Multi-resolution Attention with Signal Splitting for Multivariate Time Series Classification

Rheeya Uppaal [1] **Bryon Kucharski** [1] **Bhanu Pratap Singh** [1] **Iman Deznabi** [1] **Madalina Fiterau** [1]

Abstract

Real world multivariate time series pose three significant challenges: irregularity in sampling, missing values and varying sampling frequencies among signals. Recent work for inference on such data aims at solving one of these issues, however a unified model is still lacking. We present a unified method which handles all three: Multi-resolution Attention with Signal Splitting (MASS). Our method is model-agnostic and can be applied to any existing model, significantly boosting predictive performance. MASS uses parallel multi-resolution block to model different resolution data streams, in addition to splitting signals into components of specific resolutions, to provide approximately a 3% improvement on the Physionet Challenge 2012 Dataset. We also compare to the state of the art TBM and GRU-D models, showcasing promising results against them.

1. Introduction

Multivariate time series analysis is an essential component of gaining knowledge in multiple domains, particularly in the medical, financial and networking domains. While data in these fields is abundant, it has certain properties which makes inference on such data challenging. For example, in the medical domain, vital signs like heart rate are regularly captured, but other features such as blood platelet count would only be recorded based on the patient's condition, introducing the problem of *irregularity*. Furthermore, different signals can be captured at different frequencies, making the multivariate time series *multi-resolutional* as well. Vital signals like heart rate are recorded at a higher frequency than signals like cholesterol. Apart from this, there is always the possibility of having *missing values* which can be introduced because of issues like device limitations.

Current work on multivariate time series classification targets one of these three problems individually, however a unified method is still lacking. Recent work focuses on either using simple methods like mean value imputation, or using sequential models which ignore missing values altogether. We present a method which tackles all three issues at once: Multi-resolution Attention with Signal Splitting (MASS). MASS combines two methods we introduce: Multi-resolution Networks and Signal Splitting.

Multi-resolution networks have dedicated 'sub models' for signals of a particular resolution bandwidth. These sub models or 'blocks' operate separately to create different representations, which are then fused before the final downstream prediction task. A sparsity score is used to identify the resolutions of signals, and map them to their appropriate block. For example, a frequently sampled vital signal like heart rate would be sampled to a 'fast' block, while an infrequently sampled signal like cholesterol would be sent to the 'slow' block. Standard models instead impute the slow signals at each time step. MASS equips a model to handle data with different frequencies without this imputation, thus significantly increasing predictive performance. We further introduce a unique form of attention in each block, which learns which hidden representations are most useful to query, which results in an additional boost in performance. Multi-resolution networks are extended through the concept of Signal Splitting, which introduces the sharing of information between signals from different blocks. As a signal may be a composition of a set of signals with different resolutions, Signal Splitting uses a simple method based on averaging to divide a signal into components of different resolutions. Following the signal splitting, each signal component is sent to the block for its specific resolution. We show the effectiveness of our method on the Physionet 2012 dataset (Silva et al., 2012), where data is irregular, and has missing values and multiple resolutions. MASS, being a method that can be added to any existing model, is added to two different models and has been shown to boost predictive accuracy.

[1] College of Information and Computer Science, University of Massachusetts, Amherst. Correspondence to: Rheeya Uppaal <ruppaal@cs.umass.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

2. Related Work

Past work on multivariate time series classification does not take into account the irregular sampling of values from a time series, or the fact that different features in the multivariate time series are sampled at varying resolutions. These methods focus on imputing missing values, and passing the imputed series to a sequential model. The imputation methods can be as basic as mean imputation and interpolation (Kreindler & Lumsden, 2016), or can be more complicated, such as using kernel methods, (Rehfeld et al., 2011), multiple imputation (Galimard et al., 2016) or expectation maximization (EM) (García-Laencina et al., 2010). However, all these methods suffer from the weakness of not being able to handle sparse datasets.

