Published in final edited form as:

Conf Proc IEEE Eng Med Biol Soc. 2018 July ; 2018: 4106–4109. doi:10.1109/EMBC.2018.8513303.

Comparison of Gaussian Processes Methods to Linear methods for Imputation of Sparse Physiological Time Series

Paul Nickerson,

University of Florida, Gainesville, FL, 32611 USA.

Raheleh Baharloo,

University of Florida, Gainesville, FL, 32611 USA.

Anis Davoudi,

University of Florida, Gainesville, FL, 32611 USA.

Azra Bihorac, and

University of Florida, Gainesville, FL, 32611 USA.

Parisa Rashidi

University of Florida, Gainesville, FL 32611 USA.

Abstract

Physiological timeseries such as vital signs contain important information about a patient and are used in different clinical application; however, they suffer from missing values and sampling irregularity. In recent years, Gaussian Processes have been used as sophisticated nonlinear value imputation methods on time series, however there is a lack of comparison to other simpler methods. This paper compares the ability of five methods that can be used in missing data imputation in physiological time series. These models are linear interpolation as the baseline, cubic spline interpolation, and three non-linear methods: Single Task Gaussian Processes, Multi-Task Gaussian Processes, and Multivariate Imputation Chained Equations. We used seven intraoperative physiological time series from 27,481 patients. Piecewise aggregate approximation was employed as a dimensionality reduction and resampling strategy. Linear interpolation and cubic splining show overall superiority in prediction of the missing values, compared to the other complex models. The performance of the kernel-based methods suggest that they are highly sensitive to the kernel width and require incorporation of domain knowledge for fine-tuning.

I. Introduction

Electronic Health Records (EHR) incorporate a wide array of information ranging from demographics to high-resolution physiological timeseries (vital signs). Physiological timeseries may contain important information about a patient's condition; however, timeseries present challenges stemming from their intrinsic high dimensionality.

Most of the inference methods for time series data assume regular intervals. They also require data points representing the random process in different time indices. However, physiological time series such as vital signs often suffer from missing data points and sampling irregularity. Furthermore, a variety of noise sources, including human error, are

present in the data. Therefore, preprocessing steps like data imputation are important to achieve high-quality data for reliable inferences. There are different data imputation methods ranging from linear interpolation and cubic splining to more complex, kernel-based methods. Linear interpolation uses linear polynomials to build new points in the gap interval among the known data points. Cubic splining exploits piecewise polynomials, thus making a smoother interpolation, with less error in approximation compared with linear polynomials. Kernel-based methods including Single-Task Gaussian Processes and Multi-Task Gaussian Process produce a latent representation of the data by transferring each data point to a high dimensional feature space and then exploiting the statistical properties of data in the new space. These methods have previously been utilized to infer missing values as a preprocessing step prior to deep learning [1]. They have also been used to incorporate clinical domain knowledge of sinusoidal temporal patterns and correlate multiple time series to impute missing data [2]. A comparison of various data imputation methods in the medical domain and their strengths is absent from the literature. In this paper, we compare various solutions for imputing missing data within vital signs time series, with an eye toward developing preprocessing strategies that can be implemented prior to further modeling.

II. Methods

A. Data

The inclusion criteria were defined as patients aged 18 years or older admitted to the UF Shands Hospital for a stay of at least 48 hours following any type of operative procedure between January 1, 2000, and November 30, 2010. The final cohort included 27,481 patients. Vital signs data were extracted and truncated to within the boundary of surgery start and end time. Thus, all examined vital signs time series were measured intraoperatively.

B. Methods

B.1. Single-Task Gaussian Processes (STGP)—A Gaussian Process is a probability distribution over functions [3]. In most cases, only a single time series (task) is considered. Let $x \in \mathcal{R}^n$ be a vector of training indices and $y \in \mathcal{R}^n$ be the associated values, where n is the number of data points. Our goal is to learn a regression model $y = f(x) + \mathcal{N}(0, \sigma^2)$, where $f(x)$ represents a latent function and $\mathcal{N}(0, \sigma^2)$ is a noise term. The function $f(x)$ can be interpreted as a probability distribution $f(x) \sim GP(m(x), k(x, x'))$, where $m(x)$ is a mean function and $k(x, x')$ is a covariance function (kernel) representing a similarity score between two values of x . In these experiments, we used squared-exponential kernel:

$$k_{SE}(x, x') = \theta_s^2 \exp\left(-\frac{\|x - x'\|^2}{2\theta_s^2}\right) \quad (1)$$

