Missing Value Imputation for Mixed Data via Gaussian Copula

Yuxuan Zhao, Madeleine Udell
Cornell University
{yz2295,udell}@cornell.edu

ABSTRACT

Missing data imputation forms the first critical step of many data analysis pipelines. The challenge is greatest for mixed data sets, including real, Boolean, and ordinal data, where standard techniques for imputation fail basic sanity checks: for example, the imputed values may not follow the same distributions as the data. This paper proposes a new semiparametric algorithm to impute missing values, with no tuning parameters. The algorithm models mixed data as a Gaussian copula. This model can fit arbitrary marginals for continuous variables and can handle ordinal variables with many levels, including Boolean variables as a special case. We develop an efficient approximate EM algorithm to estimate copula parameters from incomplete mixed data. The resulting model reveals the statistical associations among variables. Experimental results on several synthetic and real datasets show the superiority of our proposed algorithm to state-of-the-art imputation algorithms for mixed data.

CCS CONCEPTS

• **Mathematics of computing → Expectation maximization**; **Maximum likelihood estimation**; **Multivariate statistics**; • **Computing methodologies → Learning latent representations**.

KEYWORDS

mixed data, ordinal data, Gaussian copula, missing values, imputation

1 INTRODUCTION

Mixed data sets, including real, Boolean, and ordinal data, are a fixture of modern data analysis. Ordinal data is particularly common in survey datasets. For example, Netflix users rate movies on a scale of 1-5. Social surveys may roughly bin respondents' income or level of education as an ordinal variable, and ordinal Likert scales measure how strongly a respondent agrees with certain stated opinions. Binary variables may be considered a special case of an ordinal with two levels. Health data often contains ordinals that result from patient surveys or from coarse binning of continuous data into, e.g., cancer stages 0–IV or overweight vs obese patients.

In all of these settings, missing data is endemic due to nonresponse and usually represents a large proportion of the full dataset. Missing value imputation generally precedes other analysis, since most machine learning algorithms require complete observations. Imputation quality can strongly influence subsequent analysis. To full exploit the information in mixed data, imputation should take into account the interaction between continuous and ordinal variables. Thus imputation separately for each type is undesirable, while a direct model for the joint distribution can be complex. Existing parametric models are either too restrictive [19] or require priori knowledge on data distribution [35].

Moreover, many available imputation methods treat ordinal data either as continuous or categorical. For example, much of the work on low rank matrix completion for movie rating datasets uses a quadratic loss [5, 13, 17, 26] , which implicitly treats ratings encoded as 1-5 as numerical values. However, for ordinal data, the differences between encoded values are misleading: is the difference between the ratings 4 and 5 the same as the difference between ratings 3 and 4? On the other hand, methods that treat ordinal data as categorical [2, 29] throw away the ordering information. Further, imputation algorithms can usually afford only a limited number of categories. Hence users of these methods may be tempted to treat ordinal data with many levels as continuous, to their detriment. For example, the ordinal variable "Weeks Worked Last Year" from the General Social Survey takes 48 levels, but 74% of the population worked either 0 or 52 weeks. Imputation with the mean works terribly!

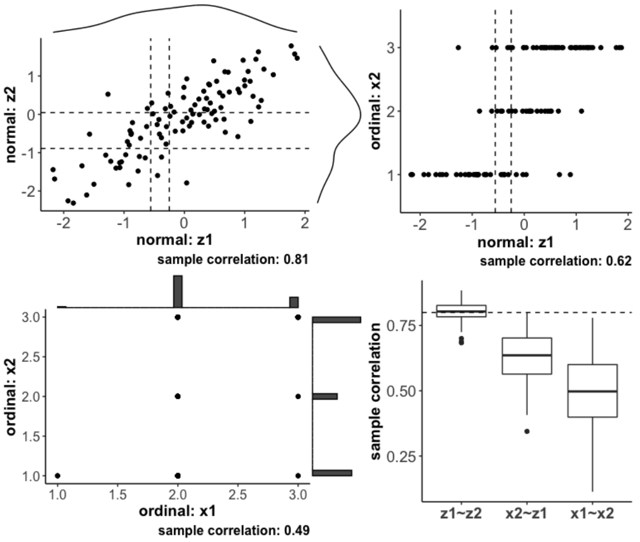

Figure 1: Draw (z_1, z_2) from a binormal with correlation 0.8. Discretize z_1 to x_1, z_2 to x_2 on random cutoffs. Top two and bottom left panels plot one repetition. Dashed lines mark the cutoffs. Bottom right panel plots the sample correlation over 100 repetitions. Dashed line marks the truth.

A more sensible, but still powerful, model, treats ordinal data as generated by thresholding continuous data, as in [27, 28]. In this paper, we associate each ordinal variable with a continuous latent variable. Each ordinal level corresponds to an interval of continuous values. When data follows this model, a naive interpretation of ordinal data can obscure correlation between variables, while correlations can be correctly computed if we can estimate the latent continuous values; see Figure 1.

2 RELATED WORK AND OUR CONTRIBUTION

Gaussian Copula for Mixed Data. Hoff et al. [15] proposes to model mixed data using a Gaussian copula and develops a Bayesian (MCMC) framework to fit the model with incomplete data. Our model matches Hoff's, but our EM algorithm runs substantially faster. Later Fan et al. [8] and Feng and Ning [9] proposed more rigorous versions of Hoff's model for complete observation and examined theoretical properties, with particular focus on discovering a graphical model of the dependency structure. When all variables are ordinal, this model is equivalent to the probit graphical model [11]. However, their corresponding parameter estimation methods are not applicable when there are missing values, and thus imputation is unavailable. To our knowledge, our EM algorithm is the first frequentist approach to fit the Gaussian copula model with incomplete mixed data and to impute missing values using the fitted Gaussian copula. The model we consider reduces to that of [8, 9] when there are no missing values and to [11] when there are no missing values nor continuous dimensions. Murray et al. [23] and Cui et al. [6] propose the Bayesian Gaussian copula factor model and corresponding MCMC algorithms that allows missing data, but focus on model estimation rather than imputation. The sensitivity of this method to internal hyperparameters (eg, number of factors) makes this method a poor choice in practice.

The implementation of [15] is still the best method available to fit a Gaussian copula model for incomplete mixed data. Hollenbach et al. [16] provides an important case study of this method for use in multiple imputation with an application to sociological data analysis. However, the method is slow and sensitive: the burn-in and sampling period must be carefully chosen for MCMC to converge, and many iterations are often required, so the method does not scale to even moderate size data, which limits its use in practice.

Mixed Data Imputation. Several other methods for mixed data imputation are available. Parametric methods [19, 35] make strong distributional assumptions that are generally unwarranted. Nonparametric methods, such as MissForest [29], based on random forests, and imputeFAMD [2], based on principal components analysis, tend to perform better.

In the low rank models literature, the generalized low rank models framework [32] handles missing values imputation for mixed data using a low rank model with appropriately chosen loss functions to ensure proper treatment of each data type. However, choosing the right loss functions for mixed data is challenging. A few papers in the low rank matrix completion literature share our motivation: for example, early papers by Rennie and Srebro [27, 28] proposed a thresholding model to generate ordinals from real low rank matrices. Ganti et al. [10] estimate monotonic transformations of a latent low rank matrix, but the method performs poorly in practice. Anderson-Bergman et al. [1] posits that the mixed data follows a Gaussian copula model with a low rank Gaussian mean and a diagonal covariance. The authors optimize a conditional likelihood function over both model parameters and the missing data rather than marginalizing over missing data; marginalizing is widely understood to produce superior results [19, Chapter 6.3]. Their setup cannot identify the correlations between variables nor the uncertainty of prediction, and empirically underperforms.

Contribution. In this paper, we propose an efficient EM algorithm to estimate a Gaussian copula model with incomplete mixed data and show how to use this model to impute missing values. Our method outperforms many state-of-art imputation algorithms for various real datasets including social survey data (columns with varying number of ordinal levels), movie rating data (high missing ratio), music tagging data (all binary columns), and etc. The proposed method has several advantages: the method has no hyperparameters to tune and is invariant to coordinate-wise monotonic transformations in the data. Moreover, our proposed algorithm is much faster than the existing MCMC algorithm of [15]; given the same time budget, our method produces substantially more accurate estimates. The fitted copula model is interpretable since it can reveal the statistical association among variables, which is useful for social survey studies.

3 NOTATION

Define $[p] = \{1, \ldots, p\}$ for $p \in \mathbb{Z}$. Let $\mathbf{x} = (x_1, \ldots, x_p) \in \mathbb{R}^p$ be a random vector. We use $\mathbf{x}_I$ to denote the subvector of $\mathbf{x}$ with entries in subset $I \subset [p]$. Let $\mathcal{M}, \mathcal{C}, \mathcal{D} \subset [p]$ denote missing, observed continuous, and observed discrete (or ordinal) dimensions, respectively. The observed dimensions are $\mathcal{O} = \mathcal{C} \cup \mathcal{D}$, so $\mathbf{x} = (\mathbf{x}_\mathcal{C}, \mathbf{x}_\mathcal{D}, \mathbf{x}_\mathcal{M}) = (\mathbf{x}_\mathcal{O}, \mathbf{x}_\mathcal{M})$.

Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ be a matrix whose rows correspond to observations and columns to variables. We refer to the i-th row, j-th column, and (i, j)-th element as $\mathbf{x}^i, \mathbf{X}_j$ and x^i_j, respectively.

Two random variables x and $y \in \mathbb{R}$ satisfy $x \overset{d}{=} y$ if their cumulative distribution functions (CDF) match. The elliptope $\mathcal{E} = \{Z \succeq 0 : \mathrm{diag}(Z) = 1\}$ is the set of correlation matrices.

4 GAUSSIAN COPULA

The Gaussian copula models complex multivariate distributions through transformations of a latent Gaussian vector. We call a random variable $x \in \mathbb{R}$ continuous when it is supported on an interval. We can match the marginals of any continuous random vector $\mathbf{x}$ by applying a strictly monotone function to a random vector $\mathbf{z}$ with standard normal marginals. Further, the required function is unique, as stated in Lemma 1.

Lemma 1. Suppose $\mathbf{x} \in \mathbb{R}^p$ is a continuous random vector with CDF F_j for each coordinate $j \in [p]$, and $\mathbf{z} \in \mathbb{R}^p$ is a random vector with standard normal marginals. Then there exists a unique elementwise strictly monotone function $\mathbf{f}(\mathbf{z}) := (f_1(z_1), \ldots, f_p(z_p))$ such that

$$x_j \overset{d}{=} f_j(z_j) \quad \text{and} \quad f_j = F_j^{-1} \circ \Phi, \quad j \in [p] \tag{1}$$

where Φ is the standard normal CDF.

All proofs appear in the supplementary materials. Notice the functions $\{f_j\}_{j=1}^p$ in Eq. (1) are strictly monotone, so their inverses exist. Define $\mathbf{f}^{-1} = (f_1^{-1}, \ldots, f_p^{-1})$. Then $\mathbf{z} = \mathbf{f}^{-1}(\mathbf{x})$ has standard normal marginals, but the joint distribution of $\mathbf{z}$ is not uniquely determined. The Gaussian copula model (or equivalently nonparanormal distribution [20]) further assumes $\mathbf{z}$ is jointly normal.

