Mixture-based Multiple Imputation Models for Clinical Data with a Temporal Dimension

Ye Xue*
Northwestern University
Evanston, IL
ye.xue@u.northwestern.edu

Diego Klabjan
Northwestern University
Evanston, IL
d-klabjan@northwestern.edu

Yuan Luo
Northwestern University
Chicago, IL
yuan.luo@northwestern.edu

ABSTRACT

The problem of missing values in multivariable time series is a key challenge in many applications such as clinical data mining. Although many imputation methods show their effectiveness in many applications, few of them are designed to accommodate clinical multivariable time series. In this work, we propose multiple imputation models that capture both cross-sectional information and temporal correlations. We integrate Gaussian processes with mixture models and introduce individualized mixing weights to handle the variance of predictive confidence of Gaussian process models. The proposed models are compared with several state-of-the-art imputation algorithms on both real-world and synthetic datasets. Experiments show that our best model can provide more accurate imputation than the benchmarks on all of our datasets.

CCS CONCEPTS

• **Information systems → Data mining**; • **Theory of computation → Mathematical optimization**; • **Applied computing → Health informatics**.

KEYWORDS

Machine learning, imputation, missing data, EHR, Gaussian process, data mining

ACM Reference Format:
Ye Xue, Diego Klabjan, and Yuan Luo. 2019. Mixture-based Multiple Imputation Models for Clinical Data with a Temporal Dimension. In *ACM Comference.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The computational modeling in clinical applications attracts growing interest with the realization that the quantitative understanding of patient pathophysiologic progression is crucial to clinical studies [45]. With a comprehensive and precise modeling, we can have a better understanding of a patient's state, offer more precise diagnosis and provide better individualized therapies [22]. Researchers are increasingly motivated to build more accurate computational models from various kinds of clinical data. However, missing values in clinical data challenge researchers using analytic techniques for modeling, as many of the techniques are designed for complete data.

Traditional strategies used in clinical studies to handle missing values include deleting records with missing values and imputing missing entries by mean values. However, deleting records with missing values and some other filtering strategies can introduce biases [44] that can impact modeling in many ways, such as limiting its generalizability. Mean imputation is widely used by researchers to handle missing values. However, it is shown to yield less effective estimates than many other modern imputation techniques [1, 4, 43, 46], such as maximum likelihood (ML) approaches and multiple imputation (MI) methods (e.g. multivariable imputation by chained equations (MICE) [5]). The ML and MI methods are based on solid statistical foundations and become standard in the last few decades [13, 34].

In recent years, additional imputation methods are proposed. Although many imputation methods [5, 8, 14, 18, 23, 26, 27, 31, 32, 37, 38, 41–43, 50] show their effectiveness in many applications, few of them are designed for time series-based clinical data. These clinical data are usually multivariable time series, where patients have measurements of multiple laboratory tests at different times. Many methods are designed for cross-sectional imputation (measurements taken at the same time point) and do not consider temporal information that is useful in making predictions or imputing missing values. Ignoring informative temporal correlations and only capturing cross-sectional information may yield less effective imputation.

In order to address the limitations mentioned above, we present mixture-based multiple imputation models for clinical time series. Our models capture both cross-sectional information and temporal correlations to estimate missing values using mixture models. We model the distribution of measurements using a mixture model. The mixture is composed of linear regression to model cross-sectional correlations and Gaussian processes (GPs) to capture temporal correlations. The problem of integrating GP within a standard mixture model is that GP models in all patient cases get the same mixing weights, while the confidence of predictions by GP models can vary largely across different patient cases. We overcome this problem by introducing individualized mixing weights for each patient cases, instead of assigning a fixed weight. We train our models using the Expectation-Maximization (EM) algorithm. We demonstrate the effectiveness of our models by comparing them with several state-of-the-art imputation algorithms on multiple clinical datasets.

Our main contributions are summarized as follows.

*Ye Xue is the corresponding author.

1. To the best of our knowledge, we are the first to build imputation models for time series by integrating GP within mixture models. We overcome the problem that all GP models in all patient cases get a fixed mixing weight by introducing individualized mixing weights.

2. We test the performance of our models on two real-world clinical datasets and several synthetic datasets. Our best model outperforms all comparison models including several state-of-the-art imputation models. Using synthetic datasets, we also explore and discover the properties of the data that benefit our models and/or comparison models. Experiments show that our best model is robust to the variation of these properties and outperforms comparison models on all synthetic datasets.

The remainder of this paper is structured as follows. Section 2 discusses related work while in Section 3, the proposed method is described. The experimental setup, including dataset collection and evaluation procedure, is described in Section 4. Section 5 discusses the computational results and underlying analyses. The conclusions are drawn in Section 6.

2 RELATED WORK

Research in designing imputation methods for multivariable time series attracts growing interest in recent decades. Previous studies generally fall into two categories. One comes from methods using linear or other simple parametric functions to estimate missing values. The other is the methods treating time series as smooth curves and estimating missing values using GP or other nonparametric methods.

In the first category, multivariable time series are modeled based on either linear models [36], linear mixed models [24, 35] or autoregressive models [3, 16]. However, in these methods, the potential trajectories of variables are only limited to linear or other simple parametric functions. Alternatively, many authors choose GPs or other nonparametric functions to model time series. Compared to linear models, GPs only have locality constraints in which close time points in a time series usually have close values. Therefore, GPs bring in more flexibility in capturing temporal trajectories of variables.

The straightforward way of applying GPs to the imputation on multivariable time series is to fit a single GP model on each time series and then make predictions for missing entries separately. However, without taking into account similarities and correlations across multiple time series, only fitting a single GP model on each time series may yield less effective imputation. Many researchers attempt to extend GP-based methods to multivariable settings. Hori et al. [17] apply Multi-Task Gaussian Processes (MTGP), a multiple-outcome modeling approach in the context of GPs, to impute missing values in longitudinal data. However, the quality of estimating missing values relies on the estimation of covariance structure among variables when using MTGP or other multi-task functional approaches [7, 15, 21]. To make a confident estimation of the covariance, a large amount of time points with shared observations of these variables are required by these multi-task approaches. Due to the fact that many patients only have records with a limited number of time points, time series of inpatient clinical laboratory tests fall short of such a requirement. Therefore, these multi-task approaches are not applicable to inpatient clinical data.

Recently, Luo et al. [28] explore the application of GPs in clinical data and propose an algorithm, 3-dimensional multiple imputation with chained equations (3D-MICE), that combines the imputation from GP and traditional MICE based on weighting equations. However, the weighting equations are calculated only based on the standard deviations of GP and MICE predictions for missing values. The weighting strategy is static and not optimized. We postulate that calculating weights through an optimization problem can help to improve the imputation quality. In our work, instead of the predictive mean matching used in [28], we choose linear regression as one component of our model. Our model is also grounded by a statistical model and thus statistically justified which is not the case for [28]. Additionally, in order to effectively model the interaction between different aspects, we represent the data as a tensor with each aspect being one mode. For that reason, our method is also considered as a tensor completion approach.

Tensor completion problems are extensively studied. However, the classic tensor completion methods [9, 25, 40] focus on general tensors and usually do not consider temporal aspects. In recent years, many studies explore the application of temporal augmented tensor completion on imputing missing values in time series [2, 6, 11, 39, 49]. These methods discretize time into evenly sampled intervals. However, due to the fact that inpatient clinical laboratory tests are usually measured at varying intervals, assembling clinical data over regularly sampled time periods might have several drawbacks, such as leading to sparse tensors if discretizing time at fine granularity (e.g. every minute) while some laboratory tests are measured less frequently (e.g. daily). Furthermore, extending these methods to the case, where time is not regularly sampled, is not easy and straightforward, requiring changing design details and the objective functions to be optimized. Recently, Yang et al. [48] propose a tensor completion method that can deal with irregularly sampled times. They extend the PACIFIER imputation framework [52] and propose a time-aware matrix decomposition method to estimate missing values in predicting septic shock. However, most components of this approach are tailored to the characteristics of septic patients. In this work, we implement the imputation approach proposed in [48] with only the time-aware mechanism, which is general and applicable to our experimental settings. However, this approach is not so effective in our experiments and thus it is not included as a benchmark in this paper. Lately, Zhe et al. [51] propose a Bayesian nonparametric tensor decomposition model that captures temporal correlations between interacting events. However, this approach is not directly applicable to continuous multivariable time series because it focuses on discrete events and captures temporal correlations between the occurrences of events.

3 METHODOLOGY

3.1 Imputation Framework

In many predictive tasks on temporal clinical data, time series are oftern aligned into the same-length sequences to derive more robust patient phenotypes through matrix decomposition or discover feature groups by applying sequence/graph mining techniques [29, 47, 48]. We model this assumption. In this work, we use

tensor representation, in which patients have the same number of time points. We represent the data collected from P patients with V laboratory tests and B time points as two 3D tensors $\mathcal{X} \in \mathbb{R}^{P\times V\times B}$ and $\mathcal{T} \in \mathbb{R}^{P\times V\times B}$, shown in Figure 1. Each laboratory test measurement $x_{p,v,b}$ is stored in the measurement tensor $\mathcal{X}$. Each time $t_{p,v,b}$, when $x_{p,v,b}$ is measured, is stored in the time tensor $\mathcal{T}$.

