Probabilistic Missing Value Imputation for Mixed Categorical and Ordered Data

Anonymous Author(s)
Affiliation
Address
email

Abstract

Many real-world datasets contain missing entries and mixed data types including categorical and ordered (e.g. continuous and ordinal) variables. Imputing the missing entries is necessary, since many data analysis pipelines require complete data, but challenging especially for mixed data. This paper proposes a probabilistic imputation method using an *extended Gaussian copula model* that supports both single and multiple imputation. The method models mixed categorical and ordered data using a latent Gaussian distribution. The unordered characteristics of categorical variables is explicitly modeled using the argmax operator. The method makes no assumptions on the data marginals nor does it require tuning any hyperparameters. Experimental results on synthetic and real datasets show that imputation with the extended Gaussian copula outperforms the current state-of-the-art for both categorical and ordered variables in mixed data.

1 Introduction

Modern datasets from healthcare, the sciences, and the social sciences often contain missing entries and mixed data types such as continuous, ordinal, and categorical. Social survey datasets, for example, are typically mixed because they include variables like age (continuous), demographic group (categorical), and Likert scales (ordinal) measuring how strongly a respondent agrees with certain stated opinions. Continuous variables are encoded as real numbers and sometimes called numeric. we refer to variables that admit a total order (e.g. continuous and ordinal) as *ordered* variables. In contrast, a categorical variable, also called nominal, can take one of a fixed number of *unordered values* such as “A", “B", “AB", or "O" for blood type.

Most data analysis techniques require a complete dataset, so missing data imputation is an essential preprocessing step. It is also often of interest to propagate imputation uncertainty into subsequent analyses through multiple imputation, which generates several potentially different imputed datasets [23]. An imputation method should ideally use all the collected data—regardless of data type—to impute any missing data entries. However, most imputation approaches, whether explicitly or implicitly, assume that each variable admits a total order and, as a result, cannot impute categorical variables without proper preprocessing.

There is no satisfying, successful, and widely adopted method for imputing categorical variables, especially in mixed datasets. It is tempting to reduce categorical imputation to ordinal imputation using an *integer encoding* in which each category is assigned a number; however, this encoding requires choosing an arbitrary ordering for the categories that may affect the results of downstream (e.g., predictive) models. (This problem is most severe for linear and neural network models, whereas tree-based models are less sensitive [28].) Instead, it is more common to use *one-hot encoding* to represent a categorical variable using a binary vector with one entry for each category. With this encoding, imputation methods developed for continuous or binary data, such as the popular low rank

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

matrix completion methods (LRMC) [22, 15, 26], can be jerry-rigged for categorical imputation. However, LRMC is only proper when the encoded matrix (both ordered and encoded categorical variables) is approximately low rank. It has been observed that LRMC performs poorly on long skinny (many samples n and few features p) datasets because the low rank assumption fails [8, 31].

Iterative imputation methods including MICE [4] and missForest [25] can directly operate on the unordered categories. MICE learns the conditional distribution of each variable (whether ordered or unordered) using all other variables via linear and logistic regression. However, the learned conditional models may be incompatible in the sense that the joint distribution cannot exist [4]. MissForest uses random forest to predict each variable and yields more accurate single imputations [25], but cannot provide multiple imputation. Both methods converge slowly (or sometimes diverge) for large datasets because they train many models on (possibly) ill-conditioned data.

This paper expands on the idea of modeling categorical data and multivariate data interaction using a latent continuous space [10, 5]. This approach can explicitly modeling the categorical distribution, which is critical for providing multiple imputation. This paper models the latent space as a Gaussian distribution and models each categorical variable as the argmax of a latent Gaussian vector. This choice is inspired by the Gaussian copula model [13, 7, 9], which yields high quality imputations for ordered data [31]. However, it cannot be used when the data contains categorical variables. This paper proposes the *extended Gaussian copula* to overcome this limitation by explicitly modeling each data type. In contrast, other imputation methods typically require bespoke preprocessing for the input data to perform well. For mixed categorical and ordered data, the extended Gaussian copula model generates each categorical variable using a latent Gaussian vector and each ordered variable using a latent Gaussian scalar. The latent Gaussian correlations capture the dependence structure.

Contributions. This paper makes three main contributions to the literature:

1. A probabilistic model, the *extended Gaussian copula*, for mixed data including continuous, ordinal, and categorical variables. The model is free of hyperparameters and makes no assumptions on the marginal distribution of each data type.

2. A *single imputation* method that empirically provides state-of-the-art accuracy for mixed data and a *multiple imputation* method that can quantify imputation uncertainty by measures such as the category probability for categorical variables and confidence intervals for continuous variables.

3. A robust and efficient parameter fitting algorithm for the extended Gaussian copula that supports further acceleration through parallelization, mini-batch training, and low rank structure.

Related work. Modeling a categorical variable as the argmax of a latent continuous vector is a classical technique used in the multinomial probit model [3, 20] in the context of supervised learning. Instead, our model is more similar to [5], which proposed one way to accommodate categorical variables in the Gaussian copula. However, the model in [5] has many redundant parameters and thus is unidentifiable, while the extended Gaussian copula is identifiable and can match any categorical marginal distribution.

2 Methodology

Notation. For $f : \mathbb{R}^K \to \mathbb{R}$, we define the (possibly set-valued) pre-image of f as $f^{-1}(y) := \{x : f(x) = y\}$. We define $\operatorname{argmax}(\mathbf{z})$ of a vector $\mathbf{z} = (z_1, \ldots, z_p)$ as the index of the maximum entry so that $\mathbf{z}_{\operatorname{argmax}(\mathbf{z})} = \max_{i=1,\ldots,p} z_i$. We use $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ to denote the multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance Σ. We use $\mathbf{0}$ to denote the all-zero vector and $\mathbf{1}$ for the all-one vector, where the context determines the length of the vector.

We introduce our model in Section 2.1, then derive the model-based imputation methods in Section 2.2 and finally present model estimation algorithms in Section 2.3. Throughout the paper, we assume the missing complete at random (MCAR) mechanism: missing values are uniform and independent of any data. Nevertheless, we show our method performs reasonably well under missing at random (MAR) and missing not at random (MNAR) assumptions through experiments.

2.1 Extended Gaussian copula with categorical variables

We first show how to model a categorical variable by transforming a latent Gaussian vector. Then we extend this model to generate categorical vectors and mixed categorical and ordered vectors. To ease the notation, we assume that all categorical variables have K categories encoded as $\{\text{“1”}, \ldots, \text{“}K\text{”}\}$. It is straightforward to allow categorical variables with different numbers of categories.

