obs} of new row $x \in \mathbf{R}^p$, obs $\subset \{1, \ldots, p\}$
- missing entries mis $= \{1, \ldots, p\} \setminus$ obs

impute missing entries using normality of z_{mis}:

- map observations to latent $z_{\text{obs}} = \{f_j^{-1}(x_j) : j \in \text{obs}\}$
- latent missing z_{mis} are normal given z_{obs}:

$$z_{\text{mis}} | z_{\text{obs}} \sim \mathcal{N}(\Sigma_{\text{mis,obs}} \Sigma_{\text{obs,obs}}^{-1} z_{\text{obs}}, \Sigma_{\text{mis,mis}} - \Sigma_{\text{mis,obs}} \Sigma_{\text{obs,obs}}^{-1} \Sigma_{\text{obs,mis}})$$

- predict with mean $\hat{z}_{\text{mis}} = \mathbb{E}[z_{\text{mis}} | z_{\text{obs}}] = \Sigma_{\text{mis,obs}} \Sigma_{\text{obs,obs}}^{-1} z_{\text{obs}}$

Given parameters, imputation is easy

- invertible marginals $f = (f_1, \ldots, f_p)$ (or estimates)
- copula Σ
- observed entries x_{obs} of new row $x \in \mathbf{R}^p$, obs $\subset \{1, \ldots, p\}$
- missing entries mis $= \{1, \ldots, p\} \setminus$ obs

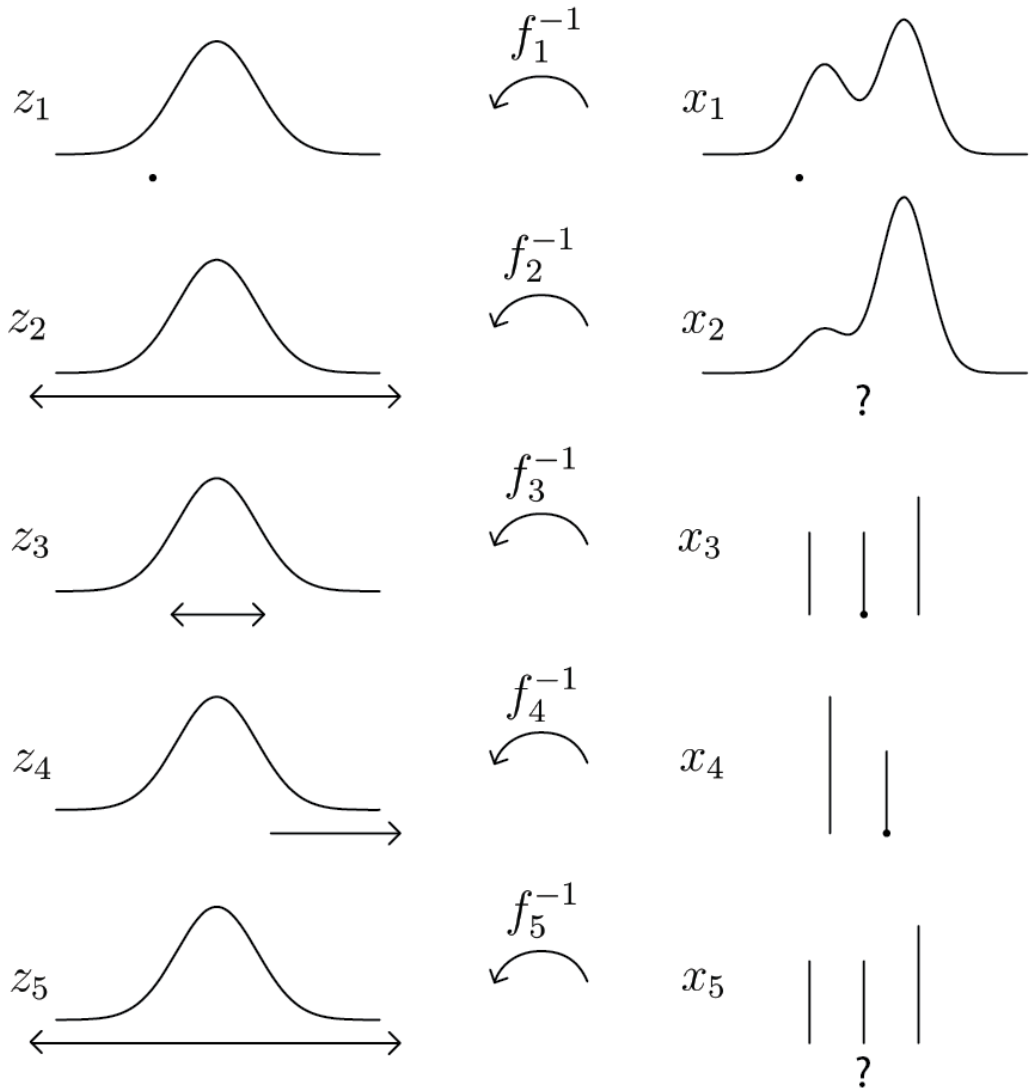
impute missing entries using normality of z_{mis}:

- map observations to latent $z_{\text{obs}} = \{f_j^{-1}(x_j) : j \in \text{obs}\}$
- latent missing z_{mis} are normal given z_{obs}:

$$z_{\text{mis}} | z_{\text{obs}} \sim \mathcal{N}(\Sigma_{\text{mis,obs}} \Sigma_{\text{obs,obs}}^{-1} z_{\text{obs}}, \Sigma_{\text{mis,mis}} - \Sigma_{\text{mis,obs}} \Sigma_{\text{obs,obs}}^{-1} \Sigma_{\text{obs,mis}})$$

- predict with mean $\hat{z}_{\text{mis}} = \mathbb{E}[z_{\text{mis}} | z_{\text{obs}}] = \Sigma_{\text{mis,obs}} \Sigma_{\text{obs,obs}}^{-1} z_{\text{obs}}$
- map back to observed space using marginals $\hat{x}_{\text{mis}} = f(\hat{z}_{\text{mis}})$

Missing or mixed data: latent vector z not known



latent z is constrained to product of intervals

Given parameters, imputation is easy

- marginals $f = (f_1, \ldots, f_p)$ (or estimates) (not invertible)
- copula Σ
- observed entries x_{obs} of new row $x \in \mathbf{R}^p$
- missing entries mis $= \{1, \ldots, p\} \setminus$ obs

Given parameters, imputation is easy

- marginals $f = (f_1, \ldots, f_p)$ (or estimates) (not invertible)
- copula Σ
- observed entries x_{obs} of new row $x \in \mathbf{R}^p$
- missing entries $\text{mis} = \{1, \ldots, p\} \setminus \text{obs}$

impute missing entries using normality of z_{mis}:

- latent missing z_{mis} are normal given z_{obs}:

$$z_{\text{mis}} | z_{\text{obs}} \sim \mathcal{N}(\Sigma_{\text{mis,obs}} \Sigma_{\text{obs,obs}}^{-1} z_{\text{obs}}, \Sigma_{\text{mis,mis}} - \Sigma_{\text{mis,obs}} \Sigma_{\text{obs,obs}}^{-1} \Sigma_{\text{obs,mis}})$$

- predict with mean:

$$\begin{aligned} \hat{z}_{\text{mis}} &= \mathbb{E}[z_{\text{mis}} \mid x_{\text{obs}}] = \mathbb{E}[\Sigma_{\text{mis,obs}} \Sigma_{\text{obs,obs}}^{-1} z_{\text{obs}} \mid x_{\text{obs}}] \\ &= \Sigma_{\text{mis,obs}} \Sigma_{\text{obs,obs}}^{-1} \mathbb{E}[z_{\text{obs}} \mid x_{\text{obs}}] \end{aligned}$$

- map back to observed space using marginals $\hat{x}_{\text{mis}} = f(\hat{z}_{\text{mis}})$

Outline

Estimating the model

Gaussian copula model

- $z \sim \mathcal{N}(0, \Sigma)$
- $x = f(z)$, $f = (f_1, \ldots, f_n)$ entrywise monotonic

two parameters to estimate

- *marginals:* monotone functions $f_1, \ldots, f_p$
- *copula:* correlation matrix $\Sigma \in \mathbf{R}^{p \times p}$

Estimating the marginals

- Gaussian matrix Z:
  - columns $\mathcal{N}(0, 1)$
- (partially) observed matrix X:
  - column j has observed empirical distribution with CDF F_j

Estimating the marginals

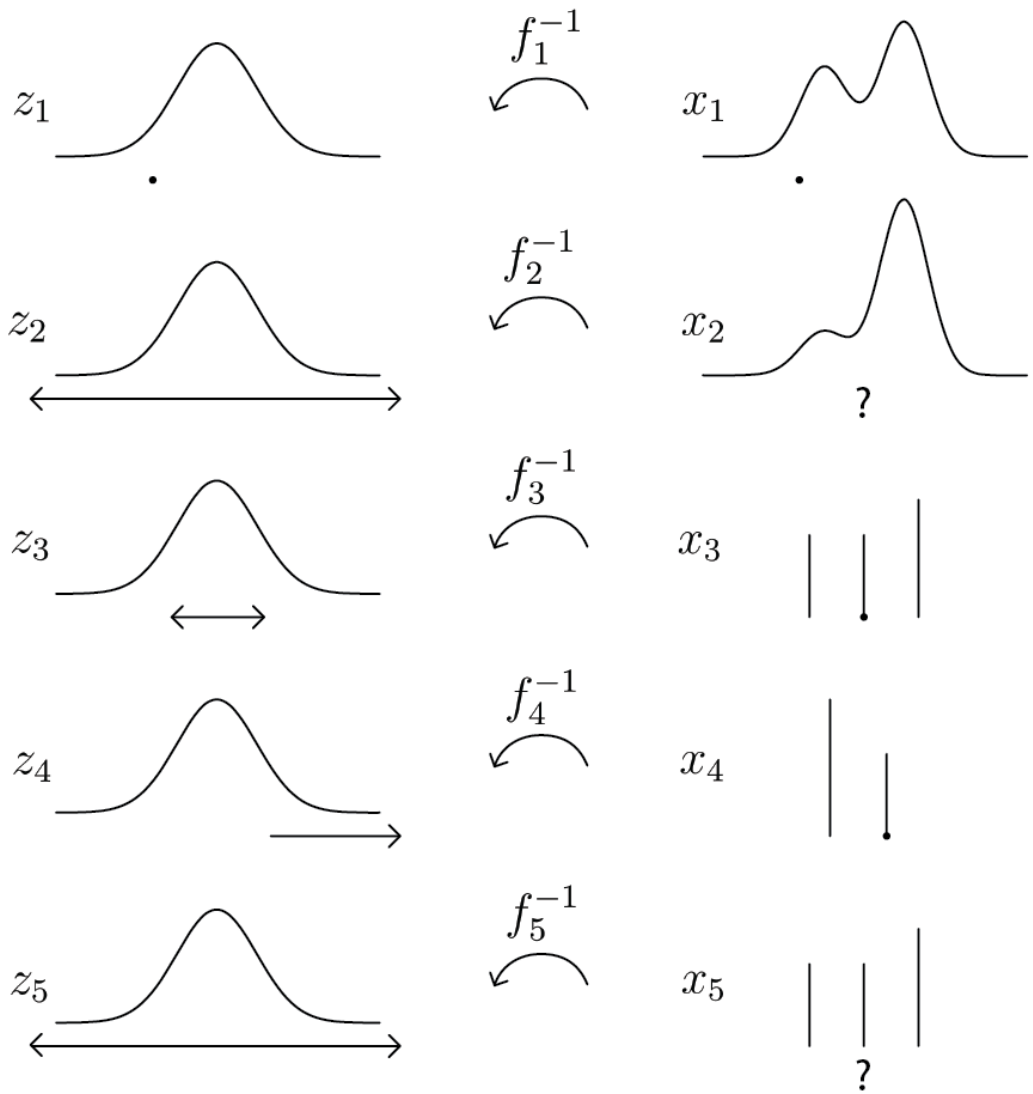
- Gaussian matrix Z:
  - columns $\mathcal{N}(0, 1)$
- (partially) observed matrix X:
  - column j has observed empirical distribution with CDF F_j