Table 1 lists main symbols we use throughout the paper. Missing values in the measurement tensor are denoted as $x^{mis}_{p,v,b}$, showing that the value of test v at time index b for patient p is missing. Correspondingly, $x^{obs}_{p,v,b}$ denotes an observed value. The time tensor $\mathcal{T}$ is complete, since we only collect patient records at the time when at least one laboratory test measurement is available. We assume we know the "prescribed" time a missing measurement should have been taken. In the matrix $x_{:,:,b}$ at time index b, the measurement time $t_{p,v,b}$ and $t_{q,v,b}$ can be different when $p \neq q$, whereas for a given patient p, we have $t_{p,v,b} = t_{p,u,b}$ for $v, u \in [1:V]$. That is, all tests for a particular patient are taken at the same time.

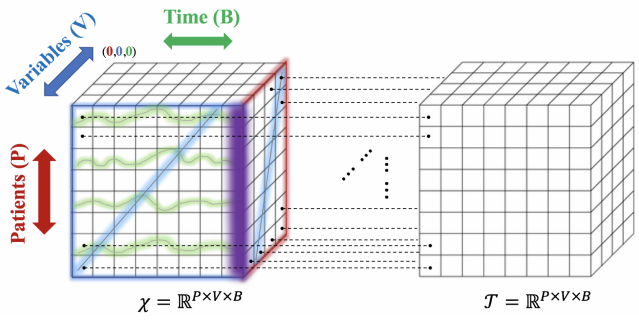


Figure 1: Measurement and time tensor. An example of the inputs and output of the mixture model $Mix_{V,B}$ is shown as the colorful fiber and matrices. In $Mix_{V,B}$, the target output is $x_{:,V,B}$ shown in purple and the inputs are $x_{:,-V,B}$ and $x_{:,V,-B}$ shown in red and blue matrices, respectively, excluding the purple fiber. We model the output with a mixture model, where we train a linear regression on the red matrix and train GPs or/and another linear regression on the blue matrix.

Disregarding the temporal dimension, the imputation problem is well studied. If one dimension is time, in order to apply imputation methods that are not designed for time series, we need to disregard the temporal aspect or ignore temporal correlations of the data. However, temporal trajectories can reveal patient's underlying pathophysiological evolution, modeling which can help to better estimate missing values. For the reason that both cross-sectional information and temporal aspects can impact the estimation of missing measurements, we explore mixture models, which are composed of several base models through either a cross-sectional or temporal view. We introduce these base models in Section 3.2.

In our imputation framework, a mixture model is trained for each variable and time index. We use $Mix_{v,b}$ to denote the mixture model to impute missing values of variable v at time index b. The missing values $x^{mis}_{:,v,b}$ in the fiber $x_{:,v,b}$ are imputed by the optimized $Mix_{v,b}$. In each iteration of the algorithm, it is assumed

Table 1: Main symbols and definitions

Symbol	Definition
$\mathcal{X}, \mathcal{T}$	Measurement and time tensor
$x_{p,v,-b}$	Measurements in fiber $x_{p,v,:}$ excluding $x_{p,v,b}$
$t_{p,v,-b}$	Times in fiber $t_{p,v,:}$ excluding $t_{p,v,b}$
$Mix_{v,b}$	Mixture model for v and b.
$V_{v,b}$	Concatenation of $x_{:,-v,b}$ and $x_{:,v,-b}$
$\mathcal{N}(\boldsymbol{\mu}^{(k)}_{v,b}, \Sigma^{(k)}_{v,b})$	The kth prior multivariate normal distribution in $Mix_{v,b}$
$m^G(\cdot)$, $\Sigma^G(\cdot)$	Predictive mean and variance of a GP model
$\gamma_{v,b}$	The set of all trainable parameters of $Mix_{v,b}$

that all other values are known and only $x^{mis}_{:,v,b}$ is imputed. This is wrapped in an outer loop. We call this procedure simple-pass tensor imputation, i.e., one pass through all v, b. Since several simple-pass tensor imputations are conducted, our approach is also considered as an iterative imputation [12], which can also be regarded as a sampling-based approach where a Gibbs sampler is used to approximate convergence. The convergence of iterative imputation methods can be quite fast with a few iterations [5].

In detail, the iterative imputation approaches start by replacing all missing data with values from simple guesses; we fill in all missing values with initial estimates by taking random draws from observed values. This procedure is called an initial imputation. Then we perform iterative tensor imputation on each copy separately. The training procedure and imputation for fibers are introduced in Section 3. We also rely on the concept of multiple imputations, where several iterative imputations are performed and the imputed values are averaged at the end. Each iterative imputation starts with a different iterative imputation tensor and/or uses a different order of v,b.

In summary, the algorithm creates M different copies $\mathcal{X}^1, \ldots, \mathcal{X}^M$ of $\mathcal{X}$, each one filled with different random $\mathcal{X}^{mis}$. For each $i = 1, \ldots, M$, we then perform K simple-pass tensor imputations. Each simple-pass has a loop over all v,b, which uses $Mix_{v,b}$ to adjust $\mathcal{X}^{i,mis}$. At the end of the M imputations, $\mathcal{X}^{i,mis}$ are averaged across all i to yield the final imputed tensor.

The whole imputation process involves $M\times K\times V\times B$ imputation models. We next first focus on the base models behind $Mix_{v,b}$ and then on the actual mixture model.

3.2 Base Models

Our mixture models are composed of three components that are derived from two base models, linear regression and Gaussian processes. One component consists of GP models and the other two components are linear models through two different views of the measurement tensor. Through a cross-sectional view, the tensor can be considered as a vector of patient-by-variable matrices at different time indices. Through a temporal view, we can view the tensor as a vector of patient-by-time matrices for different variables.

3.2.1 Linear model through cross-sectional view. We can view the measurement tensor $\mathcal{X}$ as a vector of patient-by-variable matrices. On the slice $x_{:,:,b}$, we use a linear regression model to fit the target

variable v as a function of the other variables except v. The target values $x_{:,v,b}$ are modeled as

$$x_{:,v,b} = x_{:,-v,b}\beta^{(1)}_{v,b} + \epsilon^{(1)}_{v,b}, \epsilon^{(1)}_{v,b} \sim \mathcal{N}(0, {\sigma^{(1)}_{v,b}}^2 I) \tag{1}$$

where $\beta^{(1)}_{v,b}$ is the column vector of coefficients and $\sigma^{(1)}_{v,b}$ is the standard deviation of the error $\epsilon^{(1)}_{v,b}$, regarding to the regression model through cross-sectional view for variable v and time index b.

The likelihood distribution of $x_{:,v,b}$ is then given by

$$x_{:,v,b}|x_{:,-v,b}, \beta^{(1)}_{v,b}, {\sigma^{(1)}_{v,b}}^2 \sim \mathcal{N}(x_{:,-v,b}\beta^{(1)}_{v,b}, {\sigma^{(1)}_{v,b}}^2 I). \tag{2}$$

The training data consists of observed target values $(x_{p,v,b})_{p\in P^{tr}_{v,b}}$ and input data $(x_{p,-v,b})_{p\in P^{tr}_{v,b}}$, where $P^{tr}_{v,b}$ is the training patient set and includes p only if $x_{p,v,b}$ is observed.

3.2.2 Linear model through temporal view. In addition to the cross-sectional view, we can also view the measurement tensor $\mathcal{X}$ as a vector of patient-by-time matrices. On matrix $x_{:,v,:}$, we use linear regression to model the measurements at time index b against those at other indices.

The target values $x_{:,v,b}$ are modeled as

$$x_{:,v,b} = x_{:,v,-b}\beta^{(2)}_{v,b} + \epsilon^{(2)}_{v,b}, \epsilon^{(2)}_{v,b} \sim \mathcal{N}(0, {\sigma^{(2)}_{v,b}}^2 I) \tag{3}$$

where $\beta^{(2)}_{v,b}$ is the column vector of coefficients and $\sigma^{(2)}_{v,b}$ is the standard deviation of the error $\epsilon^{(2)}_{v,b}$, regarding the linear regression model though temporal view for variable v and time index b.

The likelihood distribution of $x_{:,v,b}$ is given by

$$x_{:,v,b}|x_{:,v,-b}, \beta^{(2)}_{v,b}, {\sigma^{(2)}_{v,b}}^2 \sim \mathcal{N}(x_{:,v,-b}\beta^{(2)}_{v,b}, {\sigma^{(2)}_{v,b}}^2 I). \tag{4}$$

The training data consists of observed target values $(x_{p,v,b})_{p\in P^{tr}_{v,b}}$ and input data $(x_{p,v,-b})_{p\in P^{tr}_{v,b}}$.

3.2.3 Gaussian processes through temporal view. Gaussian processes are commonly used to capture trajectories of variables, thus used in our mixture model to capture temporal correlations. Through the same temporal view as introduced above, on matrix $x_{:,v,:}$, we fit GPs on time series for each patient.