2.1.1 Univariate categorical variable

We model a univariate categorical variable x with K categories as the argmax of a K-dim latent Gaussian $\mathbf{z} = (z_1, \ldots, z_K)$ with some mean $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$ and identity covariance. That is,

$$x := \operatorname{argmax}(\mathbf{z} + \boldsymbol{\mu}), \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K), \tag{1}$$

We call the distribution in Eq. (1) the *Gaussian-Max* distribution. Without loss of generality, we assume $\mu_1 = 0$, as the argmax is invariant under translations, i.e., $\mu \leftarrow \mu + \alpha\mathbf{1}$ for $\alpha \in \mathbb{R}$. While a dense covariance matrix is sometimes used for $\mathbf{z}$ [5], we prefer the model of Eq. (1) as it is identifiable. Theorem 1 states that any categorical distribution corresponds to a unique choice of μ under Eq. (1). All proofs are in the supplement. In Section 2.3.1 we describe an algorithm to estimate $\boldsymbol{\mu}$ for a given categorical distribution.

Theorem 1 (Existence and Uniqueness)**.** *For any categorical distribution* $\mathbb{P}[x = \text{“}k\text{”}] = p_k > 0$ *for* $k = 1, \ldots, K$ *such that* $\sum_{k=1}^{K} p_k = 1$*, there is a unique* $\boldsymbol{\mu} \in \mathbb{R}^K$ *with* $\mu_1 = 0$ *such that*

$$\mathbb{P}_{\mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_K)}[\operatorname{argmax}(\mathbf{z} + \boldsymbol{\mu}) = k] = p_k, \quad k = 1, \ldots, K. \tag{2}$$

2.1.2 Multivariate categorical vector

We model a categorical vector $\mathbf{x} = (x_1, \ldots, x_p)$ by supposing each of its entries x_j follows the Gaussian-Max distribution. That is, $x_j = \operatorname{argmax}(\mathbf{z}^{(j)} + \boldsymbol{\mu}^{(j)})$ for some $\boldsymbol{\mu}^{(j)}$ and isotropic Gaussian $\mathbf{z}^{(j)}$. Additionally, the latent Gaussian variables $\mathbf{z}^{(j)}$ corresponding to different categorical variables may be correlated. We call this model the *categorical latent Gaussian (CLG)* model (see Definition 1). For $\mathbf{z} := (\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(p)})$, we use $[j]$ to denote the indices of $\mathbf{z}^{(j)}$ in $\mathbf{z}$, so $\mathbf{z}_{[j]} = \mathbf{z}^{(j)}$ for $j = 1, \ldots, p$.

Definition 1 (Categorical latent Gaussian)**.** *For a categorical vector* $\mathbf{x} = (x_1, \ldots, x_p)$*, we say* $\mathbf{x}$ *follows the categorical latent Gaussian* $\mathbf{x} \sim \mathrm{CLG}(\Sigma, \boldsymbol{\mu})$*, if there exists a correlation matrix* Σ *and* $\boldsymbol{\mu}$ *such that (1) for* $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \Sigma)$*,* $x_j = \operatorname{argmax}(\mathbf{z}_{[j]} + \boldsymbol{\mu}_{[j]})$*; (2)* $\Sigma_{[j],[j]} = \mathbf{I}_K$*, for every* $j = 1, \ldots, p$*.*

In the CLG model, the value of $\boldsymbol{\mu}_{[j]}$ suffices to determine the marginal distribution of each categorical x_j, as the marginal distribution of $\mathbf{z}_{[j]}$ is $\mathcal{N}(\mathbf{0}, \mathbf{I}_K)$ and independent of Σ. The correlation matrix Σ introduces dependencies between different categorical variables in $\mathbf{x}$.

Consider two categorical variables x_{j_1} and x_{j_2} whose joint distribution is described by $\mathbf{P} \in \mathbb{R}^{K\times K}$, where $p_{kl} = \mathbb{P}[x_{j_1} = k, x_{j_2} = l]$. The CLG model captures the dependence between x_{j_1} and x_{j_2} by the correlation submatrix $\Sigma_{[j_1],[j_2]}$. Note that $\mathbf{P}$ has only $(K-1)^2$ free parameters as the rows and columns must sum to the associated marginal probabilities. This means that $\Sigma_{[j_1],[j_2]}$ in the CLG model is not identifiable. We develop an identifiable variant of the CLG model in the supplement that uses only $(K-1)^2$ parameters in $\Sigma_{[j_1],[j_2]}$. Both models share similar imputation performance, but the identifiable model is not invariant under permutation of the categorical labels while the model in Definition 1 is invariant. For the rest of this paper, we use the permutation-invariant CLG model.

2.1.3 Mixed categorical and ordered vector

We model mixed data that contains both categorical and ordered variables by combining the CLG model for categorical variables with the Gaussian copula model for ordered variables [9, 31].

To model $\mathbf{x}$ with ordered variables, the Gaussian copula assumes $\mathbf{x} \sim \mathrm{GC}(\Sigma, \mathbf{f})$ is generated as an elementwise transformed Gaussian, i.e., $\mathbf{x} = \mathbf{f}(\mathbf{z}) = (f_1(z_1), \ldots, f_p(z_p))$, where each f_j is a monotonic increasing function and $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \Sigma)$. Denoting the cumulative distribution function (CDF) of x_j as F_j, the transformation f_j is uniquely determined as $f_j = F_j^{-1} \circ \Phi$, where Φ denotes the standard Gaussian CDF and $F_j^{-1}(y) := \inf\{x \in \mathbb{R} : F_j(x) \geq y\}$. Specifically, ordinals result from thresholding the latent Gaussian variable. For an ordinal x_j, the transformation $f_j(z)$ has the

form $\sum_{s \in S} \mathbb{1}(z > s)$ where $\mathbb{1}(z > s)$ is 1 if $z > s$ and 0 otherwise. Here, the set S is determined by F_j. The CLG model has the same form as $\mathbf{x} = \mathbf{f}(\mathbf{z})$ by writing

$$x_j = f_j(\mathbf{z}_{[j]}; \boldsymbol{\mu}_{[j]}) := \operatorname{argmax}(\mathbf{z}_{[j]} + \boldsymbol{\mu}_{[j]}). \tag{3}$$

Hence, we propose the *extended Gaussian copula (EGC)* model.