estimate f_j for each column $j = 1, \ldots, p$ by matching quantiles:

$$\hat{f}_j^{-1}(x_j^i) = \Phi^{-1}\left(\frac{n_j}{n_j + 1}\hat{F}_j(x_j^i)\right).$$

our convention: for discrete data, $\hat{F}_j$ is set-valued.

Matching quantiles



Warmup: estimate copula for complete cts data

given

- observed matrix $X \in \mathbf{R}^{n \times p}$
- invertible marginals $f = (f_1, \ldots, f_p)$ (or estimates)

estimate copula Σ using maximum likelihood estimation:

Warmup: estimate copula for complete cts data

given

- observed matrix $X \in \mathbf{R}^{n \times p}$
- invertible marginals $f = (f_1, \ldots, f_p)$ (or estimates)

estimate copula Σ using maximum likelihood estimation:

- compute latent $z^i = f^{-1}(x^i)$ for each row x^i

Warmup: estimate copula for complete cts data

given

- observed matrix $X \in \mathbf{R}^{n \times p}$
- invertible marginals $f = (f_1, \ldots, f_p)$ (or estimates)

estimate copula Σ using maximum likelihood estimation:

- compute latent $z^i = f^{-1}(x^i)$ for each row x^i
- log likelihood is

$$\ell(\Sigma; X) = c - \frac{\log(|\Sigma|)}{2} - \frac{1}{2}\operatorname{Tr}(\Sigma^{-1} \sum_{i=1}^{n} \frac{1}{n} z^i (z^i)^\top),$$

maximized by the empirical covariance $G = \frac{1}{n}\sum_{i=1}^{n} z^i (z^i)^\top$

Warmup: estimate copula for complete cts data

given

- observed matrix $X \in \mathbf{R}^{n \times p}$
- invertible marginals $f = (f_1, \ldots, f_p)$ (or estimates)

estimate copula Σ using maximum likelihood estimation:

- compute latent $z^i = f^{-1}(x^i)$ for each row x^i
- log likelihood is

$$\ell(\Sigma; X) = c - \frac{\log(|\Sigma|)}{2} - \frac{1}{2}\operatorname{Tr}(\Sigma^{-1} \sum_{i=1}^{n} \frac{1}{n} z^i (z^i)^\top),$$

  maximized by the empirical covariance $G = \frac{1}{n}\sum_{i=1}^{n} z^i (z^i)^\top$
- scale $\hat{\Sigma}$ to correlation matrix:
  set $D = \mathbf{diag}(G)$ and scale $\hat{\Sigma} \leftarrow D^{-1/2} G D^{-1/2}$

Estimating the copula: missing or discrete

given

- observed entries x^i_{obs} for each row $x^i \in \mathbf{R}^p$, $i = 1, \ldots, n$
- marginals $f = (f_1, \ldots, f_p)$ (or estimates)

idea: estimate copula Σ using maximum likelihood estimation?

Estimating the copula: missing or discrete

given

- observed entries x^i_{obs} for each row $x^i \in \mathbf{R}^p$, $i = 1, \ldots, n$
- marginals $f = (f_1, \ldots, f_p)$ (or estimates)

idea: estimate copula Σ using maximum likelihood estimation? log likelihood is

$$\ell(\Sigma; \{x_{\mathcal{O}_i^i}\}_{i=1}^n) = \frac{1}{n} \sum_{i=1}^n \log \left(\int_{z^i \in f^{-1}(x^i)} \phi(z^i; 0, \Sigma) dz^i \right),$$

where $\phi(z; 0, \Sigma)$ is the pdf of a Gaussian with mean 0 and correlation Σ.