The $m(x)$ and $k(x, x')$ models are learned from the data. For notational simplicity, $m(x)$ is generally set to zero, which can be done without loss of generality. Given observed data x and y , as well as unobserved hypothetical data x^* and y^* , we can represent the joint distribution by combining covariance matrices as follows:

$$\begin{bmatrix} y \\ y^* \end{bmatrix} \sim N \left(0, \begin{bmatrix} K_{11} & \cdots & K_{1n} \\ \vdots & \ddots & \vdots \\ K_{n1} & \cdots & K_{nn} \end{bmatrix} \begin{bmatrix} K_{*1} \\ \vdots \\ K_{n*} \end{bmatrix} \right) = N \left(0, \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix} \right) \quad (2)$$

Where $K_{ij} = k(x_i, x_j)$, $K_{*} = k(x_i, x^*)$, and $K_{**} = k(x^*, x^*)$ make up the covariance matrices calculated using our kernel [4]. Note the covariance expressions in the two lines are equivalent, but the submatrices have been replaced with variable representations for notational simplicity. The goal is to calculate $\bar{y}^* \sim N(\bar{y}^*, \sigma^*)$. Using the Schur complement [4, 5], we get the following result:

$$\bar{y}^* = K_*^T K^{-1} y \quad (3)$$

$$\sigma^* = -K_*^T K^{-1} K_* + K_{**} \quad (4)$$

Thus, we can make predictions on the output of the latent function for unseen data points. A particularly useful property is the ability to quantify uncertainty using σ^* , allowing us to bound predictions with a confidence interval. Hyperparameter values θ can be inferred, for example, by starting with an initial guess, then minimizing the negative log marginal likelihood (NLML) [3]:

$$NLML = -\log p(y|x, \theta) = \frac{1}{2} \log |K| + \frac{1}{2} y^T K^{-1} y + \frac{n}{2} \log(2\pi) \quad (5)$$

B.2. Multi-Task Gaussian Processes (MTGP)—Gaussian Process methodology can be extended to the case of multiple tasks with a few adjustments [2]. Let m denote the number of tasks, $x = (x^{(1)}, \dots, x^{(m)})$ and $y = (y^{(1)}, \dots, y^{(m)})$, with the i^{th} task containing n_i points. To specify that $x^{(i)}$ and $y^{(i)}$ belong to task i , label vectors $(\ell^{(1)}, \dots, \ell^{(m)})$ are added alongside x , with all components of $\ell^{(i)} = i$. Different tasks may have different training indices. This means this model is suited for imputing missing data either using existing data from other related tasks or auto-correlating with existing data in the same task.

Two independent covariance functions can be assumed, $K^c(I,)$ for correlation between tasks, and $K^t(x, x')$ for autocorrelation within a task at different indices. Let K^c and K^t be the covariance matrices formed from the aforementioned kernels. For notational simplicity, we assume the same length for each task, $n_i = n \forall i \in \{1, \dots, m\}$; thus $K^c \in R^{m \times m}$ and $K^t \in R^{n \times n}$. The Cholesky decomposition of K^c is used to construct a valid positive semidefinite

matrix in order to satisfy the Mercer theorem, which is needed to form a valid kernel. The full covariance matrix for the training data is

$$K_{MTGP} = K_c \otimes K_t \quad (6)$$

Where $\otimes$ is the Kronecker product, and $K_{MTGP} \in R^{mn \times mn}$. To account for the correlations between any given pairs being different from each other, two or more covariance functions, each having their own hyperparameters, can be convolved into a new covariance function. This also satisfies Mercer's theorem. Thus, individual correlation hyperparameters may be learned for each pair of tasks.

B.3. Multivariate Imputation Chained Equations (MICE)—MICE is a commonly used method for imputing missing data in multivariate time series [6]. It has several advantages over other multivariate time series imputation methods. It can create many different plausible datasets in which different values are imputed. Imputed values are based on other observed values for the individual time series as well as on its relations to other time series. This makes MICE an ideal choice for imputing values in multivariate time series, where individual tasks may be correlated with others.

MICE method assumes that missing data are Missing At Random (MAR), i.e. the probability that a value is missing depends only on the observed values and not on unobserved values [7]. If this assumption is violated, then imputed values may be biased. It is unclear if it would hold in real world, where the fact that vital signs are currently being measured may indicate a higher probability of clinically-significant abnormal values. In these experiments, values are MAR.