Definition 1. We say a continuous random vector $\mathbf{x} \in \mathbb{R}^p$ follows the Gaussian copula $\mathbf{x} \sim \mathrm{GC}(\Sigma, \mathbf{f})$ with parameters Σ and $\mathbf{f}$ if there exists a correlation matrix Σ and elementwise strictly monotone function $\mathbf{f} : \mathbb{R}^p \to \mathbb{R}^p$ such that $\mathbf{f}(\mathbf{z}) = \mathbf{x}$ for $\mathbf{z} \sim \mathcal{N}_p(\mathbf{0}, \Sigma)$.

This model is semiparametric: it comprises nonparametric functions $\mathbf{f}$ and parametric copula correlation matrix Σ. The monotone $\mathbf{f}$ establishes the mapping between observed $\mathbf{x}$ and latent $\mathbf{z}$, while Σ fully specifies the distribution of $\mathbf{z}$. Further, the correlation Σ is invariant to elementwise strictly monotone transformation of $\mathbf{x}$. Concretely, if $\mathbf{x} \sim \mathrm{GC}(\Sigma, \mathbf{f})$ and $\mathbf{y} = \mathbf{g}(\mathbf{x})$ where $\mathbf{g}$ is elementwise strictly monotone, then $\mathbf{y} \sim \mathrm{GC}(\Sigma, \mathbf{f} \circ \mathbf{g}^{-1})$. Thus the Gaussian copula separates the multivariate interaction Σ from the marginal distribution $\mathbf{f}$.

When f_j is strictly monotone, x_j must be continuous. On the other hand, when f_j is monotone but not strictly monotone, x_j takes discrete values in the range of f_j and can model ordinals. Thus for ordinals, f_j will not be invertible. For convenience, we define a set-valued inverse $f_j^{-1}(x_j) := \{z_j : f_j(z_j) = x_j\}$. When the ordinal x_j has range $[k]$, Lemma 2 states that the only monotone function f_j mapping continuous z_j to x_j is a cutoff function, defined for some parameter $S \subset \mathbb{R}$ as

$$\mathrm{cutoff}(z; \mathrm{S}) := 1 + \sum_{s \in \mathrm{S}} \mathbb{1}(z > s) \text{ for } z \in \mathbb{R}.$$

Lemma 2. *Suppose $x \in \mathbb{R}$ is an ordinal random variable with range $[k]$ and probability mass function $\{p_l\}_{l=1}^k$ and $z \in \mathbb{R}$ is a continuous random variable with CDF F_z. Then $f = \mathrm{cutoff}(z; \mathrm{S})$ is the unique monotone function f that satisfies $x \overset{d}{=} f(z)$, where $\mathrm{S} = \{s_l = F_z^{-1}\left(\sum_{t=1}^l p_t\right) : l \in [k-1]\}$.*

For example, in recommendation system we can think of the discrete ratings as obtained by rounding some ideal real valued score matrix. The rounding procedure amounts to apply a cutoff function. See Figure 2 for an example of cutoff function.

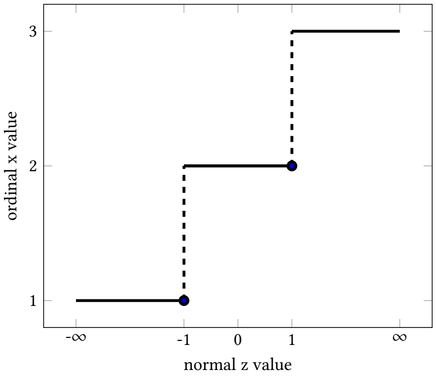

Figure 2: Cutoff function $f(\cdot)$ with cutoffs $\{-1, 1\}$ maps continuous z to ordinal $x \in \{1, 2, 3\}$.

To extend the Gaussian copula to mixed data, we simply specify that f_j is strictly monotone for $j \in \mathcal{C}$ and that f_j is a cutoff function for $j \in \mathcal{D}$. As before, the correlation Σ remains invariant to elementwise strictly monotone transformations. The main difference is that while $f_j^{-1}(x_j)$ is a single number when $j \in \mathcal{C}$ is continuous, it is an interval when $j \in \mathcal{D}$ is discrete. See Figure 3 for illustration.

So far we have introduced a very flexible model for mixed data, which has been explored in graphical model with complete observation [8, 9]. Our interest is to investigate missing value imputation under this model. Suppose the data matrix $\mathbf{X}$ has rows $\mathbf{x}^1, \ldots, \mathbf{x}^n \overset{i.i.d.}{\sim} \mathrm{GC}(\Sigma, \mathbf{f})$ and $\mathbf{x}^i = (\mathbf{x}^i_{\mathcal{C}_i}, \mathbf{x}^i_{\mathcal{D}_i}, \mathbf{x}^i_{\mathcal{M}_i})$ for $i \in [n]$. Define $\mathbf{f}_I = (f_j)_{j \in I}$ for $I \subset [p]$ and $f_j^{-1}(x^i_j) = \mathbb{R}$ for $j \in \mathcal{M}_i$. Given estimates for $\mathbf{f}$ and Σ, we impute in three steps:

(1) Compute the constraints $\mathbf{z}^i \in \mathbf{f}^{-1}(\mathbf{x}^i)$.
(2) Impute $\hat{\mathbf{z}}^i_{\mathcal{M}_i}$ using Σ and constraints on $\mathbf{z}^i_{\mathcal{O}_i}$.
(3) Impute $\hat{\mathbf{x}}^i_{\mathcal{M}_i} = \mathbf{f}_{\mathcal{M}_i}(\hat{\mathbf{z}}^i_{\mathcal{M}_i})$ using imputed $\hat{\mathbf{z}}^i_{\mathcal{M}_i}$.

We show how to estimate $\mathbf{f}$ in Section 5, how to estimate Σ in Section 6, with details in Algorithm 2. The missing completely at random (MCAR) assumption is needed to consistently estimate $\mathbf{f}$. If the true $\mathbf{f}$ is known, the missing at random (MAR) assumption suffices to consistently estimate Σ. We discuss this issue further later in the paper.

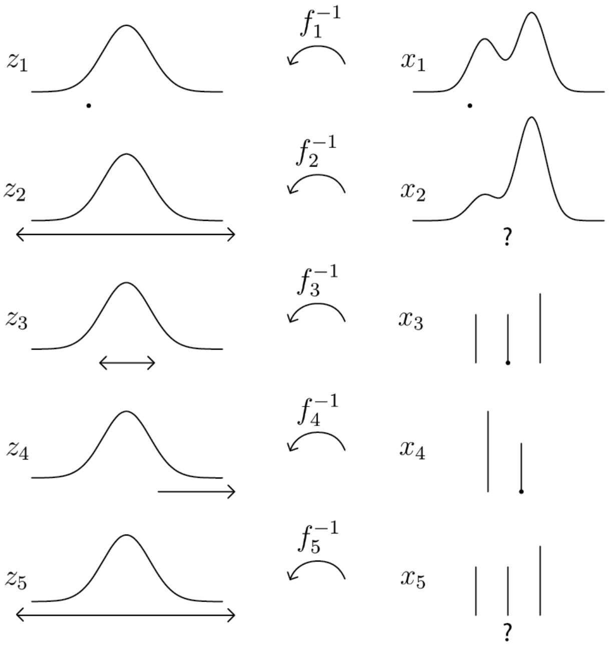

Figure 3: An example of 5 dimensional Gaussian copula model. For observed continuous x_1, the corresponding z_1 takes a fixed value. For observed ordinal x_3 and x_4, the corresponding z_3 and z_4 take values from an interval. For missing continuous x_2 and missing ordinal x_5, the corresponding z_2 and z_5 can take any value.

5 MONOTONIC FUNCTION ESTIMATION

To map between $\mathbf{x}$ and $\mathbf{z}$, we require both $\mathbf{f}^{-1}$ and $\mathbf{f}$. It is easier to directly estimate $\mathbf{f}^{-1}$. For $j \in \mathcal{C}$, we have $f_j^{-1} = \Phi^{-1} \circ F_j$, as shown in Eq. (1). While the true CDF F_j is usually unavailable, it is natural to estimate it by the empirical CDF of $\mathbf{X}_j$ on the observed entries, denoted as $\hat{F}_j$. Let n_j be the observed length of $\mathbf{X}_j$. We use the following estimator:

$$\hat{f}_j^{-1}(x_j^i) = \Phi^{-1}\left(\frac{n_j}{n_j+1}\hat{F}_j(x_j^i)\right). \tag{2}$$

The scale constant $n_j/(n_j+1)$ ensures the output is finite. MCAR assumption guarantees the observed entries of $\mathbf{X}_j$ are from the distribution of F_j. Consider a case when MCAR is violated: an entry is observed if and only if it is smaller than a constant c, then the observed entries are actually from the distribution $\tilde{F}_j$:

$$\tilde{F}_j(x_j) = \begin{cases} F_j(x_j)/F_j(c), & \text{when } x \le c \\ 1, & \text{when } x > c \end{cases}$$

Thus we assume MCAR in this section. This assumption may be relaxed to MAR or even missing not at random by carefully modeling F_j. We leave that to our future work. Lemma 3 shows this estimator converges to f_j^{-1} in sup norm on the observed domain.

Lemma 3. Suppose the continuous random variable $x \in \mathbb{R}$ with CDF F_x and normal random variable $z \in \mathbb{R}$ satisfy $f(z) \overset{d}{=} x$ for a strictly monotone f. Given $x^1, \dots, x^n \overset{i.i.d.}{\sim} F_x$, $m = \min_i x^i$, and $M = \max_i x^i$, the inverse $\hat{f}^{-1}$ defined in Eq. (2) satisfies

$$\mathrm{P}\left(\sup_{m \le x \le M} |\hat{f}^{-1}(x) - f^{-1}(x)| > \epsilon\right) \le 2e^{-c_1 n \epsilon^2}$$

for any ϵ in $a_1 n^{-1} < \epsilon < b_1$, where $a_1, b_1, c_1 > 0$ are constants depending on $F_x(m)$ and $F_x(M)$.

For an ordinal variable $j \in \mathcal{D}$ with k levels, $f_j(z_j) = \text{cutoff}(z_j; \mathbf{S}^j)$. Since $\mathbf{S}^j$ is determined by the probability mass function $\{p_l^j\}$ of x_j, we may estimate cutoffs $\hat{\mathbf{S}}^j$ as a special case of Eq. (2) by replacing p_l^j with its sample mean:

$$\mathbf{S}^j = \left\{\Phi^{-1}\left(\frac{\sum_{i=1}^{n_j} \mathbb{1}(x_j^i \le l)}{n_j+1}\right),\ l \in [k-1]\right\} \tag{3}$$

Lemma 4 shows that $\hat{\mathbf{S}}^j$ consistently estimates $\mathbf{S}^j$.

