Online Missing Value Imputation and Change Point Detection with the Gaussian Copula

Yuxuan Zhao[*]**, Eric Landgrebe**[*]**, Eliot Shekhtman, Madeleine Udell**

Cornell University, Ithaca, NY 14853, USA
{yz2295, ecl93,ess239,udell}@cornell.edu

Abstract

Missing value imputation is crucial for real-world data science workflows. Imputation is harder in the online setting, as it requires the imputation method itself to be able to evolve over time. For practical applications, imputation algorithms should produce imputations that match the true data distribution, handle data of mixed types, including ordinal, boolean, and continuous variables, and scale to large datasets. In this work we develop a new online imputation algorithm for mixed data using the Gaussian copula. The online Gaussian copula model meets all the desiderata: its imputations match the data distribution even for mixed data, improve over its offline counterpart on the accuracy when the streaming data has a changing distribution, and on the speed (up to an order of magnitude) especially on large scale datasets. By fitting the copula model to online data, we also provide a new method to detect change points in the multivariate dependence structure with missing values. Experimental results on synthetic and real world data validate the performance of the proposed methods.

1 Introduction

Many modern datasets contain missing values; yet many machine learning algorithms require complete data. Hence missing value imputation is an important preprocessing step. The progress in low rank matrix completion (LRMC) (Candes and Plan 2010; Recht, Fazel, and Parrilo 2010) has led to widespread use in diverse applications (Bell and Koren 2007; Yang et al. 2019). LRMC succeeds when the data matrix can be well approximated by a low rank matrix. While this assumption is often reasonable for sufficiently large data matrices (Udell and Townsend 2019), it usually fails when one dimension of the data matrix is much larger than the other. We refer to such matrices as long skinny datasets, or high rank matrices. When a long skinny dataset has *mixed* type, consisting of a combination of ordinal, binary, and continuous (or real-valued) variables, the imputation challenge is even greater, and successful methods must account for the different distribution of each column. For example, survey dataset may contain millions of respondents but only dozens of questions. The questions may include both real-valued responses such as age and weight, and ordinal responses on a Likert scale measuring how strongly a respondent agrees with certain stated opinions. A Gaussian copula imputation model, which adapts to the distribution of values in each column, has recently shown state-of-the-art performance on a variety of long skinny mixed datasets (Zhao and Udell 2020b). Our work builds on the success of this model, which we describe in greater detail below.

Missing values also appear in online data, generated by sensor networks, or ongoing surveys, as sensors fail or survey respondents fail to respond. In this setting, online (immediate) imputation for new data points is important to facilitate online decision-making processes. However, most missing value imputation methods, including missForest (Stekhoven and Bühlmann 2012) and MICE (Buuren and Groothuis-Oudshoorn 2010), cannot easily update model parameters with new observation in the online setting. Reapplying offline methods after seeing every new observation consumes too much time and space. Online methods, which incrementally update the model parameters every time new data is observed, enjoy lower space and time costs and can adapt to changes in the data, and hence are sometimes preferred even in the offline setting.

Another common interest for online data (or time series) is change point detection: does the data distribution change abruptly, and can we pinpoint when the change occurs? While there are many different types of temporal changes, we focus on changes in the dependence structure of the data, a crucial issue for many real world applications. For example, classic Markowitz portfolio design uses the dynamic correlation structure of exchange rates and market indexes to design a portfolio of assets that balances risk and reward (Markowitz 1991). In practice, the presence of missing values and mixed data handicaps most conventional change point detection approaches.

In this paper, we address all these challenges: our online algorithm can impute missing values and detect changes in the dependency structure of long skinny mixed data, including real-valued data and ordinal data as special cases. Our online imputation method builds on the offline Gaussian copula imputation model (Zhao and Udell 2020b). This model posits that each data point is generated by drawing a latent Gaussian vector. This latent Gaussian vector is then transformed to match the marginal distribution of each ob-

---

[*]Equal contribution

served variable. Ordinals are assumed to result from thresholding a real-valued latent variable. In the case of, say, product ratings data, we can imagine the observed ordinal values result from thresholding the customer's (real-valued) affinity for a given product.

Contribution   We make three major contributions: (1) We propose an online algorithm for missing value imputation using the Gaussian copula model, which incrementally updates the model and thus can adapt to a changing data distribution. (2) We develop a mini-batch Gaussian copula fitting algorithm to accelerate the training in the offline setting. Compared to the offline algorithm (Zhao and Udell 2020b), our methods achieve nearly the same imputation accuracy but being an order of magnitude faster, which allows the Gaussian copula model to scale to larger datasets. (3) We propose a Monte Carlo test for dependence structure change detection *at any time*. The method tracks the magnitude of the copula correlation update and reports a change point when the magnitude exceeds a threshold. Inheriting the advantages of the Gaussian copula model, all our proposed methods naturally handle long skinny mixed data with missing values, and have no model hyperparameters except for common online learning rate parameters. This property is crucial in the online setting, where the best model hyperparameters may evolve.

Related work   The Gaussian copula has been used to impute incomplete mixed data in the offline setting using an expectation maximization (EM) algorithm (Zhao and Udell 2020b). Here, we develop an online EM algorithm to incrementally update the copula correlation matrix, following (Cappé and Moulines 2009), and an online method to estimates the marginals, so that there is no need to store historical data except for the previous model estimate.

Existing online imputation methods mostly rely on matrix factorization (MF). Online LRMC methods (Balzano, Nowak, and Recht 2010; Dhanjal, Gaudel, and Clémençon 2014) assume a low rank data structure. Consequently, they work poorly for long skinny data. Online KFMC (Fan and Udell 2019) first maps the data to a high dimensional space and assumes the mapped data has a low rank structure. It learns a nonlinear structure and outperforms online LRMC for long skinny data. However, its performance is sensitive to a selected rank r, which should be several times larger than the data dimension p and thus needs to be carefully tuned in a wide range. As p increases, it also requires increasing r to outperform online LRMC methods; for moderate p, the $O(r^3)$ computation time of online KFMC becomes prohibitive. For all aforementioned MF methods, their underlying continuity assumptions can lead to poor performance on mixed data. Moreover, the sensitivity to the rank poses a difficulty in the online setting, as the best rank may vary over time, and the rank chosen by cross-validation early on can lead to poor performance or even divergence later.