The specific MICE process used here – known as Predictive Mean Matching, PMM [8, 9] – can be broken down into the following six steps: 1) When a timeseries has no missing data, estimate a linear regression between that timeseries and another timeseries with missing values, producing a set of coefficients. 2) Produce a random sample from a distribution modeled on the coefficients in step 1, producing a new set of coefficients. This allows for an infinite number of possible artificial coefficients, and consequently an infinite number of possible generated datasets. 3) Using the coefficients from step 2, generate cases for all of the values of the time series with missing values, including regions with missing data. 4) Use cases produced in step 3 where observed data is close to predicted data, to impute missing data. 5) From among the cases in step 4, randomly choose one to use for imputing missing data. 6) Repeat steps 2–5 for each completed data set.

I. Experiments

A series of experiments using STGP, MTGP, MICE, and cubic spline interpolation were compared to the performance of simple linear interpolation as the baseline.

For this study, we needed to impute missing points to produce a data set from which features can be extracted to make clinical decisions. Towards this end, we selected a set of vital signs

traditionally used for feature extraction in applications such as acuity scores [10, 11]. These variables have the possibility of demonstrating inter-relationships and correlations, which may be utilized by the imputation method. Seven time series were selected: end-tidal CO₂ (EtCO₂), FiO₂/SpO₂ ratio, systolic blood pressure (Sys), mean arterial blood pressure (MAP), heart rate (HR), respiratory rate (RR), and temperature (Temp).

The minimum measurement resolution was one second. However, the data were very sparse at this resolution. The imputation methods depend on a constant/regular measurement interval; therefore, we employed Piecewise Aggregate Approximation (PAA) as a dimensionality reduction and resampling strategy to create a downsampled representation which closely preserves the properties of the underlying stochastic process [12]. From the original vital signs time series, a new set of PAA representations were derived utilizing a frame width of five minutes, meaning each data point of the new time series was calculated as the average value of a five-minute window of original data. This window length resulted in a maximum of 300 original measurements. However, the number of measurements included per PAA point was generally lower.

For each vital sign, three batches of 300 random windows were extracted, each with two hours of data (i.e. 24 PAA data points). These batches corresponded to experiments in which one, three, and five consecutive PAA time points would be erased (see Figure I). To qualify, each window had to contain extant values both in the target area to be erased as well as neighboring time points. We refer to the number of points deleted as the erasure time span. These erasure time span values were chosen to investigate model performance with varying lengths of imputation intervals. Each model's ability to impute missing data values were investigated in terms of Root Mean Square Error (RMSE) between the imputed values and the known values pre-deletion. Here, in model training and evaluation, each attempt to predict missing values for a single window is called a "trial".

A. STGP

The STGP Model was implemented using the Scikit-Learn Python library [13]. We used an exponential kernel for the covariance matrix (as is general practice). This matrix was convolved with a white noise kernel to estimate and account for dataset noise. Kernel parameters are learned during model optimization. However, the initial values can influence the quality of the local minima selection. To improve the local minima selection, an STGP was trained for each trial with default initial kernel parameters. Next, the median of the optimal kernel parameters for all trials was used as the starting kernel parameters for retraining. This resulted in a modest improvement in RMSE. Missing data values were predicted as the value of the GP mean function evaluated at their indices.

STGPs are not computationally expensive to train. We evaluated both "STGP-Windowed" trials, in which windows were examined, as well as "STGP-Whole" trials, in which the entire vital signs time series were examined. The same locations were erased for both STGP-whole and STGP-windowed. The only difference was that, for STGP-whole, the rest of the data outside of the window were available for training as well.

B. MTGP

Due to computational complexity, training the MTGP using the whole timeseries is intractable. Thus, only windowed training was examined. The MTGP MatLab [14] library was utilized for training and prediction. Each trial utilized an MTGP with convolved exponential kernels such that each task/time series had its own kernel parameters. For each of 1000 repetitions, kernel parameters were initialized with random values and optimized to the local minimum. The model with the lowest negative log marginal likelihood (NLML) was chosen as the final model. As with the STGP experiments, the predicted value was calculated as the output of the mean function evaluated at missing data indices.

C. MICE

The MICE experiments were designed similarly to the STGP experiments, and the MICE R Package was used for modeling [15]. Both windowed (MICE-Window) and full time series (MICE-Whole) experiments were performed. Each trial consisted of generating 100 imputation possibilities, and the missing value was predicted as the average of each of these at missing time points.