Estimating the copula: missing or discrete

given

- observed entries x^i_{obs} for each row $x^i \in \mathbf{R}^p$, $i = 1, \ldots, n$
- marginals $f = (f_1, \ldots, f_p)$ (or estimates)

idea: estimate copula Σ using maximum likelihood estimation? log likelihood is

$$\ell(\Sigma; \{x_{\mathcal{O}_i^i}\}_{i=1}^n) = \frac{1}{n} \sum_{i=1}^n \log \left(\int_{z^i \in f^{-1}(x^i)} \phi(z^i; 0, \Sigma) dz^i \right),$$

where $\phi(z; 0, \Sigma)$ is the pdf of a Gaussian with mean 0 and correlation Σ.

- $z^i \in f^{-1}(x^i)$ is product of intervals
- truncated Gaussian integral is hard to compute or optimize

Estimating the copula: missing or discrete

given

- observed entries x^i_{obs} for each row $x^i \in \mathbf{R}^p$, $i = 1, \ldots, n$
- marginals $f = (f_1, \ldots, f_p)$ (or estimates)

idea: estimate copula Σ using maximum likelihood estimation?
log likelihood is

$$\ell(\Sigma; \{x_{\mathcal{O}_i^i}\}_{i=1}^n) = \frac{1}{n} \sum_{i=1}^n \log \left(\int_{z^i \in f^{-1}(x^i)} \phi(z^i; 0, \Sigma) dz^i \right),$$

where $\phi(z; 0, \Sigma)$ is the pdf of a Gaussian with mean 0 and correlation Σ.

- $z^i \in f^{-1}(x^i)$ is product of intervals
- truncated Gaussian integral is hard to compute or optimize
- instead: use EM

EM algorithm for Gaussian Copula

EM algorithm for Gaussian Copula:

- **Input:** observed entries X_{obs}, marginals f
- **Initialize:** $t = 0$, $\hat{\Sigma} = I$.
- For $t = 0, 1, 2, \ldots$
  1. E-step: $G = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}\left[zz^T | x_{\text{obs}}^i, \hat{\Sigma}, f \right]$.
  2. M-step: $D = \textbf{diag}(G)$; scale $\hat{\Sigma} \leftarrow D^{-1/2} G D^{-1/2}$.

  until convergence.
- **Output:** $\hat{\Sigma}$.

EM algorithm for Gaussian Copula

EM algorithm for Gaussian Copula:

- **Input:** observed entries X_{obs}, marginals f
- **Initialize:** $t = 0$, $\hat{\Sigma} = I$.
- For $t = 0, 1, 2, \ldots$
  1. E-step: $G = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}\left[zz^T | x_{\text{obs}}^i, \hat{\Sigma}, f \right]$.
  2. M-step: $D = \mathbf{diag}(G)$; scale $\hat{\Sigma} \leftarrow D^{-1/2} G D^{-1/2}$.

  until convergence.
- **Output:** $\hat{\Sigma}$.

properties:

- every step increases likelihood $\mathbb{P}(X_{\text{obs}}; \hat{\Sigma})$
- consistent estimate for Σ if data MAR
- E step is challenging; our approximate EM method uses a fast approximation based on 1D integrals.

Outline

Copula-EM imputes well and quickly

Synthetic data experiment:

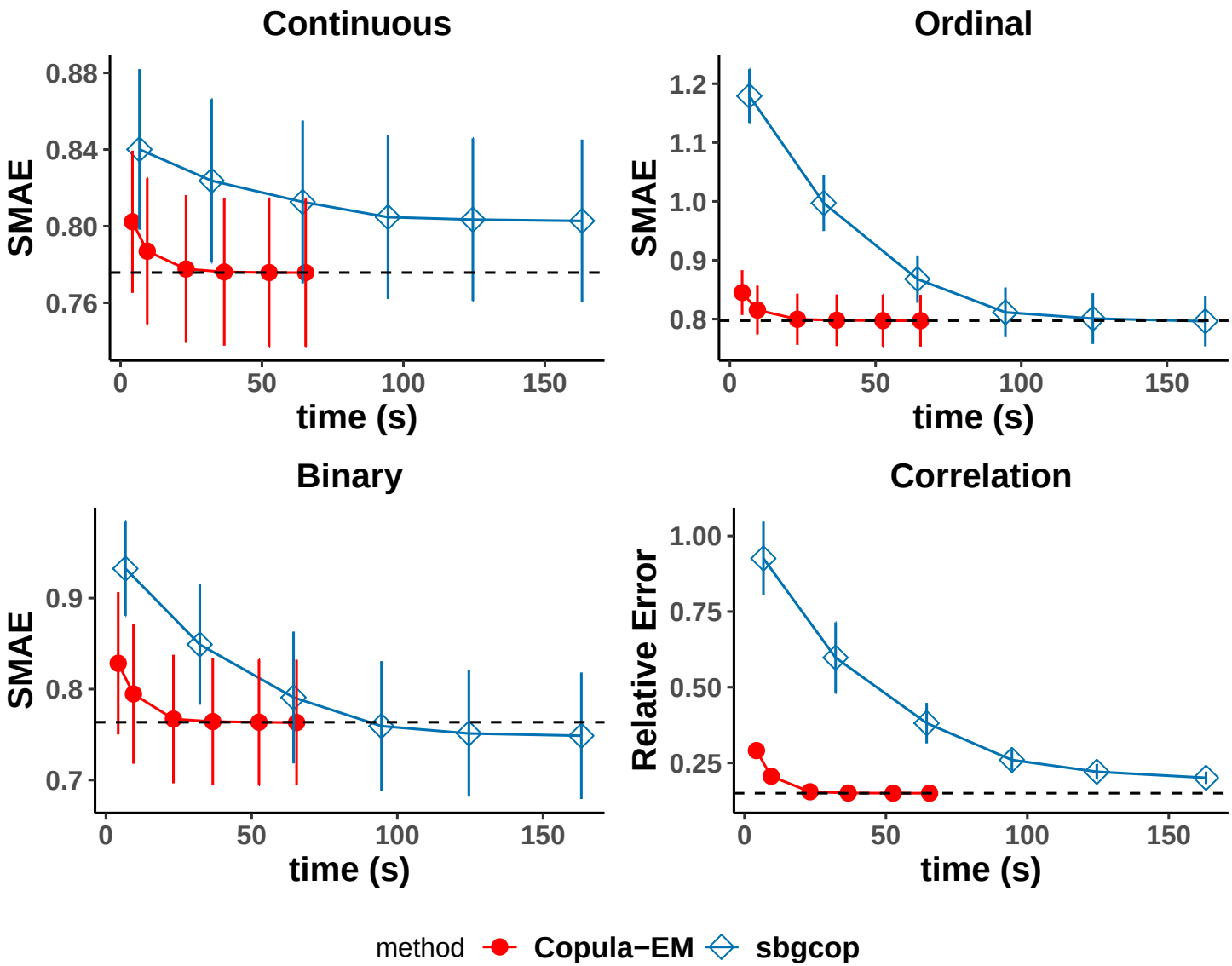
- $p = 15$, $n = 2000$. Correlation Σ is random. First five columns have exponential distributions, next five are binary, and last five are ordinal with five levels.

Define scaled mean absolute error

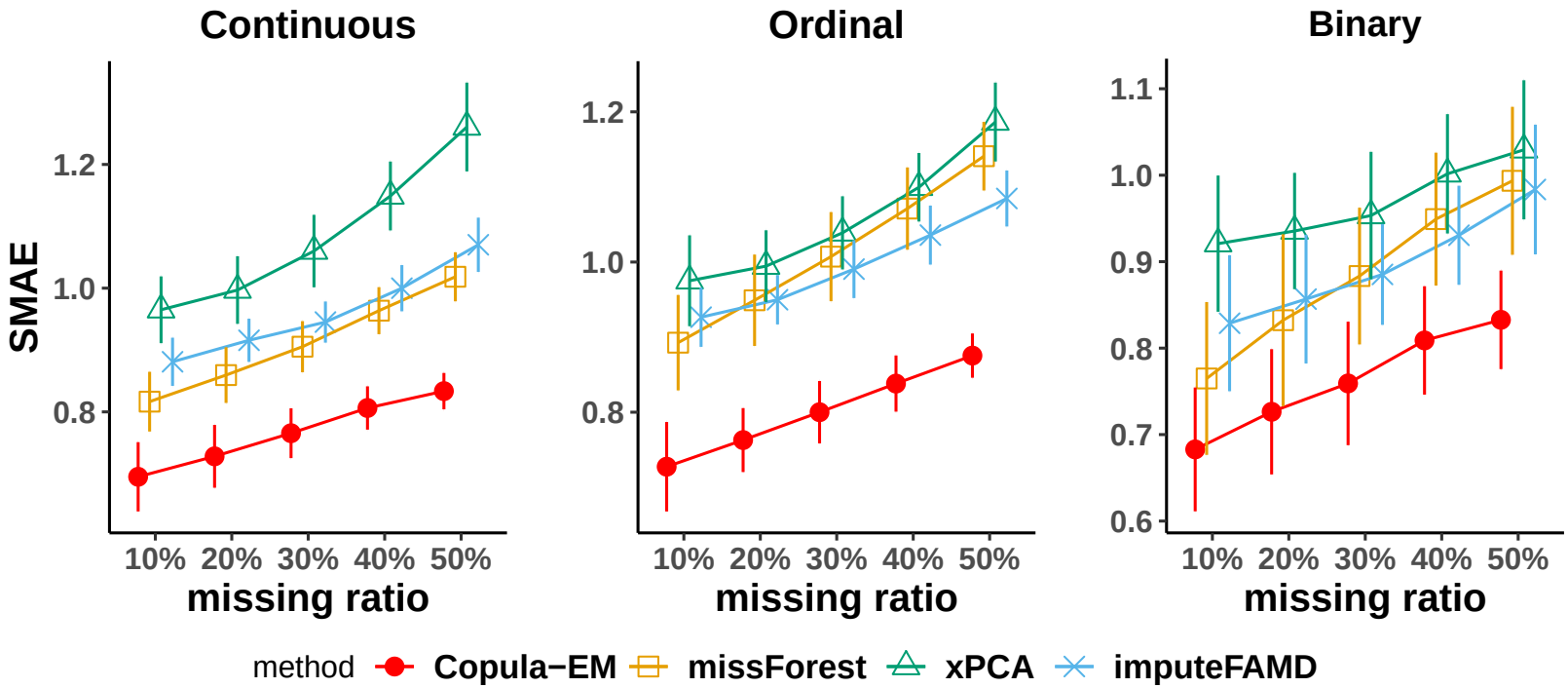
$$\text{SMAE}(\hat{x}, x, \text{obs}) := \frac{||\hat{x} - x||_1}{||\text{median}(x) - x||_1}$$

(Median imputation has SMAE $= 1$.)