While recent deep generative imputation methods (Yoon, Jordon, and Schaar 2018; Mattei and Frellsen 2019) look like online methods (due to the SGD update), they actually require lots of data, and are slow to adapt to changes in the data stream, which are unsatisfying for real-time tasks. Deep

time series imputation methods (Cao et al. 2018; Fortuin et al. 2020) use the future to impute the past, and thus do not suit the considered online imputation task.

Change point detection (CPD) is an important topic with a long history. See Aminikhanghahi and Cook (2017) for an expansive review. Online CPD seeks to identify change points in real-time, before seeing all the data. Missing data is also a key challenge for CPD: most CPD algorithms require complete data. The simplest fix for this problem, imputation followed by a complete-data CPD method, can hallucinate change points due to the changing missingness structure or imputation method used. Our proposed method avoids these difficulties. Another workaround, Bayesian online CPD methods (Adams and MacKay 2007; Fearnhead and Liu 2007), can fill out the missing entries by sampling from its posterior distribution given all observed entries.

2   Methodology

Gaussian copula has two parameters: the transformation function and the copula correlation matrix. We first review Gaussian copula imputation with known model parameters. Online imputation differs from offline imputation only in how we estimate the model parameters. We assume the missing mechanism is missing completely at random (MCAR) throughout the paper, same as in the offline setting (Zhao and Udell 2020b), but show our method is robust to missing not at random (MNAR) mechanism empirically in the supplement. We show how to estimate the transformation online in Section 2.1 and how to estimate the copula correlation online in Section 2.2 with a given marginal estimate.

Notation   Define $[p] = \{1, \ldots, p\}$ for $p \in \mathbb{N}^+$. We use capital letters $\mathbf{X}$ to denote matrices and lower-case letters $\mathbf{x}$ to denote vectors. For a matrix $\mathbf{X}$, we refer to the i-th row, j-th column, and (i, j)-th entry as $\mathbf{x}^i, \mathbf{X}_j$ and x^i_j, respectively. We use columns to represent variables and rows to represent examples. For a vector $\mathbf{x} \in \mathbb{R}^p$, we use $\mathbf{x}_I$ to denote the subvector of $\mathbf{x}$ with entries in subset $I \subset [p]$. For each row vector $\mathbf{x}^i$, we use $\mathcal{O}_i, \mathcal{M}_i \subset [p]$ to denote the observed and missing locations respectively, and thus $\mathbf{x}^i_{\mathcal{O}_i}$ is observed and $\mathbf{x}^i_{\mathcal{M}_i}$ is missing. We use $\phi(\cdot; \mu, \Sigma)$ for the PDF of a normal vector with mean μ and covariance matrix Σ.

Gaussian copula imputation   We now formally introduce the Gaussian copula model for mixed data (Hoff et al. 2007; Fan et al. 2017; Feng and Ning 2019; Zhao and Udell 2020b). We say a random vector $\mathbf{x} \in \mathbb{R}^p$ follows the Gaussian copula model, $\mathbf{x} \sim \mathrm{GC}(\Sigma, \mathbf{f})$, if $\mathbf{x} = \mathbf{f}(\mathbf{z}) := (f_1(z_1), \ldots, f_p(z_p))$ with $\mathbf{z} \sim N(\mathbf{0}, \Sigma)$, for correlation matrix $\Sigma \in \mathbb{R}^{p \times p}$ and elementwise monotone $\mathbf{f} : \mathbb{R}^p \to \mathbb{R}^p$. In other words, we generate a Gaussian copula random vector $\mathbf{x}$ by first drawing a latent Gaussian vector $\mathbf{z}$ with mean 0 and covariance Σ, and then applying the elementwise monotone function $\mathbf{f}$ to $\mathbf{z}$ to produce $\mathbf{x}$. If the cumulative distribution function (CDF) for x_j is given by F_j, then f_j is uniquely determined: $f_j = F_j^{-1} \circ \Phi$ where Φ is the standard Gaussian CDF. For ordinal x_j, the CDF F_j and thus f_j are step functions, so $f_j^{-1}(x_j) := \{z_j : f_j(z_j) = x_j\}$ is an interval. If $\mathbf{x} \sim \mathrm{GC}(\Sigma, \mathbf{f})$ is observed at indices $\mathcal{O}$, we map the
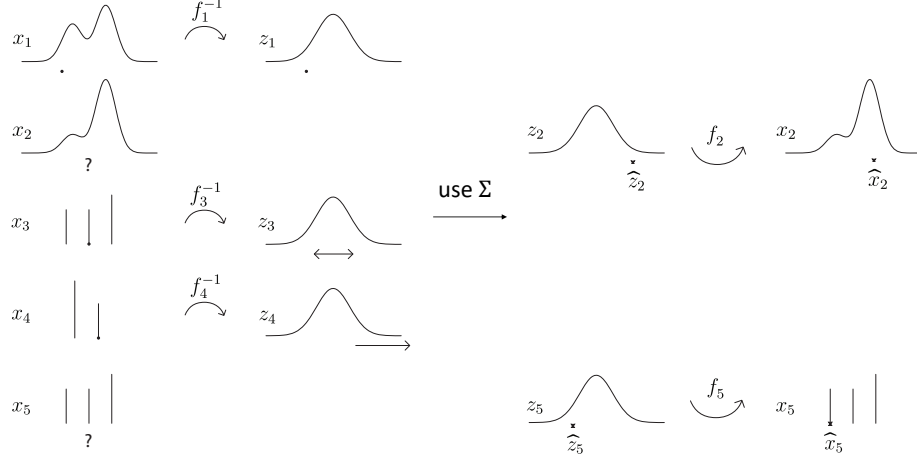
Figure 1: Gaussian copula imputation for a 5-dim partially observed mixed vector. Curves indicate the marginal probability density functions (for continuous) or probability mass function (for ordinal). First, compute the set of the latent normal vector which maps to the observation (x_1, x_3 and x_4) through $\mathbf{f}^{-1}$. Second, compute the conditional mean of the latent normal vector at missing locations ($\hat{z}_2$ and $\hat{z}_5$) given the copula correlation Σ and that z_1, z_3 and z_4 only take values from the computed inverse set. Lastly, map the conditional mean through $\mathbf{f}$ to obtain the imputations $\hat{x}_2$ and $\hat{x}_5$.

