Recurrent Neural Networks for Mortality Prediction

Satya Narayan Shukla, Benjamin Marlin
College of Information and Computer Sciences
University of Massachusetts Amherst
snshukla@cs.umass.edu

Abstract

Irregularly sampled time series is quite common in various practical applications such as healthcare, geoscience, astronomy etc. In this paper, we present a new approach to model irregularly sampled time series data. We introduce interpolation models to first transform the input time series data to a fixed set of reference points and then it is fed to a Gated Recurrent Unit (GRU). We also present an approach to learn the correlation across variables to improve the interpolated output. Experiments show that the proposed model outperforms strong deep learning models based on GRU for the mortality prediction task.

1 Introduction

Irregularly sampled time series is commonly found in clinical data. For example, physiological vitals such as blood pressure, heart rate are measured repeatedly but irregularly during a patient's hospital episode. Modeling such time series is important for several practical applications ranging from health care, geoscience, astronomy, to biology and others.

Recurrent neural Networks (RNNs) such as LSTMs ((Hochreiter & Schmidhuber, 1997) and GRUs (Chung et al. 2014) have shown great power in many applications with sequential data such as machine translation (Bahdanau et al., 2014; Sutskever et al., 2014) and speech recognition (Hinton et al., 2012). RNNs have this intrinsic capability to capture long-term dependencies and variable-length observations. Recent works (Che et al. 2016) tried to handle irregularly sampled/ missing data with RNNs by introducing decay in input as well as hidden layer. Instead of just using the last observation (as done in some of the previous work) for the missing variable, it is decayed over time towards the empirical mean. Since at training, we have the data available for all the time points, we can try to interpolate missing variable using not only the past observations but also the future observations.

In this paper, we develop a novel architecture based on GRU, to effectively exploit both the past and the future observations to accurately model the behavior of missing variables in input time series. We interpolate the irregularly sampled input time series to a fixed set of reference points before inputting it to GRU. We jointly train all the model parameters using backpropagation. Our proposed model not only captures the long term dependencies from the past but also tries to learn from future to create a smooth interpolation. Our model also tries to learn from other correlated variables present in the time series. Experiments on real-world clinical dataset demonstrate that our proposed model outperforms strong deep learning models built on GRU with imputation as well as other strong baselines.

2 Models

We represent the time series data with d variables of length T as $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T\}^T \in \mathbb{R}^{T*d}$ where $\boldsymbol{x}_t \in \mathbb{R}^d$ represents the measurements of all variables recorded at time t and x_t^j represents the

value of jth variable at time t .

$$x_t^j = \begin{cases} x_t^j, & \text{if } x_t^j \text{ is recorded.} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

We use masking vector $m_t \in \{0,1\}^d$ to represent which variables were measured at time t.

$$m_t^j = \begin{cases} 1, & \text{if } x_t^j \text{ is observed.} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

We introduce another variable $\boldsymbol{z}$ to store the timestamps of the recorded variables. $\boldsymbol{z}$ is union of all the recorded timestamps of the d variables for a particular example. We note that the length $\boldsymbol{z}$ can vary across the examples. In this paper, we are interested in mortality prediction problem on irregularly sampled time series data. We represent the given time series as $\{X_n, \boldsymbol{z}_n, M_n, l_n\}_{n=1}^N$ where $X_n = \{\boldsymbol{x}_1^n, \boldsymbol{x}_2^n, \cdots, \boldsymbol{x}_{T_n}^n\}^T, \boldsymbol{z}_n = \{z_1^n, \cdots, z_{T_n}^n\}, M_n = \{\boldsymbol{m}_1^n, \boldsymbol{m}_2^n, \cdots, \boldsymbol{m}_{T_n}^n\}^T$ and $l_n \in \{0,1\}$.

We investigate the use of recurrent neural networks(RNN) for time series classification because their recursive nature allow them to handle variable length sequences intrinsically. Among the various variants of RNN, we consider an RNN with gated recurrent units (Chung et al., 2014).

