A Survey on Principles, Models and Methods for Learning from Irregularly Sampled Time Series: From Discretization to Attention and Invariance

Satya Narayan Shukla, Benjamin M. Marlin
College of Information and Computer Sciences
University of Massachusetts Amherst
Amherst, MA 01003
{ snshukla, marlin }@cs.umass.edu

Abstract

Irregularly sampled time series data arise naturally in many application domains including biology, ecology, climate science, astronomy, and health. Such data represent fundamental challenges to many classical models from machine learning and statistics due to the presence of non-uniform intervals between observations. However, there has been significant progress within the machine learning community over the last decade on developing specialized models and architectures for learning from irregularly sampled univariate and multivariate time series data. In this survey, we first describe several axes along which approaches differ including what data representations they are based on, what modeling primitives they leverage to deal with the fundamental problem of irregular sampling, and what inference tasks they are designed to perform. We then survey the recent literature organized primarily along the axis of modeling primitives. We describe approaches based on temporal discretization, interpolation, recurrence, attention and structural invariance. We discuss similarities and differences between approaches and highlight primary strengths and weaknesses.

1 Introduction

An irregularly sampled time series is a sequence of time-value pairs with non-uniform intervals between successive time points. This paper surveys approaches for modeling and learning from irregularly sampled time series generated from underlying continuously varying, real-valued univariate or multivariate functions. Such time series data naturally occur in domains where the observation process is constrained to a degree that prohibits regular observation of continuously varying phenomena.

Irregularly sampled time series occur in a number of scientific and industrial domains and is also a prominent feature of some types of data in the health domain. For example, observational studies of free-living animals in biology and ecology inevitably lack data points due to movement of subjects, weak transmitter reception, or poor weather or lighting conditions that hinder observation [Ruf, 1999]. In astronomy, measurements of properties such as the spectra of celestial objects are taken at times determined by seasonal and weather conditions as well as the availability of time on required instruments [Scargle, 1982]. In climate science, paleoclimate time series data are often irregular due to the difficulty in obtaining historical information [Schulz and Stattegger, 1997]. In electronic health records data, vital signs and other measurements are recorded at time points determined by a number of complex factors that may include the type of measurement, the patient's state of health, and the availability of clinical staff [Marlin et al., 2012]. Irregularly sampled time series data also commonly occur in a range of other areas with similar complex observation processes including
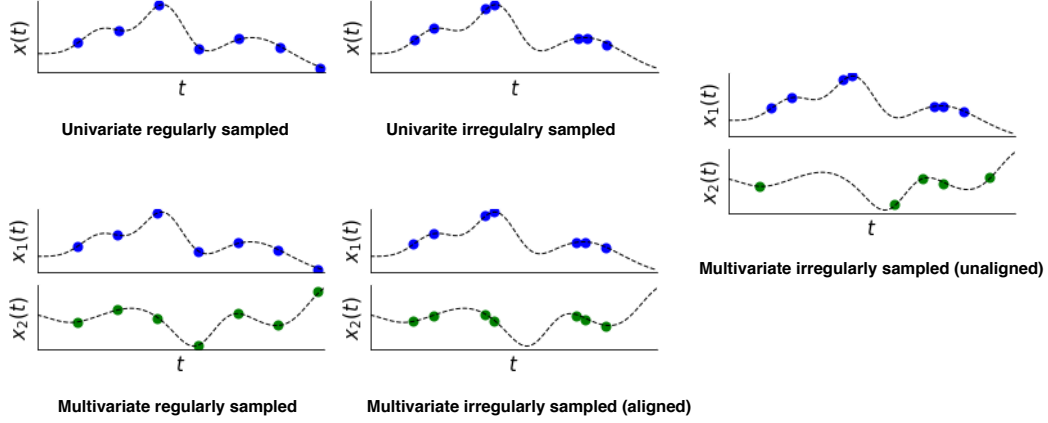
Figure 1: Illustration of regularly and irregularly sampled univariate and two-dimensional multivariate time series. Univariate irregularly sampled time series data are characterized by variable time intervals between observations. Multivariate irregularly sampled time series are also characterized by variable time intervals between observations, but can have aligned or unaligned observation times across dimensions. When observation time points are unaligned across dimensions, different dimensions also often contain different numbers of observations. Finally, while this figure only illustrates single univariate or multivariate time series, it is important to note that in data sets containing multiple univariate or multivariate time series as data cases, the observation times in different data cases are also typically unaligned and different data cases often also have different numbers of observations.

geology [Rehfeld et al., 2011], finance [Manimaran et al., 2006], economics [Kling and Bessler, 1985], meteorology [Mudelsee, 2002], and traffic analysis [Ye et al., 2010].

Irregularly sampled time series data (as shown in Figure 1) present fundamental challenges to many classical models from machine learning and statistics. To illustrate these challenges, consider the case of a supervised learning task where a model takes as input an irregularly sampled time series and must predict a scalar output. To learn such a model, we assume access to a data set $\mathcal{D}$ where each instance is a tuple $(\mathbf{s}_n, y_n)$. Here $\mathbf{s}_n$ represents the irregularly samples time series and y_n represents the corresponding prediction target. The first challenge with modeling such data is that the dimensionality of the inputs $\mathbf{s}_n$ can be different for different data cases n. This occurs naturally due to the fact that different data cases are likely to include different numbers of observations. The second challenge is that even if all data cases contain the same number of observations, we would expect different irregularly sampled instances to have observations recorded at different times. This results in a lack of temporal alignment across data cases.

For multivariate irregularly sampled time series data, we may also have a situation where different dimensions (variables or channels) within the same multivariate time series are observed at a different collection of time points. These features of irregularly sampled time series data invalidate the assumption of a coherent fixed-dimensional feature space that underlies most basic supervised and unsupervised learning models including K-nearest neighbours [Altman, 1992], decision trees [Quinlan, 1986], linear and logistic regression [Hastie et al., 2001, Hosmer Jr et al., 2013], linear, polynomial and RBF-kernel support vector machines [Cortes and Vapnik, 1995], multi-layer perceptrons [Hastie et al., 2001], K-means [Lloyd, 1982], mixtures models (with standard component distributions) [Mclachlan and Basford, 1988], factor analysis [Harman, 1976], PCA [Hotelling, 1933], autoencoders [Kramer, 1991], and more.

However, there has been significant progress over the last decade on developing specialized models and architectures for learning from irregularly sampled multivariate time series data within the machine learning community. This paper presents a survey of many of these approaches that focuses on categorizing methods in terms of a number of key properties including the underlying data representation they use, the fundamental mechanism they use to accommodate irregular sampling, and the types of machine learning tasks to which they can be applied.

We identify three primary underlying data representations for irregularly sampled time series that we refer to as *series-based*, *vector-based*, and *set-based* representations. In the series-based representa-

Modeling Primitives

- **Discretization** — Marlin et al. [2012], Lipton et al. [2016], Song et al. [2018], Harutyunyan et al. [2019], Bahadori and Lipton [2019], Bianchi et al. [2019], Fortuin et al. [2020]
- **Interpolation**
  - **Deterministic Interpolation** — Shukla and Marlin [2019, 2020], Li and Marlin [2020]
  - **Probabilistic Interpolation** — Ghassemi et al. [2015], Liu and Hauskrecht [2016], Li and Marlin [2016], Futoma et al. [2017], Soleimani et al. [2018]
  - **Similarity** — Lu et al. [2008], Li and Marlin [2015]
- **Recurrence**
  - **RNN-based** — Choi et al. [2016a], Neil et al. [2016], Pham et al. [2017], Baytas et al. [2017], Yoon et al. [2018b], Che et al.[2018a], Kim and Chi [2018], Cao et al. [2018], Che et al. [2018b], Luo et al. [2018], Yoon et al. [2019], Li and Xu [2019], Luo et al. [2019], Guo et al. [2019], Tan et al. [2020]
  - **ODE-based** — Chen et al. [2018], Rubanova et al. [2019], De Brouwer et al. [2019]
- **Attention** — Choi et al. [2016b], Song et al. [2018], Zhang et al. [2019], Xu et al. [2019], Horn et al. [2020], Shukla and Marlin [2020]
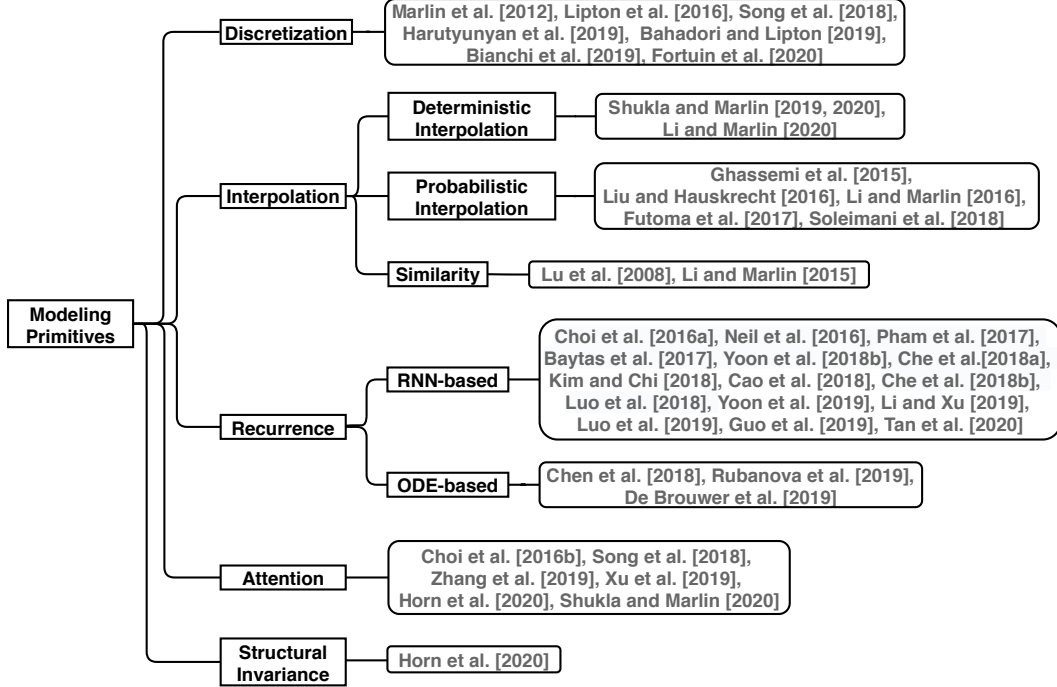- **Structural Invariance** — Horn et al. [2020]

Figure 2: This figure illustrates a taxonomy of methods based on modeling primitives. We define modeling primitives to be the fundamental approach that methods use to accommodate irregular sampling. We identify five high-level modeling primitives including discretization, interpolation, recurrence, attention and structural invariance.

tion, a multivariate irregularly sampled time series is viewed as consisting of a collection of univariate time series, each with its own collection of observation times and values. In the vector-based representation, a multivariate time series is viewed as a time series of vector-valued observations. When observations for different dimensions are not temporally aligned, the result is missing values in the vector-valued observations. The set-based representation views a multivariate irregularly sampled time series as a set of tuples of the form (t, d, x) where t is a time point, d is a dimension, and x is the observed value of dimension d at time t. As we will see, different approaches are based on different of these representations, leading to methods with different capabilities and limitations.

Different approaches also leverage different "modeling primitives" for accommodating irregular sampling. We define modeling primitives to be the basic building blocks or modules leveraged in larger and more complex models. We identify several such modeling primitives that specifically provide the interface between irregularly sampled time series data and more standard model components. These modeling primitives fall into several categories including approaches based on temporal discretization, interpolation, similarity, recurrence, attention and structural invariance. We categorize approaches in terms of their use of such modeling primitives, which can sometimes be obscured in larger and more complex models. In Figure 2, we provide a categorization of different approaches under a taxonomy based on modeling primitives.

The third categorization that we focus on is the set of inference tasks that a given approach is designed to solve. These tasks include detection, prediction, filtering, smoothing, interpolation, and forecasting. Some approaches can be applied to multiple of these tasks, while others can not. Understanding the range of problems a given approach can be applied to in a valid way is obviously important, but can again be unclear for some larger and more complex models.