The target value $x_{p,v,b}$ is modeled as

$$\begin{aligned} x_{p,v,b} &= \mu_{p,v,b} + f(t_{p,v,b}), \\ f(t_{p,v,b}) &\sim \mathcal{GP}(0, \mathcal{K}(t_{p,v,b}, t_{p,v,b'})) \end{aligned} \tag{5}$$

where $\mu_{p,v,b}$ is the overall mean of the model, $f(\cdot)$ is a Gaussian process with mean of 0 and a covariance matrix $\mathcal{K}(t,t')$ of time pairs (t,t'). Then the likelihood distribution of $x_{p,v,b}$ is written as

$$\begin{aligned} x_{p,v,b}|\alpha_{p,v,b} &\sim \mathcal{N}(m^G(\alpha_{p,v,b}), \Sigma^G(\alpha_{p,v,b})) \\ \alpha_{p,v,b} &= (\theta_{v,b}, x_{p,v,-b}, t_{p,v,-b}) \end{aligned} \tag{6}$$

where $\theta_{v,b}$ are the kernel parameters of the GP models, and the predictive mean and variance are given by $m^G(\cdot)$ and $\Sigma^G(\cdot)$; see more details in Appendix B. For a certain v and b, all GP models share the same kernel parameters $\theta_{v,b}$.

3.3 The Mixture Model

Given the likelihood distribution of all three components, we model the joint mixture distribution, regarding the variable v and time index b, in the following way

$$\begin{aligned} &p(x_{:,v,b}, V_{v,b}) \\ &= \pi^{(1)}_{v,b}\mathcal{N}(V_{v,b}|\boldsymbol{\mu}^{(1)}_{v,b}, \Sigma^{(1)}_{v,b})\mathcal{N}(x_{:,v,b}|x_{:,-v,b}\beta^{(1)}_{v,b}, {\sigma^{(1)}_{v,b}}^2 I) \\ &+ \pi^{(2)}_{v,b}\mathcal{N}(V_{v,b}|\boldsymbol{\mu}^{(2)}_{v,b}, \Sigma^{(2)}_{v,b})\mathcal{N}(x_{:,v,b}|x_{:,v,-b}\beta^{(2)}_{v,b}, {\sigma^{(2)}_{v,b}}^2 I) \\ &+ \pi^{(3)}_{v,b}\mathcal{N}(V_{v,b}|\boldsymbol{\mu}^{(3)}_{v,b}, \Sigma^{(3)}_{v,b})\mathcal{N}(x_{:,v,b}|m^G(\boldsymbol{\alpha}_{v,b}), \mathrm{diag}(\Sigma^G(\boldsymbol{\alpha}_{v,b}))) \end{aligned} \tag{7}$$

where we define $V_{v,b} = [x_{:,-v,b}\ x_{:,v,-b}]$ and $\boldsymbol{\alpha}_{v,b} = (\alpha_{p,v,b})_{p\in P}$. This model can be interpreted as the joint distribution between observed data $V_{v,b}$ and missing values $x_{:,v,b}$, consisting of a mixture of three distributions. The first one $p_1(x_{:,v,b}, V_{v,b})$ is modeled as

$$\begin{aligned} &p_1(x_{:,v,b}, V_{v,b}) \\ &\quad = p(x_{:,v,b}|V_{v,b})p(V_{v,b}) \\ &\quad = \mathcal{N}(x_{:,v,b}|\mu_1(V_{v,b}), \sigma_1(V_{v,b}))\mathcal{N}(V_{v,b}|\boldsymbol{\mu}^{(1)}_{v,b}, \Sigma^{(1)}_{v,b}) \\ &\quad = \mathcal{N}(x_{:,v,b}|x_{:,-v,b}\beta^{(1)}_{v,b}, {\sigma^{(1)}_{v,b}}^2 I)\mathcal{N}(V_{v,b}|\boldsymbol{\mu}^{(1)}_{v,b}, \Sigma^{(1)}_{v,b}) \end{aligned} \tag{8}$$

the remaining two follow the same logic.

By marginalizing over $x_{:,v,b}$, the prior probability distribution $p(V_{v,b})$ is written as

$$p(V_{v,b}) = \sum_{k=1}^{3} \pi^{(k)}_{v,b}\mathcal{N}(V_{v,b}|\boldsymbol{\mu}^{(k)}_{v,b}, \Sigma^{(k)}_{v,b}) \tag{9}$$

which is a mixture of Gaussians. It also follows that

$$\begin{aligned} &p(x_{:,v,b}|V_{v,b}) \\ &= \frac{p(x_{:,v,b}, V_{v,b})}{p(V_{v,b})} \\ &= \frac{\pi^{(1)}_{v,b}\mathcal{N}(V_{v,b}|\delta^{(1)}_{v,b})}{\sum_{j=1}^{3}\pi^{(j)}_{v,b}\mathcal{N}(V_{v,b}|\delta^{(j)}_{v,b})}\mathcal{N}(x_{:,v,b}|x_{:,-v,b}\beta^{(1)}_{v,b}, {\sigma^{(1)}_{v,b}}^2 I) \\ &+ \frac{\pi^{(2)}_{v,b}\mathcal{N}(V_{v,b}|\delta^{(2)}_{v,b})}{\sum_{j=1}^{3}\pi^{(j)}_{v,b}\mathcal{N}(V_{v,b}|\delta^{(j)}_{v,b})}\mathcal{N}(x_{:,v,b}|x_{:,v,-b}\beta^{(2)}_{v,b}, {\sigma^{(2)}_{v,b}}^2 I) \\ &+ \frac{\pi^{(3)}_{v,b}\mathcal{N}(V_{v,b}|\delta^{(3)}_{v,b})}{\sum_{j=1}^{3}\pi^{(j)}_{v,b}\mathcal{N}(V_{v,b}|\delta^{(j)}_{v,b})}\mathcal{N}(x_{:,v,b}|m^G(\boldsymbol{\alpha}_{v,b}), \mathrm{diag}(\Sigma^G(\boldsymbol{\alpha}_{v,b}))) \end{aligned} \tag{10}$$

where we define $\delta^{(k)}_{v,b} = (\boldsymbol{\mu}^{(k)}_{v,b}, \Sigma^{(k)}_{v,b})$.

We train our mixture model on observed target values by maximizing the log likelihood of the joint mixture distribution

$$\hat{\gamma}_{v,b} = \arg\max_{\gamma_{v,b}} \ln p(x^{obs}_{:,v,b}, V^{tr}_{v,b}; \gamma_{v,b})$$

where $V^{tr}_{v,b}$ is the training input data and defined as the concatenation of $(x_{p,-v,b})_{p\in P^{tr}_{v,b}}$ and $(x_{p,v,-b})_{p\in P^{tr}_{v,b}}$, and $\gamma_{v,b}$ is the set of all trainable parameters

$$\beta^{(1)}_{v,b}, {\sigma^{(1)}_{v,b}}^2, \beta^{(2)}_{v,b}, {\sigma^{(2)}_{v,b}}^2, \theta_{v,b} \text{ and } \pi^{(k)}_{v,b}, \boldsymbol{\mu}^{(k)}_{v,b}, \Sigma^{(k)}_{v,b} \text{ for } k = 1, 2, 3.$$

After training, the missing values are imputed using individualized mixing weights that are derived from the conditional distribution. The missing value $x^{mis}_{p,v,b}$ is imputed by

$$\Pi^{(1)}_{p,v,b} x_{p,-v,b} \hat{\beta}^{(1)}_{v,b} + \Pi^{(2)}_{p,v,b} x_{p,v,-b} \hat{\beta}^{(2)}_{v,b} + \Pi^{(3)}_{p,v,b} m^G(\hat{\alpha}_{p,v,b})$$

where the individualized mixing weight of the kth component for patient p, variable v and time index b is defined as

$$\Pi^{(k)}_{p,v,b} = \frac{\hat{\pi}^{(k)}_{v,b} \mathcal{N}(V_{p,v,b} | \hat{\boldsymbol{\mu}}^{(k)}_{v,b}, \hat{\Sigma}^{(k)}_{v,b})}{\sum_{j=1}^{3} \hat{\pi}^{(j)}_{v,b} \mathcal{N}(V_{p,v,b} | \hat{\boldsymbol{\mu}}^{(j)}_{v,b}, \hat{\Sigma}^{(j)}_{v,b})}, k = 1, 2, 3 \quad (11)$$

where $V_{p,v,b}$ is the observed data and defined as the concatenation of $x_{p,-v,b}$ and $x_{p,v,-b}$.