conditional mean of $\mathbf{z}_{\mathcal{M}}$ given observations $\mathbf{x}_{\mathcal{O}}$ through $\mathbf{f}$ to impute the missing values $\mathbf{x}_{\mathcal{M}}$ (Zhao and Udell 2020b), as in Eq. (1) and visualized in Fig. 1.

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

$\Sigma_{I,J}$ denotes the submatrix of Σ with rows in I and columns in J. The expectation $\mathbb{E}[\mathbf{z}_{\mathcal{O}}|\mathbf{x}_{\mathcal{O}}, \Sigma, \mathbf{f}]$ is the mean of a normal vector $\mathbf{z}_{\mathcal{O}}$ truncated to the region $\prod_{j \in \mathcal{O}} f_j^{-1}(x_j)$, which can be estimated efficiently. For an incomplete matrix $\mathbf{X}$, we assume that the rows are iid samples from $\mathrm{GC}(\Sigma, \mathbf{f})$ with some $\mathbf{f}$ and Σ. Each row is imputed separately as described above with $\mathbf{f}$ and Σ replaced by their estimates. We now turn to the problem of estimating the parameters $\mathbf{f}$ and Σ.

2.1 Online Marginal Estimation

In the offline setting, we estimate the transformation $\mathbf{f}$ based on the observed empirical distribution (Liu, Lafferty, and Wasserman 2009; Zhao and Udell 2020b,a): for $j \in [p]$, using observations in $\mathbf{X}_j$, we construct the estimates as:

$$\hat{f}_j = \hat{F}_j^{-1} \circ \Phi, \quad \hat{f}_j^{-1} = \Phi^{-1} \circ \hat{F}_j. \quad (2)$$

where F_j and F_j^{-1} are the empirical CDF and quantile function on the observed entries of the j-th variable. In the online setting, we simply update the observation set as new data comes in for each column $\mathbf{X}_j$. Specifically, we store a running window matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{k \times p}$ which records the k most recent observations for each column, and update $\tilde{\mathbf{X}}$ as new data comes in. The window size is an online learning rate hyperparameter that should be tuned to improve accuracy. A longer window works better when the data distribution is

mostly stable but has a few abrupt changes. If the data distribution changes rapidly, a shorter window is needed. Domain knowledge should also inform the choice of window length.

2.2 Online Copula Correlation Estimation

We estimate copula correlation matrix Σ through maximum likelihood estimation (MLE). The existing offline method (Zhao and Udell 2020b) applies EM algorithm to find the Σ that maximizes the likelihood value. The key idea of our online estimation is to replace each offline EM iteration with an online EM variant, which incrementally updates the likelihood objective as new data comes in. This online approach does not need to retain all data to perform updates. We first present the offline likelihood objective to be maximized and then show how to update it in the online setting.

First when the data matrix $\mathbf{X}$ is fully observed with all continuous columns, we can compute exactly the Gaussian latent variable $\mathbf{z}^i = \mathbf{f}^{-1}(\mathbf{x}^i) \in \mathbb{R}^p$ for each row i. The likelihood objective is simply Gaussian likelihood $\sum_i \log \phi(\mathbf{z}^i; 0, \Sigma)$, and the MLE of the correlation matrix Σ is the empirical correlation matrix $P_{\mathcal{E}}\left(\frac{1}{n}\sum_{i=1}^n \mathbf{z}^i(\mathbf{z}^i)^{\top}\right)$, where $P_{\mathcal{E}}$ scales its argument to output a correlation matrix: for $D = \mathrm{diag}(\Sigma)$, $P_{\mathcal{E}}(\Sigma) = D^{-1/2}\Sigma D^{-1/2}$.

If there exist missing entries and ordinal variables in the data $\mathbf{X}$, the likelihood given observation $\mathbf{x}_{\mathcal{O}_i}^i$ is the integral over the latent Gaussian vector $\mathbf{z}_{\mathcal{O}_i}^i$ that maps to $\mathbf{x}_{\mathcal{O}_i}^i$ through $\mathbf{f}$: $z_j \in f_j^{-1}(x_j)$ for $j \in \mathcal{O}_i$ and $\mathbf{z}_{\mathcal{M}_i}^i \in \mathbb{R}^{|\mathcal{M}_i|}$. For simplicity, we write $\mathbf{z}^i \in \mathbf{f}^{-1}(\mathbf{x}^i)$ by defining that f_j^{-1} maps missing values to $\mathbb{R}$. Thus the observed likelihood objective we seek to maximize (over Σ) is:

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

The EM algorithm in Zhao and Udell (2020b) avoids maximizing this difficult integral; instead, the M-step at iteration $l+1$ maximizes the expectation of the complete likelihood $\ell(\Sigma; \{\mathbf{z}^i, \mathbf{x}^i_{\mathcal{O}_i}\}^n_{i=1})$, conditional on the observations $\mathbf{x}^i_{\mathcal{O}_i}$, the previous estimate Σ^l and the marginal estimate $\hat{\mathbf{f}}$, computed in the E-step. We denote this objective function as below:

$$Q(\Sigma; \Sigma^l, \{\mathbf{x}^i_{\mathcal{O}_i}\}^n_{i=1}) = \frac{1}{n}\sum_{i=1}^n \mathbb{E}[\ell(\Sigma; \mathbf{z}^i, \mathbf{x}^i_{\mathcal{O}_i})|\mathbf{x}^i_{\mathcal{O}_i}, \Sigma^l, \hat{\mathbf{f}}]. \quad (3)$$