Definition 2 (Extended Gaussian copula)**.** *Write a mixed data vector as* $\mathbf{x} = (\mathbf{x}_{\text{cat}}, \mathbf{x}_{\text{ord}})$ *where* $\mathbf{x}_{\text{cat}}$ *collects all categorical variables and* $\mathbf{x}_{\text{ord}}$ *collects all ordered variables. We say* $\mathbf{x}$ *follows the extended Gaussian copula* $\mathbf{x} \sim \text{EGC}(\Sigma, \mathbf{f}^{\text{ord}}, \boldsymbol{\mu})$ *if there exists a correlation matrix* Σ*, an elementwise monotone* $\mathbf{f}^{\text{ord}}$*, and a* $\boldsymbol{\mu}$ *such that* $\mathbf{x}_{\text{cat}} \sim \text{CLG}(\Sigma_{\text{cat, cat}}, \boldsymbol{\mu})$ *and* $\mathbf{x}_{\text{ord}} \sim \text{GC}(\Sigma_{\text{ord,ord}}, \mathbf{f}^{\text{ord}})$*, where* $\Sigma = \begin{bmatrix} \Sigma_{\text{cat, cat}} & \Sigma_{\text{cat, ord}} \\ \Sigma_{\text{ord, cat}} & \Sigma_{\text{ord,ord}} \end{bmatrix}$.

We can also write $\mathbf{x} = \mathbf{f}(\mathbf{z}) = (\mathbf{f}^{\text{cat}}(\mathbf{z}_{\text{cat}}; \boldsymbol{\mu}), \mathbf{f}^{\text{ord}}(\mathbf{z}_{\text{ord}}))$ for $\mathbf{z} = (\mathbf{z}_{\text{cat}}, \mathbf{z}_{\text{ord}}) \sim \mathcal{N}(\mathbf{0}, \Sigma)$ where $\mathbf{f}$ is defined by all transformation parameters, i.e., $\mathbf{f}^{\text{ord}}$ and $\boldsymbol{\mu}$. Unlike the original Gaussian copula, the dimension of the latent $\mathbf{z}$ does not match that of the data $\mathbf{x}$; instead, it is larger as subvectors of $\mathbf{z}$ correspond to categorical entries of $\mathbf{x}$. The correlation matrix Σ must also obey constraints on the submatrix $\Sigma_{\text{cat, cat}}$ according to Definition 1.

2.2 Missing data imputation

Now we show how to impute missing values given an EGC model with known parameters. For model parameter estimation, see Section 2.3.

Note each categorical x_j corresponds to K consecutive entries in $\mathbf{z}$: the subvector $\mathbf{z}_{[j]}$ that generates x_j. We use $[I]$ to denote the set of indices in $\mathbf{z}$ that corresponds to a set of indices I from $\mathbf{x}$. For example, $[j]$ follows this notation with $I = \{j\}$. Two sets of indices are particularly important: the observed entries $\mathcal{O}$ and missing entries $\mathcal{M}$ for a given mixed data vector $\mathbf{x}$. Thus $[\mathcal{O}]$ and $[\mathcal{M}]$ are the latent dimensions in $\mathbf{z}$ that generate $\mathbf{x}_{\mathcal{O}}$ and $\mathbf{x}_{\mathcal{M}}$, respectively.

Our imputation strategy follows [31]. First, the algorithm identifies a region of the latent space $\mathbf{z}_{[\mathcal{O}]}$ constrained by the observed entries. Then, it imputes the latent dimensions by exploiting the Gaussian distribution of $\mathbf{z}_{[\mathcal{M}]}$ conditional on $\mathbf{z}_{[\mathcal{O}]}$. Finally, the algorithm uses the marginal transformation $\mathbf{f}$ to produce imputations in the data space $\mathbf{x}_{\mathcal{M}}$. Pictorially, we can visualize this process as

$$\mathbf{x}_{\mathcal{O}} \xrightarrow{\mathbf{f}_{\mathcal{O}}^{-1}} \mathbf{z}_{[\mathcal{O}]} \xrightarrow{\Sigma} \mathbf{z}_{[\mathcal{M}]} \xrightarrow{\mathbf{f}_{\mathcal{M}}} \mathbf{x}_{\mathcal{M}}.$$

Multiple imputation. Multiple imputation creates several imputed datasets by sampling from the distribution of missing entries conditional on the observations. For the EGC model, Algorithm 1 samples from the distribution of $\mathbf{x}_{\mathcal{M}}$ using two facts: (1) given $\mathbf{x}_{\mathcal{O}}$, the variable $\mathbf{z}_{[\mathcal{O}]}$ follows a truncated Gaussian distribution truncated to $\mathbf{f}_{\mathcal{O}}^{-1}(\mathbf{x}_{\mathcal{O}}) = \prod_{j \in \mathcal{O}} f_j^{-1}(x_j)$; and (2) the random variable $\mathbf{z}_{[\mathcal{M}]} | \mathbf{z}_{[\mathcal{O}]}$ is Gaussian distributed. Section 2.3.3 shows how to sample from the truncated Gaussian random variable given by $\mathbf{z}_{[\mathcal{O}]} | \mathbf{x}_{\mathcal{O}}$. Using the empirical distribution of drawn samples, we can estimate the category probability for a missing categorical variable and build confidence intervals for a missing continuous variable.

Algorithm 1 Multiple imputation via the extended Gaussian Copula

1: **Input:** # of imputations m, data vector $\mathbf{x}$ observed at $\mathcal{O}$, model parameters $\mathbf{f}$ and Σ.
2: **for** $s = 1, 2, \ldots, m$ **do**
3: Sample $\hat{\mathbf{z}}_{[\mathcal{O}]}^{(s)} \sim \mathbf{z}_{[\mathcal{O}]} | \mathbf{x}_{\mathcal{O}} : \quad \mathcal{N}(\mathbf{0}, \hat{\Sigma}_{[\mathcal{O}],[\mathcal{O}]})$ truncated to $\mathbf{f}_{\mathcal{O}}^{-1}(\mathbf{x}_{\mathcal{O}})$
4: Sample $\hat{\mathbf{z}}_{[\mathcal{M}]}^{(s)} \sim \mathbf{z}_{[\mathcal{M}]} | \mathbf{z}_{[\mathcal{O}]} : \quad \mathcal{N}\left(\Sigma_{[\mathcal{M}],[\mathcal{O}]} \hat{\mathbf{z}}_{[\mathcal{O}]}^{(s)}, \Sigma_{[\mathcal{M}],[\mathcal{M}]} - \Sigma_{[\mathcal{M}],[\mathcal{O}]} \Sigma_{[\mathcal{O}],[\mathcal{O}]}^{-1} \Sigma_{[\mathcal{O}],[\mathcal{M}]}\right)$
5: Compute $\hat{\mathbf{x}}_{\mathcal{M}}^{(s)} = \mathbf{f}_{\mathcal{M}}(\hat{\mathbf{z}}_{[\mathcal{M}]}^{(s)})$
6: **end for**
7: **Output:** $\{\hat{\mathbf{x}}_{\mathcal{M}}^{(s)} | s = 1, .., m\}$.