The rest of this survey is organized as follows. In Section 2 we describe the categorization of representations for irregularly sampled time series. In Section 3, we define modeling primitives for irregularly sampled time series. In Section 4, we define modeling tasks for irregularly sampled time series. In Sections 5 to 9, we present a detailed discussion of specific models and approaches organized by their primary modeling primitives including temporal discretization, interpolation,
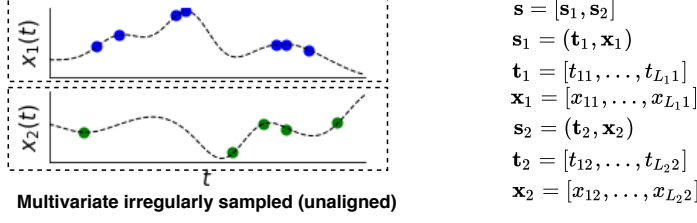
$$\mathbf{s} = [\mathbf{s}_1, \mathbf{s}_2]$$
$$\mathbf{s}_1 = (\mathbf{t}_1, \mathbf{x}_1)$$
$$\mathbf{t}_1 = [t_{11}, \ldots, t_{L_11}]$$
$$\mathbf{x}_1 = [x_{11}, \ldots, x_{L_11}]$$
$$\mathbf{s}_2 = (\mathbf{t}_2, \mathbf{x}_2)$$
$$\mathbf{t}_2 = [t_{12}, \ldots, t_{L_22}]$$
$$\mathbf{x}_2 = [x_{12}, \ldots, x_{L_22}]$$

Figure 3: Illustration of the series-based representation for a two-dimensional multivariate irregularly sampled time series. In the series-based representation, a d-dimensional multivariate irregularly sampled time series $\mathbf{s} = [\mathbf{s}_1, ..., \mathbf{s}_D]$ is represented as a collection of univariate irregularly sampled time series, one per dimension. Here $\mathbf{s}_d = (\mathbf{t}_d, \mathbf{x}_d)$ indicates the time series for dimension d. $\mathbf{t}_d$ indicates the collection of time points with observed values for dimension d while $\mathbf{x}_d$ indicates the corresponding collection of observed values.

similarity, recurrence, attention and invariance. In Section 10 we discuss data sets commonly used to evaluate models for irregularly sampled time series and in Section 11 we discuss the predictive performance of methods. In Appendix A, we provide pointers to open source implementations of many of the approaches described in this survey.

2 Data Representations for Irregularly Sampled Time Series

There are several possible data representations for multivariate irregularly sampled time series. While these representations are equivalent, they expose different properties and suggest different approaches to modeling. In all cases we will assume that a data set contains N data cases and that each data case consists of observations of D different variables through time. We being by describing what we will refer to as the series-based representation followed by the vector-based representation and the set-based representation.
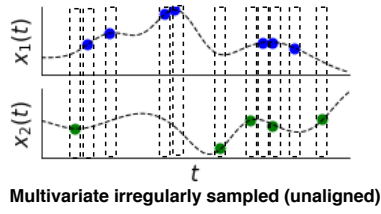
2.1 Series-Based Representation

As shown in Figure 3, the series-based representation of a multivariate irregularly sampled time series data set views the data for a single data case n as a collection of D univariate irregularly sample time series with one item in the collection for each of the D dimensions. We define L_{dn} to be the number of observations of variable d for data case n. We define $\mathbf{t}_{dn} = [t_{1dn}, ..., t_{L_{dn}dn}]$ to be the collection of time points at which variable d is observed for data case n. We define $\mathbf{x}_{dn} = [x_{1dn}, ..., x_{L_{dn}dn}]$ to be the corresponding collection of observed values. The data for time series n and variable d is then a univariate irregularly sampled time series $\mathbf{s}_{dn} = (\mathbf{t}_{dn}, \mathbf{x}_{dn})$. We define $\mathbf{s}_n = [\mathbf{s}_{1n}, ..., \mathbf{s}_{Dn}]$ to be the complete multivariate irregularly sampled time series for data case n.

We note that in this representation, there is no missing data. We represent only the observations available for each dimension. Further, this structure exposes the fact that in the fully general case of multivariate irregular sampling, different dimensions can be observed at different collections of time points with different total numbers of observations for the same data case. We also note that this representation can be wasteful in scenarios where all variables are observed at each time point. This motivates the vector-based representation, which we introduce next.
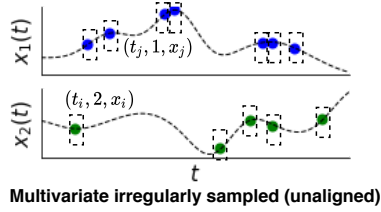
2.2 Vector-Based Representation

As shown in Figure 4, the vector-based representation of a multivariate irregularly sampled time series data set views the data for a single data case n as a single series of D dimensional vectors. We define L_n to be the number of time points with observations for data case n. We define $\mathbf{t}_n = [t_{1n}, ..., t_{L_nn}]$ to be the collection of time points with observations for data case n. Next, we define $\mathbf{x}_n = [\mathbf{x}_{1n}, ..., \mathbf{x}_{L_nn}]$ to be the corresponding collection of D-dimensional vector-valued observations. We again let x_{idn} be the observed value of dimension d at time point t_{in}. We define $\mathbf{s}_n = (\mathbf{t}_n, \mathbf{x}_n)$ to be the irregularly sampled time series for data case n. Under this representation, different data cases n can still have different observation time points as well as different numbers of observation time points. However, the observations across dimensions are assumed to be aligned in time.

4

$$\mathbf{s} = (\mathbf{t}, \mathbf{x}, \mathbf{r})$$
$$\mathbf{t} = [t_1, \ldots, t_L]$$
$$\mathbf{x} = [\mathbf{x}_1, \cdots, \mathbf{x}_L]$$
$$\mathbf{r} = [\mathbf{r}_1, \cdots, \mathbf{r}_L]$$

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix} \qquad x_{id} = \begin{cases} x_{id}, & \text{if } x_{id} \text{ is observed} \\ \text{NA}, & \text{otherwise} \end{cases}$$

$$\mathbf{r}_i = \begin{bmatrix} r_{i1} \\ r_{i2} \end{bmatrix} \qquad r_{id} = \begin{cases} 1, & \text{if } x_{id} \text{ is observed} \\ 0, & \text{otherwise} \end{cases}$$

Figure 4: Illustration of the vector-based representation for multivariate irregularly sampled time series in the case of two dimensions with unaligned observation times. In this representation, there is a single collection of time points $\mathbf{t}$. At each time point t_i, we define a D-dimensional vector-valued observation $\mathbf{x}_i$. In the general case, not all dimensions of $\mathbf{x}_i$ are observed, leading to the need to explicitly represent which dimensions are observed and which are missing. Following Little and Rubin [2014], we introduce a D-dimensional binary response indicator vector $\mathbf{r}_i$ at each time point t_i to indicate which dimensions are observed and which are missing. The complete representation for the time series is thus $\mathbf{s} = (\mathbf{t}, \mathbf{x}, \mathbf{r})$ where $\mathbf{t}$ is the collection of observation times, $\mathbf{x}$ is the collection of vector-valued observations (with missing values), and $\mathbf{r}$ is the collection of response indicators.

$$\mathbf{s} = \{(t_1, 2, x_1), (t_2, 1, x_2), (t_3, 1, x_3), \cdots\}$$

Figure 5: Illustration of the set-based representation for multivariate irregularly sampled time series in the case of two dimensions. In this representation, a D-dimensional multivariate irregularly sampled time series is represented as a set of (time, dimension, value) tuples, one for each observation.

When the vector $\mathbf{x}_{in}$ is not fully observed at time t_{in}, the vector-based representation results in the need to represent missing data. Following Little and Rubin [2014], the standard approach to representing which elements of $\mathbf{x}_n$ are observed and which are missing is to introduce an auxiliary response indicator series $\mathbf{r}_n = [\mathbf{r}_{1n}, \ldots, \mathbf{r}_{L_n n}]$ where $r_{idn} = 1$ if the value of x_{idn} is observed and $r_{idn} = 0$ otherwise. The full vector-based representation of an irregularly sampled time series with incomplete observations is thus $\mathbf{s}_n = (\mathbf{t}_n, \mathbf{x}_n, \mathbf{r}_n)$. Compared to the series-based representation, the vector-based representation can thus be substantially more space efficient, but only if all vectors are completely observed.

2.3 Set-Based Representation

As shown in Figure 5, the set-based representation of a multivariate irregularly sampled time series data set views the data for a single data case n as set of time-dimension-value tuples of the form (t_{in}, d_{in}, x_{in}). Here we let L_n represent the total number of observations across all time points and dimensions. The irregularly sampled time series for data case n is thus represented as $\mathbf{s}_n = \{(t_{in}, d_{in}, x_{in}) | 1 \le i \le L_n\}$. Like the series-based representation, the set-based representation does not require explicit representation of missing data. Unlike both the the series and vector-based representation, the time ordering of the data is not explicitly reflected in the structure of the set-based representation.

3 Modeling Primitives for Irregularly Sampled Time Series

While there has been a significant volume of work in recent years on modeling sparse and irregularly sampled time series data in the context of different tasks and applications, the number of fundamental approaches for accommodating irregular sampling itself is much more limited. In this section, we describe several "modeling primitives" for accommodating irregular sampling. These modeling prim-
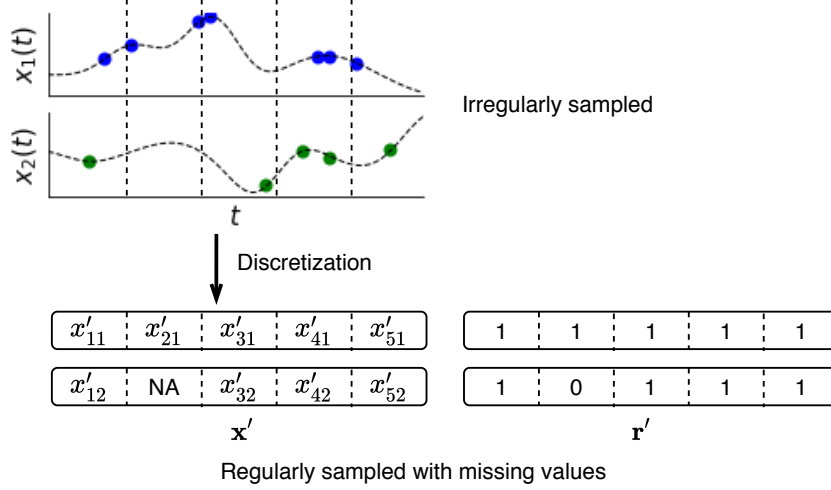
Figure 6: This figure illustrate the discretization of irregularly sampled time series into regularly spaced time series with missing values. The discretization process requires dividing the time axis into equal sized non-overlapping intervals and defining a value within each time interval based on the observed values falling within that interval. In cases where a given interval contains no observations on a particular dimension, the result is missing data. We can again represent these missing values via auxiliary response indicator vectors.

itives are the basic building blocks or modules for dealing with irregular sampling that are leveraged in larger and more complex models and methods for solving specific tasks using irregularly sampled time series. We discuss a number of modeling primitives including discretization, interpolation, similarity, recurrence, attention and structural invariance. Figure 2 provides a taxonomy of modeling primitives.

3.1 Discretization

Temporal discretization is a basic modeling primitive used to convert irregularly sampled time series data into a regularly sampled time series, as shown in Figure 6. To apply this primitive, one first defines a sequence of $K + 1$ regularly spaced reference time points $\tau_0, ..., \tau_K$. Given an irregularly sampled time series $\mathbf{s}_n$, we define a new regularly sampled time series in the vector-based representation with missing data indicators $\mathbf{s}'_n = (\mathbf{t}'_n, \mathbf{x}'_n, \mathbf{r}'_n)$. To begin, we let $\mathbf{t}'_n = [t'_1, ..., t'_K]$ for all n where $t'_i = (\tau_{i-1} + \tau_i)/2$.[1]

Next, we define a function that maps the observed values of $\mathbf{s}_n$ falling within the discretization window $[\tau_{i-1}, \tau_i)$ to the vectors $\mathbf{x}'_{in}$ and $\mathbf{r}'_{in}$. Assume there are $m > 0$ observations for dimension d in $\mathbf{s}_n$ within the interval $[\tau_{i-1}, \tau_i)$ and let their indices be $\mathbf{o} = [o_1, ..., o_m]$. We require a function of the form $\mathbf{x}'_{idn} = f([\mathbf{t}_{o_1 n}, ..., \mathbf{t}_{o_m n}], [\mathbf{x}_{o_1 dn}, ..., \mathbf{x}_{o_m dn}])$ that maps the set of observations within the discretization window to a single representative value. A common approach is to set this function to be the average of the m values. To complete the representation for this interval, we set $\mathbf{r}'_{idn} = 1$ to indicate that the interval has an observed value.

In the case where $m = 0$, the original time series has no observations of dimension d within the interval $[\tau_{i-1}, \tau_i)$. In this case we set $\mathbf{r}'_{idn} = 0$ to indicate no observations are available. The value set in $\mathbf{x}'_{idn}$ will typically be 0 or nan, but is not consequential as any correct algorithm for this representation will not use values in $\mathbf{x}'_{idn}$ when $\mathbf{r}'_{idn} = 0$ [Little and Rubin, 2014].

As we can see, the discretization primitive provides a reduction from the problem of modeling irregularly sampled time series to the problem of modeling regularly sampled time series that may contain missing data. As discretization windows become larger, the volume of missing data will generally decrease. However, more values will potentially also fall within the same discretization

---

[1]We choose the midpoint of the interval $[\tau_{i-1}, \tau_i)$ to represent the interval, but the starting point or any other consistent choice could also be made.
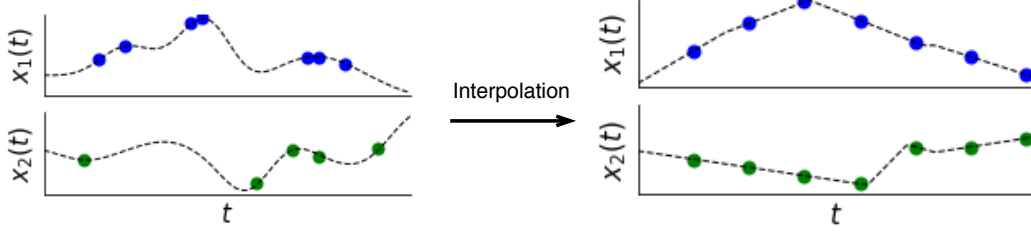
Figure 7: Illustration of interpolation in the case of a two-dimensional irregularly sampled time series. In this example, we illustrate the basic use of linear interpolation against a fixed set of reference time points. .

window leading to more aggregation. This may remove information needed to perform some tasks. As discretization windows become shorter, aggregation effects are reduced, but the length of the time series increase as does the volume of missing data. Thus, the window size becomes an important hyperparameter of methods based on this principle.