Now the maximizer for Eq. (3) is simply the expected "empirical covariance matrix" of the latent variables $\mathbf{z}^i$:

$$\Sigma^{l+1} = \sum_{i=1}^n \frac{1}{n}\mathbb{E}[\mathbf{z}^i(\mathbf{z}^i)^\top|\mathbf{x}^i_{\mathcal{O}_i}, \Sigma^l, \hat{\mathbf{f}}]. \quad (4)$$

The expectation weights these $\mathbf{z}^i$ by their conditional likelihood value. These expectations are fast to approximate (Zhao and Udell 2020b; Guo et al. 2015). At last the the obtained estimate is scaled to have unit diagonal to satisfy the copula model constraints: $\Sigma^{l+1} \leftarrow P_{\mathcal{E}}(\Sigma^{l+1})$.

Now we show how to adjust and maximize the objective Q in the online setting. When data points come in different batches, i.e. rows S_{t+1} observed at time $t+1$, Cappé and Moulines (2009) propose to update the objective function Q with new rows as:

$$Q_{t+1}(\Sigma) = (1-\gamma_t)Q_t(\Sigma) + \gamma_t Q(\Sigma; \Sigma^t, \{\mathbf{x}^i_{\mathcal{O}_i}\}_{i\in S_{t+1}}), \quad (5)$$

with $Q_1(\Sigma) = Q(\Sigma; \Sigma^0, \{\mathbf{x}^i_{\mathcal{O}_i}\}_{i\in S_1})$ given initial estimate Σ^0 and a monotonically decreasing stepsize $\gamma_t \in (0,1)$. Using Eq. (5), we derive a very natural update rule, stated as Lemma 1: in each step we simply take a weighted average of the previous covariance estimate and the estimate we get with a single EM step on the next batch of data. We require the batch size to be larger than the data dimension p to obtain a valid update. One can still make an immediate prediction for each new data point, but to update the model we must wait to collect enough data or use overlapping data batches.

Lemma 1. For data batches $\{\mathbf{x}^i\}_{i\in S_1}, ..., \{\mathbf{x}^i\}_{i\in S_t}$ with $\mathbf{x}^i \in \mathbb{R}^p$ and $\min_{l\in[t]}|S_l| > p$, and objective $Q_t(\Sigma)$ as in Eq. (5) for $\gamma_t \in (0,1)$. Given a marginal estimate $\hat{\mathbf{f}}$, for $l = 1,...,t$, $\Sigma^l := \arg\max_\Sigma Q_l(\Sigma)$ satisfies

$$\Sigma^{t+1} = (1-\gamma_t)\Sigma^t + \frac{\gamma_t}{|S_{t+1}|}\sum_{i\in S_{t+1}} \mathbb{E}[\mathbf{z}^i(\mathbf{z}^i)^\top|\mathbf{x}^i_{\mathcal{O}_i}, \Sigma^t, \hat{\mathbf{f}}]. \quad (6)$$

We also project the resulting matrix to a correlation matrix as in the offline setting. The update takes $O(\alpha p^3|S_t|)$ time with missing fraction α and $|S_t|$ rows. The proof (in the supplement) shows that online EM formally requires a weighted update to the expectation computed in the E-step. But for our problem, the parameter Σ, computed as the maximizer (in the M-step), is a *linear* function of the computed expectation (from the E-step). Hence the maximizer also evolves according to the same simple weighted update. A weighted update rule for the parameter fails — leading to divergence — for more general models, when the maximizer is not linear in

the expectation, such as for the low-rank-plus-diagonal copula correlation model of Zhao and Udell (2020a).

Cappé and Moulines (2009) prove an online EM algorithm converges to the stationary points of the KL divergence between the true distribution of the observation π (not necessarily the assumed model) and the learned model distribution, under some regularity conditions. We adapt their result to Theorem 1.

Theorem 1. Let $\pi(\mathbf{x}_{\mathcal{O}})$ be the distribution function of the true data-generating distribution of the observations and $g_\Sigma(\mathbf{x}_{\mathcal{O}})$ be the distribution function of the observed data from $\text{GC}(\Sigma, \mathbf{f})$, assuming data is missing uniformly at random (MCAR). Suppose the step-sizes $\gamma_t \in (0,1)$ satisfy $\sum_{t=1}^\infty \gamma_t^2 < \sum_{t=1}^\infty \gamma_t = \infty$. Let $\mathcal{L} = \{\Sigma \in S_{++}^p : \nabla_\Sigma\text{KL}(\pi\|g_\Sigma) = 0\}$ be the set of stationary points of $\text{KL}(\pi\|g_\Sigma)$ for a fixed $\mathbf{f}$. Under two regularity conditions on π (see the supplement), the iterates Σ^t produced by online EM (Eq. (6)) converge to $\mathcal{L}$ with probability 1 as $t \to \infty$.

The conditions on stepsize γ_t are standard for stochastic approximation methods. If the true correlation Σ generating the data evolves over time, a constant stepsize $\gamma_t \in (0,1)$ should be used to adapt the estimate to the changing correlation structure. We find using $\gamma_i = c/(i+c)$ with $c = 5$ for the offline setting and $\gamma_i = 0.5$ for the online setting gives good results throughout our experiments.