D. Cubic splining

In this method, the whole time series was used in all cases. The reason for this is that a localized region in the vicinity of the erased area was used to fit the spline. In the windowed version, this may have been truncated off. This localized region consisted of two points on either side of the erased area. These points were used to define a cubic spline, from which the missing data was imputed. Splining was implemented using the Numpy Python library [16].

“Coverage” as defined below, can be used to measure the timeseries sparsity:

$$\text{coverage} = 100 \times \frac{\sum_{i=1}^m \sum_{j=1}^n 1(y_{ij} \text{ exists})}{mn} \quad (7)$$

Where $1(y_{ij} \text{ exists})$ denotes the indicator function, returning one if vital signs time series y_i has a measurement at index j , and zero otherwise. m denotes the number of vital signs, and n denotes time series length, which is constant for all vital signs in a patient’s record.

Coverage is an appropriate metric for both the original timeseries and their PAA representations.

III. Results and Discussion

Prior to PAA dimensionality reduction, coverage was 2.04%. The coverage of the PAA representation with five-minute frame-width was 80.98%, representing a 3969.61% increase in coverage – a substantial improvement in measurement regularity and a significant decrease in sparsity.

Regarding the various imputation models, Table I lists the prediction accuracies for each experiment.

For erasure time span of one, cubic splining and linear interpolation were superior in all instances. With increasing the erasure time spans, MICE-Windowed appears to start showing superiority for some vital signs. For erasure time span of three, MTGP was superior in one case. The best-performing model overall was cubic splining. Performance sharply worsened for longer erasure time spans.

MTGP was superior to both STGP-Windowed and STGP-Whole in about half of the experiments with significant difference; however, cases where a STGP experiment surpassed MTGP, the results were usually very close. This suggests the advantage of incorporating information from adjacent timeseries. This is also demonstrated by the performance of MICE-Whole versus MICE-Windowed. MICE only utilizes observed values from other time series to impute missing data and does not take into account vicinity from the unobserved data. Therefore, the more data available to the model and the longer the time series, the more likely MICE is to impute using properties of a distant time region. MICE-Windowed utilized a smaller region than MICE-Whole, therefore missing data were more likely to be imputed using properties of localized regions. MICE-Windowed was far superior to MICE-Whole in nearly all cases, suggesting a strong influence of localized measurements. MTGP's performance and computational complexity suggest that, without major fine-tuning and incorporation of domain knowledge into kernel selection, it is probably not a superior method for general imputation. It may prove more useful in cases where a single time series is being imputed based on information observed in another time series for which domain knowledge suggests a clear relationship, and for which temporal patterns are already known [2].

In the future, the performance of deep learning models to learn interrelationships and influence of localized versus distant measurements needs to be investigated. Specifically, some variation of a recurrent neural network, such as a Long Short Term Memory (LSTM) or Gated Recurrent Unit (GRU) architecture may prove useful for automatic imputation [17]. We also suspect that a simple machine learning regression model – e.g. Elastic Net or Support Vector Regression – could be trained to utilize features engineered from linear- or spline-interpolated values nearby within the same timeseries and from adjacent timeseries. This model may be adept at learning interdependencies, an advantage over linear interpolation, which allows it to have the best of all worlds, while being computationally inexpensive and trivial to implement in a real-time system.

Acknowledgment

We acknowledge the University of Florida Integrated Data Repository (IDR) and the UF Health Office of the Chief Data Officer for providing the analytic data set for this project. Additionally, the Research reported in this publication was supported by the National Center for Advancing Translational Sciences of the National Institutes of Health under University of Florida Clinical and Translational Science Awards UL1 TR000064 and UL1TR001427, and NIH/NIGMS RO1 GM-110240.

References

- [1]. Lasko TA, Denny JC, and Levy MA, "Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data," (in eng), PLoS One, vol. 8, no. 6, p. e66341, 2013. [PubMed: 23826094]