Lemma 4. Suppose the ordinal random variable $x \in [k]$ with probability mass function $\{p_l\}_{l=1}^k$ and normal random variable $z \in \mathbb{R}$ satisfy $f(z) = \text{cutoff}(z; \mathbf{S}) \overset{d}{=} x$. Given samples $x^1, \cdots, x^n \overset{i.i.d.}{\sim} \{p_l\}_{l=1}^k$, the cutoff estimate $\hat{\mathbf{S}}$ from Eq. (3) satisfies

$$\mathrm{P}\left(||\hat{\mathbf{S}} - \mathbf{S}||_1 > \epsilon\right) \le 2^k e^{-c_2 n \epsilon^2/(k-1)^2}$$

for any ϵ in $(k-1)a_2 n^{-1} < \epsilon < (k-1)b_2$, where $a_2, b_2, c_2 > 0$ are constants depending on $\{p_1, p_k\}$.

6 COPULA CORRELATION MATRIX ESTIMATION

We first consider maximum likelihood estimation (MLE) for Σ with complete continuous observation, then generalize the estimation method to incomplete mixed observation.

6.1 Complete Continuous Observations

We begin by considering continuous, fully observed data: $\mathcal{D} = \mathcal{M} = \emptyset$. The density of the observed variable $\mathbf{x}$ is

$$p(\mathbf{x}; \Sigma, \mathbf{f})\ d\mathbf{x} = \phi_p(\mathbf{z}; \Sigma) d\mathbf{z}$$

where $\mathbf{z} = \mathbf{f}^{-1}(\mathbf{x})$, $d\mathbf{z} = \left|\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right| d\mathbf{x}$, $\phi_p(\cdot; \Sigma)$ is the PDF of the p-dimensional normal with mean $\mathbf{0}$ and covariance Σ. The MLE of Σ maximizes the likelihood function defined as:

$$\begin{aligned} \ell(\Sigma; \mathbf{x}^i) &= \frac{1}{n}\sum_{i=1}^n \log \phi_p(\mathbf{f}^{-1}(\mathbf{x}^i); \Sigma) \\ &= c - \frac{1}{2}\log\det\Sigma - \frac{1}{2}\mathrm{Tr}\left(\Sigma^{-1}\frac{1}{n}\sum_{i=1}^n \mathbf{z}^i(\mathbf{z}^i)^\intercal\right) \end{aligned} \tag{4}$$

over $\Sigma \in \mathcal{E}$, where $\mathbf{z}^i = \mathbf{f}^{-1}(\mathbf{x}^i)$ and c is a universal constant (We omit here and later the constant arising from $\left|\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right|$ after the log transformation). Thus the MLE of Σ is the sample covariance of $\mathbf{Z} := \mathbf{f}(\mathbf{X}) = [f_1(\mathbf{X}_1), \dots, f_p(\mathbf{X}_p)]$. When we substitute $\mathbf{f}$ by its empirical estimation in Eq. (2), the resulting covariance matrix $\tilde{\Sigma}$ of $\hat{\mathbf{Z}} := \hat{\mathbf{f}}(\mathbf{X})$ is still consistent and asymptotically normal under some regularity conditions [30], which justifies the use of our estimator $\hat{\mathbf{f}}$. To simplify notation, we assume $\mathbf{f}$ is known below.

For a Gaussian copula, notice Σ is a correlation matrix, thus we update $\hat{\Sigma} = P_{\mathcal{E}}\tilde{\Sigma}$, where $P_{\mathcal{E}}$ scales its argument to output a correlation matrix: for $D = \mathrm{diag}(\Sigma)$, $P_{\mathcal{E}}(\Sigma) = D^{-1/2}\Sigma D^{-1/2}$. The obtained $\hat{\Sigma}$ is still consistent and asymptotically normal.

6.2 Incomplete Mixed Observations

When some columns are ordinal and some data is missing, the Gaussian latent vector $\mathbf{z}^i$ is no longer fully observed. We can compute the entries of $\mathbf{z}^i$ corresponding to continuous data: $\mathbf{z}^i_{C_i} = \mathbf{f}^{-1}_{C_i}(\mathbf{x}^i_{C_i})$. However, for ordinal data, $\mathbf{f}^{-1}_{\mathcal{D}_i}(\mathbf{x}^i_{\mathcal{D}_i})$ is a Cartesian product of intervals; we only know that $\mathbf{z}^i_{\mathcal{D}_i} \in \mathbf{f}^{-1}_{\mathcal{D}_i}(\mathbf{x}^i_{\mathcal{D}_i})$. The entries corresponding to missing observations, $\mathbf{z}^i_{\mathcal{M}_i}$, are entirely unconstrained. Hence the matrix $\hat{\mathbf{Z}}$ is only incompletely observed, and it is no longer possibly to simply compute its covariance.

We propose an expectation maximization (EM) algorithm to estimate Σ for incomplete mixed observation. Proceeding in an iterative fashion, we replace unknown $\mathbf{z}^i(\mathbf{z}^i)^\intercal$ with their expectation conditional on observations $\mathbf{x}^i_{O_i}$ and an estimate $\hat{\Sigma}$ in the E-step, then in the M-step we update the estimate of Σ as the conditional expectation of covaraince matrix:

$$G(\hat{\Sigma}, \mathbf{x}^i_{O_i}) = \frac{1}{n}\sum_{i=1}^n \mathrm{E}[\mathbf{z}^i(\mathbf{z}^i)^\intercal | \mathbf{x}^i_{O_i}, \hat{\Sigma}] \tag{5}$$

Similar to the case of complete continuous data, we further scale the estimate to a correlation matrix. We first present the EM algorithm in Algorithm 1, then provide precise statements in Section 6.3. Computation details of Algorithm 1 appear in Section 6.4 and Section 6.5.

6.3 EM algorithm

We first write down the marginal density of observed values by integrating out the missing data. Since $\mathbf{x}^i \sim \mathrm{GC}(\Sigma, \mathbf{f})$, there exist

Algorithm 1 EM algorithm for Gaussian Copula

```
Input: observed entries $\mathbf{x}_O$.
Initialize: $t = 0$, $\Sigma^{(0)}$.
For $t = 0, 1, 2, \ldots$
    (1) E-step: Compute $G^{(t)} = G(\Sigma^{(t)}, \mathbf{x}_O)$.
    (2) M-step: $\Sigma^{(t+1)} = G^{(t)}$.
    (3) Scale to correlation matrix: $\Sigma^{(t+1)} = P_{\mathcal{E}}(\Sigma^{(t+1)})$
until convergence.
Output: $\hat{\Sigma} = \Sigma^{(t)}$.
```

latent $\mathbf{z}^i$ satisfying $\mathbf{f}(\mathbf{z}^i) = \mathbf{x}^i$ and $\mathbf{z}^i \sim \mathcal{N}_p(\mathbf{0}, \Sigma)$. The observed likelihood function is:

$$\ell_{\text{obs}}(\Sigma; \mathbf{x}^i_{O_i}) = \frac{1}{n}\sum_{i=1}^{n} \int_{\mathbf{z}^i \in \mathbf{f}^{-1}(\mathbf{x}^i)} \phi_p(\mathbf{z}^i; \mathbf{0}, \Sigma)\, d\mathbf{z}^i \tag{6}$$

Notice the integral region is $\mathbb{R}^{|\mathcal{M}_i|}$ for missing dimensions. With known $\mathbf{f}$, MAR mechanism guarantees the maximizer of the observed likelihood function in Eq. (6) shares the consistency and asymptotic normality of standard maximum likelihood estimate, according to the classical theory [19, Chapter 6.2].

However, the maximizer has no closed form expression. Even direct evaluation of $\ell_{\text{obs}}(\Sigma; \mathbf{x}^i_{O_i})$ is challenging since it involves multivariate Gaussian integrals in a truncated region and the missing locations $\mathcal{M}_i$ varies for different observations i. Instead, the proposed EM algorithm is guaranteed to monotonically converge to a local maximizer according to classical EM theory [22, Chapter 3].

Now we derive the proposed EM algorithm in detail. Suppose we know the values of the unobserved $\mathbf{z}^i$. Then the joint likelihood function is the same as in Eq. (4). Since the values of $\mathbf{z}^i$ are unknown, we treat $\mathbf{z}^i$ as latent variables and $\mathbf{x}^i_{O_i}$ as observed variables. Substituting the joint likelihood function by its expected value given observations $\mathbf{x}^i_O$ and an estimate $\hat{\Sigma}$:

$$\begin{aligned} Q(\Sigma; \hat{\Sigma}, \mathbf{x}^i_{O_i}) &:= \frac{1}{n}\sum_{i=1}^{n} \mathrm{E}[\ell(\Sigma; \mathbf{x}^i_{O_i}, \mathbf{z}^i) | \mathbf{x}^i_{O_i}, \hat{\Sigma}] \\ &= c - \frac{1}{2}\left(\log\det(\Sigma) + \mathrm{Tr}\left(\Sigma^{-1} G(\hat{\Sigma}, \mathbf{x}^i_{O_i})\right)\right) \end{aligned}$$

EM theory [22, Chapter 3] guarantees the updated $\tilde{\Sigma} = \mathrm{argmax}_{\Sigma\in\mathcal{E}}\, Q(\Sigma; \hat{\Sigma}, \mathbf{x}^i_{O_i})$ improves the likelihood with $\hat{\Sigma}$,

$$\ell_{\text{obs}}(\tilde{\Sigma}; \mathbf{x}^i_{O_i}) \geq \ell_{\text{obs}}(\hat{\Sigma}; \mathbf{x}^i_{O_i}),$$

and that by iterating this update, we produce a sequence $\{\Sigma^{(t)}\}$ that converges monotonically to a local maximizer of $\ell_{\text{obs}}(\Sigma; \mathbf{x}^i_{O_i})$. At the t-th iteration, for the E step we compute $\mathrm{E}[\mathbf{z}^i(\mathbf{z}^i)^\intercal | \mathbf{x}^i_{O_i}, \Sigma^{(t)}]$ to express $Q(\Sigma; \Sigma^{(t)}, \mathbf{x}^i_{O_i})$ in terms of Σ. For the M step, we find $\Sigma^{(t+1)} = \mathrm{argmax}_{\Sigma}\, Q(\Sigma; \Sigma^{(t)}, \mathbf{x}^i_{O_i})$. In practice, we resort to an approximation, as in [11]. Notice that the unconstrained maximizer is $\tilde{\Sigma} = G(\Sigma^{(t)}, \mathbf{x}^i_{O_i})$. We update $\Sigma^{(t+1)} = P_{\mathcal{E}}\tilde{\Sigma}$.

6.4 Conditional Expectation Computation

Suppressing index i, we now show how to compute $\mathrm{E}[\mathbf{z}\mathbf{z}^\intercal | \mathbf{x}_O, \Sigma]$ in Eq. (5). With $\mathbf{z}_C = \mathbf{f}_C^{-1}(\mathbf{x}_C)$, it suffices to compute the following terms:

(1) the conditional mean and covariance of observed ordinal dimensions $\mathrm{E}[\mathbf{z}_{\mathcal{D}} | \mathbf{x}_O, \Sigma], \mathrm{Cov}[\mathbf{z}_{\mathcal{D}} | \mathbf{x}_O, \Sigma]$.
(2) the conditional mean and covariance of missing dimensions $\mathrm{E}[\mathbf{z}_{\mathcal{M}} | \mathbf{x}_O, \Sigma], \mathrm{Cov}[\mathbf{z}_{\mathcal{M}} | \mathbf{x}_O, \Sigma]$.
(3) the conditional covariance between missing and observed ordinal dimensions $\mathrm{Cov}[\mathbf{z}_{\mathcal{M}}, \mathbf{z}_{\mathcal{D}} | \mathbf{x}_O, \Sigma]$.