Single imputation. To provide an accurate single imputation under the EGC model, we impute based on the conditional mean of the latent Gaussian, i.e.,

$$\hat{\mathbf{x}}_{\mathcal{M}} = \mathbf{f}_{\mathcal{M}}(\mathbb{E}[\mathbf{z}_{[\mathcal{M}]}|\mathbf{x}_{\mathcal{O}}, \Sigma, \mathbf{f}]) = \mathbf{f}_{\mathcal{M}}(\Sigma_{[\mathcal{M}],[\mathcal{O}]}\Sigma^{-1}_{[\mathcal{O}],[\mathcal{O}]}\mathbb{E}[\mathbf{z}_{[\mathcal{O}]}|\mathbf{x}_{\mathcal{O}}, \Sigma, \mathbf{f}]). \tag{4}$$

To compute $\mathbb{E}[\mathbf{z}_{[\mathcal{O}]}|\mathbf{x}_{\mathcal{O}}, \Sigma, \mathbf{f}]$, see Section 2.3.3. With a known mean $\mathbb{E}[\mathbf{z}_{[\mathcal{O}]}|\mathbf{x}_{\mathcal{O}}, \Sigma, \mathbf{f}]$ and marginal distribution $\mathbf{f}_{\mathcal{M}}$, computing the imputation $\hat{\mathbf{x}}_{\mathcal{M}}$ is straightforward. Specifically, for categorical missing x_j, we first compute the K-dim $\hat{\mathbf{z}}_{[j]} := \mathbb{E}[\mathbf{z}_{[j]}|\mathbf{x}_{\mathcal{O}}, \Sigma, \mathbf{f}]$ and then impute as $\hat{x}_j = \mathrm{argmax}(\hat{\mathbf{z}}_{[j]} + \boldsymbol{\mu}_{[j]})$.

Online imputation. Online imputation is the task to impute missing entries in data streams arriving at different times. We can conduct online imputation using the EGC model following [29]: upon observing an incomplete data batch, impute the missing entries with a current saved model, and then update the model using a new observation. See the supplement on how to update the model.

2.3 Parameter estimation

Parameter estimation for the EGC model consists of two steps: (1) we form an estimate $\hat{\mathbf{f}}$ of the marginal transformation $\mathbf{f}$, and (2) we estimate the copula correlation Σ given $\hat{\mathbf{f}}$. The marginal transformation differs for ordered and categorical variables. For an ordered x_j, since $f_j = F_j^{-1} \circ \Phi$, we can simply estimate F_j^{-1} using the empirical quantile function on the observed entries of x_j as in [18, 31]. For a categorical variable x_j, we show how to estimate $f_j(\cdot; \boldsymbol{\mu}_{[j]})$ in Section 2.3.1 and how to estimate Σ in Eq. (3) by extending estimation algorithms from [31].

2.3.1 Marginal estimation for categorical variables

Here we estimate the mean $\boldsymbol{\mu}$ for which the EGC model, $\mathrm{EGC}(\Sigma, \mathbf{f}^{\mathrm{ord}}, \boldsymbol{\mu})$, matches the observed categorical marginal distribution. Each subvector $\boldsymbol{\mu}_{[j]}$ corresponding to categorical x_j may be estimated independently of all the others, so we drop the variable subscript j to ease the notation.

We must find $\boldsymbol{\mu}$ in Eq. (2) so that the resulting category frequency p_k matches the observed frequency, subject to $\mu_1 = 0$: this setup gives $K-1$ nonlinear equations with $K-1$ unknowns. Unfortunately, there is no closed form solution. Thus we resort to iterative root finding algorithms, which use as oracles the value and derivative of $\mathbb{P}[\mathrm{argmax}(\mathbf{z} + \boldsymbol{\mu}) = k]$ in Eq. (2). We use two approximation techniques to evaluate these oracles efficiently.

First we approximate the probability of argmax as the expection of softmax by noticing

$$\mathbb{P}_{\mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_K)}[\mathrm{argmax}(\mathbf{z} + \boldsymbol{\mu}) = k] = \lim_{\beta\to\infty} \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_K)} \left[\mathrm{softmax}(\beta(z_k + \mu_k); \beta(\mathbf{z} + \boldsymbol{\mu}))\right],$$

where $\mathrm{softmax}(z_k; \mathbf{z}) = \exp(z_k)/\sum_{l=1}^{K}\exp(z_l)$. This approximation is accurate for large β. Second we approximate the expectation using Monte Carlo samples with the reparameterization trick [17]:

$$\mathbb{E}\left[\mathrm{softmax}(\beta(z_k + \mu_k); \beta(\mathbf{z} + \boldsymbol{\mu}))\right] \approx \frac{1}{M}\sum_{i=1}^{M}\mathrm{softmax}(\beta(\bar{z}_k^i + \mu_k); \beta(\bar{\mathbf{z}}^i + \boldsymbol{\mu})), \tag{5}$$

where $\bar{\mathbf{z}}^i \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$. With these approximations, we solve for $\boldsymbol{\mu}$ subject to $\mu_1 = 0$ in Eq. (6),

$$\frac{1}{M}\sum_{i=1}^{M}\mathrm{softmax}(\beta(\bar{z}_k^i + \mu_k); \beta(\bar{\mathbf{z}}^i + \boldsymbol{\mu})) = p_k, \tag{6}$$

for $k = 2, \ldots, K$. The modified Powell method implemented in `SciPy` computes an accurate solution. These approximations are similar to ones used for the Gumbel-softmax distribution [14], but the Gaussianity used here better models a mixed distribution; see supplement for further comparison.