3.4 Mixture Parameter Estimation

Let $\ell(\gamma_{v,b})$ be the log likelihood $\ln p(x^{obs}_{:,v,b}, V^{tr}_{v,b}; \gamma_{v,b})$. Explicitly maximizing $\ell(\gamma_{v,b})$ is hard. Instead, we use the EM algorithm to repeatedly construct a lower-bound on $\ell(\gamma_{v,b})$ and then optimize that lower-bound $\mathcal{L}(\gamma_{v,b})$. We first define a latent indicator variable $q_{v,b} \in \{1, 2, 3\}$ that specifies which mixing component that data points come from. Then we use Jensen's inequality to get the lower-bound $\mathcal{L}(\gamma_{v,b})$, which is given by

$$\begin{aligned}
&\mathcal{L}(\gamma_{v,b}) \\
&= \sum_{p \in P^{tr}_{v,b}} \sum_{q_{v,b}} Q_p(q_{v,b}) \ln \frac{p(x_{p,v,b}, V_{p,v,b}, q_{v,b}; \gamma_{v,b})}{Q_p(q_{v,b})} \\
&\leq \sum_{p \in P^{tr}_{v,b}} \ln \sum_{k=1}^{3} p(q_{v,b} = k) p(x_{p,v,b}, V_{p,v,b}; \gamma_{v,b} | q_{v,b} = k) \\
&= \ell(\gamma_{v,b})
\end{aligned} \quad (12)$$

where

$$Q_p(q_{v,b} = k) = \frac{p(q_{v,b} = k) p(x_{p,v,b}, V_{p,v,b} | q_{v,b} = k)}{\sum_{j=1}^{3} p(q_{v,b} = j) p(x_{p,v,b}, V_{p,v,b} | q_{v,b} = j)}. \quad (13)$$

In (12) and (13), the marginal distribution $p(q_{v,b} = k)$ over $q_{v,b}$ is specified by the mixing coefficients $\pi^{(k)}_{v,b} = p(q_{v,b} = k)$. We can view $Q_p(q_{v,b} = k)$ as the responsibility that component k of the mixture model $Mix_{v,b}$ takes to "explain" $x_{p,v,b}$. We use the standard EM algorithm to maximize the lower-bound $\mathcal{L}(\gamma_{v,b})$; see more details about the estimation of the parameters in Appendix A.

3.5 Special Cases and An Ensemble Model

Our imputation model provides flexibility in changing base models. The mixture model mentioned above consists of three base models. Two of them are applied through the same temporal view of the measurement tensor. We can drop one of the models through temporal view to yield a new mixture model. For example, if the GP model through temporal view is removed from the mixture model for variable v and time point b, the new mixture model is now a mixture of two linear models.

Each mixture model can be a mixture of all three base models (the Linear-Linear-GP [LLG] model), a mixture of two linear models (the Linear-Linear [LL] model) or a mixture of linear and GP models.

To further improve the imputation quality, we build an ensemble model (En-LLG) by allowing each mixture model to be a mixture of either two linear models or all three base models.

In En-LLG, for each variable and time index, we train the mixture model with two linear models and the mixture model with all three base models, and then select the one with less training error as the final mixture model, which is used to do imputation. Although we train our mixture models by maximizing the likelihood, we use the absolute training error as the selection criteria because the likelihood of the mixture model with two linear models is not in the same scale as the mixture model with three base models.

4 EXPERIMENTAL SETUP

4.1 Real-world Datasets

We collect two real-world datasets from the Medical Information Mart for Intensive Care (MIMIC-III) database [20] and the Northwestern Medicine Enterprise Data Warehouse (NMEDW). Each dataset contains inpatient test results from 13 laboratory tests. These tests are quantitative and frequently measured on hospital inpatients. They are the same as those used in [28] in their imputation study. We organize the data by unique admissions. We distinguish multiple admissions of the same patient. Each admission consists of time series of the 13 laboratory tests.

In both MIMIC-III and NMEDW datasets, the length of time series varies across admissions. To apply our imputation models on these datasets, we truncate time series so that they have the same length. The length is the average number of time points across all admissions. Before truncating, the average number of time points in MIMIC-III dataset is 11. We first exclude admissions that have less than 11 time points, and then we truncate time series by removing measurements taken after the 11-th time point. We also exclude admissions that contain time series that have no observed values. Our MIMIC-III dataset includes 26,154 unique admissions and the missing rate is about 28.71%. The same data collection procedure is applied on the NMEDW dataset where we end up with 13,892 unique admissions that have 7 time points, as the average number of time points of patients in NMEDW is 7. The missing rate of the NMEDW dataset is 24.22%.

4.2 Synthetic Datasets

We create synthetic datasets to explore and discover the properties of the data that might benefit our models and/or comparison models. In synthetic datasets, we augment the correlation between measurements and times. We do not augment correlations by imposing strong constraints on time series where closer measurements have closer values. Instead, we generate synthetic times by altering real times so that the constraints in synthetic data are "slightly" stronger than real-world data. We also introduce a scaling factor d to control the strength of the constraints in synthetic data.

The synthetic datasets are generated based on the real-world MIMIC-III dataset. We move two consecutive times of a time series closer, if the relative difference $\Delta\tilde{x}$ in two consecutive measurements is smaller than the relative difference $\Delta\tilde{t}$ in two consecutive times. The relative differences $\Delta\tilde{x}$ and $\Delta\tilde{t}$ of a time series are given

Table 2: Overall MASE by dataset and imputation model. The bold numbers are the significantly best values among all imputation models.

Dataset	MICE	GP	3D-MICE	LL	En-LLG
Real-world MIMIC (d=0)	0.11763	0.13072	0.11186	0.09304	**0.09285**
Synthetic MIMIC (d=0.5)	0.11573	0.10466	0.09087	0.08612	**0.08448**
Synthetic MIMIC (d=1)	0.11561	0.09220	0.07715	0.08427	**0.07538**
NMEDW	0.13718	0.18353	0.13624	0.11600	**0.11589**

Table 3: MASE on the real-world MIMIC dataset by variable and imputation model. The bold numbers are the best values among all imputation models.

Variable	MICE	GP	3D-MICE	LL	En-LLG
Chloride	0.10575	0.12993	0.10836	0.08664	**0.08603**
Potassium	0.10997	0.11533	0.10822	0.09453	**0.09442**
Bicarbonate	0.12275	0.13196	0.11984	0.10302	**0.10254**
Sodium	0.10138	0.12525	0.10727	**0.08787**	0.08807
Hematocrit	0.06558	0.11436	0.06726	0.05486	**0.05482**
Hemoglobin	0.05772	0.14168	0.06301	0.05117	**0.05103**
MCV	0.13474	0.14215	0.13340	**0.11634**	0.11657
Platelets	0.14236	0.13855	0.12815	0.10090	**0.10070**
WBC count	0.14068	0.13963	0.13060	0.10934	**0.10913**
RDW	0.15836	0.14592	0.13897	0.11340	**0.11340**
Blood urea nitrogen	0.15189	0.12358	0.11814	0.09479	**0.09410**
Creatinie	0.13212	0.13341	0.12217	0.10067	**0.10014**
Glucose	0.11794	0.12491	0.11921	**0.10493**	0.10501

by

$$\Delta\tilde{x}_i = \frac{|x_i - x_{i-1}|}{\sum_{i=2}^{B} |x_i - x_{i-1}|}$$
$$\Delta\tilde{t}_i = \frac{|t_i - t_{i-1}|}{\sum_{i=2}^{B} |t_i - t_{i-1}|}.$$

The scaling factor $d \in (0, 1)$ controls how farther/closer we move times. If $d = 0$, we do not move times. In other words, the synthetic dataset at $d = 0$ is the same as the real-world MIMIC-III dataset. As d increases, stronger constraints are introduced to synthetic data. The synthetic time t' for a time series is generated as follows:

$$t_i' = \begin{cases} t_1, & \text{if } i = 1 \\ t_i + \sum_{j=2}^{i} [d(\Delta\tilde{x}_j - \Delta\tilde{t}_j)S], & \text{otherwise} \end{cases}$$
$$S = \sum_{j=2}^{B} (|t_j - t_{j-1}|).$$

If a time series has missing values, we first calculate the synthetic times for the observed measurements. Then we perform a linear interpolation between real times and synthetic times for observed measurements to generate synthetic times for missing measurements.

4.3 Evaluation of Imputation Quality

We randomly mask 20% observed measurements in a data set as missing and treat the masked values as the test set. The remaining observed values are used in training. We impute originally missing and masked values together, and compare the imputed values with the ground truth for masked data to evaluate imputation performance.

We use Mean Absolute Scaled Error (MASE) [19] to measure the quality of imputation on the test set. MASE is a scale-free measure of the accuracy of predictions and recommended by Hyndman et al [10, 19] to measure the accuracy of predictions for series. In this work, we calculate MASE for all tests (variables) and take a weighted average, according to the number of masked values of a variable, to get an overall MASE per dataset.