We show that with the results from (1), we can compute (2) and (3). Computation for (1) is put in Sec 6.5.

Suppose we can know the ordinal values $\mathbf{z}_{\mathcal{D}}$ and thus $\mathbf{z}_O$. Conditional on $\mathbf{z}_O$, the missing dimensions $\mathbf{z}_{\mathcal{M}}$ follows normal distribution with mean $\mathrm{E}[\mathbf{z}_{\mathcal{M}} | \mathbf{z}_O, \Sigma] = \Sigma_{\mathcal{M},O}\Sigma_{O,O}^{-1}\mathbf{z}_O$. Further taking expectation of $\mathbf{z}_O$ conditional on observation, we obtain

$$\mathrm{E}[\mathbf{z}_{\mathcal{M}} | \mathbf{x}_O, \Sigma] = \mathrm{E}\left[\mathrm{E}[\mathbf{z}_{\mathcal{M}} | \mathbf{z}_O, \Sigma] \middle| \mathbf{x}_O, \Sigma\right] = \Sigma_{\mathcal{M},O}\Sigma_{O,O}^{-1}\mathrm{E}\left[\mathbf{z}_O | \mathbf{x}_O, \Sigma\right]$$

The above equation also shows how to impute $\mathbf{z}^i_{\mathcal{M}_i}$ using its conditional mean given observed $\mathbf{x}^i_{O_i}$ and estimated $\hat{\Sigma}$.

One can compute $\mathrm{Cov}[\mathbf{z}_{\mathcal{M}} | \mathbf{x}_O, \Sigma]$ and $\mathrm{Cov}[\mathbf{z}_{\mathcal{M}}, \mathbf{z}_{\mathcal{D}} | \mathbf{x}_O, \Sigma]$ similarly: deferring details to the supplement, we find

$$\mathrm{Cov}[\mathbf{z}_{\mathcal{M}}, \mathbf{z}_O | \mathbf{x}_O, \Sigma] = \Sigma_{\mathcal{M},O}\Sigma_{O,O}^{-1}\mathrm{Cov}[\mathbf{z}_O | \mathbf{x}_O, \Sigma],$$

$$\begin{aligned} \mathrm{Cov}[\mathbf{z}_{\mathcal{M}} | \mathbf{x}_O, \Sigma] = \Sigma_{\mathcal{M},\mathcal{M}} &- \Sigma_{\mathcal{M},O}\Sigma_{O,O}^{-1}\Sigma_{O,\mathcal{M}} \\ &+ \Sigma_{\mathcal{M},O}\Sigma_{O,O}^{-1} \cdot \mathrm{Cov}[\mathbf{z}_O | \mathbf{x}_O, \Sigma] \cdot \Sigma_{O,O}^{-1}\Sigma_{O,\mathcal{M}}, \end{aligned}$$

where $\mathrm{Cov}[\mathbf{z}_O | \mathbf{x}_O, \Sigma]$ has $\mathrm{Cov}[\mathbf{z}_{\mathcal{D}} | \mathbf{x}_O, \Sigma]$ as its submatrix and 0 elsewhere, $\mathrm{Cov}[\mathbf{z}_{\mathcal{M}}, \mathbf{z}_O | \mathbf{x}_O, \Sigma]$ has $\mathrm{Cov}[\mathbf{z}_{\mathcal{M}}, \mathbf{z}_{\mathcal{D}} | \mathbf{x}_O, \Sigma]$ as its submatrix and 0 elsewhere.

6.5 Approximating Truncated Normal Moments

Now it remains to compute $\mathrm{E}[\mathbf{z}_{\mathcal{D}} | \mathbf{x}_O, \Sigma]$ and $\mathrm{Cov}[\mathbf{z}_{\mathcal{D}} | \mathbf{x}_O, \Sigma]$, which are the mean and covariance of a $|\mathcal{D}|$-dimensional normal truncated to $\mathbf{f}_{\mathcal{D}}^{-1}(\mathbf{x}_{\mathcal{D}})$, a Cartesian product of intervals. The computation involves multiple integrals of a nonlinear function and only admits a closed form expression when $|\mathcal{D}| = 1$. Direct computational methods [4] are very expensive and can be inaccurate even for moderate $|\mathcal{D}|$. One can sample from truncated normal distribution [24] to evaluate the empirical moments. However, the sampling is still expensive because it needs to be done for each data point at each iteration separately due to varying truncated normal parameters. Instead, we use an approximate method that scales well to large datasets, following [11].

Suppose all but one element of $\mathbf{z}_{\mathcal{D}}$ is known. Then we can easily compute the resulting one dimensional truncated normal mean: for $j \in \mathcal{D}$, if $\mathbf{z}_j$ is unknown and $\mathbf{z}_{\mathcal{D}-j}$ is known, let $\mathrm{E}[z_j | \mathbf{z}_{\mathcal{D}-j}, \mathbf{x}_O, \Sigma] =: g_j(\mathbf{z}_{\mathcal{D}-j}; x_j, \Sigma)$ define the nonlinear function $g_j : \mathbb{R}^{|\mathcal{D}|-1} \to \mathbb{R}$, parameterized by x_j and Σ, detailed in the supplement. We may also use g_j to estimate $\mathrm{E}[z_j | \mathbf{x}_O, \Sigma]$ if $\mathrm{E}[\mathbf{z}_{\mathcal{D}-j} | \mathbf{x}_O, \Sigma]$ is known:

$$\begin{aligned} &\mathrm{E}[z_j | \mathbf{x}_O, \Sigma] = \mathrm{E}[\mathrm{E}[z_j | \mathbf{z}_{\mathcal{D}-j}, \mathbf{x}_O, \Sigma] | \mathbf{x}_O, \Sigma] \\ =&\mathrm{E}[g_j(\mathbf{z}_{\mathcal{D}-j}; x_j, \Sigma) | \mathbf{x}_O, \Sigma] \approx g_j(\mathrm{E}[\mathbf{z}_{\mathcal{D}-j} | \mathbf{x}_O, \Sigma]; x_j, \Sigma) \end{aligned} \tag{7}$$

if g_j is approximately linear. Suppose we have an estimate $\hat{\mathbf{z}}_{\mathcal{D}}^{(t)} \approx \mathrm{E}[\mathbf{z}_{\mathcal{D}}|\mathbf{x}_O, \Sigma^{(t)}]$ at iteration t. Then at iteration $t+1$ we compute

$$\mathrm{E}[z_j|\mathbf{x}_O, \Sigma^{(t+1)}] \approx \hat{z}_j^{(t+1)} := g_j(\hat{\mathbf{z}}_{\mathcal{D}-j}^{(t)}; x_j, \Sigma^{(t+1)}). \tag{8}$$

We use a diagonal approximation for $\mathrm{Cov}\left[\mathbf{z}_{\mathcal{D}}|\mathbf{x}_O, \Sigma\right]$: we approximate $\mathrm{Cov}\left[z_j, z_k|\mathbf{x}_O, \Sigma\right]$ as 0 for $j \neq k \in \mathcal{D}$. This approximation performs well when z_j and z_k are nearly independent given all observed information. We approximate the diagonal entries $\mathrm{Var}\left[z_j|\mathbf{x}_O, \Sigma^{(t+1)}\right]$ for $j \in \mathcal{D}$ using a recursion similar to Eq. (8), detailed in the supplement.

We point out the estimated covariance matrix in Eq. (5) is the sum of the sample covariance matrix of the imputed $\mathbf{z}^i$ using its conditional mean and the expected covariance brought by the imputation. The diagonal approximation only applies to the second term, while the first term is dense. Consequently, the estimator in Eq. (5) is dense and can fit a large range of covariance matrices. Empirical evidence indicates that our approximation works well, shown in Section 8.1.

6.6 Computation Cost

The complexity of each EM iteration is $O(\alpha n p^3)$ with observed entry ratio α. The overall complexity is $O(T\alpha n p^3)$, where T is the number of EM steps required for convergence. We found $T \leq 50$ in most of our experiments. On a laptop with Inter-i5-3.1GHz Core and 8 GB RAM, it takes 1.2min for our algorithm to converge on a dataset with size 2000×60 and 25% missing entries (generated as in Section 8.1 when $p = 60$). Scaling our algorithm to large p is important future work. However, our algorithm is usually faster than many start-of-the-art imputation algorithms for large n small p. Speed comparison on a dataset with size 6039×207 is shown in Section 8.3.

7 IMPUTATION

We have shown how to estimate the model parameters $\mathbf{f}$ and Σ. We summarize the complete imputation approach in Algorithm 2.

Algorithm 2 Imputation via Gaussian Copula

```
Input: $\mathbf{x}_O$, observed entries of $\mathbf{X} \in \mathbb{R}^{n\times p}$.
  (1) Estimate $\hat{\mathbf{f}}^{-1}$ using Eqs. (2) and (3).
  (2) Compute constraints $\mathbf{z}^i_{O_i} \in \hat{\mathbf{f}}^{-1}_{O_i}(\mathbf{x}^i_{O_i}), i \in [n]$.
  (3) Compute $\hat{\Sigma}$ using Algorithm 1.
  (4) For $i = 1, \dots, n$,
        • Impute $\hat{\mathbf{z}}^i_{\mathcal{M}_i} = \mathrm{E}[\mathbf{z}^i_{\mathcal{M}_i}|\mathbf{x}^i_{O_i}, \hat{\Sigma}]$.
        • Impute $\hat{\mathbf{x}}^i_{\mathcal{M}_i} = \hat{\mathbf{f}}_{\mathcal{M}_i}(\hat{\mathbf{z}}^i_{\mathcal{M}_i})$.
Output: $\hat{\mathbf{x}}^i_{\mathcal{M}_i}$ for $i \in [n]$ and $\hat{\Sigma}$.
```

While most applications require just a single imputation, multiple imputations are useful to describe the uncertainty due to imputation. Our method also supports multiple imputation: in step (4) of Algorithm 2, replace the conditional mean imputation with conditional sampling and then impute $\hat{\mathbf{x}}^i_{\mathcal{M}_i}$ for each sample. The conditional sampling consist of two steps: (1) sample $\mathbf{z}^i_{\mathcal{D}_i}$ conditional on $\mathbf{x}^i_{O_i}$ and $\hat{\Sigma}$; (2) sample $\mathbf{z}^i_{\mathcal{M}_i}$ conditional on $\mathbf{x}^i_{O_i}, \mathbf{z}^i_{\mathcal{D}_i}$ and $\hat{\Sigma}$. The first step samples from truncated normal distribution, for which efficient sampling methods have been proposed [24]. The second step samples from normal distribution.