Multivariate time series, especially in the medical domain, can be comprised of complicated distributions. Additionally, the patterns of sampling of the data can also provide information for inference. For example, frequent sampling of a patient's vitals can indicate a more serious condition. Recent models make better use of such information. Temporal Belief Memory Network (Kim & Chi, 2018) is one such method. TBM imputes a time series with decaying values, making use of the average value and last observed value of that series. The imputed set of signals are then passed through a sequential model. Another recent method, GRU-D (Che et al., 2018), works under the assumption that patterns of missing values are often correlated with the target labels. Based on the Gated Recurrent Unit (GRU), the model uses two time based representations to impute values, and provides a greater capability to capture complicated patterns. However, despite the strength of TBM and GRU-D, neither effectively handles multi-resolution data.

3. Method

3.1. Multi-Resolution Networks

Recent work in multivariate time series processing focuses on using a monolithic model for learning the distribution of the entire data. However, such an approach may have disadvantages. In a multivariate setting, each feature is sampled from a different true distribution P^*, where distributions can have varying resolutions. In such case, a single model may not be able to effectively learn the overall distribution of the data.

While it is not reasonable to have a dedicated model for each feature, it is possible to have one for a cluster of similar features. A considerable number of multivariate time series datasets originate from the medical domain. Curating these entails performing a battery of tests on a patient to record various statistics, however, these samples are taken with varying frequencies for each feature. Thus, we cluster the features of the multivariate time series on the basis of a 'sparsity score', which serves as an indicator of how frequently a feature is sampled. This allows each model to learn from a separate fine grained distribution, thus better capturing the structure of the data.

The sparsity score calculation is designed to classify signals with large missing chunks of data as slower signals. For this, a two dimensional matrix C is created for every data point x to keep track of how many time steps a signal has been missing. M is a matrix of indicator random variables, keeping track of time steps in which a signal is present (where 1 represents a missing value, and 0 represents a recorded value). C is constructed as,

$$C_{i,j} = \begin{cases} 0 & \text{if } M_{i,j} = 0 \\ 1 + C_{i,j-1} & \text{if } M_{i,j} = 1 \end{cases}$$

Where i represents the feature and j represents the time step of each data point. After the construction of C, the score for each feature x_i is calculated as follows, where T is the length of the time series for the given data point.

$$score(x_i) = \frac{\sum_{j=0}^{T} C_{i,j}}{\sum_{t=0}^{T} t}$$

Figure 2 shows the sparsity scores for all features in the Physionet 2012 Dataset. The figure shows three clear clusters of signals with different resolutions. For this reason, we modeled MASS to have three dedicated resolution blocks: a block for 'fast' signals, one for 'moderate' frequency signals and one for 'slow' signals. It is possible that data from other sources might have a different clustering order of signals, however, it is safe to assume that clustering features into three groups of slow, moderate and fast is a paradigm that would generalize well to other domains.

Each block creates a resolution-specific representation of the signals passed to it, allowing the representations to be more fine grained and specific. An additional feature we add in multi resolution networks is an attention mechanism, where the query is learned from the data, rather than being extracted from it. This is added after the LSTM creates an initial representation of the data, and it helps the model identify which hidden features of this representation the model would most benefit to learn from. This is similar to self attention, with the main distinction being that the model uses a specific part of the input as a query, rather than the entire input. The three self-aware representations are then concatenated and passed through a fully connected layer, which calculates the final probabilities for binary classification.

3.2. Residual Signal Splitting

While multi-resolution networks are a helpful first step in solving the problem of multi-resolution between signals,