We note that when discretization does result in significant volumes of missing data, the problem of dealing with the resulting missing data can itself be highly non-trivial as the presence of incomplete data also violates the assumptions of most standard discriminative machine learning methods. Imputation methods can be used as a final stage of pre-processing to address the problem of missing data, but care is needed when reflecting uncertainty is important. A number of recent deep learning approaches have addressed the problem of providing flexible single and multiple imputation methods [Yoon et al., 2018a, Li et al., 2019, Mattei and Frellsen, 2019, Śmieja et al., 2018]. Many simpler methods are also commonly used for time series including mean imputation, zero imputation and forward-filling.

3.2 Interpolation

Interpolation primitives provide an approach to accommodating irregular sampling that is closely related to discretization but provides improved flexibility with potentially fewer ad-hoc assumptions. Interpolation methods are often used to provide an interface between irregularly sampled time series data and models for either fixed-dimensional feature spaces or for variable length sequences.

Similar to discretization, we begin by defining a set of K reference time points $\boldsymbol{\tau} = [\tau_1, ..., \tau_K]$. In the case of a length L_n univariate irregularly sampled time series $\mathbf{s}_n = (\mathbf{t}_n, \mathbf{x}_n)$, the interpolated output $\mathbf{x}'_{in}$ at reference time point τ_i is computed as shown in Equation 1. The interpolated time series is then given by $\mathbf{s}'_n = (\boldsymbol{\tau}, \mathbf{x}'_n)$. The function $\kappa_\theta()$ is a similarity kernel that puts higher weight on pairs of time points $\tau_i, \mathbf{t}_{jn}$ that are closer together in time. This function can depend on a set of learnable parameters θ. The squared exponential kernel function $\kappa_\theta(t, t') = \alpha \exp(-\beta(t - t')^2)$ is a common choice in the literature for non-linear interpolation (with $\theta = [\alpha, \beta]$).

$$\mathbf{x}'_{in} = \frac{\sum_j \kappa_\theta(\tau_i, \mathbf{t}_{jn}) \, \mathbf{x}_{jn}}{\sum_j \kappa_\theta(\tau_i, \mathbf{t}_{jn})} \tag{1}$$

For multivariate irregularly sampled time series, there are more possibilities in terms of the construction of interpolation methods as these methods can account for both correlation in time and correlation across different dimensions. One basic approach motivated by the series-based view of a multivariate time series is to separately interpolate each dimension of the time series using a univariate interpolation approach as shown in Equation 1 and in Figure 7, but with the same set of reference time points used across all dimensions. This ignores cross-dimension correlations, but provides regularly spaced output to which further modeling can be applied to account for cross-dimensional correlations.

One potential issue when inter-observation times exhibit significant variability is variable uncertainty in interpolated values. The use of Gaussian process regression (GPR) models can provide a better alternative to deterministic interpolation methods in such cases [Rasmussen and Williams, 2006]. The primary strength of GPR models is that they provide a joint posterior distribution over an arbitrary collection of reference time points $\boldsymbol{\tau} = [\tau_1, ..., \tau_K]$ conditioned on an input irregularly sampled time

series $\mathbf{s}_n$. This posterior distribution is a joint Gaussian defined in terms of a mean $\mathbf{m}_n$ over $\boldsymbol{\tau}$ and a corresponding covariance matrix $\boldsymbol{S}_n$ [Rasmussen and Williams, 2006].

This approach is most straightforward to apply in the univariate case where $\mathbf{s}_n = (\mathbf{t}_n, \mathbf{x}_n)$. The model is defined in terms of a covariance function $\kappa_\theta(\cdot, \cdot)$ with parameters θ. Unlike the case of basic deterministic interpolation methods mentioned previously, GPR models require the covariance function to be a valid Mercer kernel. The GPR model also includes a noise variance term σ^2 and a mean function $\mu_\phi()$. Under this model, the posterior probability density over the values $\mathbf{x}'_n = [x'_{1n}, ..., x'_{Kn}]$ is given by $p(\mathbf{x}'_n | \tau, \mathbf{s}_n, \theta, \phi, \sigma) = \mathcal{N}(\mathbf{x}'_n; \mathbf{m}_n, \mathbf{S}_n)$ where the conditional mean and covariance matrices $\mathbf{m}_n$ and $\mathbf{S}_n$ are shown below. $\mathbf{K}_{\mathbf{t}, \mathbf{t}'}$ denotes the covariance matrix defined by $[\mathbf{K}_{\mathbf{t}, \mathbf{t}'}]_{ij} = \kappa_\theta(\mathbf{t}_i, \mathbf{t}'_j)$.

$$\boldsymbol{m}_n = \mu_\phi(\boldsymbol{\tau}) + \mathbf{K}_{\boldsymbol{\tau}, \mathbf{t}_n}(\mathbf{K}_{\mathbf{t}_n, \mathbf{t}_n} + \sigma^2 \mathbf{I})^{-1}(\mathbf{x}_n - \mu_\phi(\mathbf{t}_n)) \tag{2}$$

$$\boldsymbol{S}_n = \mathbf{K}_{\boldsymbol{\tau}, \boldsymbol{\tau}} - \mathbf{K}_{\boldsymbol{\tau}, \mathbf{t}_n}(\mathbf{K}_{\mathbf{t}_n, \mathbf{t}_n} + \sigma^2 \mathbf{I})^{-1}\mathbf{K}_{\mathbf{t}_n, \boldsymbol{\tau}} \tag{3}$$

One possible application of GPR to the interpolation problem is to define $\mathbf{s}'_n = (\boldsymbol{\tau}, \boldsymbol{m}_n)$. This approach simply uses the posterior mean on τ as the interpolated values. This approach has no particular advantage over deterministic interpolation methods using direct smoothing kernels introduced above as it discards all posterior uncertainty. A potentially better approach is to draw J samples of the form $\mathbf{x}'_{nj} \sim \mathcal{N}(\mathbf{x}'_n; \mathbf{m}_n, \mathbf{S}_n)$ for $1 \leq j \leq J$ and to construct J interpolated time series using $\mathbf{s}'_{nj} = (\boldsymbol{\tau}, \mathbf{x}'_{nj})$. The ensemble of interpolants $\{\mathbf{s}'_{nj}\}_{j=1:J}$ will exhibit greater cross-sample variation in regions where there are few observations in $\mathbf{s}_n$, reflecting natural uncertainty due to observation sparsity. Such an ensemble can be used in down-stream analyses in a way that is analogous to multiple imputations in a missing data problem.

However, as we can see, applying GPR in this way can be quite expensive as the naive computation of the matrix inverse in the posterior mean $\mathbf{m}_n$ and covariance matrix $\mathbf{S}_n$ computations requires time cubic in L_n. While faster approximations are available for these computations as well as for drawing samples from these posteriors, GPR-based approaches are typically still much slower than deterministic interpolation methods. We also note that as with deterministic interpolation methods, GPR has a number of parameters to estimate. These can be fit in a number of ways including the use of marginal likelihood maximization when given a data set consisting of irregularly sampled time series.

The discussion of GPR thus far has focused only on the case of univariate irregularly sampled time series. Like with deterministic interpolation methods, GPR-based methods can also be applied to the case of multivariate irregularly sampled time series. The two basic approaches are again to leverage the series-based view and produce a posterior distribution or set of samples over each dimension separately, or to define a joint covariance function that applies both over time and across dimensions. A full GPR model over all dimensions (leveraging a multi-task or multi-output Gaussian Process) has the advantage of being able to leverage covariance structure across all dimensions while producing a posterior distribution for any individual dimension at any time point. However, defining flexible and valid Mercer kernels for multivariate problems can be quite challenging.

3.2.1 Similarity

The previous section discusses the use of similarity functions and kernels within individual irregularly sampled time series. These approaches essentially aim to interpolate a single time series by considering the local similarity of time points and/or dimensions. However, we can also consider applying similarity to pairs of multivariate irregularly sampled time series $\mathbf{s}$ and $\mathbf{s}'$, providing a similarity-based modeling primitive.

At a minimum, a similarity kernel between irregularly sampled time series $\mathcal{K}_\theta(\mathbf{s}, \mathbf{s}')$ needs to provide non-negative values and needs to encode the notion that when $\mathbf{s}$ and $\mathbf{s}'$ are more "similar" in some sense, $\mathcal{K}_\theta(\mathbf{s}, \mathbf{s}')$ takes larger values. Any such kernel function $\mathcal{K}_\theta(\mathbf{s}, \mathbf{s}')$ can then be used to solve machine learning tasks that can be formulated in terms of similarity between data points. Basic examples include K-nearest neighbor classification and regression methods [Altman, 1992]. If the function $\mathcal{K}_\theta(\mathbf{s}, \mathbf{s}')$ also satisfies the requirements of a Mercer kernel function, a wider set of classical kernel methods can be applied including support vector classification [Cortes and Vapnik, 1995], support vector regression [Smola and Schölkopf, 2004], kernel ridge regression [Hastie et al., 2001], kernel logistic regression [Wahba, 1999], kernel principle components analysis [Schölkopf et al., 1998], etc.
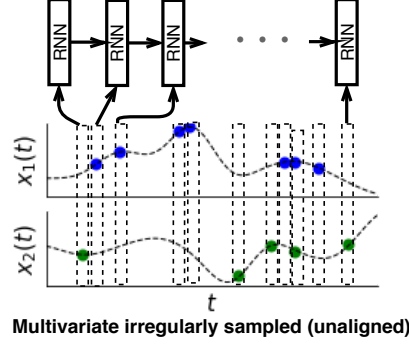
Multivariate irregularly sampled (unaligned)

Figure 8: This figure illustrates the recurrence based modeling primitive for a two-dimensional irregularly sampled time series with unaligned observations. In this case, we show the use of recurrent neural network (RNN) cell that integrates the input at each time point with the latent state from the previous time point. Basic RNN models view time series as general sequences and require completely observed input vectors. However, many RNN-based models have been developed that explicitly represent time to deal with irregular sampling and integrate methods for dealing with missing data.

A number of kernels have been proposed for irregularly sampled time series, some of which are valid Mercer kernels. For example, Lu et al. [2008] proposed an approach for the univariate case that can be thought of as constructing a latent function conditioned on an observed time series **s** using a temporal similarity kernel. The construction of this latent function is identical to the construction of the Gaussian process mean function described in the previous section. Lu et al. [2008] then show that a kernel between two such latent functions induced by two different irregularly sampled time series **s** and **s**′ can be computed only in terms of the observed values and time points that define the two time series. This construction is quite elegant in that it avoids the need to materialize the latent functions at reference time points. However, since it measures the similarity between deterministic latent functions, it discards all uncertainty due to sparse sampling. Other approaches include kernels that consider alignment and warping when assessing similarity [Shimodaira et al., 2001, Cuturi, 2011].

3.3 Recurrence

As noted in the previous sections, one of the issues when dealing with time series in general is the possibility of different data cases containing different numbers of observations. This problem can be dealt with by defining models that appeal to primitives based on recurrence. The key property of such primitives is that they use a fixed, finite set of parameters to model sequences of arbitrary length. Examples include classical autoregressive structures used in Markov models, hidden Markov models [Rabiner, 1990], conditional random fields [Lafferty et al., 2001], and recurrent neural networks [Rumelhart et al., 1986]. In this section, we will focus on recurrent neural network (RNN) models as an example recurrent primitive as shown in Figure 8.

An RNN provides a basic building block for modeling fully observed multivariate sequences $\mathbf{z}_n = [\mathbf{z}_{1n}, ..., \mathbf{z}_{L_n n}]$. The key is to define a cell whose hidden state $\mathbf{h}_i$ is updated based on the previous state $\mathbf{h}_{i-1}$ and the current input $\mathbf{z}_{in}$ through a non-linear function $f_\theta()$ with parameters θ. The model can also optionally output values $\hat{\mathbf{y}}_{in}$ via a second function $o_\phi()$ applied to the hidden state. The model is thus able to process sequences of arbitrary length due to the fact that it processes elements of the sequence one at a time and the parameters θ and ϕ are position-independent.

$$\mathbf{h}_i = f_\theta(\mathbf{h}_{i-1}, \mathbf{z}_{in}) \tag{4}$$
$$\hat{\mathbf{y}}_{in} = o_\phi(\mathbf{h}_i) \tag{5}$$

Due to the ability to process sequences of arbitrary length, RNNs are a useful building block for modeling time series in general. In the case of univariate irregularly sampled time series or multivariate series with completely observed vector-valued observations, an RNN can be directly applied simply by processing the data in time order and discarding the values of the time points. Such an approach could be suitable for problems where the variation in inter-observation intervals is relatively small. When that is not the case, discarding the time values makes the model invariant to time gaps in a way that may be harmful to performance on tasks of interest.
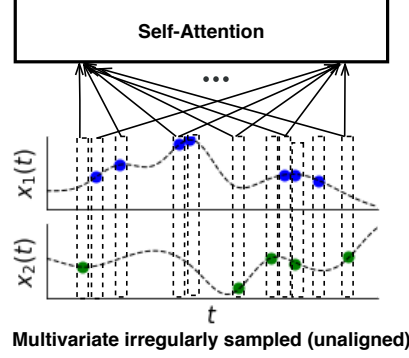
9

Figure 9: This figure illustrates the attention modeling primitive for two dimensional irregularly sampled time series. The attention-based primitive processes time series in parallel instead of sequentially as for an RNN. Attention models learn which regions of an input time series to attend to when computing outputs at different points in time by leveraging positional or time encodings.