Algorithm 1: Online Imputation with the Gaussian Copula

Input: Window size k, step size γ_t for $t \in [T]$.
1: Initialize Σ^0 and running window matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{k\times p}$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3:     Obtain new data batch $\{\mathbf{x}^i\}_{i\in S_t}$, with $\mathbf{x}^i$ partially observed at $\mathcal{O}_i$ and missing at $\mathcal{M}_i$.
4:     Replace the oldest point in $\tilde{\mathbf{X}}_j$ with x_j^i for $j \in \mathcal{O}_i, i \in S_t$.
5:     Estimate marginals $\hat{\mathbf{f}}, \hat{\mathbf{f}}^{-1}$ using $\tilde{\mathbf{X}}$ as in Eq. (2) .
6:     EM step update: obtain Σ^{t+1} as in Eq. (6).
7:     Scale to a correlation matrix: $\Sigma^{t+1} = P_{\mathcal{E}}(\Sigma^{t+1})$.
8:     Impute $\hat{\mathbf{x}}^i_{\mathcal{M}_i}$ using Σ^{t+1} and $\hat{\mathbf{f}}$ as in Eq. (1) for $i \in S_t$.
9: **end for**
Output: Imputation $\{\hat{\mathbf{x}}^i_{\mathcal{M}_i}\}_{i\in S_t}$ and Σ^t for $t \in [T]$.

Online versus offline implementation We may estimate $\hat{\mathbf{f}}$ in Eq. (6) either online or offline. The decision entails some tradeoffs. When the storage limit is the main concern, as in the streaming data setting, we can employ the online marginal estimate, storing only a running window and a correlation matrix estimate. We call such an implementation fully *online EM*. When the data marginal distribution evolves over time, it is also important to use *online EM* to forget the old data. On the other hand, when training time is the main concern but the whole dataset is available, the online EM algorithm can be implemented as an offline minibatch EM algorithm to accelerate convergence. In that setting, the offline marginals are used to provide more accurate and stable estimates as well as to reduce the time for estimating the marginals. We call this implementation *(offline)*

mini-batch EM. We present the fully online algorithm in Algorithm 1 with data batches observed sequentially.

Parallelization Noting the computation of expectation in Eq. (4) and Eq. (6) are separable over the rows, we have developed a parallel algorithm to accelerate the both the offline and the online EM algorithms. For the long skinny datasets we target, this parallel algorithm allows for faster imputation by exploiting multiple computational cores.

2.3 Online Change Point Detection

We first outline the change point detection (CPD) problem in the context of the Gaussian copula model. Consider a sequence of incomplete mixed data observations $\mathbf{x}^1, \ldots, \mathbf{x}^T \sim \mathrm{GC}(\Sigma, \mathbf{f})$, where $\mathbf{x}^i$ is observed at locations $\mathcal{O}_i$ for $i \in [T]$. We wish to identify whether there is a change point t_0 — a time when the copula correlation Σ changes substantially — and if so, when this change occurs. We formulate the single CPD problem as the following hypothesis test, for fixed t_0: $\mathbf{x}^1, \ldots, \mathbf{x}^{t_0} \sim \mathrm{GC}(\Sigma, \mathbf{f})$, and $\mathbf{x}^{t_0+1}, \ldots, \mathbf{x}^T \sim \mathrm{GC}(\tilde{\Sigma}, \mathbf{f})$,

$$\mathrm{H}_0 : \tilde{\Sigma} = \Sigma \text{ versus } \mathrm{H}_1 : \tilde{\Sigma} \neq \Sigma. \tag{7}$$

We assume time-invariant marginal $\mathbf{f}$. In practice, it suffices for $\mathbf{f}$ to be stable in a small local window. The latent correlation matrix changes, reflecting the changing dependence structure. To detect a change-point, a test statistic is computed for each point to measure the deviation of new points from old distribution. A change is detected if the test statistic exceeds a certain threshold. We consider the online detection problem instead of a retrospective analysis with all data available. Specifically, to test whether a change occurs at time t_0, we may use only the data $\mathbf{x}^{t_0+1}, \ldots, \mathbf{x}^T$ for a small window length $T - t_0$ and the fitted model at time t_0.

To derive a test statistic, notice that $\Sigma^{-1/2} \tilde{\Sigma} \Sigma^{-1/2} = \mathrm{I}_p$ under H_0. Thus for some matrix norm h, we use the matrix distance $d(\Sigma, \tilde{\Sigma}; h) = h(\Sigma^{-1/2} \tilde{\Sigma} \Sigma^{-1/2} - \mathrm{I}_p)$ to measure the deviation of new points from old distribution. While Σ and $\tilde{\Sigma}$ are unknown, we replace them with the estimates Σ^{t_0} and Σ^T, generated by the EM iteration up to time t_0 and time T, respectively. Thus we construct our test statistic as $d(\Sigma^{t_0}, \Sigma^T; h)$: large values indicate high probability of a change point. Experimentally, we find that different choices of h give very similar trends. Hence below we report results using the Frobenius norm as h, to reduce computation.

The change point is detected when $d(\Sigma^{t_0}, \Sigma^T; h)$ exceeds some threshold b_α, which is chosen to control the false-alarm rate α. Calculating b_α analytically requires the asymptotic behaviour of the statistic under the null distribution, which is generally intractable including our case. We use Monte Carlo (MC) methods to simulate the null distribution of our test statistic and select the threshold. This method is similar to the permutation test for CPD (Matteson and James 2014). We present our test for the hypothesis in Eq. (7) as Algorithm 2 . Notice comparing $d(\Sigma^{t_0}, \Sigma^T; h)$ to b_α is equivalent to comparing the returned empirical p-value with the desired false-alarm rate α. See (Davison and Hinkley 1997; North, Curtis, and Sham 2002) for the use of empirical p-

values. In practice, α can be regarded as a hyperparameter to tune the false positive/negative rate.

Algorithm 2: Monte Carlo test for Gaussian copula correlation change point detection

---

Input: New data $\{\mathbf{x}^i\}_{i=t_0+1}^T$, the number of samples B, estimated model Σ^{t_0}, Σ^T and $\mathbf{f}^{t_0}$.

1: Compute the test statistic $s = d(\Sigma^{t_0}, \Sigma^T)$.
2: **for** $j = 1, 2, \ldots, B$ **do**
3:   Sample $\mathbf{y}^i \sim \mathrm{GC}(\Sigma^{t_0}, \mathbf{f}^{t_0})$ and mask $\mathbf{y}^i$ at where $\mathbf{x}^{i+t_0}$ is missing for $i = 1, \ldots, T - t_0$.
4:   Update the model at t_0 with new points $\{\mathbf{y}^i\}_{i=1}^{T-t_0}$.
5:   Compute $s_j = d(\Sigma^{t_0}, \Sigma^{T,j})$ with the updated correlation $\Sigma^{T,j}$.
6: **end for**

Output: The p-value $(|\{s_j : s \leq s_j\}| + 1)/(B + 1)$.