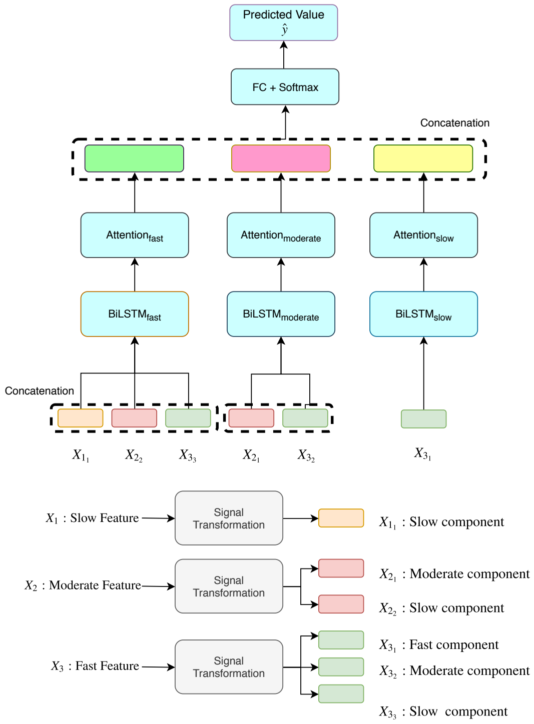

Figure 1. Multi-resolution Attention with Signal Splitting (MASS) for a vanilla BiLSTM with Attention. x_1 is a feature that has been categorized as 'slow', based on the sparsity score. Similarly, x_2 and x_3 represent 'moderate' and 'fast' signals respectively. The Signal Transformer divides each x_i into its slow, moderate and fast components, should they exist (x_{i_j}, where $j \leq$ number of blocks in the model). All components of a particular resolution across all signals are concatenated and sent to the model block for that resolution, which creates a self-aware representation of the signals. The representations from all blocks are then concatenated and passed through a fully connected layer to make the final binary prediction.

they can be improved upon further. Since all blocks function independently, a particular block receives no information from signals sent to other blocks. Although there is some amount of information passed between blocks due to backward gradient flows, it is not sufficient. Furthermore, each signal can consist of multiple other signals which have different resolutions. For example, Figure 3 (a) shows two signals that have been assigned to the 'fast' block and 'slow' blocks, based on their sparsity score. The signals in (b) and (c) sum up to make the original signal, and are thus components of it. However, they have different resolutions from the original signal.

For this reason, we introduce the second major component of MASS: signal splitting. Every signal is split into a maximum of three components with different resolutions, one each for 'slow', 'medium' and 'fast', should they all exist. The Signal Transformer in Figure 1 performs this resolution based division of signals using the Signal Transformation

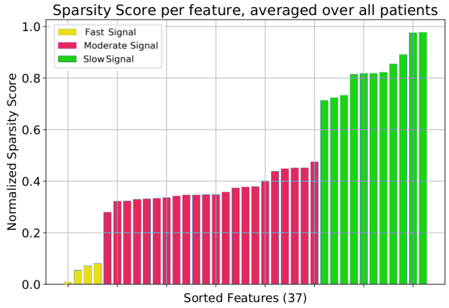

Figure 2. A histogram of sparsity scores for each feature of the Physionet 2012 Dataset. The signals form three clear clusters based on their sparsity: 'fast' (yellow), 'moderate' (pink) and 'slow' (green).

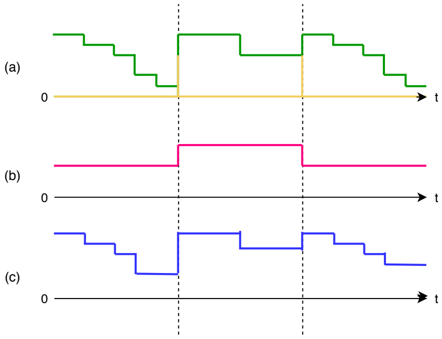

Figure 3. An example of a composite signal with multiple resolutions. (a) shows a 'fast' signal (green) and 'slow' signal (yellow), based on their sparsity score. (b) shows the averaged value, or the 'slow' component of the green signal, while (c) shows the residual 'fast' and sparse component. The component signals have different resolutions from the original signal.

described in Algorithm 1 (Appendix). The algorithm covers a simplified case where only a single 'slow' signal x_1 and a single 'fast' signal x_2 are considered.

Let x_k be a particular signal from the dataset. Its components of different resolutions are then defined as $x_{k_1}, x_{k_2} \ldots x_{k_l}$, where l is the number of resolution blocks used in the model. Before the Signal Transformation algorithm is applied, mean imputation is performed on all signals to ensure that a time step has values for either all signals, or none of the signals. This resolves the irregularity problem. Forward imputation, or other imputation methods can also be used at this step.

