The Impact of Missing Data in User-Generated mHealth Time Series

Li-Fang Cheng[1], **David Stück**[2], **Tom Quisel**[2], **Luca Foschini**[2]
[1]Princeton University, Princeton, NJ
[2]Evidation Health, Santa Barbara, CA

Abstract

Mobile wearable devices and apps have created new pathways for individuals to collect health and well-being data outside the point of care and over time. However, data quality issues such as missing data due to the data collection happening in free-living conditions may impact the utility of the data and has severely limited adoption of consumer-generated data in clinical settings. In this work, we take a first step at quantifying the impact of data missingness in mHealth time series and propose ways to mitigate it via imputation. First, we compare the performance of different imputation strategies in reconstructing known portions of mHealth time series. Second, we investigate the benefits of performing imputation as a pre-processing step of a classification pipeline using a multi-task convolutional neural network (MT-CNN) to predict self-reported chronic conditions from the mHealth time series. We additionally study changes in classification performance as a function of artificially increasing data missing rate in the mHealth time series. We find that imputers based on Gaussian Processes (GP) outperform simpler baseline when they include time kernels that allow learning user-specific behavioral patterns. We also observe that the performance of MT-CNN classifiers is very robust to missingness in the data and may benefit only marginally from the imputation pre-processing.

1 Introduction

Mobile wearable technologies are increasingly giving their users the opportunity to continuously collect and analyze data about their health and wellness. However, data collected in real-life settings cannot undergo the same quality control that is imposed on data collected in a controlled environment, such as at a point of care. User behavior, battery life, device malfunctioning etc. are all factors that affect the quality of the stream of data. Even the resulting decreased data quality may be acceptable from a consumer's perspective, it becomes an issue when the data is used for research purposes, or even more so, in clinical applications.

Missing data, the lack of reported data for extended periods of times, is a major limitation in the use of consumer-generated data for clinical research. In the time series context, when the signal measured is smooth over time, missing data is usually imputed linearly as a preprocessing step to any further analysis. Several data science tools and libraries come with a flurry of imputation options available out of the box [1]. Linear interpolation works on the assumption that the missing data over the gap matches a linear interpolation between the two observations immediately straddling the gap. However, in the case of time-series data where each data point represent a daily measurement from a user (e.g., step counts, slept hours) using additional knowledge learned from the users' history may provide a better guess for the missing data.

In this work, we explore several techniques to impute daily time series of user behavior. In particular, to capture the behavioral patterns of users better, we focus on exploring nonlinear methods, especially

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Gaussian processes (GPs), to impute the mHealth time series. GPs have recently been applied to time series analysis due to their flexibility and ability to dealing with missing data naturally [2]. In our experiment, we first consider an mHealth dataset with very low rate of missing data. We use GPs and leverage subject-specific patterns to reconstruct segments of data that have been artificially removed. Subsequently, we assess the value of using GP-based imputation as a preprocessing step in a multi-task CNN pipeline to predict which of 9 chronic conditions a user has self-reported from step- and sleep- related daily aggregate of minute-level data [3]. In doing so, we artificially increase the level of missing data and study how the classification performance varies as a function of the fraction of data removed, with independent and correlated removal patterns.

To the best of our knowledge, this is the first work that investigates the impact of missing data in mHealth time series.

2 Methods

Gaussian processes By definition, a Gaussian process (GP) is a collection of random variables, where any finite collection of which have a joint multivariate Gaussian distribution [4]. A GP can also be described as a distribution over functions $f(x)$, written as:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')).$$

Here, $m(x)$ is the *mean function*, and $k(x, x')$ is the *covariance function*, or *kernel*. The kernel of a GP decides the properties of the functions sampled from it. For instance, a ***RBF*** kernel is a kernel with two hyperparameters σ and ℓ: $k_{\text{RBF}}(x, x') = \sigma^2 \exp\left(-\frac{||x-x'||^2}{2\ell^2}\right)$. A function drawn from a GP with a RBF kernel is infinitely differentiable, so is usually considered as smooth functions over the input domain. The hyperparameter ℓ decides how fast the function changes with x. In practice, we want to learn kernel hyperparameters such as σ and ℓ from the data [1].