We have shown how to test if a change point happens at a time t_0. Repeating this test across time points may detect multiple change points, but also yield many false positives. We discuss in the supplement how to alleviate this issue using recent development from online FDR (Javanmard, Montanari et al. 2018; Ramdas et al. 2017, 2018).

3 Experiments

The experiments are divided into two parts: online datasets (rows obtained sequentially) and offline datasets (rows obtained simultaneously). The online setting examine the ability of our methods to detect and learn the changing distribution of the steaming data. The offline setting evaluate the speedups and the potential accuracy lost due to minibatch training and online marginal estimation compared to offline EM. See the supplement for more experimental details and more experiments under different data dimension, missing ratio and missing mechanisms.

Algorithm implementation: we implement the *offline EM* algorithm (Zhao and Udell 2020b), the minibatch EM with online marginal estimate denoted by *online EM*, and the minibatch EM with offline marginal estimate denoted by *minibatch EM*. For imputation comparison, we implement GROUSE (Balzano, Nowak, and Recht 2010) and online KFMC (Fan and Udell 2019). For fair comparison, we use 1 core for all methods, but report the acceleration brought by parallelism for all Gaussian copula methods in the supplement. We also implement the online Bayesian change point detection (BOCP) algorithm (Adams and MacKay 2007), one of the best performing CPD method according to a recent evaluation paper (van den Burg and Williams 2020). The norm of subspace fitting residuals for GROUSE can also serve for CPD: a sudden peak of large residual norm indicates abrupt changes. We compare our test statistic, defined in Algorithm 2, with the residual norms from GROUSE, to see which identifies change points more accurately.

Tuning parameters selection: we do not use tuning parameter for offline EM and minibatch EM. We use 1 tuning parameter for online EM: the window size m for online marginal estimates, 2 tuning parameters for GROUSE, the

rank and the step size, and 2 tuning parameters for online KFMC, the rank in a latent space and the regularization parameter. BOCP requires 4 hyperparameters for its priors and its hazard function (Adams and MacKay 2007).

We note one other issue. For online algorithms, it is typical to choose hyperparameters during an initial "burn-in" period. For example, in GROUSE, choosing the step-size from initial data can result in divergence later on as the data distribution changes. As a result, the maximum number of optimization iterations is also difficult to choose: the authors' default settings are often insufficient to give good performance, while allowing too many iterations may lead to (worse) divergence. We will report and discuss an example of divergence in our online real data experiment.

Imputation evaluation: for ordinal or real valued data, we use mean absolute error (MAE) and root mean squared error (RMSE). For mixed data, we use the scaled MAE (SMAE), the MAE divided by the MAE of the median imputation. A method that imputes better than the median has SMAE < 1.

Offline synthetic experiment We construct a dataset consisting of 6000 i.i.d. data points drawn from a 15-dimensional Gaussian Copula, with 5 continuous, 5 ordinal with 5 levels, and 5 binary entries. We randomly mask 40% entries as missing. Shown in Table 1, the minibatch and online variants of the EM algorithm converge substantially faster than offline EM and provide similar imputation accuracy. The results are especially remarkable for online EM, which estimates the marginals using only 200 points. The minibatch variant is three times faster than offline EM with the same accuracy. All EM methods outperform online KFMC and GROUSE, and even median imputation outperforms GROUSE. Interestingly, the best rank for GROUSE is 1. The results here show LRMC methods fit poorly for long skinny datasets, although the selected best rank, 1, misleadingly indicates the existence of low rank structure.

Method	Runtime (s)	Continuous	Ordinal	Binary
Offline EM	187.7(0.8)	0.79(.04)	0.84(.03)	0.63(.07)
Minibatch EM	48.2(0.5)	0.79(.04)	0.83(.03)	0.63(.07)
Online EM	54.5(3.4)	0.80(.04)	0.84(.02)	0.63(.07)
Online KFMC	79.6(1.6)	0.92(.03)	0.92(.02)	0.67(.08)
GROUSE	7.7(.3)	1.17(.03)	1.67(.05)	1.10(.07)

Table 1: Mean(sd) for runtime, imputation error of each data type for synthetic offline data over 10 trials.

Online synthetic experiment Now we consider streaming data from a changing distribution. To do this, we generate and mask the dataset similar to Section 3, but set two change points at which a new correlation matrix is chosen: $\mathbf{x}^1, \ldots, \mathbf{x}^t \sim \text{GC}(\Sigma_1, \mathbf{f})$, $\mathbf{x}^{t+1}, \ldots, \mathbf{x}^{2t} \sim \text{GC}(\Sigma_2, \mathbf{f})$ and $\mathbf{x}^{2t+1}, \ldots, \mathbf{x}^{3t} \sim \text{GC}(\Sigma_3, \mathbf{f})$, with $t = 2000$. We implement all online algorithms from a cold start and make only one pass through the data, to mimic the streaming data setting. For comparison, we also implement offline EM and miss-Forest (Stekhoven and Bühlmann 2012) (also offline) and allow them to make multiple passes.