8 EXPERIMENTS

Our first experiment demonstrates that our method, `Copula-EM`, is able to estimate a well-specified Gaussian copula model faster than the MCMC method `sbgcop` [14, 15]. Our other experiments compare the accuracy of imputations produced by `Copula-EM` with `missForest`[29], `xPCA`[1] and `imputeFAMD`[2], state-of-the-art nonparametric imputation algorithms for mixed data; and the low rank matrix completion algorithms `softImpute` [21] and `GLRM`[32], which scale to large datasets. All tuning parameters such as rank and regularization are selected through 5-fold cross validation (5CV), as detailed in the supplement, unless otherwise specified. For real datasets, we report results from our `Copula-EM` but not from `sbgcop`, since `Copula-EM` outperforms on all evaluation metrics and converges substantially faster.

To measure the imputation error on columns in I, we define a scaled mean absolute error (SMAE):

$$\mathrm{SMAE} := \frac{1}{|I|}\sum_{j\in I}\frac{||\hat{\mathbf{X}}_j - \mathbf{X}_j||_1}{||\mathbf{X}_j^{\mathrm{med}} - \mathbf{X}_j||_1},$$

where $\hat{\mathbf{X}}_j, \mathbf{X}_j^{\mathrm{med}}$ are the imputed values and observed median for j-th column, respectively. The estimator's SMAE is smaller than 1 if it outperforms column median imputation. For each data type, the SMAE can be computed on corresponding columns. To evaluate the estimated correlation, we use relative error $||\hat{\Sigma} - \Sigma||_F / ||\Sigma||_F$, where $\hat{\Sigma}$ is the estimated correlation matrix.

8.1 Synthetic Data

The first experiment compares the speed of the two algorithms to estimate Gaussian copula models: `Copula-EM` and `sbgcop`. Note `Copula-EM` is implemented in pure R, while the computational core of `sbgcop` is implemented in C. Hence further acceleration of `Copula-EM` is possible.

We generate 100 synthetic datasets with $n = 2000$ observations and $p = 15$ variables from a well-specified Gaussian copula model with random Σ generated [25]. For each Σ, first generate rows of $\mathbf{Z} \in \mathbb{R}^{n\times p}$ as $\mathbf{z}^1, \cdots, \mathbf{z}^n \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \Sigma)$. Then generate $\mathbf{X} = \mathbf{f}(\mathbf{Z})$ using monotone functions $\mathbf{f}$ such that $\mathbf{X}_1, \dots, \mathbf{X}_5$ have exponential distributions, $\mathbf{X}_6, \dots, \mathbf{X}_{10}$ are binary and $\mathbf{X}_{11}, \dots, \mathbf{X}_{15}$ are ordinal with 5 levels.

We randomly remove 30% of the entries of $\mathbf{X}$, train `Copula-EM` and `sbgcop`, and compute the imputation error on the held-out set. We plot the imputation accuracy and correlation estimation accuracy versus runtime of each algorithm in Figure 4. `Copula-EM` converges quickly, in about 25s, while `sbgcop` takes much longer and suffers high error at shorter times. `Copula-EM` estimates correlations and continuous imputations at convergence more accurately than `sbgcop` even when the latter algorithm is given 6 times more runtime. Interestingly, `Copula-EM` recovers the correlation matrix better than `sbgcop` even asymptotically. These results demonstrate the impact of the approximate EM algorithm 6.5 compared to the (fully accurate) MCMC model of sbgcop: the approximation allows faster convergence, to an estimate of nearly the same quality.

For ordinal data imputation, Copula-EM reaches the same performance as sbgcop 6 times faster. For binary data imputation, sbgcop is four times slower than Copula-EM at reaching the final performance of Copula-EM, but sbgcop outperforms Copula-EM given even more time. We conjecture that the drop in imputation accuracy of Copula-EM for binary data could be mitigated using multiple imputation [19, Chapter 5.4], as outlined in Sec 7 by combining the imputations (using mean or median) into a single imputation to reduce the effect of our approximation to the truncated normal distribution.

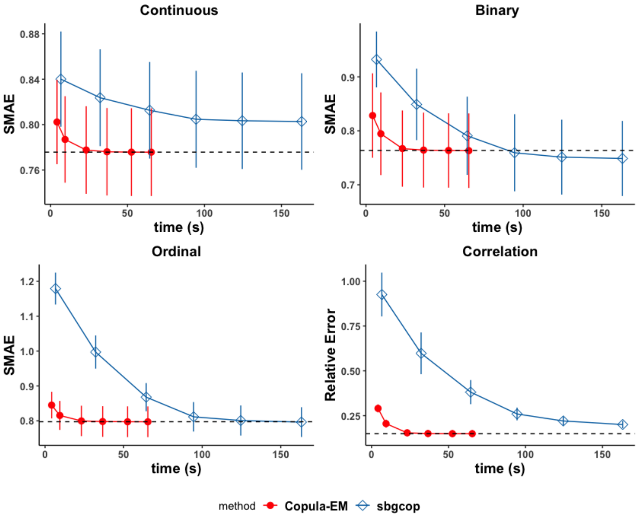

Figure 4: Copula-EM vs sbgcop: The imputation error for each data type and estimated correlation error over time cost. Dashed line indicates the final error of Copula-EM.

The second experiment compares the imputation accuracy of Copula-EM and nonparametric algorithms. Using the same data generation mechanism, we randomly remove 10% – 50% of the entries of $\mathbf{X}$. The optimal rank selected using 5CV is 3 for xPCA and 6 for imputeFAMD. Shown in Figure 5,Copula-EM substantially outperforms all nonparametric algorithms for all data types.

8.2 General Social Survey (GSS) Data

We chose 18 variables with 2538 observations from GSS dataset in year 2014. 24.9% of the entries are missing. The dataset consists of 1 continuous (AGE) and 17 ordinal variables with 2 to 48 levels. We investigate the imputation accuracy on five selected variables: INCOME, LIFE, HEALTH, CLASS[1] and HAPPY. For each variable, we sample 1500 observation and divide them into 20 folds. We mask one fold of only one variable as test data in each experiment. The selected rank is 2 for both xPCA and imputeFAMD. We report the SMAE for each variable in Table 1. Our method performs the best for all variables. Further our method always performs better than median imputation. In contrast, the other three methods perform worse than median imputation for some variables. Our method also provides estimated variable correlation, which is usually desired in social survey study. We plot high correlations from the copula correlation matrix as a graph in Figure 6.

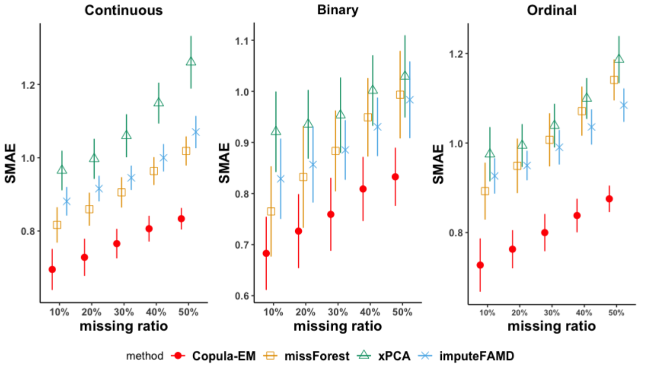

Figure 5: Copula-EM vs nonparametric algorithms: The imputation error for each data type on synthetic data.

Table 1: Imputation Error on Five GSS Variables

Variable	Copula-EM	missForest	xPCA	imputeFAMD
CLASS	**0.735(0.10)**	0.782(0.09)	0.795(0.08)	0.797(0.10)
LIFE	**0.759(0.12)**	0.828(0.17)	0.783(0.11)	0.821(0.11)
HEALTH	**0.877(0.09)**	1.143(0.18)	0.908(0.10)	0.947(0.04)
HAPPY	**0.896(0.08)**	1.079(0.15)	1.003(0.15)	1.001(0.10)
INCOME	**0.869(0.07)**	0.944(0.18)	1.090(0.15)	0.996(0.01)

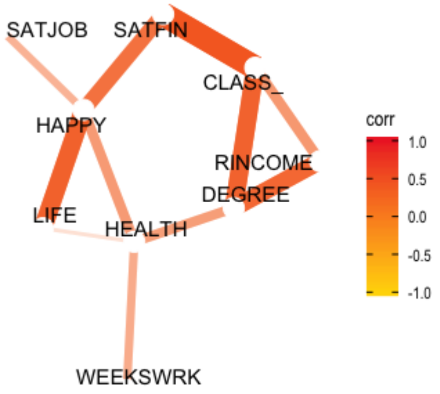

Figure 6: High Correlations ($|\cdot| > 0.3$) of 5 interesting variables from GSS data are plotted.

8.3 MovieLens 1M Data

Recall our method scales cubicly in the number of variables. Hence for this experiment, we sample the subset of the MovieLens 1M data [12] consisting of the 207 movies with at least 1000 ratings and all users who rate at least one of those 207 movies. On this subset,

[1] Subjective class identification from lower to upper class

Table 2: 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	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)

75.6% of entries are missing. The selected rank using 5CV is 99 for `softImpute`, 6 for `xPCA` and 8 for `GLRM` with bigger-vs-smaller loss. For `imputeFAMD`, we select the best rank 3 under time limit 5 hours. Then we manually mask 10% of the data for the test set and use the remaining data to train the model, and repeat 20 times. We round the imputed values to a 1-5 integer for `softImpute`.

As for the time required, `Copula-EM` takes 9 mins and `missForest` takes 25 mins. These two methods have no parameters to tune. With fixed tuning parameters, low rank matrix completion methods are substantially faster. For example, `softImpute` only takes 33s. However, those methods need additional time to select tuning parameters. Using 5CV, `softImpute` takes 16mins to select the optimal tuning parameters with regularization path length 50, which is already more expensive than `Copula-EM`. Interestingly, the ranks selected by 5CV are quite different even when the models perform similarly: `GLRM` chooses rank 8 while `softImpute` chooses rank 99. `imputeFAMD` takes more than 4 hours, suggesting the current implementation does not scale to even moderate size data.

We report both mean absolute error (MAE) and RMSE in Table 2. Our method outperforms all others in both MAE and RMSE. This result is notable, because `Copula-EM` does not directly minimize MAE or RMSE, while `softImpute` directly minimizes RMSE.

8.4 Music Auto-tagging: CAL500exp Data

The CAL500 expansion (CAL500exp) dataset [36] is an enriched version of the well-known CAL500 dataset [31]. This dataset consists of 67 binary tags (including genre, mood and instrument, labeled by experts) to 3223 music fragments from 500 songs. Music auto-tagging is a multi-label learning problem. A feature vector is usually computed first based on the music files and then a classifier is trained for each tag. This procedure is expensive and neglects the association among known labels. We treat this task as a missing data imputation problem and only use observed labels to impute unknown labels. This dataset is completely observed. We randomly remove some portions of the observed labels as a test set and repeat 20 times. The selected optimal rank is 4 for `xPCA` and 15 for `imputeFAMD`. Shown in Figure 7, `Copula-EM` performs the best in terms of SMAE. The superiority of `Copula-EM` over other algorithms substantially grows as the missing ratio increases. Moreover, `Copula-EM` yields very stable imputations: the standard deviation of its SMAE is imperceptibly small.