Kernel design based on user behaviors. Some human behaviors are predictable, and they are reflected by predictable temporal patterns in mHealth data. For example, daily stepcounts may change significantly on weekend vs. weekdays. To take these bahaviors into account we experimented several common kernels, including ***RBF*** kernels and other popular kernels [2]. We also considered the *spectral mixture (SM)* kernel proposed in [5] that can capture empirically a wide array of kernels. The formulation of an one-dimensional ***SM*** kernel is written as:

$$k_{\text{SM}}(x, x') = \sum_{q=1}^{Q} w_q \exp\left(-2\pi^2\tau^2 v_q\right) \cos\left(2\pi\tau\mu_q\right), \quad \tau = x - x'$$

where x, x' represent time, Q is the number of mixtures and μ_q and v_q controls period and smoothness of each k_q specifically. One of the advantages of using ***SM*** kernel is to discovery different kernel components, especially potential periodic patterns useful for prediction in practice [5, 6]. We further add priors on μ_q and v_q to encode our belief in activity patterns. For setup ***SM-Q2*** we set $Q = 2$, and chose $\mu_1 \sim \log\mathcal{N}(u_1, 0.1)$, $v_1 \sim \log\mathcal{N}(l_1, 0.1)$ to capture weekly patterns [3]. For the second kernel, we set $\mu_2 \sim \log\mathcal{N}(u_2, 0.1)$, $v_2 \sim \log\mathcal{N}(l_2, 0.1)$ to encode aperiodic decaying temporal dependency [4].

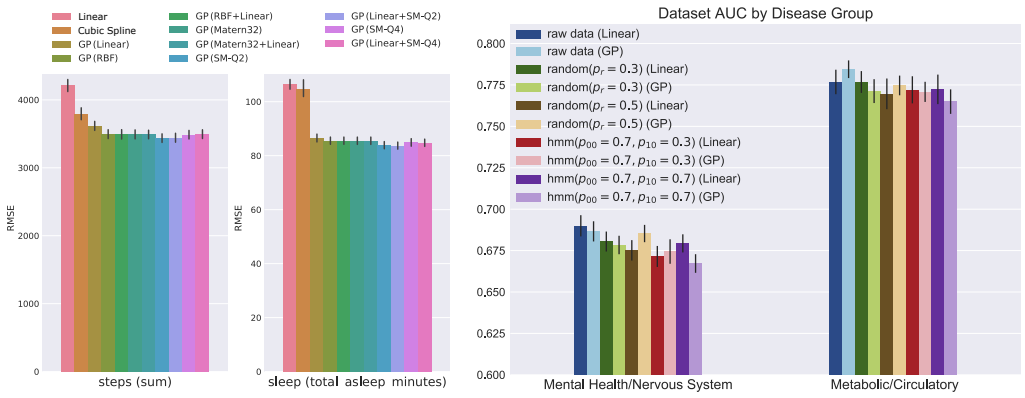
We additionally considered more complex kernels, including four kernel components ($Q = 4$), denoting as ***SM-Q4***, which includes the same prior setup as in ***SM-Q2*** described above, plus two additional kernel components with priors to capture potential monthly and seasonality patterns. Finally, we considered a ***Linear*** kernel to account for possible monotonic changes in behaviors, such as the decrease in interest of using the devices. We denote those kernels as ***Linear+SM-Q2*** and ***Linear+SM-Q4***, respectively. For all experiments, we adapted GPy [7] for implementation.

[1] Note that as in most previous work, we set mean function as zero without loss of generality [4]

[2] ***Linear***: $k_{\text{LIN}}(x, x') = \sigma_b^2 + \sigma_v^2 x x'$; ***Matern32***: $k_{\text{MAT32}}(x, x') = \sigma^2(1 + \frac{\sqrt{3}||x-x'||}{\ell}) \exp(-\frac{\sqrt{3}||x-x'||}{\ell})$.