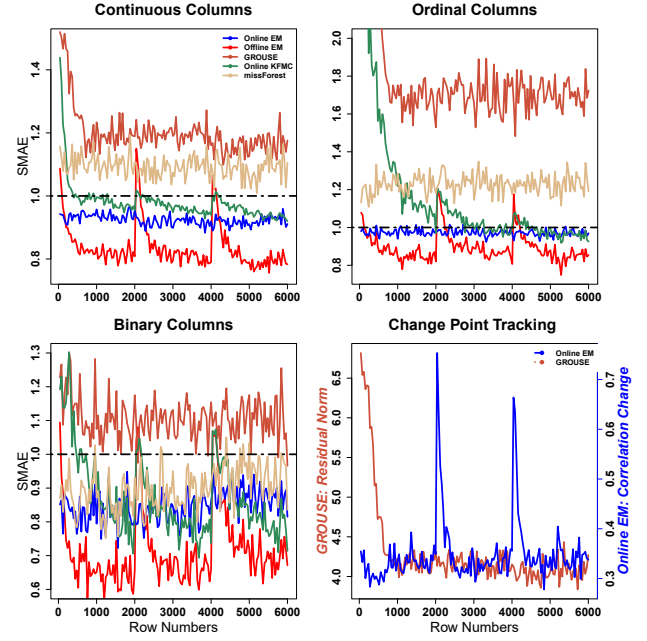Shown in Fig. 2, online EM clearly outperforms the of-

Figure 2: Mean imputation error and change point tracking statistics over 10 trials for online synthetic datasets. Each point stands for an evaluation over a data batch of 40 points.
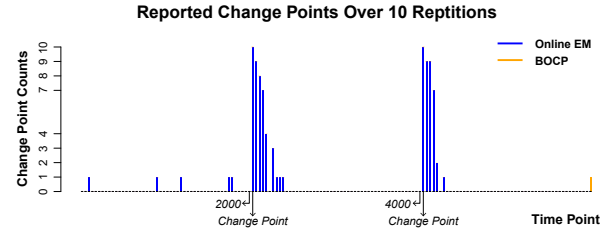

Figure 3: Change points from online EM detection (ours) and BOCP over 10 trials in online synthetic experiments. Each bar stands for a decision over a data batch of 40 points.

fline EM on average, by learning the changing correlation. Online EM has a sharp spike in error as the correlation abruptly shifts, but the error rapidly declines as it learns the new correlation. Both online EM and online KFMC outperform missForest. Surprisingly, online KFMC cannot even outperform offline EM. GROUSE performs even worse in that it cannot outperform median imputation as in the offline setting. The results indicate online imputation methods can fail to learn the changing distribution when their underlying model does not fit the data well. Our correlation deviation statistic provides accurate prediction for change points, while the residual norms from GROUSE remains stable after the burn-in period for model training, which verifies GROUSE cannot adapt to the changing dependence here.

Show in Fig. 3, online EM successfully detects both change points in all repetitions. In fact, the algorithm detects a change point (of decreasing magnitude) during sev-

Method	Runtime (s)	MAE	RMSE
Offline EM	1690(9)	0.583(.002)	0.883(.004)
Minibatch EM	252(2)	0.585(.003)	0.886(.003)
Online EM	269(3)	0.590(.002)	0.890(.003)
Online KFMC	176(21)	0.631(.005)	0.905(.006)
GROUSE	27(2)	0.634(.003)	0.933(.004)

Table 2: Mean(sd) for runtime and imputation error on a subset of MovieLens1M data over 10 trials.

eral batches after each true change, showing how long it takes to finally learn the new dependence structure. To avoid the repeated false alarms, one could set a burn-in period following each detected change point. In contrast, BOCP only reports one false discovery, showing its inability to detect the changing dependence structure.

Offline real data experiment To further show the speedup of the minibatch algorithms, we evaluate on a subset of the MovieLens 1M dataset (Harper and Konstan 2015) that consists of all movies with more than 1000 ratings, with 1-5 ordinal ratings of size 6939×207 with over 75% entries missing. Section 3 shows that the minibatch and online EM still obtain comparable accuracy to the offline EM. The minibatch EM is around 7 times faster than the offline EM. All EM methods significantly outperform online KFMC and GROUSE. Interestingly, as the dataset gets wider, online KFMC loses its advantage over GROUSE. The results here indicate the nonlinear structure learned by online KFMC fails to provide better imputation than the linear structure learned by GROUSE. In contrast, the structural assumptions of our algorithm retain their advantage over GROUSE even on wider data.

Online real data experiment We now evaluate both imputation and CPD on the daily prices and returns of 30 stocks currently in the Dow Jones Industrial Average (DJIA) across 5030 trading days. We consider two tasks: predicting each stock's price (or log return) today using only yesterday's data and a learned model. After prediction, we reveal today's data to further update the model.

In Fig. 4, the left 2 plots show that all methods predict prices well early on, but GROUSE and online KFMC both diverge eventually. The residuals norm from GROUSE also indicate divergence. In contrast, online EM has robust performance throughout. Although the imputation error peaks around the start of 2020, online EM is able to quickly adjust to the changing distribution: the imputation error quickly falls back. Thus online EM stands out in that it obviates the need of online hyperparameter selection to have stable performance. The right 2 plots show that online EM and GROUSE perform similarly on log returns: their error curves almost overlap each other. Online KFMC underperforms: it makes large errors more often. We conjecture GROUSE and online KFMC perform better on the log returns than on the price data because the scale of the data is stable, so that hyperparameters chosen early on still exhibit good performance later. The good performance of
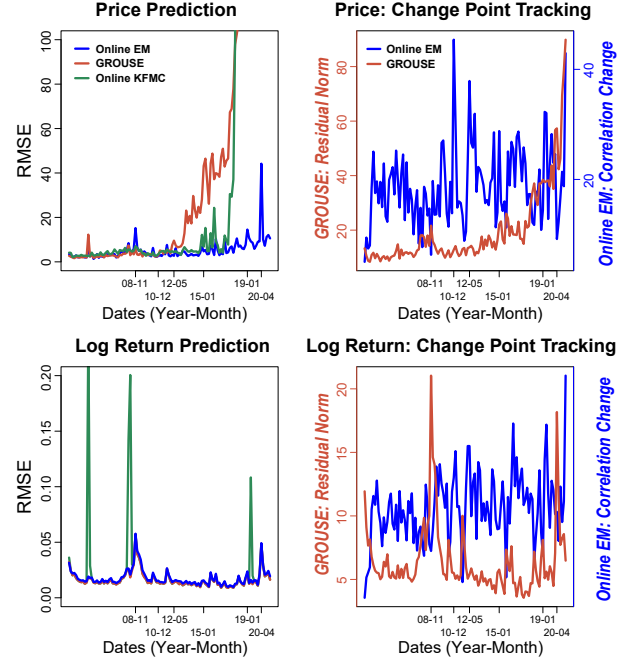

Figure 4: The imputation error and change tracking statistics for the DJIA daily price (left 2) and the DJIA daily log-returns (right 2), averaged over all stocks. Each point stands for an evaluation over a time interval of 40 points.