Copula-EM imputes faster than MCMC



Gaussian copula has best accuracy



Gaussian copula for survey data

- General Social Survey data
- 1 continuous, 17 ordinal variables with 2–48 levels
- 25% of data missing
- mask additional 10% of data

Table: Imputation Error (SMAE) on Five GSS Variables

Variable	Copula-EM	sbgcop	missForest	xPCA	imputeFAMD
CLASS	**0.736(0.11)**	1.098(0.15)	0.782(0.09)	0.795(0.08)	0.797(0.10)
LIFE	**0.758(0.13)**	0.994(0.19)	0.828(0.17)	0.783(0.11)	0.821(0.11)
HEALTH	**0.886(0.08)**	1.257(0.17)	1.143(0.18)	0.908(0.10)	0.947(0.04)
HAPPY	**0.894(0.08)**	1.332(0.13)	1.079(0.15)	1.003(0.15)	1.001(0.10)
INCOME	**0.897(0.05)**	0.952(0.03)	0.944(0.18)	1.090(0.15)	0.996(0.01)

Gaussian copula yields insights

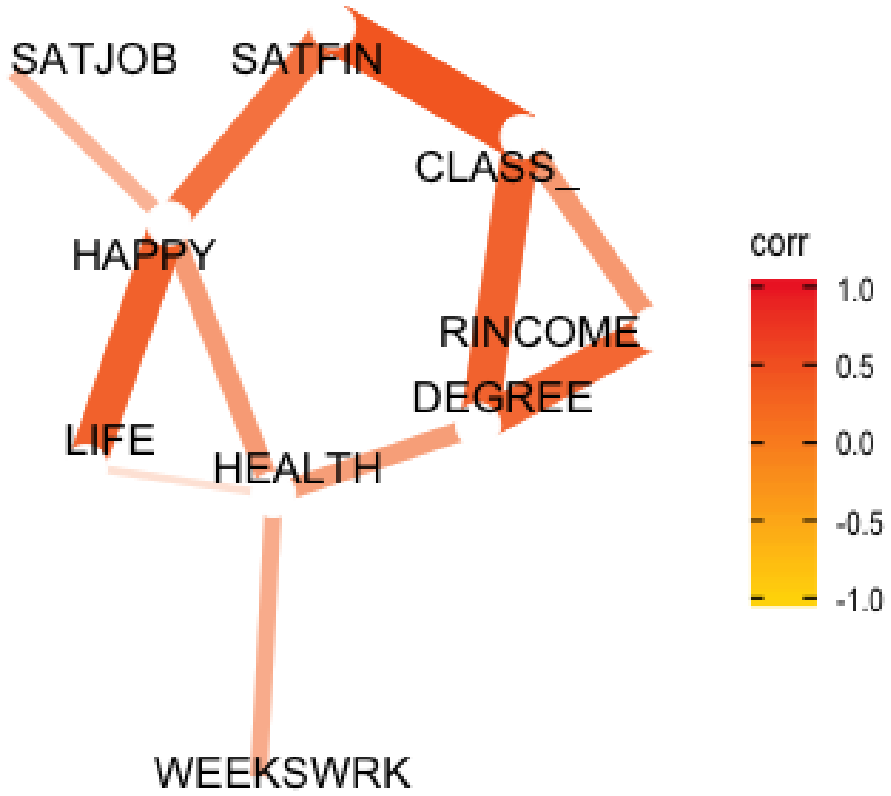


Figure: Large correlations ($|\cdot| > 0.3$) between a few interesting variables from GSS data.

Gaussian copula model outperforms on MovieLens

MovieLens data:

- 207 movies with $\geq$ 1000 ratings
- all users who rate at least one of those 207 movies
- 75% missing rate
- mask additional 10% of data

Table: Imputation Error on 207 Movies

Algorithm	MAE	RMSE
Column Median	0.702(0.004)	1.001(0.004)
Copula-EM	**0.579(0.004)**	**0.880(0.005)**
GLRM (BvSLoss)	0.595(0.004)	0.892(0.004)
softImpute	0.602(0.004)	0.883(0.004)
xPCA	0.613(0.004)	0.897(0.004)
imputeFAMD	0.646(0.005)	0.991(0.005)
missForest	0.669(0.004)	1.015(0.006)

True rank?

selecting the best rank using 5-fold cross validation,

- xPCA uses rank 6
- GLRM model uses rank 8
- softImpute model uses rank 99
- Copula EM model uses full rank covariance Σ

Outline

Low-rank Gaussian copula

Low-rank Gaussian copula model:

- $z \sim \mathcal{N}(0, WW^\top + \sigma I)$
- $x = f(z)$, $f = (f_1, \ldots, f_n)$ entrywise monotonic

where $W \in \mathbf{R}^{p \times r}$, $r \ll p$

Low-rank Gaussian copula

Low-rank Gaussian copula model:

- $z \sim \mathcal{N}(0, WW^\top + \sigma I)$
- $x = f(z)$, $f = (f_1, \ldots, f_n)$ entrywise monotonic

where $W \in \mathbf{R}^{p \times r}$, $r \ll p$

- low rank + diagonal covariance $\Sigma = WW^\top + \sigma I$

Low-rank Gaussian copula

Low-rank Gaussian copula model:

- $z \sim \mathcal{N}(0, WW^\top + \sigma I)$
- $x = f(z)$, $f = (f_1, \ldots, f_n)$ entrywise monotonic

where $W \in \mathbf{R}^{p \times r}$, $r \ll p$

- low rank + diagonal covariance $\Sigma = WW^\top + \sigma I$
- latent z follows probabilistic PCA model

Why use the low-rank Gaussian copula?

advantages of low-rank Gaussian copula:

Why use the low-rank Gaussian copula?

advantages of low-rank Gaussian copula:

- fewer parameters
  - covariance has $O(p)$ parameters for constant r

Why use the low-rank Gaussian copula?

advantages of low-rank Gaussian copula:

- fewer parameters
  - covariance has $O(p)$ parameters for constant r
- faster to estimate
  - for dense data, $O(np)$ flops per EM iteration
  - for sparse data, flops per EM iteration $\sim$ #observations

Why use the low-rank Gaussian copula?

advantages of low-rank Gaussian copula:

- fewer parameters
  - covariance has $O(p)$ parameters for constant r
- faster to estimate
  - for dense data, $O(np)$ flops per EM iteration
  - for sparse data, flops per EM iteration $\sim$ #observations
- insensitive to the chosen rank (unlike low rank models)
  - doesn't overfit
  - generally, bigger is (very modestly) better

Simulation Setting

- Generate $X = f(Z)$ where $Z = TW^\top + \sigma E$.
- $T \in \mathbf{R}^{500 \times k}, W \in \mathbf{R}^{200 \times k}, E \in \mathbf{R}^{500 \times 200}$ with standard normal entries
- Continuous data: set $k = 10, \sigma^2 = 0.1$, missing ratio 40%.
  - Low rank: $X = f(Z) = Z$
  - High rank: $X = f(Z) = Z^3$ (entrywise)
- Ordinal (1-5) and binary data: set $k = 5$, missing ratio 60%.
  - High SNR: $\sigma^2 = 0.1$
  - Low SNR: $\sigma^2 = 0.5$

Simulation: accuracy

Continuous	LRGC	PPCA	softImpute	MMMF-ℓ_2	MMC
Low Rank	.347(.004), $r = 10$	**.338**(**.004**), $r = 10$	.371(.004), $r = 117$	.364(.003)	.633(.007), $r = 130$
High Rank	**.517**(**.011**), $r = 10$	.690(.010), $r = 10$	.703(.005), $r = 104$	.696(.006)	.824(.011), $r = 137$
1-5 ordinal	LRGC	PPCA	softImpute	MMMF-BvS	MMMF-ℓ_1
High SNR	**.358**(**.008**), $r = 5$	.501(.010), $r = 6$	.582(.011), $r = 83$	.407(.007)	.410(.007)
Low SNR	**.788**(**.013**), $r = 5$	.863(.013), $r = 5$	.951(.015), $r = 38$	.850(.011)	.863(.011)
Binary	LRGC	PPCA	MMMF-hinge	MMMF-logistic	MMMF-ℓ_1
High SNR	**.103**(**.003**), $r = 5$	.116(.002), $r = 6$	.140(.002)	.117(.002)	.224(.005)
Low SNR	**.205**(**.006**), $r = 5$	.208(.005), $r = 5$	.234(.006)	.217(.005)	.265(.009)

Table: Imputation error (NRMSE for continuous and MAE for ordinal) reported over 20 repetitions, with rank r for available methods. Maximum margin matrix completion (`MMMF`) are trained at rank 199, implemented using generalized low rank model (GLRM). Monotonic matrix completion (MMC) is aimed at high rank matrix completion.

Simulation: confidence intervals are calibrated

Low Rank Data	LRGC	PPCA	LRMC	MI-PCA
Empirical coverage rate	0.927(.002)	0.940(.001)	0.878(.006)	0.933(.002)
Interval length	1.273(.004)	1.264(.004)	1.129(.015)	1.267(.004)
Run time (in seconds)	6.9(.5)	3.4(.7)	2.7(.4)	189.8(15.4)
High Rank Data	**LRGC**	**PPCA**	**LRMC**	**MI-PCA**
Empirical coverage rate	0.927(.002)	0.943(.002)	0.925(.004)	0.948(.002)
Interval length	3.614(.068)	9.086(.248)	6.546(.191)	9.307(.249)
Run time (in seconds)	7.2(1.2)	0.4(.1)	3.1(.6)	220.0(30.2)

Table: 95% Confidence intervals on synthetic continuous data over 20 repetitions. LRMC stands for low rank matrix completion (Chen et al. 2019). MI-PCA stands for multiple imputation based on PCA (Josse et al. 2011).