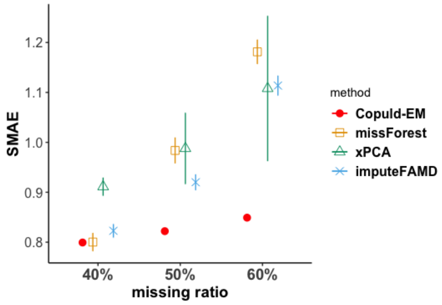

Figure 7: The imputation error on CAL500exp.

8.5 More Ordinal Data and Mixed Data

We compare mixed data imputation algorithms on two more ordinal classification datasets[2], Lecturers Evaluation *(LEV)* and Employee Selection *(ESL)*, and two more mixed datasets, German Breast Cancer Study Group *(GBSG)*[3] and Restaurant Tips *(TIPS)*[4]. Dataset descriptions appear in Table 3, and more details appear in the supplement. All datasets are completely observed.

For each dataset, we randomly remove 30% entries as a test set and repeat 100 times. For ordinal classification datasets, we evaluate the SMAE for the label and for the features, respectively. For mixed datasets, we evaluate the SMAE for ordinal dimensions and for continuous dimensions, respectively. We report results in Table 3. Our method outperforms the others in all but one setting, often by a substantial margin.

9 SUMMARY AND DISCUSSION

In this paper, we proposed an imputation algorithm that models mixed data with a Gaussian copula model, together with an effective approximate EM algorithm to estimate the copula correlation with incomplete mixed data. Our algorithm has no tuning parameter and are easy to implement. Our experiments demonstrate the success of the proposed method. Scaling these methods to larger datasets (especially, with more columns), constitutes important future work.

We end by noting a few contrasts between the present approach and typical low rank approximation methods for data imputation. Low rank approximation constructs a latent simple (low rank) object and posits that observations are noisy draws from that simple latent object. In contrast, our approach uses a parametric, but full-dimensional, model for the latent object; observations are given by a deterministic function of the latent object. In other words, in previous work the latent object is exact and the observations are noisy; in our work, the latent object is noisy and the observations are exact. Which more faithfully models real data? As evidence, we might consider whether low rank models agree on the best rank to fit a given dataset. For example, on the MovieLens dataset: (1) The low rank matrix completion methods `xPCA` and `GLRM`, implemented using alternating minimization, select small optimal ranks (6 and 8), while `softImpute`, implemented using nuclear norm minimization, selects the much larger optimal rank 99. (2) Our algorithm

[2] Available at https://waikato.github.io/weka-wiki/datasets/
[3] Available at https://cran.r-project.org/web/packages/mfp/
[4] Available at http://ggobi.org/book/

Table 3: Imputation Error on More Ordinal and Mixed Datasets.

Dataset	Size	Selected Rank	Type	Copula-EM	missForest	xPCA	imputeFAMD
ESL	488×5	1 (xPCA)	Label	**0.372(0.04)**	0.553(0.08)	0.404(0.04)	0.503(0.06)
ESL	4 features, 1 label	5 (imputeFAMD)	Feature	**0.584(0.03)**	0.873(0.06)	0.668(0.03)	0.687(0.03)
LEV	1000×5	1 (xPCA)	Label	**0.750(0.04)**	0.970(0.09)	0.860(0.06)	0.882(0.05)
LEV	4 features, 1 label	5 (imputeFAMD)	Feature	0.907(0.01)	**0.799(0.03)**	1.037(0.02)	1.085(0.04)
GBSG	686×10	2 (xPCA)	Ordinal	**0.793(0.03)**	0.887(0.05)	0.876(0.04)	0.840(0.03)
GBSG	6 continuous, 4 ordinal	2 (imputeFAMD)	Continuous	**0.876(0.01)**	1.029(0.03)	1.100(0.04)	1.038(0.03)
TIPS	244×7	2 (xPCA)	Ordinal	**0.786(0.05)**	0.928(0.09)	0.928(0.08)	0.891(0.09)
TIPS	2 continuous, 5 ordinal	6 (imputeFAMD)	Continuous	**0.755(0.04)**	0.837(0.05)	1.011(0.11)	0.892(0.13)

outperforms all the low rank matrix completion methods we tested. These observations suggest the low rank assumption commonly used to fit the MovieLens dataset may not be fundamental, but may arise as a mathematical artifact [33]. More supporting empirical results can be found in [3]: the performance of `softImpute` keeps improving as the rank increases (up to 10^3).

ACKNOWLEDGMENTS

This work was supported in part by DARPA Award FA8750-17-2-0101. The authors wish to thank Tamara G. Kolda for her help in implementing xPCA, Julie Josse for her help in implementing imputeFAMD.

REFERENCES

[1] Clifford Anderson-Bergman, Tamara G Kolda, and Kina Kincher-Winoto. 2018. XPCA: Extending PCA for a Combination of Discrete and Continuous Variables. *arXiv preprint arXiv:1808.07510* (2018).

[2] Vincent Audigier, François Husson, and Julie Josse. 2016. A principal component method to impute missing values for mixed data. *Advances in Data Analysis and Classification* 10, 1 (2016), 5–26.

[3] Haim Avron, Satyen Kale, Shiva Kasiviswanathan, and Vikas Sindhwani. 2012. Efficient and practical stochastic subgradient descent for nuclear norm regularization. *arXiv preprint arXiv:1206.6384* (2012).

[4] Manjunath BG and Stefan Wilhelm. 2009. Moments calculation for the double truncated multivariate normal density. *Available at SSRN 1472153* (2009).

[5] Emmanuel J Candès and Benjamin Recht. 2009. Exact matrix completion via convex optimization. *Foundations of Computational mathematics* 9, 6 (2009), 717.

[6] Ruifei Cui, Ioan Gabriel Bucur, Perry Groot, and Tom Heskes. 2019. A novel Bayesian approach for latent variable modeling from mixed data with missing values. *Statistics and Computing* 29, 5 (2019), 977–993.

[7] Aryeh Dvoretzky, Jack Kiefer, Jacob Wolfowitz, et al. 1956. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics* 27, 3 (1956), 642–669.

[8] Jianqing Fan, 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 (2017), 405–421.

[9] Huijie Feng and Yang Ning. 2019. High-dimensional Mixed Graphical Model with Ordinal Data: Parameter Estimation and Statistical Inference. In *The 22nd International Conference on Artificial Intelligence and Statistics*. 654–663.

[10] Ravi Sastry Ganti, Laura Balzano, and Rebecca Willett. 2015. Matrix completion under monotonic single index models. In *Advances in Neural Information Processing Systems*. 1873–1881.

[11] Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. 2015. Graphical models for ordinal data. *Journal of Computational and Graphical Statistics* 24, 1 (2015), 183–204.

[12] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2016), 19.

[13] Trevor Hastie, Rahul Mazumder, Jason D Lee, and Reza Zadeh. 2015. Matrix completion and low-rank SVD via fast alternating least squares. *The Journal of Machine Learning Research* 16, 1 (2015), 3367–3402.

[14] Peter Hoff and Maintainer Peter Hoff. 2018. Package âĂŸsbgcopâĂŹ. (2018).

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

[16] Florian M Hollenbach, Iavor Bojinov, Shahryar Minhas, Nils W Metternich, Michael D Ward, and Alexander Volfovsky. 2018. Multiple Imputation Using Gaussian Copulas. *Sociological Methods & Research* (2018), 0049124118799381.

[17] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. 2010. Matrix completion from noisy entries. *Journal of Machine Learning Research* 11, Jul (2010), 2057–2078.

[18] Michael R Kosorok. 2008. *Introduction to empirical processes and semiparametric inference.* Springer.

[19] Roderick JA Little and Donald B Rubin. 2019. *Statistical analysis with missing data*. Vol. 793. Wiley.

[20] Han Liu, John Lafferty, and Larry Wasserman. 2009. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research* 10, Oct (2009), 2295–2328.

[21] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. 2010. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research* 11, Aug (2010), 2287–2322.

[22] Geoffrey McLachlan and Thriyambakam Krishnan. 2007. *The EM algorithm and extensions*. Vol. 382. John Wiley & Sons.

[23] Jared S Murray, David B Dunson, Lawrence Carin, and Joseph E Lucas. 2013. Bayesian Gaussian copula factor models for mixed data. *J. Amer. Statist. Assoc.* 108, 502 (2013), 656–665.

[24] Ari Pakman and Liam Paninski. 2014. Exact hamiltonian monte carlo for truncated multivariate gaussians. *Journal of Computational and Graphical Statistics* 23, 2 (2014), 518–542.

[25] Weiliang Qiu and Harry Joe. 2009. clusterGeneration: random cluster generation (with specified degree of separation). *R package version* 1, 7 (2009), 75275–0122.

[26] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. 2010. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review* 52, 3 (2010), 471–501.

[27] Jasson DM Rennie and Nathan Srebro. 2005. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*. ACM, 713–719.

[28] Jason DM Rennie and Nathan Srebro. 2005. Loss functions for preference levels: Regression with discrete ordered labels. In *Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling*. Kluwer Norwell, MA, 180–186.

[29] Daniel J Stekhoven and Peter Bühlmann. 2011. MissForestâĂŤnon-parametric missing value imputation for mixed-type data. *Bioinformatics* 28, 1 (2011), 112–118.

[30] Hideatsu Tsukahara. 2005. Semiparametric estimation in copula models. *Canadian Journal of Statistics* 33, 3 (2005), 357–375.

[31] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. 2007. Towards musical query-by-semantic-description using the cal500 data set. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 439–446.

[32] Madeleine Udell, Corinne Horn, Reza Zadeh, Stephen Boyd, et al. 2016. Generalized low rank models. *Foundations and Trends® in Machine Learning* 9, 1 (2016), 1–118.

[33] Madeleine Udell and Alex Townsend. 2019. Why are Big Data Matrices Approximately Low Rank? *SIAM Journal on Mathematics of Data Science (SIMODS)* 1, 1 (2019), 144–160. https://epubs.siam.org/doi/pdf/10.1137/18M1183480

[34] Aad W Vaart and Jon A Wellner. 1996. *Weak convergence and empirical processes: with applications to statistics*. Springer.

[35] Stef Van Buuren and Karin Oudshoorn. 1999. *Flexible multivariate imputation by MICE*. Leiden: TNO.

[36] Shuo-Yang Wang, Ju-Chiang Wang, Yi-Hsuan Yang, and Hsin-Min Wang. 2014. Towards time-varying music auto-tagging based on CAL500 expansion. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1–6.