A number of simple approaches can be taken to solve this problem. One approach is to append the time points or inter-observation intervals to the vector-valued observations yielding the sequence of values $\mathbf{x}'_{in} = [\mathbf{x}_{in}, t_{in}]$ or $\mathbf{x}'_{in} = [\mathbf{x}_{in}, t_{in} - t_{i-1n}]$. An RNN model can then be applied to this modified sequence. In theory, a sufficiently powerful non-linear mapping $f_\theta()$ should be able to account for irregular sampling based on this representation. In practice, this approach can under-perform other approaches to dealing with time.

Recent work on ordinary differential equation (ODE) models [Chen et al., 2018] in machine learning provides an alternative recurrence-based solution with better properties than traditional RNNs in terms of their ability to accomodate irregularly sampled data. In these models, ODEs are used to evolve the hidden state between continuous time observations. At each observation time-point, the hidden state is updated using a standard RNN update. These models are often referred to as ODE-RNNs. The hidden state update equations are defined below:

$$\mathbf{h}'_i = \text{ODESolve}(g_\gamma, \mathbf{h}_{i-1}, (t_{i-1n}, t_{in})) \tag{6}$$

$$\mathbf{h}_i = f_\theta(\mathbf{h}'_i, \mathbf{x}_{in}) \tag{7}$$

The function g_γ is a time-invariant function that takes the value at the current time step and outputs a gradient: $\frac{\partial \mathbf{h}(t)}{\partial t} = g_\gamma(\mathbf{h}(t))$. This function is parameterized using a neural network. The ODE defined by $g_\gamma(\mathbf{z}(t))$ is then solved at the given time point t_{in} to produce a new hidden state $\mathbf{h}'_i$. This hidden state is further updated to fold in the input value at time t_{in}, yielding the final hidden state value $\mathbf{h}_i$ at time t_{in}. This model does not explicitly depend on t_{in} or $t_{in} - t_{i-1n}$ when updating the hidden state, but does depend on time implicitly through the resulting latent state of the dynamical system.

The ODE-RNN model primitive is a substantially more elegant approach to dealing with irregular sampling than adaptations of discrete time RNNs, but it can also be substantially slower due to the need to repeatedly apply an ODE solver. Further, we note that when irregularly sampled time series are multivariate, but the vector-valued observations are not fully observed, the application of both standard RNN models and ODE-RNNs becomes significantly more challenging as both models require a fully observed $\mathbf{x}_{in}$ vector in order to update the hidden state.

A common baseline approach is to perform some form of imputation by defining a new sequence of values $\mathbf{x}'_{in} = \mathbf{r}_{in} \mathbf{x}_{in} + (1 - \mathbf{r}_{in}) \tilde{\mathbf{x}}_{in}$ where $\tilde{\mathbf{x}}_{in}$ is the value to impute when the response indicator $\mathbf{r}_{in} = 0$. Forward filling, zero imputation, and mean imputation can sometimes be reasonable options when the volume of missing data is low.

3.4 Attention

Attention models have become a key modeling component for many machine learning tasks including image caption generation, speech recognition and neural translation [Vaswani et al., 2017]. In self-attention models, each element of a sequence learns which other elements of a sequence to attend to
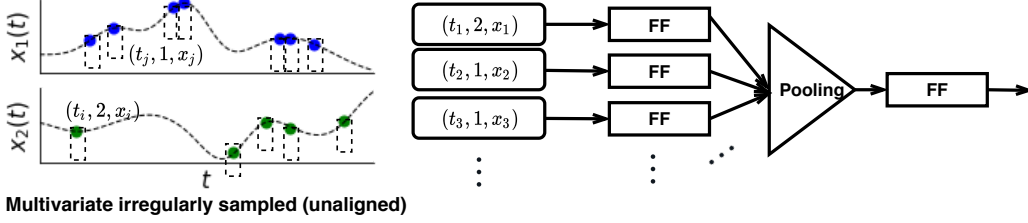
Figure 10: An illustration of the structural invariance-based modeling primitive. Inspired by the set-based view of a multivariate irregularly sampled time series, this approach processes individual (time, value, dimension) tuples via an encoding function and then pools over the output of all such tuples. The model structure thus does not reflect the time ordering of the data in any way.

without relying on recurrent network structures. Self-attention models offer computational advantages over RNNs since sequence processing can be fully parallelized. Self-attention is a particular instance of scaled dot-product attention which is defined as:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{C}}\right)\mathbf{V} \tag{8}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ denote the query, key, value representation respectively, and C is the dimension of the key or query representation. In self-attention, all of the queries, keys and values come from the input sequence. Self-attention relies on positional encoding, a vector representation for each position in the sequence to recognize and capture the sequential ordering information. The above defined $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ matrices are typically linear projections of position-value joint representations.

To provide a primitive for irregular sampling (as shown in Figure 9), time values can be converted into a vector representation similar to positional encoding and concatenated with the observation value as described for RNNs. Missing values in vector-valued observations are also problematic for attention-based modules, which (like standard RNNs) expect fully observed vectors as input. Approaches based on attention thus also need to solve the problem of missing dimensions, and a variety of imputation solutions can be used as described previously.

3.5 Structural Invariance

While the modeling primitives presented to this point all attempt to represent the time ordering of observations in the structure of the model this is not strictly required. Leveraging such structural invariance has the potential to greatly simplify learning problems.

Recently, set-based neural network approaches (as shown in Figure 10) have been employed to deal with irregularly sampled data and they exactly express this form of structural invariance. Set-based model architectures [Zaheer et al., 2017] support variable length sequences, partially observed vectors, sparse observations and irregular intervals between observation times while avoiding the need for temporal discretization, imputation and interpolation. These approaches leverage the set-based representation of an irregularly sampled time series data set $\mathbf{s}_n = \{(t_{in}, d_{in}, x_{in}) | 1 \leq i \leq L_n\}$. As shown below, such approaches produce an encoding of an input irregularly sampled time series by applying an initial encoder f_θ to individual time-dimension-value tuples.

$$\mathbf{h} = g_\phi(\text{pool}(\{f_\theta(t_{in}, d_{in}, x_{in}) | 1 \leq i \leq L_n\})) \tag{9}$$

The approach then performs a pooling operation over all initial encodings in a way that is completely invariant to the temporal structure of the data. The output of pooling is mapped through one additional set of encoding layers g_ϕ to produce the final representation. Commonly used pooling functions are max, mean and sum.

4 Inference Tasks

In this section, we define time series inference tasks. While these tasks are not specific to the case of irregularly sampled time series, carefully specifying inference tasks is necessary to properly
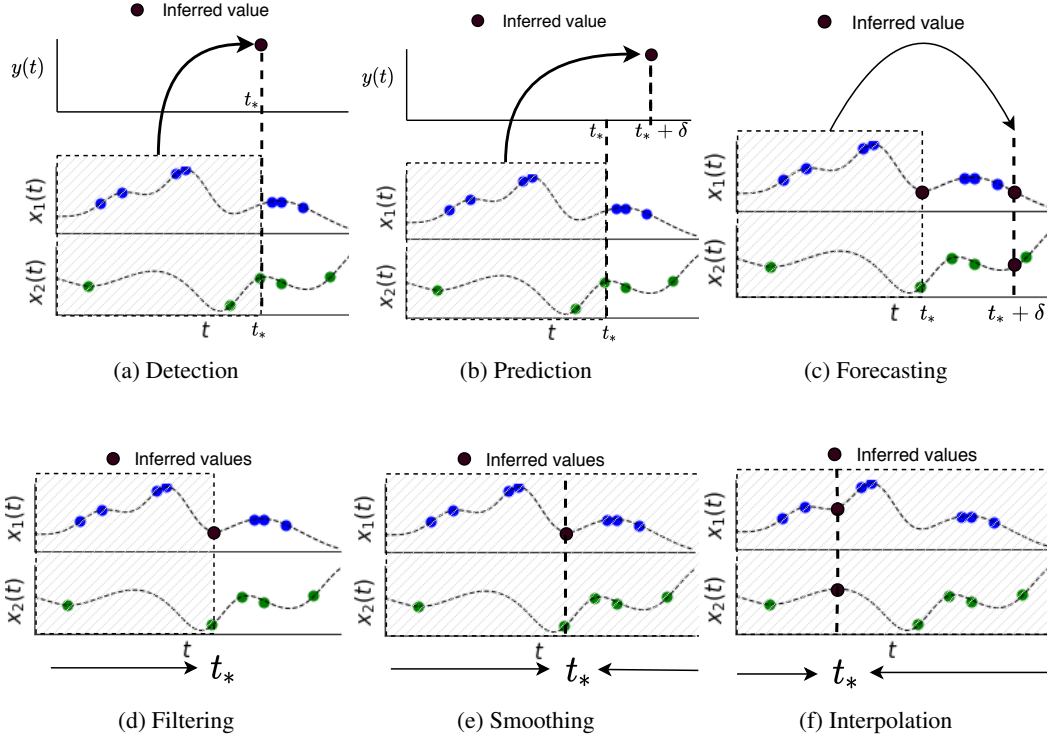
(a) Detection (b) Prediction (c) Forecasting

(d) Filtering (e) Smoothing (f) Interpolation

Figure 11: This figure illustrates time series inference tasks. (a) The detection task involves predicting the target values $y[t_*]$ at time t_* conditioning on the observations available up to and including time t_*. (b) The prediction task requires inferring the prediction target value $y[t_* + \delta]$ at time $t_* + \delta$ for $\delta > 0$ by conditioning on the observations available up to and including time t_*. (c) The forecasting task requires inferring $\mathbf{x}[t_* + \delta]$ by conditioning on the observations up to an including time t_*. (d) The filtering task requires inferring missing variables at time t_* by conditioning on the observations up to an including time t_*. (e) The smoothing task requires inferring missing variables at time t_* using all observed data. (f) Interpolation task requires inferring the values of $x[t_*]$ at time t_* using all observed data.

categorize models for learning from irregularly sampled time series in terms of the tasks that they can perform. In the definition of these tasks, it will be important to specify the time ranges that are conditioned on for a given input irregularly sampled time series. We will use the notation $\mathbf{s}[:t]$ to denote all of the data contained in time series $\mathbf{s}$ that are observed up to and including time t. We will use $\mathbf{x}[t]$ to refer to the vector of observations available at time t, some (or all) of which may be missing. For brevity, we will use $\mathbf{x}^m[t]$ to indicate the sub-set of dimensions with values that are missing at time t. Some of the tasks we consider are supervised in the sense that they involve learning to infer values that are distinct from the input irregularly sampled time series. We denote these outputs at time t by $\mathbf{y}[t]$. Below we define and discuss six inference tasks.

Definition 1 Detection: *Inferring prediction target values $\mathbf{y}[t_*]$ at time t_* conditioning on the observations $\mathbf{s}[:t_*]$ available up to and including time t_*.*

The detection task is to infer the value of the prediction target variable at time t_*. All time series data observed up to and including time t_* can be conditioned on when making this inference. In machine learning, the inference for any quantity that is not known is often referred to as a "prediction," but in this context we reserve the term "prediction" to refer to a task where the inference is for the value of the output variable at a time that is in the relative future of the time point t_* at which the inference is made, as we describe next.

Definition 2 Prediction: *Inferring prediction target values $\mathbf{y}[t_* + \delta]$ at time $t_* + \delta$ (for $\delta > 0$) conditioning on the observations $\mathbf{s}[:t_*]$ available up to and including time t_*.*

12

In the prediction problem, t_* refers to the time point at which the prediction is made and $t_* + \delta$ refers to the time point at which the value of the target variable is to be predicted. We assume $\delta > 0$. The case $\delta = 0$ recovers the detection problem.

We note that in some supervised problems, the prediction target variables may not have time stamps explicitly associated with them. Such problems include whole time series regression and classification where an input time series $\mathbf{s}$ is associated with a single scalar output y. These problems are equivalent in structure to the prediction problem defined above where all available data are allowed to be conditioned on when inferring the output value.

Definition 3 Filtering: *Inferring missing variables* $\mathbf{x}^m[t_*]$ *at time* t_* *by conditioning on the observations* $\mathbf{s}[:t_*]$ *up to an including time* t_*.