Let $mask_{p,v}$ be the set of cardinality $I_{p,v}$ of all time indices that have been masked for patient p and variable v. Also let $Y_{p,v} = (x^{obs}_{p,v,j})_j$ be the sequence of length $J_{p,v}$ of all observed values for patient p and variable v, and let $\tilde{x}_{p,v,i}$ represent the imputed value. The MASE for variable v is defined as

$$MASE(v) = \frac{1}{\sum_{\bar{p}} I_{\bar{p},v}} \sum_{p} \frac{\sum_{i \in mask_{p,v}} |\tilde{x}_{p,v,i} - x^{obs}_{p,v,i}|}{\frac{J_{p,v}}{J_{p,v}-1} \sum_{j=2}^{J_{p,v}} |Y_{p,v,j} - Y_{p,v,j-1}|}.$$

To show the effectiveness of our imputation models, we compare the MASE scores of our models (LL and En-LLG) with other three imputation methods: (a) MICE with 100 imputations, where the average of all imputations are used for evaluation; (b) the pure Gaussian processes, where a GP model is fitted to the observed data of each time series using GPfit [30] in R and missing values are replaced with the predictions from the fitted GP models; (c) 3D-MICE, a state-of-the-art imputation model [28] for which we

obtain their code and adapt it to account for our use of the tensor representation. To tune hyperparameters if any in these models, we mask out 20% observed measurements in the training set as a validation set and tune hyperparameters on the validation set.

We run our En-LLG model for $M = 3$ multiple imputations with $K = 2$ iterations and run the LL model with more multiple imputations (M=5) and more iterations (K=5). We run more iterations of the LL model, because the LL model takes less time than the En-LLG model. For 3D-MICE, we set the number of multiple imputation to 40, instead of 100 that is suggested in [28], to balance the performance and running time.

5 RESULTS

5.1 Performance Comparison

Table 2 and Figure 2 compare the 5 imputation models on 4 datasets using MASE. Table 2 shows the overall MASE score of each imputation model on all datasets. Figure 2 provides a comparison for all imputation models in the MASE score over 3D-MICE by showing the percentage deviation against 3D-MICE. We select 3D-MICE since it is the best benchmark model. We observe that our En-LLG model outperforms all comparison models on all 4 datasets. The LL model outperforms MICE and GP on all datasets, and outperforms 3D-MICE on all but the synthetic MIMIC (d=1) dataset. The En-LLG model is significantly better than the second best model (p=.001, permutation test with 1000 replicates) on all 4 datasets.

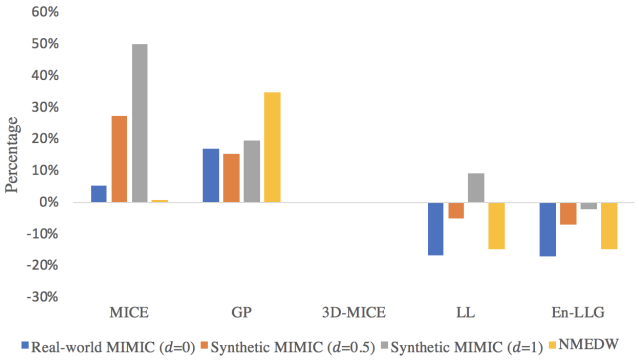


Figure 2: Percentage deviation of MASE score against 3D-MICE

Table 3 shows a variable-wise comparison of the imputation models on the real-world MIMIC (d=0) dataset. Our two imputation models outperform three comparison models on all variables. The En-LLG model is better than the LL model on most variables. All models except GP achieve a much lower error on Hematocrit and Hemoglobin than on other variables. The reason is that these two variables are highly correlated. Those methods that capture the correlation between variables can reasonably infer missing values for Hematocrit from observed measurements of Hemoglobin, and vice versa. Compared to MICE, our models achieve even lower errors on these two variables, which indicates that temporal correlations captured by our models help to make better estimation of missing values, even when there is a more dominant cross-sectional correlation.

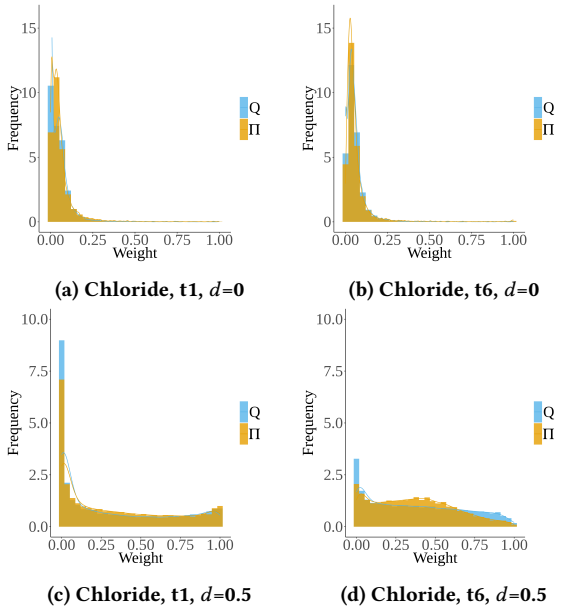


Figure 3: A comparison between the individualized mixing weights Π and the optimized responsibilities Q that the GP component should take to "explain" observed measurements. The plots are from the real-world MIMIC (d=0) and syntehtic MIMIC (d=0.5) dataset, and for the mixture models of Chloride at time point 1 and 6. The distributions of the optimized responsibilities are shown in blue and the distributions of individualized mixing weights are in yellow.

As shown in Table 2, all models benefit from the increment of d, the scaling factor when generating synthetic data. The reason is that all models take into account temporal aspects and the measurements in the synthetic time series have stronger temporal correlations as d increases. The reason that MICE also benefits from the temporal correlations is that we include time as a feature in addition to variable features. We also try to exclude times in MICE, however, experiments show that MICE performs better when times are included. As shown in Table 2, GP and 3D-MICE benefit the most as d increases from 0 to 1, MICE benefits the least and our models (LL and En-LLG) are in the middle. The LL model is outperformed by 3D-MICE when d increases to 1. However, En-LLG shows its robustness to the variation of d in our current experimental settings.

5.2 Individualized Weights

By introducing individualized mixing weights Π defined in (11), we improve the performance of our En-LLG model in the MASE score from 0.08351 to 0.07538, an improvement of 9.73% compared against the model where each mixture component has a fixed weight for all patient cases. The reason individualized weights are better than fixed weights in our model might be that they better approximate the responsibilities.

In training, we can optimize the responsibility a component should take to "explain" an observed target value $x_{p,v,b}$ for $p \in P^{tr}_{v,b}$. These correspond to Q in (13). However, when making inference,

we can not calculate the responsibility each component should take to "explain" missing values, because responsibilities depend on observed target values, according to (13). We have to use Π in (11), individualized mixing weights. In a standard mixture model, we could use $\pi_{v,b}^{(k)}$, which is the average of responsibilities of the kth component across all training patients, as a fixed weight that the kth component should contribute to impute missing values $x_{:,v,b}^{mis}$ for all test patients. However, patient time series can be very different and the confidence of predictions by the GP component can vary largely across different patient cases. In our mixture model, therefore, a fixed weight can not reflect such variation in prediction confidence.

We shall view an individualized mixing weight as an approximation of how much responsibility a component should take to impute the missing value for a particular patient case. It is tailored for each patient. As defined in (11), the individualized mixing weights only depend on the inputs, therefore, we can calculate them when making inferences on the test set.

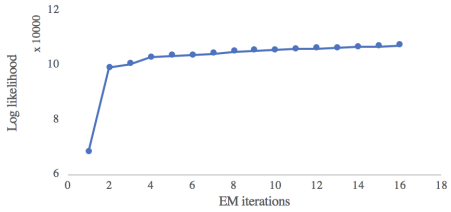
In Figure 3, we plot the distribution of the individualized weights Π of the GP component in the training set and compare it with the distribution of the optimized responsibility values Q. The responsibilities the GP component should take can vary a lot in different patient cases, especially on the synthetic dataset, which implies that it is more reasonable for patients to get individualized mixing weights than a fixed weight. We also observe that the individualized mixing weights reasonably mimic the distribution of the optimized responsibilities on the training set. The improvement of our model on the test set attests that the individualized weights approximate the responsibilities better than fixed weights.

In addition, 3D-MICE also assigns individualized weights at the same level of granularity as our models, however, weights in 3D-MICE are calculated only based on the deviations of the cross-sectional imputation and the temporal imputation, and are not optimized. As shown in Table 3, the improvement of our models over 3D-MICE implies that our models provide a more accurate weighting solution when combining the cross-sectional and temporal imputation.

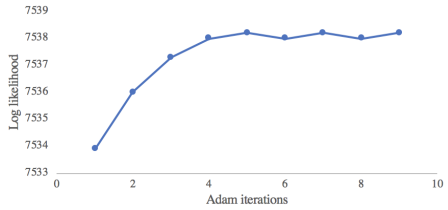
5.3 Time Complexity

We compare the running times of our proposed models and all comparison models on the real-world MIMIC dataset. All models ran on the same Linux server and each ran in parallel with 20 cores. The LL model (taking 4.2 hours) and GP (1.1 hours) are the two fastest models, the En-LLG model (109.5 hours) and 3D-MICE (156.1 hours) are the two slowest models and MICE (77.5 hours) is in the middle.