[3] $u_1 = 1/7$ for periodicity of a week, where the unit of our input dimension is day(s). $l_1 = 1/(2\pi 30)$ that corresponds to decaying temporal dependency of 30 days.

[4] $u_2 = 1/1000$, $l_1 = 1/(2\pi 30)$ that corresponds only decaying temporal dependency of 30 days.



(a) (b)

Figure 1: (a) The imputation results (7-day window) in RMSE on two selected time series: total daily step count and total sleep duration (in hours). (b) The AUCs under different levels of added missing data using different imputation methods.

Subject-specific Training. Considering the large variations in activity data across subjects, we train one GP on each individual's data separately with the same prior. We assume the noises are i.i.d. Gaussian noises, so we can compute the marginal likelihood directly and learn the kernel hyperparameters through optimizing the marginal likelihood directly using gradient ascent [4].

3 Experimental Results

3.1 Imputation

Data Sets. We start from a randomly selected cohort of 3,888 users who shared Fitbit daily steps and sleep with a reward platform between May 2016 and May 2017 (observation period). To investigate the performance of imputation methods, we select a cohort of 551 users with low rates of missing data. The cohort satisfies the inclusion criteria: (i) have more than 200 days with both step and sleep data, (ii) more than 90% days with data in the first 90 days of the observation period, and (iii) overall more than 60% of days with data over all the observation period. For each user, we consider two time series with daily cadence: The total daily step count, and the total time asleep.

We compare imputation strategies by measuring the reconstruction error (RMSE) on a 7-day window of data (hold-out window) randomly selected within the observation period year. We use the rest of the observations from the user to learn the parameters of the GP models, and then use the model to regress values in the hold-out 7-day window. We repeated the experiment on each subject five times, for five different random hold-out windows.

Baselines. We compare the GP-based methods with two baselines. The first baseline is piecewise linear interpolation. The second baseline is cubic spline interpolation. For implementation, we use the off-the-shelf functions provided in the pandas Python library [1].

Results. Figure 1 (a) summarizes the imputation errors in subject-level root-mean-square-error (RMSE). We find that using GP with the kernel *Linear+SM-Q2* has the smallest root-mean-square error (RMSE) for both step and sleep data, and is statistically significant (p-value < 0.05 for paired t-tests) compared with other kernels, except for the second best kernel *SM-Q2*.

3.2 Multi-task Classification

Data Sets. In this section, we move from quantifying reconstruction error in a high-quality dataset, to the complementary task of investigating imputation as a step that a practitioner can use to pre-process datasets with higher rates of missing data prior to performing further analyses. Specifically, we aim at quantifying the impact of imputation on a the multi-task classification technique described in Quisel et al. [3], which considers a dataset of 7,261 subjects with 5 demographic features (age, gender,

ethnicity, education level, and parental status), 4 basic health features (weight, max weight in the past, height, BMI) and 49 daily time series activity features computed from minute-level steps and minute-level sleep data (referred to as "steps" and "sleep" features, respectively). A day with missing steps (resp. sleep) data will have all the steps-related (resp. sleep-related) features missing. The MT-CNN is trained to predict targets corresponding to user's self-reported chronic conditions, from a set of 9 conditions that were grouped into two clusters: a mental health/nervous system (MH/NS) cluster composed of 6 conditions: anxiety, depression, other mental illness, chronic pain, insomnia, and sleep apnea and a metabolic/circulatory (M/C) cluster composed of 3 conditions: hypertension, type 2 diabetes, and dyslipidemia.

Our main objective is to test whether imputed data improves classification performance using the multi-task convolutional neural network (MT-CNN) at increasing levels of missingness. In a first experiment, we directly impute the activity features time series from the original dataset, which has on average users with 40% days with missing steps feature and 65% days with missing sleep features [5]. In Quisel et al. [3], the data were imputed using piecewise linear imputation in both step and sleep features. We use the same method to impute the weight features, but use GP with *Linear+SM-Q2* kernel to impute step and sleep features.