A COMPUTATION DETAIL

A.1 Details for Section 6.5

Denote the observation $\{\mathbf{x}_\mathcal{O}, \Sigma\}$ i.e. $\{\mathbf{z}_\mathcal{C} = \mathbf{f}_\mathcal{C}^{-1}(\mathbf{x}_\mathcal{C}), \mathbf{z}_\mathcal{D} \in \mathbf{f}_\mathcal{D}^{-1}(\mathbf{x}_\mathcal{D}), \Sigma\}$ as $\{*\}$. Since the task is to compute the marginal mean and variance of a multivariate truncated normal, we suppose $\mathcal{M} = \emptyset$ here without loss of generality. For each $j \in \mathcal{D}$, we use the law of total expectation by conditional on $\mathbf{z}_{\mathcal{D}-j}$ first. Given $\{*, \mathbf{z}_{\mathcal{D}-j}\}$, z_j is univariate normal with mean $\tilde{\mu}_j = \Sigma_{j,-j}\Sigma_{-j,-j}^{-1}\mathbf{z}_{-j}$ and variance $\tilde{\sigma}_j^2 = 1 - \Sigma_{j,-j}\Sigma_{-j,-j}^{-1}\Sigma_{-j,j}$, truncated to the region $f_j^{-1}(x_j)$, where indexing $-j$ means all dimensions but j i.e. $[p]-j$. The region $f_j^{-1}(x_j)$ is an interval: $f_j^{-1}(x_j) = (a_j, b_j]$. Here are three cases: (1)$a_j, b_j \in \mathbb{R}$; (2)$a_j \in \mathbb{R}, b_j = \infty$; (3)$a_j = -\infty, b_j \in \mathbb{R}$. The computation for all cases are similar. We take the first case as an example. First we introduce a lemma for univariate truncated normal.

Lemma 5. Suppose a univariate random variable $z \sim \mathcal{N}(\mu, \sigma^2)$. For constants $a < b$, let $\alpha = (a - \mu)/\sigma$ and $\beta = (b - \mu)/\sigma$. Then the mean and variance of z truncated to the interval $(a, b]$ are:

$$\mathrm{E}(z|a < z \leq b) = \mu + \frac{\phi(\alpha) - \phi(\beta)}{\Phi(\beta) - \Phi(\alpha)} \cdot \sigma$$

$$\mathrm{Var}(z|a < z \leq b) = \left(1 + \frac{\alpha\phi(\alpha) - \beta\phi(\beta)}{\Phi(\beta) - \Phi(\alpha)} - \left(\frac{\phi(\alpha) - \phi(\beta)}{\Phi(\beta) - \Phi(\alpha)}\right)^2\right)\sigma^2$$

Plugging $\mu = \tilde{\mu}_j, \sigma^2 = \tilde{\sigma}_j^2$ and $(a, b] = f_j^{-1}(x_j)$ into the above mean and variance formulas, we obtain the expression of $g_j(\mathbf{z}_{\mathcal{D}-j}; x_j, \Sigma)$ defined in Section 6.5, and the univariate truncated normal variance $\mathrm{Var}[z_j|\mathbf{z}_{\mathcal{D}-j}, \mathbf{x}_\mathcal{O}, \Sigma] =: h_j(\mathbf{z}_{\mathcal{D}-j}; x_j, \Sigma)$, a nonlinear function $\mathbb{R}^{|\mathcal{D}|-1} \longrightarrow \mathbb{R}$, parameterized by x_j and Σ. Write down the formula for marginal variance conditional on observation:

$$\begin{aligned}\mathrm{Var}[z_j|*] &= \mathrm{E}\left[\mathrm{Var}[z_j|\mathbf{z}_{\mathcal{D}-j}, *]\middle|*\right] + \mathrm{Var}\left[\mathrm{E}[z_j|\mathbf{z}_{\mathcal{D}-j}, *]\middle|*\right] \\ &= \mathrm{E}\left[h_j(\mathbf{z}_{\mathcal{D}-j}; x_j, \Sigma)\middle|*\right] + \mathrm{Var}\left[g_j(\mathbf{z}_{\mathcal{D}-j}; x_j, \Sigma)\middle|*\right]\end{aligned}$$

We approximate the first term as $h_j(\mathrm{E}[\mathbf{z}_{\mathcal{D}-j}|*]; x_j, \Sigma)$. As for the second term, Guo et al. [11] approximated it as $\mathrm{Var}[\tilde{\mu}_j|*]$ based on $\mathrm{E}\left[g_j^2(\mathbf{z}_{\mathcal{D}-j}; x_j, \Sigma)\middle|*\right] \approx g_j^2(\mathrm{E}[\mathbf{z}_{\mathcal{D}-j}|*]; x_j, \Sigma)$. However, we found in practice simply dropping the second term performs better.

In summary, given an estimate $\hat{\mathbf{z}}_\mathcal{D}^{(t)} \approx \mathrm{E}[\mathbf{z}_\mathcal{D}|\mathbf{x}_\mathcal{O}, \Sigma^{(t)}]$ and $\Sigma^{(t+1)}$, for $j \in \mathcal{D}$, we update $\mathrm{E}[z_j|\mathbf{x}_\mathcal{O}, \Sigma^{(t+1)}] \approx g_j(\hat{\mathbf{z}}_{\mathcal{D}-j}^{(t)}; x_j, \Sigma^{(t+1)})$ and $\mathrm{Var}[z_j|\mathbf{x}_\mathcal{O}, \Sigma^{(t+1)}] \approx h_j(\hat{\mathbf{z}}_{\mathcal{D}-j}^{(t)}; x_j, \Sigma^{(t+1)})$. In other words, we update the conditional mean and variance of z_j as the univariate truncated normal mean and variance with all other observed ordinal dimensions equal to their mean from last iteration, i.e. $\mathbf{z}_{\mathcal{D}-j} = \hat{\mathbf{z}}_{\mathcal{D}-j}^{(t)}$.

A.2 Details for Section 6.4

Given $\mathrm{E}[\mathbf{z}_\mathcal{O}|*]$, $\mathrm{E}[\mathbf{z}_\mathcal{M}|*]$ and $\mathrm{Cov}[\mathbf{z}_\mathcal{O}|*]$, it suffices to compute $\mathrm{E}[\mathbf{z}_\mathcal{M}\mathbf{z}_\mathcal{O}^\intercal|*]$ and $\mathrm{E}[\mathbf{z}_\mathcal{M}\mathbf{z}_\mathcal{M}^\intercal|*]$ for $\mathrm{Cov}[\mathbf{z}_\mathcal{M}, \mathbf{z}_\mathcal{O}|*]$ and $\mathrm{Cov}[\mathbf{z}_\mathcal{M}|*]$. Using the law of total expectation, we have:

$$\begin{aligned}&\mathrm{E}[\mathbf{z}_\mathcal{M}\mathbf{z}_\mathcal{O}^\intercal|*] = \mathrm{E}\left[\mathrm{E}[\mathbf{z}_\mathcal{M}\mathbf{z}_\mathcal{O}^\intercal|\mathbf{z}_\mathcal{O}, *]\middle|*\right] = \mathrm{E}\left[\mathrm{E}[\mathbf{z}_\mathcal{M}|\mathbf{z}_\mathcal{O}, *] \cdot \mathbf{z}_\mathcal{O}^\intercal\middle|*\right] \\ =&\mathrm{E}\left[\Sigma_{\mathcal{M},\mathcal{O}}\Sigma_{\mathcal{O},\mathcal{O}}^{-1}\mathbf{z}_\mathcal{O} \cdot \mathbf{z}_\mathcal{O}^\intercal\middle|*\right] = \Sigma_{\mathcal{M},\mathcal{O}}\Sigma_{\mathcal{O},\mathcal{O}}^{-1}\mathrm{E}[\mathbf{z}_\mathcal{O}\mathbf{z}_\mathcal{O}^\intercal|*].\end{aligned}$$

$$\begin{aligned}&\mathrm{E}[\mathbf{z}_\mathcal{M}\mathbf{z}_\mathcal{M}^\intercal|*] = \mathrm{E}\left[\mathrm{E}[\mathbf{z}_\mathcal{M}\mathbf{z}_\mathcal{M}^\intercal|\mathbf{z}_\mathcal{O}, *]\middle|*\right] \\ =&\mathrm{E}\left[\mathrm{Cov}[\mathbf{z}_\mathcal{M}|\mathbf{z}_\mathcal{O}, *]\middle|*\right] + \mathrm{E}\left[\mathrm{E}[\mathbf{z}_\mathcal{M}|\mathbf{z}_\mathcal{O}, *] \cdot \mathrm{E}[\mathbf{z}_\mathcal{M}^\intercal|\mathbf{z}_\mathcal{O}, *]\middle|*\right] \\ =&\Sigma_{\mathcal{M},\mathcal{M}} - \Sigma_{\mathcal{M},\mathcal{O}}\Sigma_{\mathcal{O},\mathcal{O}}^{-1}\Sigma_{\mathcal{O},\mathcal{M}} + \Sigma_{\mathcal{M},\mathcal{O}}\Sigma_{\mathcal{O},\mathcal{O}}^{-1}\mathrm{E}[\mathbf{z}_\mathcal{O}|*]\mathrm{E}[\mathbf{z}_\mathcal{O}^\intercal|*]\Sigma_{\mathcal{O},\mathcal{O}}^{-1}\Sigma_{\mathcal{O},\mathcal{M}}\end{aligned}$$

B SUPPLEMENT FOR EXPERIMENTS

B.1 Results of sbgcop on Real Datasets

For GSS data, Copula-EM takes 24s, while sbgcop with 1000 iterations takes 87s, with imputation error: CALSS, 0.992(0.13); LIFE, 0.924(0.7); HEALTH, 1.132(0.15); HAPPY, 1.231(0.11); INCOME, 0.931(0.03).

For movielens data, Copula-EM takes 9 mins, while sbgcop with 200 iterations takes 33 mins, with imputation error: MAE, 0.752(0.004); RMSE, 1.030(0.005).

For CAL500exp data, Copula-EM takes 80s, while sbgcop with 500 iterations takes 290s, with imputation error: 1.301(0.019) for 40% missing ratio; 1.328(0.015) for 50% missing ratio; 1.379(0.016) for 60% missing ratio.

For four small datasets used in Section 8.5, the time sbgcop with 1000 iterations takes is 2 times to 9 times (varying over datasets) of the time Copula-EM takes. The corresponding imputation error is: ESL label 0.466(0.04), feature 0.649(0.02); LEV label 0.849(0.03), feature 0.936(0.01); GBSG ordinal 0.992(0.03), continuous 0.953(0.02); TIPS ordinal 0.984(0.06), continuous 0.768(0.05).

B.2 Datasets Description for Section 8.5

ESL This dataset contains profiles of applicants for certain jobs. The recruiting company, based upon psychometric test results and interviews with the candidates, determined the values of the input attributes. The output is an overall score corresponding to the degree of fitness of the candidate.

LEV This dataset contains lecturer evaluations. Students evaluate their lecturers according to four attributes such as oral skills and contribution to their professional/general knowledge. The output is an overall score of the lecturerâĂŹs performance.

GBSG This dataset contains the information of women with breast cancer concerning the status of the tumours and the hormonal system of the patient.

TIPS This dataset concerns the tips given to a waiter in a restaurant collected from customers. Recording variables contains the price of the meal, the tip amount and the conditions of the restaurant meal(number of guests, time of data, etc.).