- [2]. Dürichen R, Pimentel MA, Clifton L, Schweikard A, and Clifton DA, "Multitask Gaussian processes for multivariate physiological time-series analysis," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 1, pp. 314–322, 2015. [PubMed: 25167541]
- [3]. Williams CK and Rasmussen CE, "Gaussian processes for machine learning," the MIT Press, vol. 2, no. 3, p. 4, 2006.
- [4]. Denil M, Matheson D, and De Freitas N, "Narrowing the gap: Random forests in theory and in practice," in *International conference on machine learning (ICML)*, 2014.
- [5]. Murphy KP, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012, p. 1096.
- [6]. Azur MJ, Stuart EA, Frangakis C, and Leaf PJ, "Multiple imputation by chained equations: what is it and how does it work?," *International journal of methods in psychiatric research*, vol. 20, no. 1, pp. 40–49, 2011. [PubMed: 21499542]
- [7]. Schafer JL and Graham JW, "Missing data: our view of the state of the art," *Psychological methods*, vol. 7, no. 2, p. 147, 2002. [PubMed: 12090408]
- [8]. Rubin DB, "Statistical matching using file concatenation with adjusted weights and multiple imputations," *Journal of Business & Economic Statistics*, vol. 4, no. 1, pp. 87–94, 1986.
- [9]. Allison P, "Imputation by predictive mean matching: Promise & peril," *Statistical Horizons*, 2015.
- [10]. Subbe C, Kruger M, Rutherford P, and Gemmel L, "Validation of a modified Early Warning Score in medical admissions," *Qjm*, vol. 94, no. 10, pp. 521–526, 2001. [PubMed: 11588210]
- [11]. Vincent J-L et al., "The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure," ed: Springer, 1996.
- [12]. Lin J, Keogh E, Lonardi S, and Chiu B, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, 2003, pp. 2–11: ACM.
- [13]. Pedregosa F et al., "Scikit-learn: Machine learning in Python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.
- [14]. Durichen R, Pimentel MAF, Clifton L, Schweikard A, and Clifton DA. (2015, 3/1/2018). MTGP- A Multi-task Gaussian Process Toolbox. Available: http://www.robots.ox.ac.uk/~davidc/publications_MTGP.php
- [15]. Buuren S. v. and Groothuis-Oudshoorn K, "mice: Multivariate imputation by chained equations in R," *Journal of statistical software*, pp. 1–68, 2010.
- [16]. Walt S. v. d., Colbert SC, and Varoquaux G, "The NumPy Array: A Structure for Efficient Numerical Computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.
- [17]. Goodfellow I, Bengio Y, and Courville A, *Deep learning*. MIT press Cambridge, 2016.

Figure 1.
Time span erasure illustration, involving the deletion of five consecutive PAA points.

TABLE I.

Model performance for missing data imputation

Model	EtCO ₂	FiO ₂ /SpO ₂	HR	MAP	RR	Sys	Temp
Erasure time span: 1							
STGP-wd	3.92	0.10	12.04	13.52	2.30	19.32	0.86
STGP-w	4.96	0.08	14.41	15.59	2.59	20.47	2.03
MTGP	3.42	0.12	10.89	10.18	2.62	12.05	0.81
MICE-wd	4.02	0.12	10.75	11.72	2.77	13.33	1.70
MICE-w	9.85	0.23	21.86	20.52	3.73	36.21	9.17
L-interpol	2.62	0.06	9.35	11.20	2.69	13.24	0.68
C-splining	3.17	0.06	12.75	9.38	1.84	12.82	0.67
Erasure time span: 3							
STGP-wd	5.16	0.17	12.19	12.49	2.40	17.48	1.68
STGP-w	5.78	0.12	15.07	13.50	2.49	18.42	2.25
MTGP	4.77	0.14	11.47	10.54	2.75	12.96	2.00
MICE-wd	4.80	0.15	11.13	11.13	2.66	12.55	1.93
MICE-w	10.06	0.27	20.48	16.95	3.40	24.82	7.67
L-interpol	3.73	0.10	10.40	10.72	2.07	14.84	1.66
C-splining	3.64	0.09	13.24	12.00	1.94	16.51	1.57
Erasure time span: 5							
STGP-wd	4.35	0.18	12.69	13.05	2.29	15.30	1.72
STGP-w	5.24	0.14	14.65	14.86	2.22	17.41	2.23
MTGP	4.48	0.15	12.73	10.84	2.62	12.27	1.58
MICE-wd	4.20	0.15	11.93	10.12	2.20	11.80	2.01
MICE-w	10.41	0.24	25.80	23.49	3.55	33.61	8.64
L-interpol	3.44	0.11	10.89	12.19	2.49	14.57	1.01
C-splining	4.07	0.11	13.89	16.21	4.13	19.68	1.07

STGP: single task Gaussian processes, MTGP: multi-task Gaussian processes, MICE: multivariate imputation chained equations, L-interpol: linear interpolation, C-splining: cubic splining. The prefix of each model name shows the erasure time span. Performances are reported in terms of RMSE. Suffix of 'wd' shows the 'windowed' version of models, while 'w' suffix shows the 'whole' version of the models.