The LL model is much faster than the En-LLG model because we can explicitly calculate the estimates of parameters for the mixture model with only linear components. However, it is hard to directly calculate the parameter estimates of the GP component. In En-LLG, we use the Adam optimizer to update the parameter estimates of the GP component in each EM iteration. In our experiments, the EM algorithm and the Adam optimization procedure converge in a few iterations. As shown in Figure 4(a), the increment speed of optimizing the log likelihood of the mixture model decreases dramatically after a few EM iterations. Figure 4(b) shows that the



(a) EM



(b) Adam

Figure 4: Log likelihood of the whole mixture model in EM and log likelihood of the GP component in Adam for Chloride at time point 1.

log likelihood with respect to the GP component converges within 10 Adam iterations. We observe a similar convergence property in other mixture models.

We notice that our models can have an overfitting problem due to the large amount of trainable parameters. We observe that, in many mixture models, the log likelihood keeps increasing, however, the imputation error stops decreasing after a few EM iterations. We alleviate the overfitting problem by terminating the EM algorithm earlier. We terminate it when the mean absolute error on the training set stops decreasing. In our experiment, this strategy is better than a vanilla ridge regularization for handling the overfitting problems. In addition to the improvement of imputation quality, this strategy also helps to reduce the running time of our model, as the EM algorithm stops earlier with this strategy. Although we alleviate the overfitting problem, we expect further improvements in the performance of our imputation model with better strategies for handling overfitting.

6 CONCLUSIONS

We present and demonstrate mixture-based imputation models for multivariable clinical time series. Our models can capture both cross-sectional and temporal correlations in time series. We integrate Gaussian processes with mixture models and introduce individualized mixing weights to further improve imputation accuracy. We show that our best model can provide more accurate imputation than MICE, GP and 3D-MICE, a state-of-the-art imputation model that integrates cross-sectional and longitudinal imputation. Although in this work our models are tested on inpatient clinical data, they can also be applied to other multivariable time series data in healthcare and other domains with necessary adaptation.

ACKNOWLEDGMENTS

This work is supported in part by NIH grant R21LM012618.

REFERENCES

[1] James L Arbuckle. 1996. Full information estimation in the presence of incomplete data. In *Advanced Structural Equation Modeling: Issues and Techniques*, George A. Marcoulides and Randall E. Schumacker (Eds.). Vol. 243. 277.

[2] Mohammad Taha Bahadori, Qi Rose Yu, and Yan Liu. 2014. Fast multivariate spatio-temporal analysis via low rank tensor learning. In *Advances in Neural Information Processing Systems*. 3491–3499.

[3] Faraj Bashir and Hua-Liang Wei. 2017. Handling missing data in multivariate time series using a vector autoregressive model-imputation (VAR-IM) algorithm. *Neurocomputing* (2017).

[4] Roger L Brown. 1994. Efficacy of the indirect approach for estimating structural equation models with missing data: A comparison of five methods. *Structural Equation Modeling: A Multidisciplinary Journal* 1, 4 (1994), 287–316.

[5] Stef Buuren and Karin Groothuis-Oudshoorn. 2011. mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software* 45, 3 (2011).

[6] Yongjie Cai, Hanghang Tong, Wei Fan, Ping Ji, and Qing He. 2015. Facets: Fast comprehensive mining of coevolving high-order time series. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 79–88.

[7] Jeng-Min Chiou, Yi-Chen Zhang, Wan-Hui Chen, and Chiung-Wen Chang. 2014. A functional data approach to missing value imputation and outlier detection for traffic flow data. *Transportmetrica B: Transport Dynamics* 2, 2 (2014), 106–129.

[8] Yi Deng, Changgee Chang, Moges Seyoum Ido, and Qi Long. 2016. Multiple imputation for general missing data patterns in the presence of high-dimensional data. *Scientific Reports* 6 (2016), 21689.

[9] Marko Filipović and Ante Jukić. 2015. Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidimensional Systems and Signal Processing* 26, 3 (2015), 677–692.

[10] Philip Hans Franses. 2016. A note on the mean absolute scaled error. *International Journal of Forecasting* 32, 1 (2016), 20–22.

[11] Hancheng Ge, James Caverlee, Nan Zhang, and Anna Squicciarini. 2016. Uncovering the spatio-temporal dynamics of memes in the presence of incomplete information. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 1493–1502.

[12] Andrew Gelman. 2004. Parameterization and Bayesian modeling. *J. Amer. Statist. Assoc.* 99, 466 (2004), 537–545.

[13] John W Graham. 2009. Missing data analysis: Making it work in the real world. *Annual Review of Psychology* 60 (2009), 549–576.

[14] Ofer Harel and Xiao-Hua Zhou. 2007. Multiple imputation for the comparison of two screening tests in two-phase Alzheimer studies. *Statistics in Medicine* 26, 11 (2007), 2370–2388.

[15] Yulei He, Recai Yucel, and Trivellore E Raghunathan. 2011. A functional multiple imputation approach to incomplete longitudinal data. *Statistics in Medicine* 30, 10 (2011), 1137–1156.

[16] Elizabeth E Holmes, Eric J Ward, and Kellie Wills. 2012. MARSS: Multivariate autoregressive state-space models for analyzing time-series data. *R Journal* 4, 1 (2012).

[17] Tomoaki Hori, David Montcho, Clement Agbangla, Kaworu Ebana, Koichi Futakuchi, and Hiroyoshi Iwata. 2016. Multi-task Gaussian process for imputing missing data in multi-trait and multi-environment trials. *Theoretical and Applied Genetics* 129, 11 (2016), 2101–2115.

[18] Chiu-Hsieh Hsu, Jeremy MG Taylor, Susan Murray, and Daniel Commenges. 2006. Survival analysis using auxiliary variables via non-parametric multiple imputation. *Statistics in Medicine* 25, 20 (2006), 3503–3517.

[19] Rob J Hyndman and Anne B Koehler. 2006. Another look at measures of forecast accuracy. *International Journal of Forecasting* 22, 4 (2006), 679–688.

[20] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data* 3 (2016), 160035.

[21] Stephanie Kliethermes and Jacob Oleson. 2014. A Bayesian approach to functional mixed-effects modeling for longitudinal data with binomial outcomes. *Statistics in Medicine* 33, 18 (2014), 3130–3146.

[22] Isaac S Kohane. 2015. Ten things we have to do to achieve precision medicine. *Science* 349, 6243 (2015), 37–38.

[23] Roderick Little and Hyonggin An. 2004. Robust likelihood-based analysis of multivariate data with missing values. *Statistica Sinica* (2004), 949–968.

[24] Minzhi Liu, Jeremy MG Taylor, and Thomas R Belin. 2000. Multiple imputation and posterior simulation for multivariate missing data in longitudinal studies. *Biometrics* 56, 4 (2000), 1157–1163.

[25] Yuanyuan Liu, Fanhua Shang, Licheng Jiao, James Cheng, and Hong Cheng. 2015. Trace norm regularized CANDECOMP/PARAFAC decomposition with missing data. *IEEE Transactions on Cybernetics* 45, 11 (2015), 2437–2448.

[26] Qi Long, Chiu-Hsieh Hsu, and Yisheng Li. 2012. Doubly robust nonparametric multiple imputation for ignorable missing data. *Statistica Sinica* 22 (2012), 149.

[27] Yuan Luo, Peter Szolovits, Anand S Dighe, and Jason M Baron. 2016. Using machine learning to predict laboratory test results. *American Journal of Clinical Pathology* 145, 6 (2016), 778–788.

[28] Yuan Luo, Peter Szolovits, Anand S Dighe, and Jason M Baron. 2017. 3D-MICE: integration of cross-sectional and longitudinal imputation for multi-analyte longitudinal clinical data. *Journal of the American Medical Informatics Association* 25, 6 (2017), 645–653.

[29] Yuan Luo, Yu Xin, Rohit Joshi, Leo A Celi, and Peter Szolovits. 2016. Predicting ICU mortality risk by grouping temporal trends from a multivariate panel of physiologic measurements. In *AAAI*. 42–50.

[30] Blake MacDonald, Pritam Ranjan, and Hugh Chipman. 2015. GPfit: An R package for fitting a Gaussian process model to deterministic simulator outputs. *Journal of Statistical Software* 64, 1 (2015), 1–23.

[31] Lihong Qi, Ying-Fang Wang, and Yulei He. 2010. A comparison of multiple imputation and fully augmented weighted estimators for Cox regression with missing covariates. *Statistics in Medicine* 29, 25 (2010), 2592–2604.

[32] Trivellore E Raghunathan, James M Lepkowski, John Van Hoewyk, and Peter Solenberger. 2001. A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology* 27, 1 (2001), 85–96.

[33] Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. 1989. Design and analysis of computer experiments. *Statist. Sci.* (1989), 409–423.

[34] Joseph L Schafer and John W Graham. 2002. Missing data: Our view of the state of the art. *Psychological Methods* 7, 2 (2002), 147.

[35] Joseph L Schafer and Recai M Yucel. 2002. Computational strategies for multivariate linear mixed-effects models with missing values. *Journal of Computational and Graphical Statistics* 11, 2 (2002), 437–457.