The filtering task is the analog of the detection task but where the inference of interest is about the values of the multivariate time series itself. In the irregularly sampled setting, it may be that some dimensions of $\mathbf{x}[t_*]$ are observed at time t_*. In this case, these values and any values observed before this time point can be used to infer the unobserved values $\mathbf{x}^m[t_*]$.

Definition 4 Smoothing: *Inferring the values of* $\mathbf{x}^m[t_*]$ *at time* t_* *using the observed data in* $\mathbf{s}$.

The smoothing task is similar to the filtering task except that all observed data can be used to infer the unobserved values $\mathbf{x}^m[t_*]$. This includes observations in both the relative past and future of $\mathbf{x}^m[t_*]$.

Definition 5 Interpolation: *Inferring the values of* $\mathbf{x}[t_*]$ *at time* t_* *using the observed data in* $\mathbf{s}$.

The interpolation task is similar to the filtering task except that in the interpolation task, we infer the values of all variables at time t_*, not just those that are unobserved. This distinction is only relevant in scenarios when the time point t_* coincides exactly with an observation time point contained in the time series $\mathbf{s}$.

Definition 6 Forecasting: *Inferring* $\mathbf{x}[t_* + \delta]$ *(for* $\delta > 0$*) by conditioning on the observations* $\mathbf{s}[:t_*]$ *up to an including time* t_*.

The forecasting task is the analog of the prediction task, but where we seek to predict the value of the time series itself at a future time point. In this problem, t_* again refers to the time point at which the forecast is made and $t_* + \delta$ for $\delta > 0$ refers to the time point about which the forecast is made. All observations up to and including time t_* can be used to compute a forecast at time t_*. δ is often referred to as the forecast horizon, the amount of time into the future we are forecasting the value of the time series.

5 Discretization-Based Approaches

In this section, we discuss approaches to modeling irregularly sampled time series that leverage discretization as their primary modeling primitive for accommodating irregular sampling.

Marlin et al. [2012], Lipton et al. [2016], Bahadori and Lipton [2019] and Harutyunyan et al. [2019] all apply discretization as a modeling primitive for dealing with irregular sampling. These papers all discretize time into consecutive, hour-long, non-overlapping intervals within a fixed time interval $[0, T]$ common to all data cases. The application of discretization under these conditions enables the use of models that operate on fixed-dimensional vectors. However, in all of the tasks considered in these papers, the application of discretization results in missing data that must be dealt with, as well as instances where aggregation is required within time intervals.

Marlin et al. [2012] make the assumption that the missing data is missing at random and apply probabilistic mixture models that can efficiently deal with missing data under this assumption. This allows models to be learned without requiring explicit imputation. Marlin et al. [2012] leverage this capability of the probabilistic mixture model framework to define a generative mixture of experts classifier for whole time series classification. The case of multiple values for a given dimension being defined within the same time interval is dealt with via averaging.

Lipton et al. [2016] apply discretization followed by an RNN to build a whole time series classification model. This approach requires explicit imputation of missing data created during the discretization

step. They consider two basic approaches: forward-filling and zero imputation. Missing values are replaced with zeros in the zero-imputation strategy. In the forward-filling approach, a missing value $\mathbf{x}_{idn}$ at time t_{in} on dimension d is set to the last observed value on dimension d (e.g., the value of $\mathbf{x}_{jdn}$ where j is such that t_{jn} is the largest time value satisfying $t_{jn} < t_{in}$ and $\mathbf{r}_{jdn} = 1$). In the absence of previous measurements (or if the variable is missing entirely), it is replaced with the-median estimated over all measurements in the training data.

One important feature of missing data problems is the potential for the sequence of observation times to itself be informative [Little and Rubin, 2014]. Since the set of response indicators $\mathbf{r}_{idn}$ is always observed, this information is easy to condition on as well. This approach was also used by Lipton et al. [2016] where they include as input to the RNN both the observed and imputed values of the multivariate time series and the values of the response indicator vector. In addition to the binary indicator variable, Lipton et al. [2016] add hand-engineered features derived from the response indicator time series such as mean and standard deviation of indicator variables for each time series.

Harutyunyan et al. [2019] also consider the application of discretization followed by RNNs. They apply averaging or selection of the last time point in an interval to deal with multiple observations in the discretization window, and use mean imputation or forward filling to deal with missing values. Harutyunyan et al. [2019] augment standard RNN models by predicting outputs for multiple tasks jointly. Unlike in standard RNNs where the hidden state from the last position in the sequence is decoded to predict supervised outputs, Harutyunyan et al. [2019] apply supervision at each time step. This framework has been applied to whole time series classification tasks, detection tasks and prediction tasks.

Song et al. [2018] follow Harutyunyan et al. [2019] and also discretize the sparse and irregularly sampled time-series data into hour long intervals, and use similar imputations for dealing with missing data. This model has been applied to the tasks of whole time series classification, detection and prediction. However, Song et al. [2018] adopt the multi-head attention mechanism similar to Vaswani et al. [2017] instead of an RNN as the primary model structure. They use positional encoding to incorporate the temporal order into the representation learning. They use the dense interpolation technique [Trask et al., 2015] to obtain a unified representation for a sequence.

Several approaches have also employed an encoder-decoder framework for learning with missing data in time series, which can also be applied to irregularly sampled time series after discretization. Bianchi et al. [2019] proposed an autoencoder-based approach to learn representations of multivariate time series with missing data for the whole time series classification and smoothing tasks. They use a standard RNN as the decoder. For the encoder, they use a bi-directional RNN where they combine the hidden layer output of forward and backward RNNs using a fully-connected layer. In the presence of missing data, they replace the missing values with zero or the mean while encoding, and use the decoder as a final imputer. Bianchi et al. [2019] also introduce a kernel alignment procedure to preserve the pairwise similarities of the inputs in the learned representations. These pairwise similarities are encoded in a positive semi-definite matrix that is also passed as input to the model. Fortuin et al. [2020] proposed a variational autoencoder (VAE) [Kingma and Welling, 2014, Rezende et al., 2014] approach for the task of smoothing in multivariate time series with a Gaussian process prior in the latent space to capture temporal dynamics.

In summary, temporal discretization provides a solution for learning from irregularly sampled time series that replaces the problem of irregular observation intervals with the problem of missing data. While convenient in some repects, this approach requires several ad-hoc choices including the width of the discretization windows and the aggregation function used within windows. In the next section, we turn to methods based on the use of interpolation as a modeling primitive. These approaches still require specifying a discrete set of interpolation points, but use more sophisticated global methods for defining values at the interpolation points.

6 Interpolation-Based Approaches

In this section we discuss methods for learning from irregularly sampled time series that are based on interpolation as a modeling primitive. Similar to discretization methods, interpolation methods require specifying a discrete set of reference time points. However, they materialize interpolants at these time points in a way that can leverage all available observations in the input if desired. This can make them much more powerful than discretization methods with ad-hoc local aggregation functions.

To begin, Shukla and Marlin [2019] proposed the interpolation-prediction network (IPNet) framework, which uses several semi-parametric radial basis function (RBF) interpolation layers to produce multiple interpolants given an irregularly sampled multivariate time series as input. This framework leverages the series-based view of an irregularly sampled time series. The first layer of the interpolation network separately transforms each of the D dimensions, creating several intermediate interpolants. The second interpolation layer merges information across all dimensions at each reference time point by taking into account learnable correlations across all dimensions. The prediction network takes the output of the interpolation network as its input and produces a prediction. The prediction network can consist of any standard supervised neural network architecture (fully-connected feed forward, convolutional, recurrent, etc.). This model has been applied to the tasks of whole time series regression and classification.

One of the drawbacks of the IPNet framework is that it uses fixed temporal RBF kernels with adjustable parameters. This biases the model to only being able to express certain notions of temporal similarity. Shukla and Marlin [2020] present a related continuous-time interpolation-based approach that replaces the fixed RBF kernels in the IPNet framework with learnable temporal similarity functions constructed using learnable embeddings of continuous time values and a temporal attention mechanism [Vaswani et al., 2017]. This added flexibility appears to lead to improved predictive performance. This framework has been applied to the interpolation task, whole time series classification task, and detection task.

Li and Marlin [2020] proposed an encoder-decoder architecture for modeling irregularly sampled time series data. The encoder in this framework is based on a piecewise linear function, which learns an interpolation on a fixed set of reference time points. This framework uses a kernel smoother as a decoder to interpolate at arbitrary times.

One of the disadvantages of using deterministic kernel smoothing-based interpolation approaches such as IPNets is that they do not reflect uncertainty due to long intervals with no observations. An alternative line of work on interpolation-based approaches instead leverages Gaussian processes as a building block, enabling uncertainty propagation as described previously [Rasmussen and Williams, 2006]. In response to the shortcomings of the kernel-based approach of Lu et al. [2008], Li and Marlin [2015] developed a two-step interpolation-based approach to modeling the similarity between sparse and irregularly sampled time series in a way that is sensitive to uncertainty. In the first step of this approach, they use marginal likelihood maximization to fit a Gaussian process regression model to a data set of irregularly sampled time series. Once the model is trained, they define a set of evenly spaced interpolation points and materialize the Gaussian process posterior for each time series at the same set of reference time points. In the second phase of the method, they define Mercer kernels between the Gaussian process posteriors at the reference time points. This approach allows for the use of irregularly sampled time series with classical supervised and unsupervised kernel methods such as support vector machines and kernel PCA.

In subsequent work, Li and Marlin [2016] showed how the materialized Gaussian process regression posterior representation could be extended for use as an uncertainty preserving interface between irregularly sampled time series and arbitrary deep learning model components that expect regularly spaced or fixed dimensional inputs. Further, this work showed how the Gaussian process regression model parameters and the deep learning model parameters can be learned jointly and end-to-end using the re-parameterization trick [Kingma et al., 2015]. This model structure allows for solving a variety of tasks, but was primarily evaluated in the context of whole time series classification.

While the model of Li and Marlin [2016] was only applied to univariate irregularly sampled time series in the original work, in follow-up work the model was extended to multivariate time series using a multi-output Gaussian process regression model [Futoma et al., 2017]. However, modeling multivariate time series within this framework is quite challenging due to the constraints on the covariance function used in the GP regression layer. Futoma et al. [2017] deal with this problem using a sum of separable kernel functions [Bonilla et al., 2008], which somewhat limits the expressiveness of the model. The work of Futoma et al. [2017] was also focused on the whole time series classification as a task.

In terms of the tasks of interpolation and forecasting, Ghassemi et al. [2015] proposed a multi-task Gaussian process (MTGP) model for irregularly sampled physiological signals and clinical notes, which jointly transforms the measurements into a unified latent space. They showed that MTGP models provide better performance than single task GP models for interpolation and forecasting. Liu

and Hauskrecht [2016] combine state-space models with GPs for learning forecasting models from irregularly sampled multivariate clinical data. Soleimani et al. [2018] also employed multi-output GPs to jointly model multivariate physiological signals.

In summary, interpolation as a modeling primitive provides more sophisticated approaches than the more basic discretization-based methods. The two main families of approaches explored to date in this area are deterministic kernel smoothing methods such as IPNets and probabilistic Gaussian process-based methods. Both approaches naturally accommodate continuous time observations and can provide interfaces to other modeling building blocks such as kernel machines and neural networks. The main advantage of Gaussian process-based modeling primitives is the ability to represent and propagate uncertainty into downstream model components. Their main drawback is the significantly higher run times during both training and prediction, and the added complexity of needing to define valid mercer kernels for multivariate time series. Deterministic interpolation frameworks like IPNets do not have the elegance of Gaussian processes in terms of representing uncertainty, but can be much more flexible in terms of how they express temporal and cross-dimensional relationships. Importantly, they are also one to two orders of magnitude more computationally efficient during training. In the next section we turn to the case of recurrence as a modeling primitive and explore methods based on recurrent neural networks.

7   Recurrence-Based Approaches

In this section, we discuss approaches to modeling irregularly sampled time series that leverage recurrence as their primary modeling primitive for accommodating irregular sampling. We begin with traditional discrete RNN models, and then discuss more recent work on ODE-based models. Importantly, methods in this section do not rely on discretization or interpolation prior to the application of RNNs. This is the primary distinction relative to methods presented in the previous sections, some of which also leverage RNNs after discretization or interpolation has been applied to accommodate irregular sampling.

7.1   RNN-Based Methods

Similar to Lipton et al. [2016], Che et al. [2018a] present several methods based on gated recurrent unit networks (GRUs, Chung et al. [2014]) for the whole time series classification task. Their approach is cast in the vector-based representation. Thus, even though they do not apply discretization, they still need to deal with the problem of missing data inherited from this view of the problem. In their baseline methods, Che et al. [2018a] deal with the issue of missing values in vectors observed at irregularly sampled time points using simple imputation methods including mean imputation and forward filling. They also consider augmenting inputs with binary response indicator values and per-dimension inter-observation time intervals. This last approach attempts to provide the RNN with information about how long it has been since a value on a given dimension was last observed. When applying forward filling as the imputation method, the inter-observation time interval values indicate how long it has been since a new value for the variable was last observed.