GROUSE indicates the asset log returns are approximately low rank, Still, online EM is robust to different (even changing) marginal data distributions and performs well on approximately low rank data.

As for CPD: online EM shows similar results for both the price and log returns datasets, identifying fluctuating (often large) changes but no very distinct spikes on either dataset. In fact, the algorithm classifies every (40-day) data batch as a change point, indicating the instability of stock data! In contrast, GROUSE detects two large changes in the log returns dataset and none in the price dataset. In the absence of ground-truth for change points, it is hard to compare the performance, but the improved stability of online EM to rescalings of the data is a clear advantage. BOCP quickly diverged on both the price and the log-returns dataset and did not return meaning results before divergence.

4 Conclusion

We presented an online missing data imputation algorithm and change point detection method using Gaussian copula for long skinny mixed datasets. The imputation performance can match or even exceed offline imputation, and improves on other state-of-the-art online imputations methods. Our algorithm also provides speedup for offline Gaussian copula imputation. The method can detect changes in the dependence structure, assuming the marginal remains stable over time. Developing a method to identify changes in the marginal distribution is an important future work.

Acknowledgement

References

Adams, R. P.; and MacKay, D. J. 2007. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*.

Aminikhanghahi, S.; and Cook, D. J. 2017. A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2): 339–367.

Balzano, L.; Nowak, R.; and Recht, B. 2010. Online identification and tracking of subspaces from highly incomplete information. In *2010 48th Annual allerton conference on communication, control, and computing (Allerton)*, 704–711. IEEE.

Bell, R. M.; and Koren, Y. 2007. Lessons from the netflix prize challenge. *Acm Sigkdd Explorations Newsletter*, 9(2): 75–79.

Buuren, S. v.; and Groothuis-Oudshoorn, K. 2010. mice: Multivariate imputation by chained equations in R. *Journal of statistical software*, 1–68.

Candes, E. J.; and Plan, Y. 2010. Matrix completion with noise. *Proceedings of the IEEE*, 98(6): 925–936.

Cao, W.; Wang, D.; Li, J.; Zhou, H.; Li, Y.; and Li, L. 2018. BRITS: bidirectional recurrent imputation for time series. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 6776–6786.

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

Davison, A. C.; and Hinkley, D. V. 1997. *Bootstrap methods and their application*. 1. Cambridge university press.

Dhanjal, C.; Gaudel, R.; and Clémençon, S. 2014. Online matrix completion through nuclear norm regularisation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, 623–631. SIAM.

Fan, J.; Liu, H.; Ning, Y.; and Zou, H. 2017. High dimensional semiparametric latent graphical model for mixed data. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 79(2): 405–421.

Fan, J.; and Udell, M. 2019. Online high rank matrix completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8690–8698.

Fearnhead, P.; and Liu, Z. 2007. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4): 589–605.

Feng, H.; and Ning, Y. 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.

Fortuin, V.; Baranchuk, D.; Rätsch, G.; and Mandt, S. 2020. Gp-vae: Deep probabilistic time series imputation. In *International Conference on Artificial Intelligence and Statistics*, 1651–1661. PMLR.

Guo, J.; Levina, E.; Michailidis, G.; and Zhu, J. 2015. Graphical models for ordinal data. *Journal of Computational and Graphical Statistics*, 24(1): 183–204.

Harper, F. M.; and Konstan, J. A. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4): 1–19.

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

Javanmard, A.; Montanari, A.; et al. 2018. Online rules for control of false discovery rate and false discovery exceedance. *The Annals of statistics*, 46(2): 526–554.

Liu, H.; Lafferty, J.; and Wasserman, L. 2009. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, 10(10).

Markowitz, H. M. 1991. Foundations of portfolio theory. *The journal of finance*, 46(2): 469–477.

Mattei, P.-A.; and Frellsen, J. 2019. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *International Conference on Machine Learning*, 4413–4423. PMLR.

Matteson, D. S.; and James, N. A. 2014. A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109(505): 334–345.

North, B. V.; Curtis, D.; and Sham, P. C. 2002. A note on the calculation of empirical P values from Monte Carlo procedures. *The American Journal of Human Genetics*, 71(2): 439–441.

Ramdas, A.; Yang, F.; Wainwright, M. J.; and Jordan, M. I. 2017. Online control of the false discovery rate with decaying memory. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 5655–5664.

Ramdas, A.; Zrnic, T.; Wainwright, M.; and Jordan, M. 2018. SAFFRON: an adaptive algorithm for online control of the false discovery rate. In *International conference on machine learning*, 4286–4294. PMLR.

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

Stekhoven, D. J.; and Bühlmann, P. 2012. MissForest—nonparametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1): 112–118.

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

van den Burg, G. J.; and Williams, C. K. 2020. An evaluation of change point detection algorithms. *arXiv preprint arXiv:2003.06222*.

Yang, C.; Akimoto, Y.; Kim, D. W.; and Udell, M. 2019. OBOE: Collaborative filtering for AutoML model selection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1173–1183.

Yoon, J.; Jordon, J.; and Schaar, M. 2018. Gain: Missing data imputation using generative adversarial nets. In *International Conference on Machine Learning*, 5689–5698. PMLR.

Zhao, Y.; and Udell, M. 2020a. Matrix Completion with Quantified Uncertainty through Low Rank Gaussian Copula. *Advances in Neural Information Processing Systems*, 33.

Zhao, Y.; and Udell, M. 2020b. Missing Value Imputation for Mixed Data via Gaussian Copula. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 636–646.