C PROOF OF LEMMAS

C.1 Proof of Lemma 1

For any $j \in [p]$, $x_j \overset{d}{=} f_j(z_j)$ if and only if (iff) x_j and $f_j(z_j)$ have the same CDF. For each $j \in [p]$, since f_j^{-1} exists for any strictly monotone function f_j, we can calculate the CDF of $f_j(z_j)$:

$$F_{f_j(z_j)}(t) = \mathrm{P}(f_j(z_j) \leq t) = \mathrm{P}(z_j \leq f_j^{-1}(t)) = \Phi(f_j^{-1}(t)).$$

Then $x_j \overset{d}{=} f_j(z_j)$ iff $\Phi \circ f_j^{-1} = F_j$, equivalently, $f_j = F_j^{-1} \circ \Phi$.

C.2 Proof of Lemma 2

It suffices to show for monotone function f, $x \overset{d}{=} f(z)$ iff $f(z) = \text{cutoff}(z; \mathrm{S})$ with $\mathrm{S} = \{s_l = F_z^{-1}\left(\sum_{t=1}^{l} p_t\right) : l \in [k-1]\}$. Notice $x \overset{d}{=} f(z)$ iff the range of $f(z)$ is $[k]$ and $p_l = \mathrm{P}(f(z) = l)$ for any $l \in [k]$. When $f(z) = \text{cutoff}(z; \mathrm{S})$, further define $s_k = \infty$ and $s_0 = -\infty$. Since z is continuous with CDF F_z, it suffices to show:

$$\mathrm{P}(f(z) = l) = \mathrm{P}(s_{l-1} < z \leq s_l) = F_z(s_l) - F_z(s_{l-1}) = p_l, \text{ for } l \in [k]$$

When $x \overset{d}{=} f(z)$, $f(z)$ has range $[k]$. For $l \in [k]$, define $A_l = \{z : f(z) = l\}, s_l = \sup_{z \in A_l} z$ and $s_0 = \inf_{z \in A_1} z$. Since $\mathrm{P}(f(z) = l) = p_l > 0$, we have $\inf_{z \in A_l} z < s_l$. Since f is monotone, we have $s_{l-1} \leq \inf_{z \in A_l} z$. Claim $s_{l-1} = \inf_{z \in A_l} z$. If not, there exists $s_{l-1} < z^* < \inf_{z \in A_l} z$ satisfying $(l-1) \leq f(z^*) \leq l$. Since $f(z)$ has range $[k]$, $f(z^*)$ can only be l or $l-1$. Equivalently $z^* \in A_l$ or $z^* \in A_{l-1}$, which contradicts $s_{l-1} < z^* < \inf_{z \in A_l} z$. Thus $s_{l-1} = \inf_{z \in A_l} z$, $f(z) = 1 + \sum_{l=1}^{k-1} \mathbb{1}(z > s_l)$,

$$p_l = \mathrm{P}(f(z) = l) = \mathrm{P}(z \in A_l) = \mathrm{P}(s_{l-1} \leq z \leq s_l) = F_z(s_l) - F_z(s_{l-1}),$$

Thus we have $F_z(s_l) = \sum_{t=1}^{l} p_t \Rightarrow s_l = F_z^{-1}(\sum_{t=1}^{l} p_t)$.

C.3 Proof of Lemma 3

Before we prove Lemma 3, we introduce the Dvoretzky-Kiefer-Wolfowitz inequality proposed in [7], also introduced in [18].

The Dvoretzky-Kiefer-Wolfowitz Inequality. *For any i.i.d. sample $x^1, \ldots, x^n$ with distribution F, then when $\epsilon > 0$,*

$$\mathrm{P}\left(\sup_{t \in \mathbb{R}} |\mathbb{F}_n(t) - F(t)| \geq \epsilon\right) \leq 2e^{-2n\epsilon^2}, \text{ where } \mathbb{F}_n(t) = \frac{\sum_{i=1}^{n} 1\{x^i \leq t\}}{n}$$

Proof of Lemma 3: Applying the Dvoretzky-Kiefer-Wolfowitz inequality, for any $\epsilon > 0$ with probability at least $1 - 2e^{-2n\epsilon^2}$, $\sup_{t \in \mathbb{R}} |\mathbb{F}_n(t) - F(t)| < \epsilon$.

Take $\epsilon > n^{-1}$, $\sup_{t \in \mathbb{R}} \left|\frac{n}{n+1}\mathbb{F}_n(t) - F(t)\right| \leq \frac{1}{n+1} + \epsilon < 2\epsilon$. Thus for any $t \in \mathbb{R}$, we have $F(t) - 2\epsilon < \frac{n}{n+1}\mathbb{F}_n(t) < F(t) + 2\epsilon$. When $t \in [m, M]$, we have $F(t) \in [F(m), F(M)]$. Further let $\epsilon < K_1 \triangleq \min\{\frac{F(m)}{4}, \frac{1-F(M)}{4}\}$, we have $\frac{n}{n+1}\mathbb{F}_n(t) \in [\frac{F(m)}{2}, \frac{1+F(M)}{2}]$,

$$\begin{aligned} \sup_{t \in [m,M]} \left|\hat{f}^{-1}(t) - f^{-1}(t)\right| &= \sup_{t \in [m,M]} \left|\Phi^{-1}\left(\frac{n}{n+1}\mathbb{F}_n(t)\right) - \Phi^{-1}(F(t))\right| \\ &\leq \sup_{r \in [\frac{F(m)}{2}, \frac{1+F(M)}{2}]} \left|\left(\Phi^{-1}(r)\right)'\right| \cdot \sup_{t \in [m,M]} \left|\frac{n}{n+1}\mathbb{F}_n(t) - F(t)\right| \\ &< 2\epsilon \cdot \sup_{r \in [\frac{F(m)}{2}, \frac{1+F(M)}{2}]} \left|\left(\Phi^{-1}(r)\right)'\right| \end{aligned}$$

Since $\left(\Phi^{-1}(r)\right)' = \frac{1}{\phi(\Phi^{-1}(r))}$, we get $\sup_{r \in [\frac{F(m)}{2}, \frac{1+F(M)}{2}]} \left|(\Phi^{-1}(r))'\right| = K_2 \triangleq 1/\min\left\{\phi\left(\Phi^{-1}(\frac{F(m)}{2})\right), \phi\left(\Phi^{-1}(\frac{F(M)+1}{2})\right)\right\}$. Adjusting the constants, for $2K_2 n^{-1} < \epsilon < 2K_1K_2$, we have

$$\mathrm{P}\left(\sup_{t \in [m,M]} \left|\hat{f}^{-1}(t) - f^{-1}(t)\right| > \epsilon\right) \leq 2\exp\left\{-\frac{n\epsilon^2}{2K_2^2}\right\}.$$

C.4 Proof of Lemma 4

Before we prove Lemma 4, we introduce the Bretagnolle-Huber-Carol inequality introduced in [34].

The Bretagnolle-Huber-Carol Inequality. *If the random vector $(N_1, \ldots, N_k)$ is multinomially distributed with parameters n and $(p_1, \ldots, p_k)$, then*

$$\mathrm{P}\left(\sum_{i=1}^{k} |N_i/n - p_i| \geq \epsilon\right) \leq 2^k e^{-\frac{1}{2}n\epsilon^2}, \qquad \epsilon > 0.$$

Proof of Lemma 4: According to Lemma 2, the cutoff function $f(z) = \text{cutoff}(z; \mathrm{S})$ is unique and $\mathrm{S} = \{s_l : s_l = \Phi^{-1}(\sum_{t=1}^{l} p_t), l \in [k-1]\}$. Define $s_l^* = \Phi^{-1}\left(\frac{\sum_{i=1}^{n} \mathbb{1}(x^i \leq l)}{n}\right)$ for $l \in [k-1]$, $s_0^* = -\infty, s_k^* = \infty$, and $\Delta_l^* = \Phi(s_l^*) - \Phi(s_{l-1}^*) = \sum_{i=1}^{n} \mathbb{1}(x^i = l)/n$. Notice $(n\Delta_1^*, \ldots, n\Delta_k^*)$ is multinomially distributed with parameters n and $(p_1, \ldots, p_k)$, applying the Bretagnolle-Huber-Carol inequality, for any $\epsilon > 0$, with probability at least $1 - 2^k e^{-\frac{1}{2}n\epsilon^2}$, $\sum_{l=1}^{k} |\Delta_l^* - p_l| < \epsilon$. First for each $l \in [k]$, $|\Phi(s_l^*) - \Phi(s_l)| \leq \sum_{t=1}^{k} |\Delta_t^* - p_t| < \epsilon$. Take $\epsilon > n^{-1}$, we have

$$\left|\Phi(s_l^*) \cdot \frac{n}{n+1} - \Phi(s_l)\right| \leq |\Phi(s_l^*) - \Phi(s_l)| + \frac{\Phi(s_l^*)}{n+1} < 2\epsilon$$

$$\Phi(s_l) - 2\epsilon < \Phi(s_l^*) \cdot \frac{n}{n+1} = \frac{\sum_{i=1}^{n} \mathbb{1}(x^i \leq l)}{n+1} < \Phi(s_l) + 2\epsilon$$

When $l \in [k-1]$, we have $p_1 \leq \Phi(s_l) \leq \sum_{t=1}^{k-1} p_t$. Further let $\epsilon < K_1 \triangleq \min\{\frac{p_1}{4}, \frac{p_k}{4}\}$, we have $\frac{p_1}{2} \leq \Phi(s_l^*) \cdot \frac{n}{n+1} \leq 1 - \frac{p_k}{2}$. Thus:

$$\begin{aligned} ||\hat{\mathrm{S}} - \mathrm{S}||_1 = \sum_{l=1}^{k-1} |\hat{s}_l - s_l| &= \sum_{l=1}^{k-1} \left|\Phi^{-1}\left(\frac{\sum_{i=1}^{n} \mathbb{1}(x^i \leq l)}{n+1}\right) - \Phi^{-1}(\Phi(s_l))\right| \\ &\leq \sup_{r \in [\frac{p_1}{2}, 1-\frac{p_k}{2}]} \left|\left(\Phi^{-1}(r)\right)'\right| \cdot \sum_{l=1}^{k-1} \left|\frac{\sum_{i=1}^{n} \mathbb{1}(x^i \leq l)}{n+1} - \Phi(s_l)\right| \\ &\leq \frac{1}{\min\left\{\phi\left(\Phi^{-1}(\frac{p_1}{2})\right), \phi\left(\Phi^{-1}(1 - \frac{p_k}{2})\right)\right\}} \cdot 2(k-1)\epsilon \end{aligned}$$

Let $K_2 = 1/\min\left\{\phi\left(\Phi^{-1}(\frac{p_1}{2})\right), \phi\left(\Phi^{-1}(1 - \frac{p_k}{2})\right)\right\}$. Adjusting the constants, for $2(k-1)K_2 n^{-1} < \epsilon < 2(k-1)K_1K_2$, we have

$$\mathrm{P}\left(||\hat{\mathrm{S}} - \mathrm{S}||_1| > \epsilon\right) \leq 2\exp\left\{-\frac{1}{8K_2^2} \cdot \frac{n\epsilon^2}{(k-1)^2}\right\}.$$