2.3.2 Copula correlation estimation

Now we show how to find the maximum likelihood estimator (MLE) of copula correlation Σ, which models the multivariate dependence structure. Given i.i.d. samples $\mathbf{x}^i \overset{\mathrm{iid}}{\sim} \mathrm{EGC}(\Sigma, \mathbf{f}^{\mathrm{ord}}, \boldsymbol{\mu})$ observed

at $\mathcal{O}_i$ and missing at $\mathcal{M}_i$ for $i = 1, \ldots, n$. We maximize the observed likelihood by integrating over the latent space that maps to the observation $\mathbf{x}^i_{\mathcal{O}_i}$, as in Eq. (7):

$$\ell_{\text{obs}}(\Sigma; \mathbf{x}^i_{\mathcal{O}_i}) = \frac{1}{n} \sum_{i=1}^{n} \int_{\mathbf{z}^i_{[\mathcal{O}_i]} \in \mathbf{f}_{\mathcal{O}_i}^{-1}(\mathbf{x}^i_{\mathcal{O}_i})} \phi(\mathbf{z}^i_{[\mathcal{O}_i]}; \mathbf{0}, \Sigma_{\mathcal{O}_i,\mathcal{O}_i})\, d\mathbf{z}^i_{[\mathcal{O}_i]}. \tag{7}$$

The form of this likelihood matches that of the Gaussian copula for ordered variables [31] except that the latent integration is in dimensions $[\mathcal{O}_i]$ instead of $\mathcal{O}_i$. Hence the Expectation Maximization (EM) algorithm in [31] also works in our case by integrating over the appropriate latent indices. Concretely, for the E-step at iteration $t+1$, we first compute the expectation of the joint likelihood of $(\mathbf{z}^i, \mathbf{x}^i_{\mathcal{O}_i})$ over the distribution of $\mathbf{z}^i$ conditional on the observation $\mathbf{x}^i_{\mathcal{O}_i}$ and the previous correlation estimate $\Sigma^{(t)}$ using the Gaussianity of $\mathbf{z}^i$, denoted as $Q(\Sigma; \Sigma^{(t)})$:

$$Q(\Sigma; \Sigma^{(t)}) = c - \frac{1}{2} \left(\log\det(\Sigma) + \text{tr} \left(\Sigma^{-1} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[\mathbf{z}^i (\mathbf{z}^i)^\top | \mathbf{x}^i_{\mathcal{O}_i}, \Sigma^{(t)}] \right) \right), \tag{8}$$

where c is a constant. Then we compute $\operatorname{argmax}_\Sigma Q(\Sigma; \Sigma^{(t)})$ in the M-step, which is the expected "empirical covariance matrix" of $\mathbf{z}^i$. We update $\Sigma^{(t+1)}$ to its corresponding correlation matrix to satisfy the copula parameter constraint similar to [11, 31]:

$$\Sigma^{(t+1)} \leftarrow P_{\text{cor}} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[\mathbf{z}^i (\mathbf{z}^i)^\top | \mathbf{x}^i_{\mathcal{O}_i}, \Sigma^{(t)}] \right), \tag{9}$$

where $P_{\text{cor}}(\Sigma)$ returns the correlation matrix corresponding to a covariance matrix Σ. The EM algorithm is guaranteed to strictly increase the likelihood in Eq. (7) and converges to a stationary point [21, Chapter 3]. The main computation is the expected sample covariance of $\mathbf{z}^i$. Dropping the row index, evaluating $\mathbb{E}[\mathbf{z}\mathbf{z}^\top | \mathbf{x}_{\mathcal{O}}, \Sigma^{(t)}]$ requires computing $\mathbb{E}[\mathbf{z}_{[\mathcal{O}]} | \mathbf{x}_{\mathcal{O}}, \Sigma^{(t)}]$ and $\text{Cov}[\mathbf{z}_{[\mathcal{O}]} | \mathbf{x}_{\mathcal{O}}, \Sigma^{(t)}]$. We give details in the supplement and show how to evaluate the reduced quantities in Section 2.3.3.

Recall the correlation Σ must satisfy the constraint in Definition 1: the submatrix $\Sigma_{[j],[j]}$ for each categorical index j must be the identity. The correlation computed by EM generally does not satisfy this constraint. Instead, we use the function $P_{\text{cat}}(\Sigma)$ to correct the correlation Σ:

$$\Sigma \leftarrow P_{\text{cat}}(\Sigma) = A \Sigma A^\top \text{ where } A = \text{diag}(\Sigma^{-1/2}_{[1],[1]}, \ldots, \Sigma^{-1/2}_{[p_{\text{cat}}],[p_{\text{cat}}]}, \mathbf{I}_{p_{\text{ord}}}), \tag{10}$$

by recognizing that $\text{Cov}[\Sigma^{-1/2}_{[j],[j]} \mathbf{z}_{[j]}] = \mathbf{I}_K$ for each categorical index j, when $\mathbf{x}_{\text{cat}}$ has p_{cat} entries and $\mathbf{x}_{\text{ord}}$ has p_{ord} entries.

Algorithm 2 Estimation algorithm for the extended Gaussian Copula

Input: Partial observation $\{\mathbf{x}^i_{\mathcal{O}_i}\}_{i=1}^n$, correlation initialization $\Sigma^{(0)}$, $M = 5000, \beta = 1000$.
for each variable index j **do** ▷ Marginal estimation starts
  Collect all observed data: $\mathbf{X}_j = \{x^i_j : j \in \mathcal{O}_i, i = 1, \ldots, n\}$
  if x_j is categorical **then**
    Solve Eq. (6) for $\hat{\boldsymbol{\mu}}$ with input M, β and p_k equals to the frequency of "k" in $\mathbf{X}_j$.
    Compute $\hat{f}_j(\cdot; \boldsymbol{\mu}_{[j]})$ in Eq. (3) with $\boldsymbol{\mu}_{[j]}$ as the solved solution $\hat{\boldsymbol{\mu}}$.
  else
    Compute $\hat{f}_j = \hat{F}_j^{-1} \circ \Phi$ where $\hat{F}_j^{-1}$ is the empirical quantile function of $\mathbf{X}_j$.
  end if
end for ▷ Marginal estimation ends
for $t = 0, 1, 2, \ldots$ until convergence **do** ▷ EM algorithm starts
  Compute $\mathbb{E}[\mathbf{z}^i (\mathbf{z}^i)^\top | \mathbf{x}^i_{\mathcal{O}_i}, \Sigma^{(t)}, \hat{\mathbf{f}}]$ for $i = 1, \ldots, n$.
  Update $\Sigma^{(t+1)}$ as in Eq. (9).
  Correct $\Sigma^{(t+1)} \leftarrow P_{\text{cat}}(\Sigma^{(t+1)})$ as in Eq. (10).
end for ▷ EM algorithm ends
Output: marginal estimation $\hat{\mathbf{f}}$, correlation estimation $\hat{\Sigma} = \Sigma^{(t+1)}$.