However, the primary contribution of Che et al. [2018a] is the GRU with decay (GRU-D) model that adds a temporal decay mechanism to the the input variables and the hidden states. The decay rate function is defined as:

$$\gamma(\delta) = \exp\{-\max(0, \mathbf{W}_\gamma \delta + \mathbf{b}_\gamma)\} \tag{10}$$

where $\mathbf{W}_\gamma$ and $\mathbf{b}_\gamma$ are learned jointly with GRU parameters at training time. δ represents the length of the inter-observation interval. They introduce input decay for missing variables to decay the previously observed value toward the empirical mean for a given dimension over time. For large inter-observation intervals, this can be a more sensible choice than continued forward filling as the previously observed value will become uncorrelated with the current value over sufficiently long intervals. This can make the overall mean for that dimension a more better imputed value. We describe the GRU-D layer below:

$$\mathbf{x}'_{idn} = \mathbf{r}_{idn}\,\mathbf{x}_{idn} + (1 - \mathbf{r}_{idn})\,(\gamma(\delta)\mathbf{x}_{jdn} + (1 - \gamma(\delta))\bar{\mathbf{x}}_d) \tag{11}$$

where $\mathbf{x}_{jdn}$ is the last observation $(t_j < t_i)$ for dimension d, and $\bar{\mathbf{x}}_d$ is the empirical mean of the dth dimension in multivariate time series. Importantly, while this approach can be thought of as a form of

imputation, the parameters that control imputation are learned end-to-end with the rest of the model parameters based on a supervised objective.

As a further modification, Che et al. [2018a] also consider decaying the hidden states in the absence of observations. This has the effect of partially resetting the hidden state when there is a long interval with no observations. This is also a sensible modification as any accumulated hidden state may be irrelevant following a long time interval with no observations. Choi et al. [2016a] employed a similar method where they concatenate the time gaps between observations with the actual observations and supply it directly as input to a GRU model. On the other hand, Pham et al. [2017] proposed to capture time irregularity by modifying the forget gate a LSTM [Hochreiter and Schmidhuber, 1997] using a decay combined with a parametric function of the time gap between current and past measurement.

Neil et al. [2016] proposed the phased LSTM for supervised learning from irregularly sampled time series. The phased LSTM model introduced a time-dependent gating mechanism by adding a new time gate which regulates access to the hidden and cell state of the LSTM. The time-aware LSTM (T-LSTM) [Baytas et al., 2017] instead transforms time intervals into weights using a decay function and uses them to adjust the hidden state passed from the previous time step. While these approaches allow the network to handle event-based sequences with irregularly spaced observations, they do not support incomplete multivariate observations.

Li and Xu [2019] argue that the time difference δ between the current time step and last observation used in the dynamic imputation mechanism is not sufficient for the model to fully capture the overall pattern of missingness across a time series. They present VS-GRU, which also considers the observation rate of each dimension. They adopt a similar decay mechanism to that used by GRU-D. Li and Xu [2019] also proposed another variant VS-GRU-i that adds a standard GRU layer on top of VS-GRU. They add a penalized mechanism at the input of the second GRU layer to detect if one variable is completely missing or there are only a few observed values.

Kim and Chi [2018] also propose an approach to dealing with missing data in an RNN using decay, which they refer to as a temporal belief memory. When an input is missing, the belief gate computes the belief of the last observation based on the time gap between the current time and the last observation time. If the belief is greater than a threshold, the model imputes using the last observation; otherwise, it sets the missing value to the mean value of observations for that dimension. The belief gate function $\gamma(\delta)$ for a given dimension d is defined as:

$$\gamma(\delta) = \begin{cases} 1 & \text{if } \exp(-\beta_d \delta / \tau) > \pi \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where β_d is a tuneable decay parameter, τ is a window length, and δ is the time difference from the last observation to the current observation. Imputed input can then be defined using Equation 11.

One drawback of all of the approaches mentioned so far is that they process an input time series in time order. This is clearly a disadvantage for RNN-based models that attempt to accommodate for missing values under a vector-based representation as any imputation that occurs corresponds to the application of a filtering approach. This is overly restrictive when models are applied to problems like interpolation or whole time-series classification as in these problems it is permissible to integrate information from both the relative past and relative future of each time point. Indeed, methods like GRU-D make maximally uninformative imputations immediately prior to the next observation, whereas smoothing approaches would be informed by the next as well as the previously observed values in the time series.

A separate line of work has looked at using data from future as well as from the past for imputation of missing data in RNNs. Yoon et al. [2019] and Yoon et al. [2018b] present an approach based on a multi-directional RNN (M-RNN) which operates across dimensions as well as within dimensions. It consists of two separate blocks for imputing missing values. They first construct an imputation function using a bi-directional RNN that operates within a given dimension. The second imputation block constructs an imputation function using a fully connected layer that operates only across streams at a given time point. They also concatenate the input with the response indicator vector and time gap from the last observed value to the current timestamp.

Cao et al. [2018] also proposed models for imputing the missing values for multiple correlated time series using smoothing-like methods using RNNs. Their basic approach learns the missing values

directly in a recurrent dynamical system based on the observed data. They combine the hidden decay mechanism of GRU-D [Che et al., 2018a] and the RNN missing data imputation methods proposed in Han-Gyu Kim et al. [2017]. They also propose a bidirectional variant, BRITS-I, where they model the recurrent dynamics in the forward as well as backward direction. They introduce an additional loss in the bidirectional case to enforce consistency of predictions obtained in both directions. They extend BRITS-I for correlated recurrent imputation where the imputation is a weighted sum of history-based estimation and feature-based estimation. History-based estimation is simply uncorrelated recurrent imputation while feature-based estimation is computed based on only other features available at that time. They also concatenate the input with mask variables and time gaps from the last observed value to the current time point.

Tan et al. [2020] also introduce a time-aware structure based on a GRU to handle irregular time intervals. Similar to GRU-D, they handle irregular sampling by decaying the hidden state between the successive observations. However, instead of using exponential decay as in GRU-D, they apply inverse log decay, which they claim works better:

$$\hat{\mathbf{h}}_{t-1} = \frac{1}{\log(e + \delta_t)} \odot \mathbf{h}_{t-1} \tag{13}$$

Tan et al. [2020] use a dual attention mechanism for dealing with missing values caused by partially observed vectors or misalignment of observation time points on different dimensions. They perform imputations using Gaussian processes and a separate time-aware GRU model and combine them using an embedding layer to get the transformed input.

Luo et al. [2018] and Guo et al. [2019] employed generative adversarial networks (GAN) to impute missing values in irregularly sampled time series data by leveraging GRU-D style models to take into account irregular sampling. They propose a 2-stage method for imputation of missing data in time series. In the first stage, they train a GAN with GRU-D as generator and discriminator. In the second stage, they train the input "noise" of the generator of the GAN so that the generated time series is as close as possible to the original incomplete time series and the generated data has high probability of being real. They achieve this by minimizing a weighted sum of reconstruction loss and discriminative loss.

Luo et al. [2019] propose a single-stage generative model to impute missing values in multivariate time series without the need of the two-stage process as in Luo et al. [2018]. They use a two-layer GRU-D network [Che et al., 2018a] and a fully connected layer as generator. They begin with a zero imputed input. After a recurrent processing of the input time series using GRU-D, the last hidden state is processed using a fully connected layer which outputs a compressed low-dimensional vector. The compressed low dimensional vector acts as the input to another fully connected layer which transforms it into the initial input for the second GRU-D layer. The second GRU-D layer then outputs the generated sample at every time step. The generator model is trained in a denoising autoencoder fashion where they drop out original samples and reconstruct them. Since it is not a good idea to drop out samples when the missing rates are very high, they add some noise to the input samples and train a denoising autoencoder. The discriminator model is implemented using a GRU-D layer and a fully-connected layer. The fully connected layer takes the hidden state of GRU-D and outputs the probability of being true. The task of the discriminator is to distinguish between generated sample and the input/true sample.

Che et al. [2018b] proposed a deep generative model that uses a latent hierarchical structure to capture the temporal dependencies of multivariate time series with different sampling rates. This model captures the underlying data generation process by using a VAE-based approach and learns latent hierarchical structures using learnable switches and auxiliary connections. In particular, the switches use an update-and-reuse mechanism to control the updates of the latent states of a layer based on their previous states and the lower latent layers. The auxiliary connections between time series of different sampling rates and different latent layers help the model effectively capture short-term and long-term temporal dependencies. Their inference network consists of different RNN models for time series with different sampling rates. They jointly learn the parameters of the generative model and the inference network by maximizing the ELBO. In the case of irregularly sampled time series or missing data, missing values are interpolated by adding auxiliary connections using a model trained using only the observed data points. While training, missing data points are replaced with zero in the inference network and the corresponding auxiliary connections are removed in the generative model.

In summary, a great deal of work in the area of modeling and learning from irregularly sampled time series has been based on modifications of standard recurrent neural network models. RNN-based approaches have the primary advantage that they can deal with input sequences of different lengths. However, basic RNNs have no direct ability to deal with irregular sampling and no way to deal with missing values. GRU-D and related methods attempt to solve both problems using temporal decay, but the resulting models lack true continuous time semantics. In the next section, we turn to ODE-based recurrent models, which can provide a more elegant approach to modeling continuous time data at increased computational cost.

7.2 ODE Based Methods

Chen et al. [2018] proposed a variational auto-encoder (VAE) model [Kingma and Welling, 2014, Rezende et al., 2014] for continuous time data based on the use of a neural network decoder combined with a latent ordinary differential equation (ODE) model. They model time series data via a latent continuous-time function $\mathbf{z}(t)$. This function is defined via a neural network representation of its gradient field that takes the form $\frac{\partial \mathbf{z}(t)}{\partial t} = f_\theta(\mathbf{z}(t))$. The full generative process for a single data case sampled from the neural ODE latent time series model is shown below. Note that the generative process conditions on the set of observation time points $\mathbf{t} = t_0, ..., t_L$.

$$\mathbf{z}(t_0) \sim p(\mathbf{z}(t_0)) \tag{14}$$
$$\mathbf{z}(t_1), \cdots, \mathbf{z}(t_L) = \text{ODESolve}(\mathbf{z}(t_0), f, \theta, t_0, \cdots, t_L) \tag{15}$$
$$\mathbf{x}_i \sim p(\mathbf{x}_i | g_\phi(\mathbf{z}(t_i))) \tag{16}$$

While this model has many appealing properties as a continuous time generative model, VAE's also require the specification of a recognition network or encoder function to map observations into the latent space. Due to the deterministic nature of the ODE model, it suffices to identify the distribution over the latent space at any single time t. Chen et al. [2018] use a basic RNN model as the encoder/recognition network for computing the approximate posterior. In the case of irregularly sampled time series, Chen et al. [2018] propose to apply the encoder to the sequence of observed values backward in time from the end of the time series to the beginning.

However, the RNN encoder used did not account for variable inter-observation intervals at all and also did not account for partially observed input vectors. This means that the overall model has fairly asymmetric capabilities with the ODE used only in the generator and much more basic components applied in the encoder. Rubanova et al. [2019] subsequently proposed using an ODE-RNN model as an encoder for the latent ODE model. This yields a much more capable model with the intrinsic ability to accommodate irregularly sampled time series of fully observed vectors.

However, this model still has the limitation that it does not directly accommodate irregularly sampled time series of incompletely observed vectors. In addition, the encoder used by Rubanova et al. [2019] is also applied backwards in time only. While in theory all information in the input time series can be encoded into the latent state at the start of the time series, in practice this structure can be limiting as the encoder may not be able to adequately preserve information about structures in later sections of the time series.

De Brouwer et al. [2019] proposed a related method based on a combination of ODE and GRU components that they refer to as GRU-ODE-Bayes. This model is a continuous-time version of the Gated Recurrent Unit model [Chung et al., 2014] that can be applied to irregularly sampled time series. Instead of the encoder-decoder architecture where the ODE is decoupled from the input processing, GRU-ODE-Bayes provides a tighter integration by interleaving the ODE and the input processing steps. GRU-ODE-Bayes uses the ODE to propagate the hidden state between continuous-time observations. It updates the current hidden state whenever a new observation is available.

Contrary to the previous approaches in this subsection, GRU-ODE-Bayes can also handle partially observed vectors, which are common in irregularly sampled time series. Partially observed input in the RNN hidden update equation is replaced with a transformed input defined as a function (with learnable parameters) of previous hidden state, partially observed input and mask variable, similar to the approach used in the GRU-D model. However, due to the structure of the GRU-ODE-Bayes model, it only has access to information from the relative past of individual time points when accounting for missing data. This means that it suffers from the same problem as the GRU-D model where imputed

values for missing dimensions have maximum uncertainty immediately before the next observation. This makes GRU-ODE-Bayes a poor choice for a smoothing or interpolation method, while the local structure of the hidden state updates make it better suited for a filtering tasks. The final hidden states can also be used as input to prediction and detection tasks.

In summary, ODE-based modeling primitives provide an interesting basis for learning from continuous time data. They solve the problem of accommodating irregularly sampled time series of fully observed vectors in a way that is significantly cleaner than standard RNNs. However, these models can be quite a bit slower during training and deployment due to the need to make repeated calls to an ODE solver. Finally, some of the model structures proposed to date are also limiting in their inability to cleanly accommodate incomplete observations.