[36] Amrik Shah, Nan Laird, and David Schoenfeld. 1997. A random-effects model for multiple characteristics with possibly missing data. *J. Amer. Statist. Assoc.* 92, 438 (1997), 775–779.

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

[38] Yu-Sung Su, Andrew Gelman, Jennifer Hill, Masanao Yajima, et al. 2011. Multiple imputation with diagnostics (mi) in R: Opening windows into the black box. *Journal of Statistical Software* 45, 2 (2011), 1–31.

[39] Koh Takeuchi, Hisashi Kashima, and Naonori Ueda. 2017. Autoregressive tensor factorization for spatio-temporal predictions. *IEEE International Conference on Data Mining* (2017).

[40] Ryota Tomioka, Kohei Hayashi, and Hisashi Kashima. 2010. Estimation of low-rank tensors via convex optimization. https://arxiv.org/pdf/1010.0789.pdf

[41] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. 2001. Missing value estimation methods for DNA microarrays. *Bioinformatics* 17, 6 (2001), 520–525.

[42] Stef Van Buuren, Hendriek C Boshuizen, and Dick L Knook. 1999. Multiple imputation of missing blood pressure covariates in survival analysis. *Statistics in Medicine* 18, 6 (1999), 681–694.

[43] Akbar K Waljee, Ashin Mukherjee, Amit G Singal, Yiwei Zhang, Jeffrey Warren, Ulysses Balis, Jorge Marrero, Ji Zhu, and Peter DR Higgins. 2013. Comparison of imputation methods for missing laboratory data in medicine. *BMJ Open* 3, 8 (2013), e002847.

[44] Griffin M Weber, William G Adams, Elmer V Bernstam, Jonathan P Bickel, Kathe P Fox, Keith Marsolo, Vijay A Raghavan, Alexander Turchin, Xiaobo Zhou, Shawn N Murphy, et al. 2017. Biases introduced by filtering electronic health records for patients with "complete data". *Journal of the American Medical Informatics Association* 24, 6 (2017), 1134–1141.

[45] Raimond L Winslow, Natalia Trayanova, Donald Geman, and Michael I Miller. 2012. Computational medicine: Translating models to clinical care. *Science Translational Medicine* 4, 158 (2012), 158rv11–158rv11.

[46] Werner Wothke. 2000. Longitudinal and multi-group modeling with missing data. In *Modeling longitudinal and multiplegroup data: Practical issues, applied approaches, and specific examples*, Todd D. Little, Kai U. Schnabel, and Jrgen Baumert (Eds.). 219–240.

[47] Ye Xue, Diego Klabjan, and Yuan Luo. 2018. Predicting ICU readmission using grouped physiological and medication trends. *Artificial Intelligence in Medicine* (2018).

[48] Xi Yang, Yuan Zhang, and Min Chi. 2018. Time-aware subgroup matrix decomposition: Imputing missing data using forecasting events. In *2018 IEEE International Conference on Big Data*. IEEE, 1524–1533.

[49] Rose Yu, Dehua Cheng, and Yan Liu. 2015. Accelerated online low-rank tensor learning for multivariate spatio-temporal streams. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. 238–247.

[50] Guangyu Zhang and Roderick Little. 2009. Extensions of the penalized spline of propensity prediction method of imputation. *Biometrics* 65, 3 (2009), 911–918.

[51] Shandian Zhe and Yishuai Du. 2018. Stochastic nonparametric event-tensor decomposition. In *Advances in Neural Information Processing Systems*. 6855–6865.

[52] Jiayu Zhou, Fei Wang, Jianying Hu, and Jieping Ye. 2014. From micro to macro: Data driven phenotyping by densification of longitudinal electronic medical records. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 135–144.

A PRAMETER ESTIMATION IN EM

In the E (Expectation) step, we calculate the responsibilities $w_{p,v,b}^{(k)} = Q_p(q_{v,b} = k)$ for $p \in P_{v,b}^{tr}$ using the current values of the parameters in iteration j:

$$[C_{p,v,b}^{(1)}]^{(j)} = [\pi_{v,b}^{(1)}]^{(j)}[D_{p,v,b}^{(1)}]^{(j)}\mathcal{N}(x_{p,v,b}|x_{p,-v,b}[\beta_{v,b}^{(1)}]^{(j)}, [{\sigma_{v,b}^{(1)}}^2]^{(j)})$$

$$[C_{p,v,b}^{(2)}]^{(j)} = [\pi_{v,b}^{(2)}]^{(j)}[D_{p,v,b}^{(2)}]^{(j)}\mathcal{N}(x_{p,v,b}|x_{p,v,-b}[\beta_{v,b}^{(2)}]^{(j)}, [{\sigma_{v,b}^{(2)}}^2]^{(j)})$$

$$[C_{p,v,b}^{(3)}]^{(j)} = [\pi_{v,b}^{(3)}]^{(j)}[D_{p,v,b}^{(3)}]^{(j)}\mathcal{N}(x_{p,v,b}|m^G([\alpha_{p,v,b}]^{(j)}), \Sigma^G([\alpha_{p,v,b}]^{(j)}))$$

$$[D_{p,v,b}^{(k)}]^{(j)} = \mathcal{N}(V_{p,v,b}|[\boldsymbol{\mu}_{v,b}^{(k)}]^{(j)}, [\Sigma_{v,b}^{(k)}]^{(j)}), k = 1, 2, 3$$

$$[w_{p,v,b}^{(k)}]^{(j)} = \frac{[C_{p,v,b}^{(k)}]^{(j)}}{\sum_{i=1}^{3}[C_{p,v,b}^{(i)}]^{(j)}}, k = 1, 2, 3$$

Let $Z_{v,b} = (x_{p,-v,b})_{p\in P_{v,b}^{tr}}$ and $Y_{v,b} = (x_{p,v,-b})_{p\in P_{v,b}^{tr}}$. In the M (Maximization) step, we re-estimate the parameters in iteration $(j+1)$ using the jth responsibilities:

$$[\pi_{v,b}^{(k)}]^{(j+1)} = \frac{1}{|P_{v,b}^{tr}|}\sum_{p\in P_{v,b}^{tr}}[w_{p,v,b}^{(k)}]^{(j)}, k = 1, 2, 3$$

$$[\boldsymbol{\mu}_{v,b}^{(k)}]^{(j+1)} = \frac{\sum_{p\in P_{v,b}^{tr}}[w_{p,v,b}^{(k)}]^{(j)}V_{p,v,b}}{\sum_{p\in P_{v,b}^{tr}}[w_{p,v,b}^{(k)}]^{(j)}}$$

$$[\Sigma_{v,b}^{(k)}]^{(j+1)} = \frac{1}{\sum_{p\in P_{v,b}^{tr}}[w_{p,v,b}^{(k)}]^{(j)}}\sum_{p\in P_{v,b}^{tr}}[w_{p,v,b}^{(k)}]^{(j)}[U_{p,v,b}^{(k)}]^{(j+1)}$$

$$[U_{p,v,b}^{(k)}]^{(j+1)} = \{V_{p,v,b} - [\boldsymbol{\mu}_{v,b}^{(k)}]^{(j+1)}\}\{V_{p,v,b} - [\boldsymbol{\mu}_{v,b}^{(k)}]^{(j+1)}\}'$$

$$[\beta_{v,b}^{(1)}]^{(j+1)} = \{\{Z'_{v,b}[\mathbf{w}_{v,b}^{(1)}]^{(j)}Z_{v,b}\}^{-1}Z'_{v,b}[\mathbf{w}_{v,b}^{(1)}]^{(j)}x_{:,v,b}^{obs}\}'$$

$$[{\sigma_{v,b}^{(1)}}^2]^{(j+1)} = \frac{1}{\sum_{p\in P_{v,b}^{tr}}[w_{p,v,b}^{(1)}]^{(j)}}[S_{v,b}^{(1)}]^{(j+1)}$$

$$[S_{v,b}^{(1)}]^{(j+1)} = \sum_{p\in P_{v,b}^{tr}}[w_{p,v,b}^{(1)}]^{(j)}\{x_{p,v,b} - x_{p,-v,b}[\beta_{v,b}^{(1)}]^{(j+1)}\}^2$$

$$[\beta_{v,b}^{(2)}]^{(j+1)} = \{\{Y'_{v,b}[\mathbf{w}_{v,b}^{(2)}]^{(j)}Y_{v,b}\}^{-1}Y'_{v,b}[\mathbf{w}_{v,b}^{(2)}]^{(j)}x_{:,v,b}^{obs}\}'$$

$$[{\sigma_{v,b}^{(2)}}^2]^{(j+1)} = \frac{1}{\sum_{p\in P_{v,b}^{tr}}[w_{p,v,b}^{(2)}]^{(j)}}[S_{v,b}^{(2)}]^{(j+1)}$$

$$[S_{v,b}^{(2)}]^{(j+1)} = \sum_{p\in P_{v,b}^{tr}}[w_{p,v,b}^{(2)}]^{(j)}\{x_{p,v,b} - x_{p,v,-b}[\beta_{v,b}^{(2)}]^{(j+1)}\}^2$$

$$[\theta_{v,b}]^{(j+1)} = G([\mathbf{w}_{v,b}^{(3)}]^{(j)}, [\theta_{v,b}]^{(j)}, x_{:,v,b}^{obs}, Y_{v,b}, t_{:,v,:})$$

where $[\mathbf{w}_{v,b}^{(k)}]^{(j)}$ is the vector of $[w_{p,v,b}^{(k)}]^{(j)}$ for $p \in P_{v,b}^{tr}$ in iteration j. The kernel parameters $\theta_{v,b}$ of GP models are evaluated by function G, a gradient descent method that calculates the estimates of $[\theta_{v,b}]^{(j+1)}$ to maximize $\mathcal{L}_{v,b}(\gamma)$, using $[\theta_{v,b}]^{(j)}$ as the starting point. The first order derivatives of $\mathcal{L}_{v,b}(\gamma)$ with respect to $\theta_{v,b}$ that are used in G are given in Appendix C.