Computational cost. Algorithm 2 summarizes the complete estimation algorithm for the EGC model. Marginal estimation only takes a small fraction of total runtime (less than 4% in all experiments of this paper). We observe that the EM algorithm mostly converges in less than 50 iterations. Each iteration has time complexity $O(\alpha n d^3)$ where α denotes the observed entry ratio, n denotes the number of samples and $d = p_{\text{ord}} + p_{\text{cat}} K$ denotes the latent dimension. Line 12 of Algorithm 2 takes the vast majority of the computation time. There are many ways to accelerate the computation. For large n, we can parallelize the computation in line 12 of Algorithm 2 or conduct minibatch EM training as in [29]. The minibatch EM performs model updates using randomly sampled data batch and achieves significant speedup without sacrificing accuracy. For large d, we can reduce the cubic time complexity in terms of d to linear by imposing a low rank structure on Σ following [30], that is $\Sigma = WW^\top + \sigma^2 \mathbf{I}_d$ where $W \in \mathbb{R}^{d \times k}$ with $k \ll d$ and $\sigma^2 \in (0, 1)$. See the supplement for details.

2.3.3 Truncated Gaussian with categorical variables

Finally, we show how to estimate moments and sample from $\mathbf{z}_{[\mathcal{O}]}|\mathbf{x}_{\mathcal{O}}$, as these operations are required for imputation and correlation estimation. Note $\mathbf{z}_{[\mathcal{O}]}|\mathbf{x}_{\mathcal{O}}$ follows a truncated Gaussian distribution truncated into the region $\mathbf{f}_{\mathcal{O}}^{-1}(\mathbf{x}_{\mathcal{O}})$. For each ordered variable with index $j \in \mathcal{O}$, $f_j^{-1}(x_j)$ is an interval. (For continuous x_j, this interval is a single point.) Thus without observed categoricals, $\mathbf{f}_{\mathcal{O}}^{-1}(\mathbf{x}_{\mathcal{O}})$ is a Cartesian product of intervals. We call this distribution the *interval-truncated Gaussian*, for which accurate sampling [2] and effective moment estimation [31] methods have been developed.

The truncation region is no longer a Cartesian product of intervals with observed categoricals. Fortunately, for any $\mathbf{x}_{\mathcal{O}}$ we can find an involution $A_{\mathbf{x}}$ (i.e., $A_{\mathbf{x}}^2 = \mathbf{I}$) so that $A_{\mathbf{x}}\mathbf{z}_{[\mathcal{O}]}|\mathbf{x}_{\mathcal{O}}$ follows an interval-truncated Gaussian distribution. Consequently,

$$\mathbb{E}[\mathbf{z}_{[\mathcal{O}]}|\mathbf{x}_{\mathcal{O}}] = A_{\mathbf{x}}\mathbb{E}[A_{\mathbf{x}}\mathbf{z}_{[\mathcal{O}]}|\mathbf{x}_{\mathcal{O}}], \quad \text{Cov}[\mathbf{z}_{[\mathcal{O}]}|\mathbf{x}_{\mathcal{O}}] = A_{\mathbf{x}}\text{Cov}[A_{\mathbf{x}}\mathbf{z}_{[\mathcal{O}]}|\mathbf{x}_{\mathcal{O}}]A_{\mathbf{x}}^\top. \tag{11}$$

Similarly, to sample $\mathbf{z}_i \overset{\text{iid}}{\sim} \mathbf{z}_{[\mathcal{O}]}|\mathbf{x}_{\mathcal{O}}$, we can first sample $\tilde{\mathbf{z}}_i \overset{\text{iid}}{\sim} A_{\mathbf{x}}\mathbf{z}_{[\mathcal{O}]}|\mathbf{x}_{\mathcal{O}}$ and then use $\mathbf{z}_i = A_x\tilde{\mathbf{z}}_i$. To derive $A_{\mathbf{x}}$, first consider that when a categorical variable x takes value k, by Eq. (3) we have:

$$f^{-1}(x; \boldsymbol{\mu}) = \{\mathbf{z} \in \mathbb{R}^K : \text{argmax}(\mathbf{z} + \boldsymbol{\mu}) = k\} = \{\mathbf{z} \in \mathbb{R}^K : z_j + \mu_j < z_k + \mu_k, \text{ for } j \neq k\}, \tag{12}$$

Define $\tilde{\mathbf{z}} = A_k\mathbf{z}$ and $\tilde{\boldsymbol{\mu}} = A_k\boldsymbol{\mu}$ where $A_k = -\mathbf{I}_K + \sum_{i=1}^K E_{ik} + E_{kk}$ (matrix E_{ij} has 1 at (i, j)-th entry and 0 elsewhere). We can rewrite Eq. (12) to $\{\tilde{\mathbf{z}} : \tilde{z}_j + \tilde{\mu}_j > 0, \text{ for } j \neq k\}$, a Cartesian product of intervals. We can find a matrix A_j as shown above for every categorical variable x_j. Let $A_{\mathbf{x}}$ be $\text{diag}(A_1, \ldots, A_{p_{\text{cat}}}, \mathbf{I}_{p_{\text{ord}}})$. Then $A\mathbf{z}_{[\mathcal{O}]}$ follows interval truncated Gaussian and $A_{\mathbf{x}}$ is an involution.

3 Experiments

In the experiments, we compare the accuracy of single imputations using our method `EGC` and several competitors. It is more challenging to compare the accuracy for multiple imputation (MI) as MI seeks to recover the correct *distribution* which is mostly unknown in practice. As shown in Algorithm 2, `EGC` does not require any model hyperparameter but only optimization hyperparamters M and β, for which we use fixed values $M = 5000$ and $\beta = 1000$ across all our experiments. We show in Section 3.3 that this option already accurately estimates the categorical marginals.

Competitors. We implement several imputation methods for mixed categorical and ordered data. Among iterative imputation algorithms, we implement `missForest` [25, 24] and `MICE` [4, 27]. Among LRMC algorithms, we implement `imputeFAMD` [1, 16] and `softImpute` [19, 12]. Both methods one-hot encodes categorical variables. `imputeFAMD` includes special weighting strategy incorporating different variables types, while `softImpute` simply treats the data as numerical. We also implement a `baseline` imputation: majority vote for categorical variables and median imputation for ordered variables. We only need to choose hyperparameters for `imputeFAMD` and `softImpute` to have satisfying performance, for which we further mask 20% of observed training entries as a validation set. Through all experiments in this paper, `softImpute` is the fastest method and `EGC` comes second overall. We include all implementation and runtime details in the supplement.

Evaluation. To evaluate an imputation method, we compute misclassification error (ME) for categorical variables and mean absolute error (MAE) for ordered variables at masked entries.