8 Attention-Based Approaches

Several recent models have leveraged attention mechanisms as their fundamental approach to dealing with irregular sampling [Horn et al., 2020, Song et al., 2018, Zhang et al., 2019]. Most of these approaches are similar to Vaswani et al. [2017] where they replace the positional encoding with an encoding of time and model sequences using self-attention. Instead of adding the time encoding to the input representation as in Vaswani et al. [2017], they concatenate it with the input representation. These methods use a fixed time encoding similar to the positional encoding of Vaswani et al. [2017]. Xu et al. [2019] learn a functional time representation and concatenate it with the input event embedding to model time-event interactions.

Zhang et al. [2019] proposed an attention-based time-aware approach that models the time irregularity between events by employing the attention mechanism. This approach adjusts the memory of an LSTM when accumulating previous information. Instead of reading the information from just one previous cell state, it combines values in previous cell states using attention weights and time gaps.

Choi et al. [2016b] learn an interpretable representation of irregularly sampled events using a two-level attention model. Attention vectors are generated by running RNNs backwards in time. These models are subsequently used to compute a final representation using a weighted average of the event embedding with the corresponding attention weights. They showed that performance can be improved by concatenating the corresponding timestamps with the event embedding.

Xu et al. [2019] proposed a set of time embedding methods for functional time representation learning, and demonstrate their effectiveness when combined with self-attention in continuous-time event sequence prediction. The proposed functional forms are motivated from Bochner's [Loomis, 2011] and Mercer's [Mercer, 1909] theorem.

Horn et al. [2020] employed a Transformer-based [Vaswani et al., 2017] approach for modeling irregularly sampled time series. This approach deals with irregular sampling by defining a time embedding to represent the time point of an observation. The time embedding defined here is a variant of positional encoding [Vaswani et al., 2017] which takes continuous time values as input. The transformer architecture was applied to the time series by concatenating the vectors of each time point with a mask variable indicating whether the variable is observed or missing and the corresponding time embedding. If the observation is missing, the input is set to zero for that dimension. The output of the transformer architecture is a sequence of embeddings, which can be aggregated into a final representation using mean-pooling for classification tasks.

9 Structural Invariance-Based Approaches

In this section we consider work in the area of using structural invariance as a modeling primitive. Such approaches leverage the set-based view of an irregularly sampled time series to build a model whose structure does not depend on the time order of the input time series at all.

In recent work, Horn et al. [2020] propose a set function-based approach for the task of classification and detection on time-series with irregularly sampled and unaligned observations. This approach can inherently handle irregular sampling and partially observed vectors. Similar to the positional encoding used in transformer models [Vaswani et al., 2017], Horn et al. [2020] define an embedding for continuous values of time. This approach represents the irregularly sampled time series data as a set of tuples consisting of a time embedding, value and dimension indicator to differentiate between

different dimensions. Following the approach described in Section 3.5, tuples are independently transformed using a multi-layer feed-forward network. Horn et al. [2020] suggest a weighted mean approach to aggregate the encoded tuples in order to allow the model to decide which observations are relevant and which should be considered irrelevant. This is achieved by computing attention weights over the set of input elements, and subsequently, computing the weighted sum over all elements in the set corresponding to their attention weights.

10   Data Sets

The previous sections have described methods for modeling irregularly sampled time series that can be applied to a variety of inference tasks. In this section, we describe some of the data sets that have been widely used in the machine learning community to conduct experiments with irregularly sampled time series and describe some of the commonly used tasks associated with each data set. We note that while it is possible to transform a regularly sampled time series into a sparse and irregularly sampled time series by randomly sampling a set of observations points, here we only describe data sets that naturally consist of sparse and irregularly sampled time series.

10.1   MIMIC-III

The MIMIC-III data set [Johnson et al., 2016] is a de-identified data set collected at Beth Israel Deaconess Medical Center from 2001 to 2012. It consists of approximately 58,000 hospital admission records. This data set contains sparse and irregularly sampled physiological signals, medications, diagnostic codes, in-hospital mortality, length of stay and more. Benchmark tasks on this data set include in-hospital mortality prediction, length of stay prediction and phenotype prediction. In-hospital mortality and length of stay prediction are whole time series classification and regression tasks while phenotype classification task is a multi-label classification task. The length of stay prediction task has also be framed as a classification task by discretizing the length of stay variable into a fixed number of buckets. MIMIC-III is available at `https://mimic.physionet.org/`.

10.2   PhysioNet 2012

The PhysioNet Challenge 2012 data set [Silva et al., 2012] consists of multivariate time series data with 37 variables extracted from intensive care unit (ICU) records. Each record contains sparse and irregularly spaced measurements from the first 48 hours after admission to ICU. The data set includes 4000 labeled instances and 4000 unlabeled instances. Benchmark tasks on this data set are in-hospital mortality prediction (whole time series classification) and interpolation. All 8000 instances can be used for interpolation experiments while only 4000 labeled instances are available for classification experiments. The PhysioNet dataset is available at `https://physionet.org/content/challenge-2012/`.

10.3   Human Activity

The Localization Data for Human Activity data set [Kaluža et al., 2010] consists of 3D positions of the waist, chest and ankles collected from five individuals performing several activities including walking, sitting, lying, standing, etc. Following the data preprocessing steps of Rubanova et al. [2019], a data set of $6,554$ sequences with 12 channels and 50 time points can be constructed. Labels are provided for each observation time point and denote the type of activity that the person is performing, such as walking, sitting, lying, etc. The data set consists of 11 classes. A per-time point classification task (detection task) is the benchmark task on this data set. The data set is available at `https://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity`.

10.4   eICU Collaborative Research Data Set

The eICU Collaborative Database [Pollard et al., 2018] is a database relating to patients who were treated as part of the Philips eICU program across intensive care units in the United States. eICU consists of medical records from 200,859 patients collected from 208 critical care units in the United States between 2014 and 2015. The database is deidentified, and includes sparse and irregularly sampled vital sign measurements, care plan documentation, severity of illness measures, diagnosis

information, treatment information, and more. The data set is available at `https://eicu-crd.mit.edu/`.

10.5 PhysioNet 2019

PhysioNet 2019 [Reyna et al., 2019] is a publicly available data set collected from two hospital systems. It contains 40,336 ICU patient admission records and 2,359 records of diagnosed sepsis cases. The data set consists of a set of multivariate time series that contains 40 variables including 8 vital signs, 26 laboratory values and 6 demographic variables. Each timestamp is labeled with a binary variable indicating whether the onset of sepsis has occurred. The benchmark task on this data set is the early detection of sepsis using physiological data.

11 Predictive Performance Comparison

In this section, we discuss the predictive performance of the different approaches discussed in the previous sections with respect to a variety of tasks. We provide pointers to open source implementations of many of these methods in Appendix A.

In terms of discretization-based methods, Lipton et al. [2016] showed that discretization combined with zero imputation and conditioning on response indicators outperformed LSTM models that do not condition on response indicators. Lipton et al. [2016] also showed that in presence of missing indicators, zero imputation performed better than forward filling based imputation. Harutyunyan et al. [2019] demonstrated the advantages of using channel-wise LSTMs and learning to predict multiple tasks using a single neural model. Song et al. [2018] showed that the performance of Harutyunyan et al. [2019] on the challenging MIMIC-III benchmark data set can be further improved by using self-attention instead of an LSTM. Bahadori and Lipton [2019] showed that a temporal clustering based approach achieves better performance than Song et al. [2018] and Harutyunyan et al. [2019] on classification and detection tasks.

However, discretization-based methods for irregular sampling data are typically outperformed by models with the ability to directly use an irregularly sampled time series as input. Kim and Chi [2018] showed that a simple imputation based on the time gap between the current time and the last observed time achieved better performance than imputation methods based on mean and forward-filling on classification tasks. They also showed that using response indicators further improved performance. Choi et al. [2016b] showed that the classification performance of their attention model on irregularly sampled events can be improved by concatenating the corresponding timestamps with the event embedding.

Che et al. [2018a] and Pham et al. [2017] showed that introducing a decay function based on time gap between current and past measurement inside a RNN model improves the performance on classification and prediction tasks as compared to several off-the-shelf machine learning models and RNN baselines based on mean and forward filling imputation schemes as well as RNN models based on concatenating time gaps and missing indicators. Che et al. [2018a] also showed that for classification and prediction tasks, combining exponential decay of hidden states with an imputation scheme based on the weighted average of previous values and the empirical mean further outperforms RNNs that use exponential decay on hidden state only. The T-LSTM proposed by Baytas et al. [2017] improved on Pham et al. [2017] by transforming time gaps into weights using a decay function and using them to adjust the hidden state. Li and Xu [2019] improved on GRU-D [Che et al., 2018a] by considering the observation rate in addition to the exponential decay on hidden state and weighted imputation. Zhang et al. [2019] showed that using a time aware attention model outperforms Choi et al. [2016b] and T-LSTM on detection task.

Recurrent methods based on bidirectional RNNs for modeling irregularly sampled time series typically achieve better performance than those based on single directional RNNs for both smoothing and classification tasks. Yoon et al. [2019], Yoon et al. [2018b] proposed MRNN based on bidirectional RNNs which outperforms Che et al. [2018a], Futoma et al. [2017], Lipton et al. [2016] and Choi et al. [2016a] on both smoothing and detection tasks. On the smoothing(imputation) task, it also outperforms standard imputation methods such as spline and cubic interpolation, MICE, Miss Forest, matrix completion and auto-encoder based approaches. Cao et al. [2018] show that another

bidirectional RNN-based approach for learning the missing values in vector-based representations outperforms GRU-D and MRNN on both smoothing and classification tasks.

GAN-based methods for imputing missing values in irregularly sampled time series data [Luo et al., 2018, 2019] have been shown to achieve better classification performance than GRU-D. Luo et al. [2019] also outperforms Luo et al. [2018] and Yoon et al. [2018a] on classification and smoothing tasks and Cao et al. [2018] on classification tasks. The VAE-based approach of Fortuin et al. [2020] achieves better classification performance than Luo et al. [2018] and Cao et al. [2018].

ODE-based recurrent models typically achieve better classification, imputation, detection and prediction performance on irregularly sampled time series than discrete time RNN-based recurrent models. For example, Latent-ODE [Rubanova et al., 2019] outperforms GRU-D, an RNN with exponential decay on hidden state, an RNN with imputation based on the weighted average of previous values and the empirical mean, and an RNN with values concatenated with time gaps and response indicators on both classification and detection tasks. Similarly, another related method based on the combination on ODE and RNN components [De Brouwer et al., 2019] achieves better forecasting performance than GRU-D and T-LSTM.

Among interpolation based methods, IP-Nets introduced by Shukla and Marlin [2019] outperforms the multi-output Gaussian process regression model [Futoma et al., 2017], GRU-D and RNN baselines based on mean and forward imputation schemes, and RNN models based on concatenating time gaps and missing indicators on classification and regression task. Attention-based interpolation [Shukla and Marlin, 2020] has been shown to achieve better performance than IP-Nets, GRU-D, Latent-ODE [Rubanova et al., 2019], Phased-LSTM [Neil et al., 2016] and set-based approaches [Horn et al., 2020] on classification, detection and interpolation tasks.

A dual-attention based time-aware method introduced by Tan et al. [2020] outperforms T-LSTM, GRU-D and IP-Nets on classification task. Horn et al. [2020] showed that a set-based and transformer-based approach for modeling irregularly sampled time series achieve comparable performance to GRU-D and IP-Nets on classification tasks.

12 Conclusions

In this paper we have presented a range of principles for learning from irregularly sampled time series data. We have provided a categorization of approaches into groups based on the fundamental modeling primitives used to accommodate irregular sampling. These modeling primitives include temporal discretization, interpolation, recurrence, attention and structural invariance. We have discussed the relationship between the three fundamental representations of multivariate irregularly sampled time series (series-based, vector-based, and set-based) that, while equivalent, motivate the use of different modeling primitives and result in different specific models and methods. We have also defined a range of inference tasks that can be performed using irregularly sampled time series (detection, prediction, filtering, smoothing, interpolation and forecasting). We, we have described a large number of specific models and methods with respect to the model primitives they build on, the tasks they aim to solve, and their relative strengths and weaknesses. Finally, we have discussed the relative performance of different approaches based on the available evidence from the current literature.

Acknowledgements

References

N. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *American Statistician*, 46(3):175–185, Aug. 1992.

M. T. Bahadori and Z. C. Lipton. Temporal-clustering invariance in irregular healthcare time series. *CoRR*, abs/1904.12206, 2019. Presented at the ACM Conference on Health, Inference, and Learning, Workshop Track, 2020.

I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou. Patient subtyping via time-aware lstm networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 65–74, 2017.

F. M. Bianchi, L. Livi, K. Mikalsen, M. Kampffmeyer, and R. Jenssen. Learning representations of multivariate time series with missing data. *Pattern Recognition*, 96, 2019.

E. V. Bonilla, K. M. Chai, and C. Williams. Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems*, pages 153–160, 2008.

W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li. BRITS: Bidirectional Recurrent Imputation for Time Series. In *Advances in Neural Information Processing Systems*, pages 6775–6785. 2018.

Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1), 2018a.