Now, considering the case with a 'slow' signal x_1 and 'moderate' signal x_2, $x_{i,j}$ represents the jth timestep in the signal x_i. Similarly, $x_{i,j:k}$ represents all values from the jth to kth timesteps. c_1 and c_2 are defined to mark the start and end of a series of timesteps in the slow signal, which all have missing values. The Signal Transformation algorithm extracts the slow and moderate components of the moderate signal, creating x_{2_1} and x_{2_2} respectively. For this, the average $\overline{x_{2,c_1:c_2}}$ of x_2 over the period from c_1 to c_2 is calculated as,

$$\overline{x_{2,c_1:c_2}} = \frac{\sum_{i=c_1}^{c_2} x_{2,i}}{c_2 - c_1}$$

The values for c_1 and c_2 are reset for every series of missing timesteps noted in the slower signal. When a 'fast' signal x_3 is introduced, the average $\overline{x_{3,c_1:c_2}}$ is calculated in addition to the average over x_2. This average is then subtracted to create the residual, where the average is marked as a slow component x_{3_1}. To find the moderate component x_{3_2}, c_1 and c_2 are computed for the moderate signal x_2. Following this, the new average $\overline{x_{3,c_1:c_2}}$ becomes x_{3_3}. The average is once again subtracted from x_3 to create the final residual, which becomes the fast component x_{3_3} of x_3.

4. Experiments

We experiment with the PhysioNet Challenge 2012 dataset (Silva et al., 2012), which consists of Intensive Care Unit (ICU) records of 8000 anonymous patients. 4000 of these patients have labelled data, and have been used to train our model. Up to 37 features are recorded for each patient for approximately 48 hours worth of time, during the stay of a patient. The data has been divided into two groups, based on the Survival Index, to classify patients by ICU mortality. This dataset is a classic example of a setting with multi-resolution, irregular sampling and missing values in the data. We also present results on a variant of the dataset in Appendix B.

We compare our method against two strong baselines: the Temporal Belief Memory Networks (TBM) of (Kim & Chi, 2018) and GRU-D of (Che et al., 2018) are state of the art methods in missing valued multivariate time series analysis. We also introduce a basic Bidirectional LSTM model with attention (BA-Mean) as our baseline. The BA-Mean model is run over the data after imputing missing values with averages. As we present a model-agnostic method that can be added to any existing model, we add it to our BA-Mean model, and to the TBM model, displaying a boost in performance over their vanilla variants. We also compare the simple BA-Mean model with multi-resolution against GRU-D, and showcase the comparable performance of a basic model to a state of the art model.

We further experiment with two variants of our method. Since there is the possibility that a sparsity specific block might have more information than others, for a particular patient, we add a second layer of attention after concatenating the representations from all blocks. These models are the MASS-BA+Attn and MASS-TBM++Attn in Table 2.

Hyperparameters were gauged by the best performance on the validation set, through a grid search over learning rates, and representation sizes of the LSTM blocks. Optimal hyperparameters have been included in Appendix A. Training was conducted using the ADAM optimizer, and ended using an early stopping criterion on the validation set. Our results are described in Table 1. The performance of BA-Mean and TBM are similar, due to both models being similar in architecture. MASS-BA and MASS-TBM are similar for the same reason, but the inclusion of MASS shows a significant boost in performance, of around 3% over their vanilla variants, as shown in Table 1. The addition of the second attention mechanism does not contribute to predictive performance in either model. This may be because the additional capacity given to the model makes it overfit to the training data. While GRU-D remains the strongest model, it must be noted that the addition of MASS to a simple BiLSTM model showcased performance almost equivalent to GRU-D. This highlights the potential of our method, and a scope to use it on better performing models in the future.

Table 1. Precision, Recall and F-scores on the Physionet 2012 Dataset, for Bidirectional LSTM with Attention with average filling (BA-Mean), Multi-resolution Attention with Signal Splitting Multi-resolution Attention BA-Mean with Signal Splitting on BA-Mean (MASS-BA), MASS-BA with second Attention (MASS +Attn), Temporal Belief Memory Network (TBM), Multi-resolution Attention with Signal Splitting on TBM (MASS-TBM), MASS-TBM with second Attention (MASS-TBM+Attn) and GRU-D. Each result has been averaged over three trials.