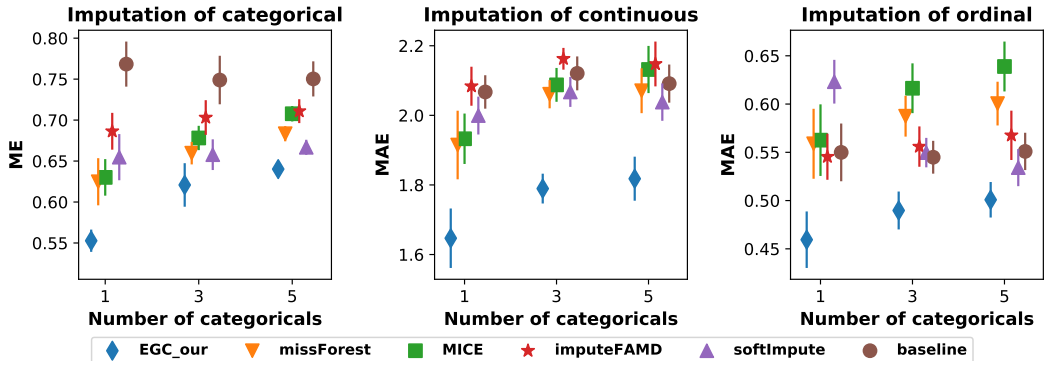

Figure 1: Imputation error on synthetic mixed data with 2000 samples. There are 5 continuous variables, 5 ordinal variables and $1/3/5$ categorical variables with three categories, reported over 10 repetitions (error bars indicate standard deviation).

3.1 Synthetic experiments

We generate synthetic datasets with 2000 samples and $p \in \{11, 13, 15\}$ variables from the EGC model in Definition 2 with randomly drawn parameters. The p variables include 5 exponentially distributed continuous variables, 5 ordinal variables with five ordinal levels, and $p_{\text{cat}} \in \{1, 3, 5\}$ categorical variables with six categories ($K = 6$). We randomly remove 30% entries of $\mathbf{X}$ for imputation evaluation and generate 10 masked datasets using different seeds.

Fig. 2 shows our method `EGC` performs the best for all variable types in all scenarios. For categorical variables, `EGC` yields a larger improvement with fewer categorical variables. For both continuous and ordinal variables, `EGC` substantially outperforms all other methods. Several methods that impute categorical variables quite well struggle to outperform a simple baseline on ordered variables. The supplement includes more experiments that differ by the number of categories for categorical variables ($K = 3, 9$), the missing ratio (20% and 40%) and missing mechanisms (MAR and MNAR). In general, these experiments show `EGC` performs well for both categorical and ordered variables in mixed data when the model is well-specified.

3.2 Real data experiments

We employ six real-world datasets from the UCI machine learning repository here [6]. Each dataset contains both categorical and ordered variables (details in the supplement). We randomly remove 20% of observed entries as a test set to evaluate imputation accuracy and generate 10 masked datasets using different seeds. Since variables in real data often have very different scales, e.g., number of categories for categorical variables and standard deviation for continuous variables, we use an evaluation metric that compensates for the imputation difficulty across variables. For an imputation method, we normalize its imputation error for each variable by the imputation error of `baseline` and then compute the average normalized error for each categorical variable and ordered variable, similar to [31]. We denote the scaled imputation error for categorical variables and ordered variables as SME and SMAE, respectively.

Fig. 2 shows that our method `EGC` is one of the best method in almost all evaluation metrics. `missForest` comes second but its total runtime is four times longer than that of `EGC`. Interestingly, `softImpute` imputes quite accurately for categorical data but poorly for ordered data. The supplement includes more experiments that differ by the missing ratio and missing mechanism, demonstrating that `EGC` consistently outperforms other methods.

3.3 Categorical marginal estimation accuracy

Here we verify that `EGC` can learn arbitrary categorical distribution, i.e. the algorithm in Section 2.3.1 finds the root $\boldsymbol{\mu}$ of Eq. (2) accurately. Concretely, we compare two sides of Eq. (2), where the right side is the empirical category probability and the left side is approximated at the estimated $\boldsymbol{\mu}$ using 10000 random samples. Across all used datasets in Section 3.1 and Section 3.2, Fig. 3 shows the

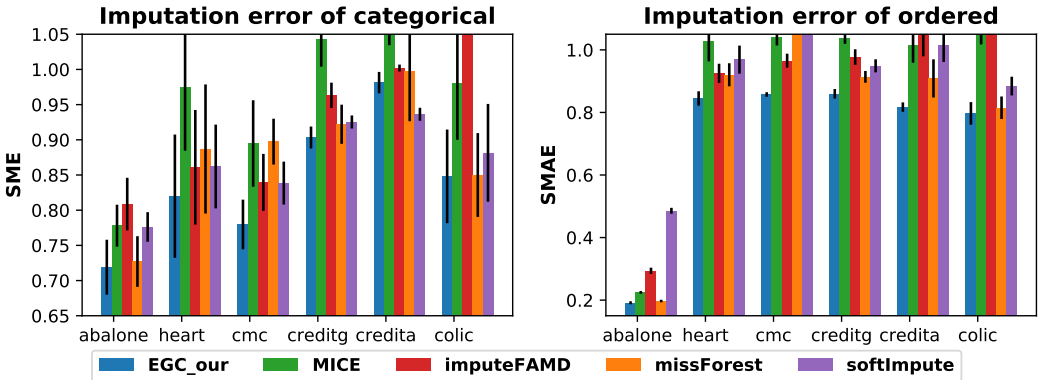

Figure 2: Imputation error of categorical variables and of ordered variables, i.e., ordinal and continuous, on 4 UCI datasets at 20% missingness. Results shown as mean $\pm$ standard deviation.