Z. Che, S. Purushotham, G. Li, B. Jiang, and Y. Liu. Hierarchical deep generative models for multi-rate multivariate time series. In *Proceedings of the 35th International Conference on Machine Learning*, pages 784–793, 2018b.

T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pages 6571–6583. 2018.

E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun. Doctor AI: Predicting Clinical Events via Recurrent Neural Networks. In *Proceedings of the 1st Machine Learning for Healthcare Conference*, pages 301–318, 2016a.

E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart. RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism. In *Advances in Neural Information Processing Systems*, pages 3504–3512, 2016b.

J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. Presented at the Deep Learning workshop at NIPS 2014.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

M. Cuturi. Fast global alignment kernels. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, page 929–936, 2011.

E. De Brouwer, J. Simm, A. Arany, and Y. Moreau. GRU-ODE-Bayes: Continuous Modeling of Sporadically-Observed Time Series. In *Advances in Neural Information Processing Systems*, pages 7379–7390. 2019.

V. Fortuin, D. Baranchuk, G. Raetsch, and S. Mandt. GP-VAE: Deep Probabilistic Time Series Imputation. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, pages 1651–1661, 2020.

J. Futoma, S. Hariharan, and K. A. Heller. Learning to detect sepsis with a multitask gaussian process RNN classifier. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1174–1182, 2017.

M. Ghassemi, M. A. Pimentel, T. Brennan, D. A. Clifton, P. Szolovits, T. Naumann, and M. Feng. A multivariate timeseries modeling approach to severity of illness assessment and forecasting in ICU with sparse, heterogeneous clinical data. In *Proceedings of the National Conference on Artificial Intelligence*, 2015.

Z. Guo, Y. Wan, and H. Ye. A data imputation method for multivariate time series based on generative adversarial network. *Neurocomputing*, 2019.

Han-Gyu Kim, Gil-Jin Jang, Ho-Jin Choi, Minho Kim, Young-Won Kim, and Jaehun Choi. Recurrent neural networks with missing information imputation for medical examination data prediction. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 317–323, 2017.

H. H. Harman. *Modern Factor Analysis*. The University of Chicago Press, 1976.

H. Harutyunyan, H. Khachatrian, D. C. Kale, and A. Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6, 2019.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. 2001.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

M. Horn, M. Moor, C. Bock, B. Rieck, and K. Borgwardt. Set functions for time series. In *Proceedings of the 25th International Conference on Machine Learning*, 2020.

D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant. *Applied logistic regression*. John Wiley & Sons, 2013.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3, 2016.

B. Kaluža, V. Mirchevska, E. Dovgan, M. Luštrek, and M. Gams. An agent-based approach to care in independent living. In *Proceedings of the 1st International Joint Conference on Ambient Intelligence*, pages 177–186, 2010.

Y. J. Kim and M. Chi. Temporal belief memory: Imputing missing data during rnn training. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2326–2332, 2018.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

D. P. Kingma, T. Salimans, and M. Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583. 2015.

J. L. Kling and D. Bessler. A comparison of multivariate forecasting procedures for economic time series. *International Journal of Forecasting*, 1:5–24, 1985.

M. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *Aiche Journal*, 37:233–243, 1991.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, page 282–289, 2001.

Q. Li and Y. Xu. VS-GRU: A Variable Sensitive Gated Recurrent Neural Network for Multivariate Time Series with Massive Missing Values. *Applied Sciences*, 2019.

S. C.-X. Li and B. Marlin. Learning from irregularly-sampled time series: A missing data perspective. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5937–5946, 2020.

S. C.-X. Li and B. M. Marlin. Classification of sparse and irregularly sampled time series with mixtures of expected Gaussian kernels and random features. In *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence*, pages 484–493, 2015.

S. C.-X. Li and B. M. Marlin. A scalable end-to-end gaussian process adapter for irregularly sampled time series classification. In *Advances In Neural Information Processing Systems*, pages 1804–1812, 2016.

S. C.-X. Li, B. Jiang, and B. Marlin. Learning from Incomplete Data with Generative Adversarial Networks. In *International Conference on Learning Representations*, 2019.

Z. C. Lipton, D. Kale, and R. Wetzel. Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In *Proceedings of the 1st Machine Learning for Healthcare Conference*, pages 253–270, 2016.

R. J. Little and D. B. Rubin. *Statistical analysis with missing data*, volume 333. John Wiley & Sons, 2014.

Z. Liu and M. Hauskrecht. Learning Adaptive Forecasting Models from Irregularly Sampled Multivariate Clinical Data. pages 1273–1279, 2016.

S. P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28: 129–137, 1982.

L. H. Loomis. *Introduction to abstract harmonic analysis*. Dover books on mathematics. 2011.

Z. Lu, T. K. Leen, Y. Huang, and D. Erdogmus. A Reproducing Kernel Hilbert Space Framework for Pairwise Time Series Distances. In *Proceedings of the 25th International Conference on Machine Learning*, pages 624–631, 2008.

Y. Luo, X. Cai, Y. Zhang, J. Xu, and X. Yuan. Multivariate time series imputation with generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 1596–1607, 2018.

Y. Luo, Y. Zhang, X. Cai, and X. Yuan. E2GAN: End-to-End Generative Adversarial Network for Multivariate Time Series Imputation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3094–3100, 2019.

P. Manimaran, J. C. Parikh, P. K. Panigrahi, S. Basu, C. M. Kishtawal, and M. B. Porecha. Modelling financial time series. In *Econophysics of Stock and other Markets*, pages 183–191. 2006.

B. M. Marlin, D. C. Kale, R. G. Khemani, and R. C. Wetzel. Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 389–398, 2012.

P.-A. Mattei and J. Frellsen. MIWAE: Deep Generative Modelling and Imputation of Incomplete Data Sets. In *Proceedings of the 36th International Conference on Machine Learning*, pages 4413–4423, 2019.

G. Mclachlan and K. Basford. *Mixture Models: Inference and Applications to Clustering*. Dekker, 1988.

J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society, London*, 209:415–446, 1909.

M. Mudelsee. TAUEST: A computer program for estimating persistence in unevenly spaced weather/climate time series. *Computers & Geosciences*, 28:69–72, 2002.

D. Neil, M. Pfeiffer, and S.-C. Liu. Phased LSTM: Accelerating Recurrent Network Training for Long or Event-based Sequences. In *Advances in Neural Information Processing Systems*, pages 3882–3890. 2016.

T. Pham, T. Tran, D. Phung, and S. Venkatesh. Predicting healthcare trajectories from medical records: A deep learning approach. *Journal of Biomedical Informatics*, 69:218–229, 2017.

T. Pollard, A. Johnson, J. Raffa, L. Celi, R. Mark, and O. Badawi. The eICU Collaborative Research Database, a freely available multi-center database for critical care research. *Scientific Data*, 5, 2018.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Readings in Speech Recognition*, page 267–296. 1990.

C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. 2006.

K. Rehfeld, N. Marwan, J. Heitzig, and J. Kurths. Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics*, 18:389–404, 2011.

M. Reyna, C. Josef, R. Jeter, S. Shashikumar, M. B. Westover, S. Nemati, and G. Clifford. Early prediction of sepsis from clinical data: The physionet/computing in cardiology challenge 2019. *Critical Care Medicine*, 48, 2019.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1278–1286, 2014.

Y. Rubanova, R. T. Q. Chen, and D. K. Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, pages 5320–5330. 2019.

T. Ruf. The lomb-scargle periodogram in biological rhythm research: analysis of incomplete and unequally spaced time-series. *Biological Rhythm Research*, 30(2):178–201, 1999.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, pages 318–362. 1986.

J. D. Scargle. Studies in astronomical time series analysis. ii-statistical aspects of spectral analysis of unevenly spaced data. *The Astrophysical Journal*, 263:835–853, 1982.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computing*, 10:1299–1319, 1998.

M. Schulz and K. Stattegger. Spectrum: Spectral analysis of unevenly spaced paleoclimatic time series. *Computers & Geosciences*, 23:929–945, 1997.

H. Shimodaira, K.-i. Noma, M. Nakai, and S. Sagayama. Dynamic time-alignment kernel in support vector machine. In *Advances in Neural Information Processing Systems*, pages 921–928, 2001.

S. N. Shukla and B. Marlin. Interpolation-prediction networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2019.

S. N. Shukla and B. M. Marlin. Multi-time attention networks for irregularly sampled time series. 2020. Presented at the 1st Workshop on the Art of Learning with Missing Values (Artemiss), ICML 2020.

I. Silva, G. Moody, D. Scott, L. Celi, and R. Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. *Computing in cardiology*, 39:245–248, 2012.

M. Śmieja, L. u. Struski, J. Tabor, B. Zieliński, and P. a. Spurek. Processing of missing data by neural networks. In *Advances in Neural Information Processing Systems*, pages 2719–2729. 2018.

A. J. Smola and B. Schölkopf. A Tutorial on Support Vector Regression. *Statistics and Computing*, 14:199–222, 2004.

H. Soleimani, J. Hensman, and S. Saria. Scalable Joint Models for Reliable Uncertainty-Aware Event Prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:1948–1963, 2018.

H. Song, D. Rajan, J. Thiagarajan, and A. Spanias. Attend and diagnose: Clinical time series analysis using attention models. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 4091–4098, 2018.

Q. Tan, M. Ye, B. Yang, S. Liu, A. J. Ma, T. C.-F. Yip, G. L.-H. Wong, and P. Yuen. DATA-GRU: Dual-Attention Time-Aware Gated Recurrent Unit for Irregular Multivariate Time Series. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 930–937, 2020.

A. Trask, D. Gilmore, and M. Russell. Modeling Order in Neural Word Embeddings at Scale. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2266–2275, 2015.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, and I. Polo-sukhin. Attention is All You Need. In *Advances in Neural Information Processing Systems*, page 6000–6010, 2017.

G. Wahba. Support vector machines, reproducing kernel hilbert spaces, and randomized gacv. In *Advances in Kernel Methods: Support Vector Learning*, page 69–88. 1999.

D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan. Self-attention with functional time representation learning. In *Advances in Neural Information Processing Systems*, pages 15915–15925. 2019.

Q. Ye, W. Szeto, and S. C. Wong. Short-term traffic speed forecasting based on data recorded at irregular intervals. *IEEE Transactions on Intelligent Transportation Systems*, 13:1727–1737, 2010.

J. Yoon, J. Jordon, and M. van der Schaar. GAIN: Missing data imputation using generative adversarial nets. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5689–5698, 2018a.

J. Yoon, W. R. Zame, and M. van der Schaar. Deep Sensing: Active Sensing using Multi-directional Recurrent Neural Networks. In *International Conference on Learning Representations*, 2018b.

J. Yoon, W. R. Zame, and M. van der Schaar. Estimating Missing Data in Temporal Data Streams Using Multi-Directional Recurrent Neural Networks. *IEEE Transactions on Biomedical Engineering*, 66:1477–1490, 2019.

M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep Sets. In *Advances in Neural Information Processing Systems*, pages 3391–3401. 2017.

Y. Zhang, X. Yang, J. Ivy, and M. Chi. ATTAIN: Attention-based Time-Aware LSTM Networks for Disease Progression Modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4369–4375, 2019.

A Source Code

Table 1: Source code for different approaches

Reference	Source Code
Baytas et al. [2017]	`https://github.com/illidanlab/T-LSTM`
Cao et al. [2018]	`https://github.com/caow13/BRITS`
Che et al. [2018a]	`https://github.com/Han-JD/GRU-D`
Chen et al. [2018]	`https://github.com/rtqichen/torchdiffeq`
Choi et al. [2016a]	`https://github.com/mp2893/doctorai`
Choi et al. [2016b]	`https://github.com/mp2893/retain`
De Brouwer et al. [2019]	`https://github.com/edebrouwer/gru_ode_bayes`
Fortuin et al. [2020]	`https://github.com/ratschlab/GP-VAE`
Harutyunyan et al. [2019]	`https://github.com/YerevaNN/mimic3-benchmarks`
Horn et al. [2020]	`https://github.com/BorgwardtLab/Set_Functions_for_Time_Series`
Kim and Chi [2018]	`https://github.com/ykim32/TBM-missing-data-handling`
Li and Marlin [2016]	`https://github.com/steveli/gp-adapter`
Li and Marlin [2020]	`https://github.com/steveli/partial-encoder-decoder`
Luo et al. [2018]	`https://github.com/Luoyonghong/Multivariate-Time-Series-Imputation-with-Generative-Adversarial-Networks`
Luo et al. [2019]	`https://github.com/Luoyonghong/E2EGAN`
Neil et al. [2016]	`https://github.com/dannyneil/public_plstm`
Pham et al. [2017]	`https://github.com/trangptm/DeepCare`
Rubanova et al. [2019]	`https://github.com/YuliaRubanova/latent_ode`
Shukla and Marlin [2019]	`https://github.com/mlds-lab/interp-net`
Song et al. [2018]	`https://github.com/khirotaka/SAnD`
Yoon et al. [2019]	`https://github.com/jsyoon0823/MRNN`
Xu et al. [2019]	`https://github.com/StatsDLMathsRecomSys/Self-attention-with-Functional-Time-Representation-Learning`