Existing work handles irregularly sampled/missing data by the following ways:

- Replacing each missing observation with the mean of the variable across the training examples - **GRU Mean**
- Assuming any missing value is same as its last measurement and use forward imputation - **GRU forward**
- Instead of computing the missing variable explicitly, input is concatenated with masking variable and time interval which tells for how long the particular variable is missing - **GRU Simple**
- Che et al. (2016) introduces input decay, where for a missing variable instead of just using the last observation, it is decayed over time towards the empirical mean. In order to capture richer information they also introduce hidden decay and argue that it has effect of decaying extracted features rather than just input variables directly- **GRU-D**

2.1 Simple Interpolation

We introduce an interpolation model to transform the irregularly sampled time points to fixed set of reference points. Let $\boldsymbol{r} = \{r_1, r_2, \cdots, r_p\}$ be the set of reference points, then interpolation model is defined as:

$$w_k(r_i, z_j) = \exp(-\alpha_k ||r_i - z_j||^2)$$

$$y_{r_i}^k = \frac{\sum_j w_k(r_i, z_j) * x_{z_j}^k}{\sum_j w_k(r_i, z_j) * m_{z_j}^k}$$

where $x_{z_j}^k$ represents the value of kth variable at timestamp z_j, $y_{r_i}^k$ is the interpolated value of kth variable at reference point r_i and α_k's are parameters of the model.

2.2 Interpolation Around Mean

Here, we fix some reference points and try to learn the nature of mean of variables across these points and then the data is interpolated around the mean.

$$w_k(r_i, z_j) = \exp(-\alpha_k ||r_i - z_j||^2)$$

$$y_{r_i}^k = \frac{\sum_j w_k(r_i, z_j) * (x_{z_j}^k - v_{z_j}^k)}{\sum_j w_k(r_i, z_j) * m_{z_j}^k} + v_{r_i}^k$$

where $v_{z_j}^k$ is the mean value of kth variable at timestamp z_j. Let $\boldsymbol{s} = \{s_1, s_2, \cdots, s_q\}$ be the set of reference points and $\{\boldsymbol{v_{s_1}}, \cdots, \boldsymbol{v_{s_q}}\}$ be the mean values that are learned. We again use a similar

interpolation model to interpolate the mean at these fixed points to the given timestamps and again to the original reference points ($\boldsymbol{r}$).

$$w'_k(r_i, s_j) = \exp(-\beta_k ||r_i - s_j||^2)$$
$$v^k_{r_i} = \frac{\sum_j w'_k(r_i, s_j) * v^k_{s_j}}{\sum_j w'_k(r_i, s_j)}$$
$$w''_k(z_i, s_j) = \exp(-\beta_k ||z_i - s_j||^2)$$
$$v^k_{z_i} = \frac{\sum_j w''_k(z_i, s_j) * v^k_{s_j}}{\sum_j w''_k(z_i, s_j)}$$

2.3 Interpolation using Multi-channel Correlation

To capture more information from the timeseries while interpolating, we introduce correlation variables to capture any correlation present across channels (or variables). If the variables are correlated then we could get some information about a missing value if other variables are recorded around same time.

$$w_k(r_i, z_j) = \exp(-\alpha_k ||r_i - z_j||^2)$$
$$y^k_{r_i} = \frac{\sum_j w_k(r_i, z_j) * x^k_{z_j}}{\sum_j w_k(r_i, z_j) * m^k_{z_j}}$$
$$\hat{y}^c_{r_i} = \frac{\sum_{c'} \rho_{cc'} * I_{r_i c} * y^c_{r_i}}{\sum_{c'} \rho_{cc'} * I_{r_i c}}$$

where $\rho_{cc'}$ is correlation coefficient of channel c and c', $\rho_{cc} = 1$, and $I_{r_i c} = \sum_j w(r_i, z_j) * m^c_{z_j}$

All the interpolation models described above are used to generate values on fixed set of reference points. The output of our interpolation model goes into GRU which learns to classify the given timeseries. The parameters of GRU as well as interpolation model are jointly learned with backpropagation.

3 Dataset

We have used the publicly available MIMIC-III dataset (Johnson et al. 2016). It contains deidentified clinical care data collected at Beth Israel Deaconess Medical Center from 2001 to 2012. It consists of around 58000 hospital admission records. We only use the first 48 hours data after admission from each time series. We extracted 12 timeseries features from 53211 records obtained after removing hospital admission record with length of stay less than 2 days. These features include heart rate, systolic and diastolic blood pressure, SpO_2, respiratory rate, temperature, glucose, pH, FiO_2, TGCS (total glasgow coma score) and CRR (capillary refill rate). We focus on predicting whether a patient dies in hospital or not by using first 48 hours of data. There are 4310 (8.1%) patients with positive mortality label.