METHOD	PRECISION	RECALL	F1-SCORE
BA-MEAN	0.806	0.854	0.812
MASS-BA	0.841	0.860	**0.846**
MASS-BA+ATTN	0.845	0.869	0.836
TBM	0.805	0.849	0.816
MASS-TBM	0.838	0.856	0.842
MASS-TBM+ATTN	0.841	0.862	**0.846**
GRU-D	0.843	0.870	**0.847**

5. Conclusion

We presented MASS, a model agnostic method which can be added to any model to improve predictive performance on multivariate time series classification, by tackling the issues of multi resolution between signals, irregular sampling and missing values in data. Experiments were performed on the Physionet Dataset which, possessing these three problems, accurately depicts real world data. The addition of MASS shows an improvement on the TBM and BA-Mean models, and even the simple BA-Mean model shows almost equivalent performance to the state of the art GRU-D, on the addition of MASS.

Our future work includes applying MASS on the state of the art GRU-D, and testing the how well our method generalizes to other models and domains. Additionally, we would like to study the effect of varying the number of resolution blocks on predictive performance.

References

Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.

Galimard, J.-E., Chevret, S., Protopopescu, C., and Resche-Rigon, M. A multiple imputation approach for mnar mechanisms compatible with heckman's model. *Statistics in medicine*, 35(17):2907–2920, 2016.

García-Laencina, P. J., Sancho-Gómez, J.-L., and Figueiras-Vidal, A. R. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2): 263–282, 2010.

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

Kreindler, D. M. and Lumsden, C. J. The effects of the irregular sample and missing data in time series analysis. In *Nonlinear Dynamical Systems Analysis for the Behavioral Sciences Using Real Data*, pp. 149–172. CRC Press, 2016.

Rehfeld, K., Marwan, N., Heitzig, J., and Kurths, J. Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics*, 18(3):389–404, 2011.

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

Table 2. Precision, Recall and F-scores on the Physionet 2012 ”Hard” Dataset, for Bidirectional LSTM with Attention with average filling (BA-Mean), Multi-resolution Attention with Signal Splitting Multi-resolution Attention BA-Mean with Signal Splitting on BA-Mean (MASS-BA), MASS-BA with second Attention (MASS +Attn), Temporal Belief Memory Network (TBM), Multi-resolution Attention with Signal Splitting on TBM (MASS-TBM), MASS-TBM with second Attention (MASS-TBM+Attn) and GRU-D. Each result has been averaged over three trials.

METHOD	PRECISION	RECALL	F1-SCORE
BA-MEAN	0.662	0.658	0.659
MASS-BA	0.673	0.672	**0.676**
TBM	0.657	0.665	0.658
GRU-D	0.676	0.684	0.667

A. Optimization and Hyperparameters

We use the ADAM optimization algorithm with a learning rate of $1e - 4$ and dropout of 0.9. Training is performed on a Titan-X GPU with a a batch size of 1 (for simplicity in the code for our method). The sizes of the representations of the LSTM were also a tuneable hyperparameter, and optimal vector lengths were found to be 200.

B. Signal Transformation Algorithm

MASS consists of a signal transformation module, shown in Figure 1. The algorithm for this method is presented in Algorithm 1.

Algorithm 1 Signal Transformation

```
Input: num timesteps over which data is recorded $T$
slow signal $x_1 \sim X_{slow}$, fast signal $x_2 \sim X_{fast}$
Initialize $c_1 \leftarrow 0$, $c_2 \leftarrow 0$, $x_{2_1} \leftarrow []$, $x_{2_2} \leftarrow []$
for $t = 1$ to $T$ do
  if $x_{1,t}$ is missing then
    $c_2 \leftarrow c_2 + 1$
  else
    $average = \frac{\sum_{i=c_1}^{c_2} x_{2,i}}{c_2 - c_1}$
    $residual \leftarrow []$
    for $j = c_1$ to $c_2$ do
      $residual$.append($x_{2,j} - average$)
    end for
    $x_{2_2}$.append($average$)
    $x_{2_1}$.append($residual$)
    $c_1 \leftarrow c_2$
  end if
end for
Return: $x_{1_1}, x_{2_1}, x_{2_2}$
```

C. Results on Physionet 2012 ”Hard” Dataset

The Physionet 2012 Dataset predicts ICU mortality. We altered the task to forecast patient survival, thus making the task of prediction more challenging. The Physionet ”Hard” Dataset also divides the patients into two roughly equal groups, with 2526 people surviving the study. Our results are described in Table 2. Here, the addition of MASS to BA-Mean clearly outperforms its vanilla variants. MASS-BA also outperforms both state of the art baselines of TBM and GRU-D.