Movielens 1M: identify entries with lowest error

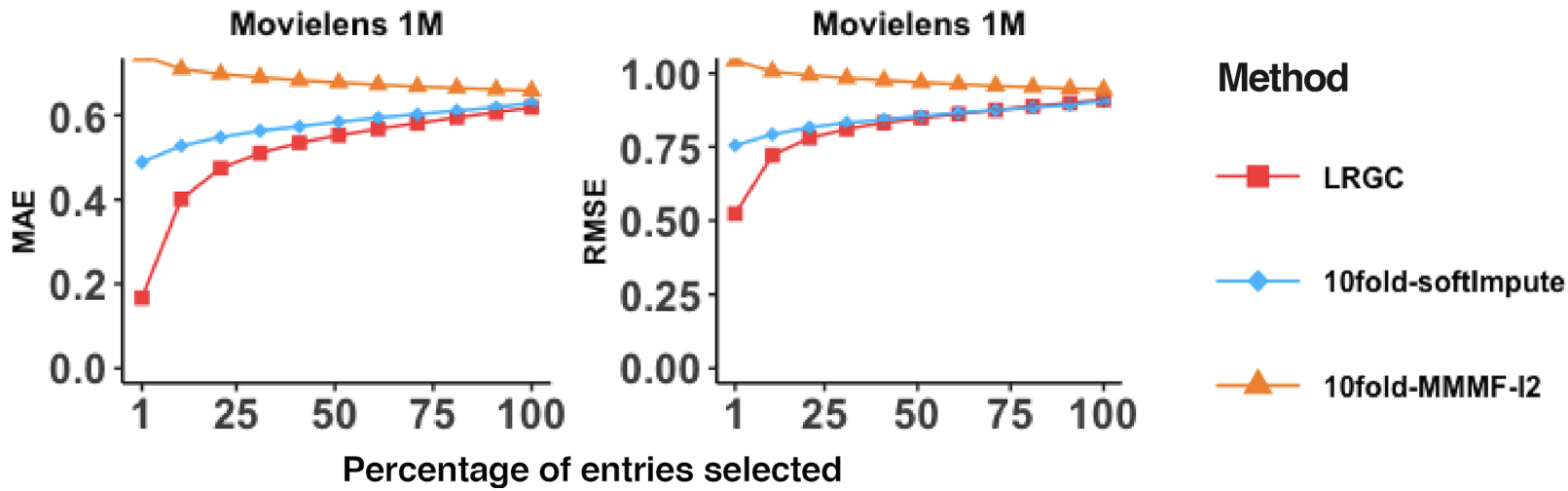


Figure: Movie rating dataset from 6040 users and 3952 movies with 4.19% observation. We select movies with at least 50 ratings, which yields size 6040×2514. The imputation error is plotted at the best tuning parameter over 5 repetitions.

Implementations

- Python: `https://github.com/udellgroup/online_mixed_gc_imp`
  - batch, minibatch, and online EM algorithm
- R: `https://github.com/udellgroup/mixedgcImp`
  - full-rank and low-rank EM algorithm

Conclusion

Gaussian copula model for data imputation:

- no hyperparameters
- outperforms fancier nonparametric methods
- yields confidence intervals for imputations
- fast and easy to estimate for moderate p or (with low rank covariance) large p

references:

- *Missing Value Imputation for Mixed Data through Gaussian Copula.* Y. Zhao and M. Udell, KDD 2020.
- *Matrix Completion with Quantified Uncertainty through Low Rank Gaussian Copula.* Y. Zhao and M. Udell, NeurIPS 2020.
- *Online Mixed Missing Value Imputation Using Gaussian Copula.* E. Landgrebe, Y. Zhao and M. Udell, ICML ARTEMISS Workshop, 2020. Full version (with E. Shekhtman): `https://arxiv.org/abs/2009.12326`.

Outline

*

References

Anderson-Bergman, C., Kolda, T. G., and Kincher-Winoto, K. (2018). XPCA: Extending PCA for a combination of discrete and continuous variables. *arXiv preprint arXiv:1808.07510*.

Cappé, O. and Moulines, E. (2009). On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):593–613.

Chen, Y., Fan, J., Ma, C., and Yan, Y. (2019). Inference and uncertainty quantification for noisy matrix completion. *Proceedings of the National Academy of Sciences*, 116(46):22931–22937.

Ganti, R. S., Balzano, L., and Willett, R. (2015). Matrix completion under monotonic single index models. In *Advances in Neural Information Processing Systems*, pages 1873–1881.

Han, F. and Liu, H. (2014). High dimensional semiparametric scale-invariant principal component analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(10):2016–2032.

Hoff, P. D. et al. (2007). Extending the rank likelihood for semiparametric copula estimation. *The Annals of Applied Statistics*, 1(1):265–283.

Josse, J., Pagès, J., and Husson, F. (2011). Multiple imputation in principal component analysis. *Advances in data analysis and classification*, 5(3):231–246.

Udell, M. and Townsend, A. (2019). Why are big data matrices approximately low rank? *SIAM Journal on Mathematics of Data Science (SIMODS)*, 1(1):144–160.