4 Experiments

Since our dataset is highly imbalanced, baseline for our model i.e. predicting all 0 gives an accuracy of 91.9%. For Non-RNN baselines, we implement Logistic Regression, Support Vector Machine, Random Forest and AdaBoost. For these experiments, we take average of each variable over the observed time points. For RNN baselines, we implement vanilla GRU (input is just time series data), Simple GRU, GRU-Mean and GRU-D (Che et al. 2016). All these models are described briefly in section 2.

For GRU models, we use a one hidden layer with 100 units to model the sequence, and then apply a soft-max regressor on top of the last hidden state to do classification. For all the interpolation models, we select timestamps where successive timestamps are separated by 5 minutes. For interpolation around mean model, the successive reference points for mean are separated by 6 hours. We divide

Table 1: Model performances for mortality prediction

Models	AUC score	Accuracy
Baseline	0.5	91.90%
Logistic Regression	0.657	92.35%
SVM	0.657	92.62%
AdaBoost	0.741	93.68%
Random Forest	0.724	93.70%
Vanilla GRU	0.797	92.24%
Simple GRU	0.827	92.12%
GRU Hidden Decay	0.833	92.41%
GRU-D	0.827	92.25%
Simple Interpolation	**0.846**	92.01%
Interpolation Around Mean	**0.844**	92.02%
Interpolation with multi-channel correlation	0.793	91.95%

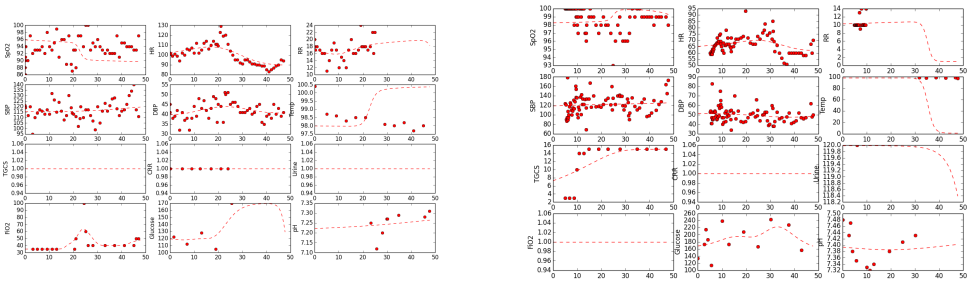

Figure 1: Output of Simple Interpolation model

our data randomly into training(60%), validation(20%), test(20%) sets, and train all the GRU models with the Adam optimization method (Kingma & Ba, 2014) and use early stopping to find the best weights on the validation dataset. We report the results in terms of accuracy and area under the ROC curve (AUC score) since accuracy alone doesn't give much insights with imbalanced datasets. As we can see in Table 1, our interpolation models without multichannel correlation performs better than other GRU and non-RNN models in terms of AUC score for mortality prediction.

Figure 1 shows the output of our Simple Interpolation model. It is clear that our model over-smooths the input time series data. This is because the α parameters learned from our model are very low in the order of 0.01 and some of them are even negative as observed in the Fig. 1.

5 Conclusions and Future Work

In this paper, we proposed novel GRU-based Interpolation models to effectively handle the irregularly sampled time series data. Experiments show that the proposed model outperforms strong deep learning models based on GRU for the mortality prediction task. Our proposed model not only captures the long term dependencies from the past but also tries to learn from the future data-points(since they are available during training) to create a smooth interpolation. Since our model over-smooths as seen from the graphs above, future goals would focus on constraining the bandwidth parameter (α) to capture more temporal information. As seen from the results our multi-channel correlation model doesn't perform as well as simple interpolation model which is not that intuitive and it requires more analysis.

References

[1] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.

[2] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, "Recurrent neural networks for multivariate time series with missing values", arXiv, 2016.

[3] AEW Johnson, TJ Pollard, L Shen, L Lehman, M Feng, M Ghassemi, B Moody, P Szolovits, LA Celi, and RG Mark. MIMIC-III, a freely accessible critical care database. Scientific Data, 2016.

[4] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[5] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. Neural computation, 9(8): 1735–1780, 1997.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

[7] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, pp. 3104–3112, 2014.

[8] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. Signal Processing Magazine, IEEE, 29(6):82–97, 2012.