B GP MODEL

We assume the GP model discussed here in a mixture model for a certain variable and time, and thus we exclude the subscripts v and b. We use $x_{p,t}$ to denote a measurement of the time series x_p at time t for patient p of a certain variable. We use $x_{p,-t}$ to denote a time series without the measurement at time t. The GP model is given by

$$x_{p,t} = \mu_{p,t} + f(t),$$
$$f(t) \sim \mathcal{GP}(0, \mathcal{K}(t, t'))$$

where $\mu_{p,t}$ is the overall mean of the model and $f(t)$ is a Gaussian process with mean of 0 and covariance of $\mathcal{K}(t, t')$. Following the maximum likelihood approach, the best linear unbiased predictor (BLUP) [33] at t and the mean squared error are

$$m^G(\theta, x_{p,-t}, \bar{t}) = (\frac{1 - r^T R^{-1}1_n}{1_n^T R^{-1}1_n}1_n^T + r^T)R^{-1}x_{p,-t}$$
$$\Sigma^G(\theta, x_{p,-t}, \bar{t}) = \sigma_f^2[1 - r^T R^{-1}r + \frac{(1 - 1_n^T R^{-1}r)^2}{1_n R^{-1}1_n}]$$

where $r_t(t') = corr(f(t), f(t'))$, r is the vector of $r_t(t')$ for all possible t, $\bar{t}$ is a vector of time except for time t, R is the $(B-1)\times(B-1)$ correlation matrix and the correlation function is given by

$$R_{t,t'} = \exp(-\theta|t - t'|^2).$$

The estimator σ^2 is given by

$$\sigma_f^2 = \frac{C^T R^{-1}C}{n}, C = x_{p,-t} - 1_n(1_n^T R^{-1}1_n)^{-1}(1_n^T R^{-1}x_{p,-t})$$

where 1_n is a vector with length $(B-1)$ of all ones.

C PARTIAL DERIVATIVES IN GP

To simplify the notations, we assume that the likelihood function L under consideration is for a mixture model for a certain variable and time. The partial derivative with respect to Gaussian process parameters θ is

$$\frac{\partial L}{\partial \theta} = \sum_{p=1}^{|p^{tr}|} w_p \frac{\partial}{\partial \theta}\ln \mathcal{N}(x_{p,t}; m^G(\theta, x_{p,-t}, \bar{t}), \Sigma^G(\theta, x_{p,-t}, \bar{t})).$$

Letting $g_p(\theta) = m^G(\theta, x_{p,-t}, \bar{t})$ and $h_p(\theta) = \Sigma^G(\theta, x_{p,-t}, \bar{t})$, we have

$$\begin{aligned}
\frac{\partial L}{\partial \theta} &= \sum_{p=1}^{|p^{tr}|} w_p \frac{\partial}{\partial \theta} \ln \mathcal{N}(x_{p,t}; g_p(\theta), h_p(\theta)) \\
&= \sum_{p=1}^{|p^{tr}|} w_p \frac{\partial}{\partial \theta} \{\ln \frac{1}{\sqrt{2\pi h_p(\theta)}} - \frac{[x_{p,t} - g_p(\theta)]^2}{2h_p(\theta)}\} \\
&= \sum_{p=1}^{|p^{tr}|} w_p \{-\frac{1}{2h_p(\theta)} \frac{\partial h_p(\theta)}{\partial \theta} - \frac{\partial}{\partial \theta} \frac{[x_{p,t} - g_p(\theta)]^2}{2h_p(\theta)}\} \\
&= \sum_{p=1}^{|p^{tr}|} w_p \{-\frac{1}{2h_p(\theta)} \frac{\partial h_p(\theta)}{\partial \theta} \\
&- \frac{1}{2h_p^2(\theta)} \{2[x_{p,t} - g_p(\theta)][-\frac{\partial g_p(\theta)}{\partial \theta}] h_p(\theta) \\
&- \frac{\partial h_p(\theta)}{\partial \theta} [x_{p,t} - g_p(\theta)]^2\}\} \\
&= \sum_{p=1}^{|p^{tr}|} w_p \{-\frac{1}{2h_p(\theta)} \frac{\partial h_p(\theta)}{\partial \theta} \\
&+ \frac{[x_{p,t} - g_p(\theta)] \frac{\partial g_p(\theta)}{\partial \theta}}{h_p(\theta)} + \frac{\frac{\partial h_p(\theta)}{\partial \theta} [x_{p,t} - g_p(\theta)]^2}{2h_p^2(\theta)}\}.
\end{aligned}$$

Then $\frac{\partial g_p(\theta)}{\partial \theta}$ and $\frac{\partial h_p(\theta)}{\partial \theta}$ are given by

$$\begin{aligned}
\frac{\partial g_p(\theta)}{\partial \theta} &= (\frac{\partial H_1}{\partial \theta} R^{-1} + H_1 \frac{\partial R^{-1}}{\partial \theta}) x_{p,-t} \\
\frac{\partial h_p(\theta)}{\partial \theta} &= \sigma_f^2 \frac{\partial H_3}{\partial \theta} + \frac{\partial \sigma_f^2}{\partial \theta} H_3
\end{aligned}$$

where H_1, $\frac{\partial H_1}{\partial \theta}$, H_3 and $\frac{\partial H_3}{\partial \theta}$ are given as follows:

$$\begin{aligned}
H_1 &= \frac{[1 - (rR^{-1}1_n)]}{1_n^T R^{-1} 1_n} 1_n^T + r \\
\frac{\partial H_1}{\partial \theta} &= \frac{-(\frac{\partial r}{\partial \theta} R^{-1} + r \frac{\partial R^{-1}}{\partial \theta}) 1_n (1_n^T R^{-1} 1_n)}{1_n^T R^{-1} 1_n^2} \\
&- \frac{(1_n^T \frac{\partial R^{-1}}{\partial \theta} 1_n)[1 - (rR^{-1}1_n)]}{1_n^T R^{-1} 1_n^2} 1_n^T + \frac{\partial r}{\partial \theta} \\
fc &= (1 - 1_n^T R^{-1} r^T)^2 \\
gc &= 1_n^T R^{-1} 1_n \\
\frac{\partial fc}{\partial \theta} &= 2(1 - 1_n^T R^{-1} r^T)[-1_n^T (\frac{\partial R^{-1}}{\partial \theta} r^T + R^{-1} \frac{\partial r^T}{\partial \theta})] \\
\frac{\partial gc}{\partial \theta} &= 1_n^T \frac{\partial R^{-1}}{\partial \theta} 1_n \\
H_2 &= \frac{(1 - 1_n^T R^{-1} r^T)^2}{1_n^T R^{-1} 1_n} \\
\frac{\partial H_2}{\partial \theta} &= \frac{\frac{\partial fc}{\partial \theta} gc - \frac{\partial gc}{\partial \theta} fc}{gc^2} \\
H_3 &= 1 - (rR^{-1} r^T) + H_2 \\
\frac{\partial H_3}{\partial \theta} &= -(\frac{\partial r}{\partial \theta} R^{-1} r^T + r \frac{\partial R^{-1}}{\partial \theta} r^T + r R^{-1} \frac{\partial r^T}{\partial \theta}) + \frac{\partial H_2}{\partial \theta} \\
H_4 &= x_{p,-t} - 1_n \frac{(1_n^T R^{-1} x_{p,-t})}{1_n^T R^{-1} 1_n} \\
\frac{\partial H_4}{\partial \theta} &= -1_n \frac{1}{(1_n^T R^{-1} 1_n)^2} [(1_n^T \frac{\partial R^{-1}}{\partial \theta} x_{p,-t})(1_n^T R^{-1} 1_n) \\
&- (1_n^T \frac{\partial R^{-1}}{\partial \theta} 1_n)(1_n^T R^{-1} x_{p,-t})] \\
\frac{\partial \sigma_f^2}{\partial \theta} &= \frac{1}{n} [(\frac{\partial H_4}{\partial \theta})^T R^{-1} H_4 + H_4^T \frac{\partial R^{-1}}{\partial \theta} H_4 + H_4^T R^{-1} \frac{\partial H_4}{\partial \theta}].
\end{aligned}$$