In a second experiment, we re-run the pipeline of the previous experiment by artificially increasing the rate of missing data. We simulated two patterns of missingness: (i) random corruption where each present daily data point is removed with probability $p_r = \{0.3, 0.5\}$, and (ii) two-state Markov model with transition matrix A, with state "0" denotes missing and state "1" denotes not missing. In our experiment, we parameterize $A = [[p_{00}, 1 - p_{00}], [p_{10}, 1 - p_{10}]]$ where p_{00} is the probability that next non-missing day is being removed given that previous non-missing day has been removed, and p_{10} is the probability that the next non-missing day is being removed even if the previous non-missing day has not been removed. We simulated two sets of parameters: $(p_{00} = 0.7, p_{10} = 0.3)$ and $(p_{00} = 0.7, p_{10} = 0.7)$.

Results. Figure 1 (b) illustrates the results of MT-CNN in area under curve (AUC) on the imputed data for two different groups of tasks. For all experiments, we use all features available. We use the same architecture of MT-CNN as that in Quisel et al. [3]. The AUCs are computed on the hold-out 25% test set and is consistent across experiments. Compared with the results pairwise by imputation methods, we find three statistically significant improvements after doing t-tests (p-value < 0.05): (i) in `M/C` group, `raw data (GP)` improves over `raw data (Linear)` by 0.007; (ii) in `MH/NS` group, `random`($p_r = 0.5$) `(GP)` improves over `random`($p_r = 0.5$) `(Linear)` by 0.010; (iii) in `MH/NS` group, `hmm`($p_{00} = 0.7, p_{10} = 0.7$) `(Linear)` improves over `hmm`($p_{00} = 0.7, p_{10} = 0.7$) `(GP)` by 0.013.

4 Discussion

When considering reconstruction error as a metric of performance, GPs with the kernels *SM-Q2* and *Linear+SM-Q2* outperform simpler methods, supporting the hypothesis that modeling user-specific weekly activity patterns is beneficial. However, the performance does not improve further by expanding the kernels to capture longer-term seasonality patterns. On the contrary, we observe that more complex imputation schemes only marginally improve performance when used as a pre-processing step of time-series classification tasks. Specifically even if in the `M/C` group, GPs improves overall AUC slightly but significantly comparing with piecewise linear imputation used in previous work [3], when the rate of missing data is increased results are mixed with no method clearly outperforming others. We believe the difference may stem from the interactions between imputation methods and patterns of missingness. In the random missing condition, even when the missing rate is high, the data is rarely contiguously missing and GPs may still learn the kernel hyperparameters effectively. On the contrary, for the `hmm`($p_{00} = 0.7, p_{10} = 0.7$) case, where there are several episodes missing, each of which could be several days long, the piecewise linear method performs better than GP, which has more parameters to learn and could be overfitting easily. In both the `MH/NS` and the `M/C` groups, the performance decrease as the rate of missing data is increased. However, `M/C` AUC decreases much more slightly, supporting the findings in Quisel et al. [3], which reported demography and basic health features are already sufficient for the `M/C` group tasks. Taken in concert, our findings indicate that the MT-CNN models seem to be already robust to missing data, thus making an imputation pre-processing only marginally beneficial.

[5] For the weight data, we use mean imputation

References

[1] Wes McKinney. pandas: a foundational python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, pages 1–9, 2011.

[2] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 371(1984), 2013. ISSN 1364-503X. doi: 10.1098/rsta.2011.0550. URL `http://rsta.royalsocietypublishing.org/content/371/1984/20110550`.

[3] Tom Quisel, David Kale, and Luca Foschini. Intra-day activity better predicts chronic conditions. In *Machine Learning for Health Care Workshop, Neural Information Processing Systems (NIPS)*, 2016.

[4] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[5] Andrew G. Wilson and Ryan P. Adams. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1067–1075, 2013.

[6] Li-Fang Cheng, Gregory Darnell, Corey Chivers, Michael Draugelis, Kai Li, and Barbara Engelhardt. Sparse Multi-Output Gaussian Processes for Medical Time Series Prediction. *ArXiv e-prints*, March 2017.

[7] GPy. GPy: A gaussian process framework in python. `http://github.com/SheffieldML/GPy`, since 2012.