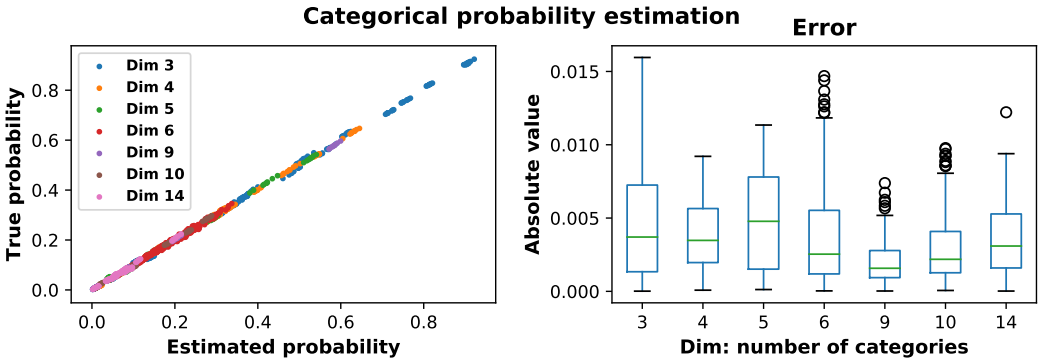

Figure 3: Performance of categorical distribution fitting. The results are collected from 70 datasets: each of 7 datasets (1 synthetic and 6 real data) is randomly masked 10 times. Dim is the number of categories for each categorical variable. *Left* plots the true probability vs the estimated probability of each categorical value, colored by dim, and *Right* plots the absolute values of the estimation error as a boxplot, grouped by dim.

estimated probability indeed matches the true probability well and the performance is robust to the number of categories. Using a larger M in Eq. (6) can further reduce the estimation error, but we find it does not significantly improve the imputation performance.

4 Discussion and conclusion

We have presented a probabilistic model, the EGC model, for mixed categorical and ordered data. We developed an imputation method based on the EGC model that empirically demonstrates state-of-the-art performance. Here we discuss two limitations of this paper. First, the estimation algorithm for the EGC model admits guarantees under MCAR mechanism. While it works reasonably well empirically under other missing mechanisms, it may be possible to improve performance by estimating and exploiting the missing mechanism for model estimation. Second, the EGC model uses a Gaussian dependence structure after a marginal transformation. This paper demonstrates the benefits of the explicit marginal modeling. Learning a more complex dependence structure, e.g., with neural networks, may yield further improvements.

References

[1] AUDIGIER, V., HUSSON, F., AND JOSSE, J. A principal component method to impute missing values for mixed data. *Advances in Data Analysis and Classification 10*, 1 (2016), 5–26.

[2] BOTEV, Z. I. The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 79*, 1 (2017), 125–148.

[3] BUNCH, D. S. Estimability in the multinomial probit model. *Transportation Research Part B: Methodological 25*, 1 (1991), 1–12.

[4] BUUREN, S. V., AND GROOTHUIS-OUDSHOORN, K. mice: Multivariate imputation by chained equations in r. *Journal of statistical software* (2010), 1–68.

[5] CHRISTOFFERSEN, B., CLEMENTS, M., HUMPHREYS, K., AND KJELLSTRÖM, H. Asymptotically exact and fast gaussian copula models for imputation of mixed data types. In *Asian Conference on Machine Learning* (2021), PMLR, pp. 870–885.

[6] DUA, D., AND GRAFF, C. UCI machine learning repository, 2017.

[7] FAN, J., LIU, H., NING, Y., AND ZOU, H. High dimensional semiparametric latent graphical model for mixed data. *Journal of the Royal Statistical Society. Series B: Statistical Methodology 79*, 2 (2017), 405–421.

[8] FAN, J., AND UDELL, M. Online high rank matrix completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 8690–8698.

[9] FENG, H., AND NING, Y. High-dimensional mixed graphical model with ordinal data: Parameter estimation and statistical inference. In *The 22nd International Conference on Artificial Intelligence and Statistics* (2019), pp. 654–663.

[10] GAL, Y., CHEN, Y., AND GHAHRAMANI, Z. Latent gaussian processes for distribution estimation of multivariate categorical data. In *International Conference on Machine Learning* (2015), PMLR, pp. 645–654.

[11] GUO, J., LEVINA, E., MICHAILIDIS, G., AND ZHU, J. Graphical models for ordinal data. *Journal of Computational and Graphical Statistics 24*, 1 (2015), 183–204.

[12] HASTIE, T., AND MAZUMDER, R. *softImpute: Matrix Completion via Iterative Soft-Thresholded SVD*, 2015. R package version 1.4.

[13] HOFF, P. D., ET AL. Extending the rank likelihood for semiparametric copula estimation. *The Annals of Applied Statistics 1*, 1 (2007), 265–283.

[14] JANG, E., GU, S., AND POOLE, B. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (2017), OpenReview.net.

[15] JOSSE, J., CHAVENT, M., LIQUET, B., AND HUSSON, F. Handling missing values with regularized iterative multiple correspondence analysis. *Journal of classification 29*, 1 (2012), 91–116.

[16] JOSSE, J., AND HUSSON, F. missMDA: A package for handling missing values in multivariate data analysis. *Journal of Statistical Software 70*, 1 (2016), 1–31.

[17] KINGMA, D. P., AND WELLING, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[18] LIU, H., LAFFERTY, J., AND WASSERMAN, L. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research 10*, 10 (2009).

[19] MAZUMDER, R., HASTIE, T., AND TIBSHIRANI, R. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research 11*, Aug (2010), 2287–2322.

[20] McCulloch, R., and Rossi, P. E. An exact likelihood analysis of the multinomial probit model. *Journal of Econometrics 64*, 1-2 (1994), 207–240.

[21] McLachlan, G., and Krishnan, T. *The EM algorithm and extensions*, vol. 382. John Wiley & Sons, 2007.

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

[23] Rubin, D. B. Multiple imputation after 18+ years. *Journal of the American statistical Association 91*, 434 (1996), 473–489.

[24] Stekhoven, D. J. *missForest: Nonparametric Missing Value Imputation using Random Forest*, 2013. R package version 1.4.

[25] Stekhoven, D. J., and Bühlmann, P. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics 28*, 1 (2011), 112–118.

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

[27] van Buuren, S., and Groothuis-Oudshoorn, K. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software 45*, 3 (2011), 1–67.

[28] Yang, C., Fan, J., Wu, Z., and Udell, M. Automl pipeline selection: Efficiently navigating the combinatorial space. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2020), pp. 1446–1456.

[29] Zhao, Y., Landgrebe, E., Shekhtman, E., and Udell, M. Online missing value imputation and change point detection with the gaussian copula. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2022).

[30] Zhao, Y., and Udell, M. Matrix completion with quantified uncertainty through low rank gaussian copula. In *Advances in Neural Information Processing Systems* (2020), vol. 33.

[31] Zhao, Y., and Udell, M. Missing value imputation for mixed data via gaussian copula. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2020), pp. 636–646.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 4
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See the supplement
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Provided in the supplemental material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See the supplement
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See the supplement

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] No license provided
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] All used data is publicly available
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] The used data does not contain any ID information or offensive content
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] Research does not involve human subjects
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Research does not involve human subjects
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Research does not involve human subjects