Using Features Extracted From Vital Time Series for Early Prediction of Sepsis

Qiang Yu, Xiaolin Huang, Weifeng Li, Cheng Wang, Ying Chen, Yun Ge

School of Electronic Science and Engineering, Nanjing University, Nanjing, China

Abstract

To get early prediction of sepsis, we propose to extract more time-dependent characteristics that retain the temporal evolvement information of the underlying biomedical dynamic system, including differential, integration, time-dependent statistics, variations and convolutions.

Considering that two categories are unbalanced in the training set, we employed easy ensemble algorithm to get multiple base learners. As for the base learner, we tried three models: random forest, XGBoost and LightGBM. By boosting the results of multiple base learners, we constructed our ensemble model.

Our team which name is njuedu ranked 25th in the official test and scored 0.282 in full test set.Since the submitted model version only used training set A to train our model, the model had a higher score of 0.401 in test set A, and 0.278 in test set B, and only -0.207 points in test set C.

1. Introduction

Prediction is of great importance in biomedical field, e.g. early goal-directed therapy provides significant benefits with respect to outcome in patients with severe sepsis and septic shock[1–3]. To achieve the real-meaning ‘pre’diction, it means the model has to rely on the history information to get a current prediction. Therefore, specific models with memory have been developed, such as hidden Markov model[4], long short-term memory recurrent neural networks[5–7], etc. However, memory-units also introduce dynamical complexity, and might impose the model on a risk of instability.

There have been several studies about early prediction of sepsis. Desautels and colleagues[8] tried to use the Insight model for severe sepsis detection and got an AUC of 0.75. Mao and colleagues[9] validated InSight based on a retrospective dataset from the mixed ward of the University of California, San Francisco (UCSF) Medical Center (San Francisco, Calif.) to detect and predict three gold standards associated with sepsis and achieve AUC of 0.92 and 0.87 on sepsis and severe sepsis respectively.Kam and Kim[10] used a deep learning model to create an early sepsis prediction system and validated its feature extraction capabilities. The best result they got was an AUC of 0.929 using the LSTM variant. They followed the feature extraction steps of [11].

In this manuscript, we tried to extract various time-dependent characteristics from the time series, and use the derived features as the input of regular machine learning model like random forest, XGBoost[12] and LightGBM[13] etc.

2. Methods

We use random forest, XGBoost and LightGBM as prediction models. However, most efforts have been paid to the pre-processing and time-dependent characteristics extraction. And all data descriptions can be found in [14].

2.1. Pre-processing

2.1.1. NAN replacement

Since we will focus on the temporal evolvement of the biomedical indices, it is necessary to replace the NAN in the original data with meaningful values in order to facilitate extraction of certain time dependent characteristics.We can see from Figure 1 that the data is missing very badly. We use the NAN replacement rules as the following:

$$\text{if} \quad a_t = NAN, \text{then}$$
$$a_t = \begin{cases} a_i & i = max(\Phi), \Phi \neq \oslash \\ \mu_a & \Phi = \oslash \end{cases} \tag{1}$$

in which a_t represents the value of characteristic a taken at time t, μ_a is the arithmetic mean of characteristic a in training set, $\Phi := \{j|a_j \neq NAN, j \in [max(1, t-3), t)\}$, and $\oslash$ denotes the empty set. This is also a combination of the forward-fill and mean-fill method.

2.1.2. Feature extraction

We will treat every record as an independent sample in the task. That means we will lose most evolvement in-

ISSN: 2325-887X DOI: 10.22489/CinC.2019.067

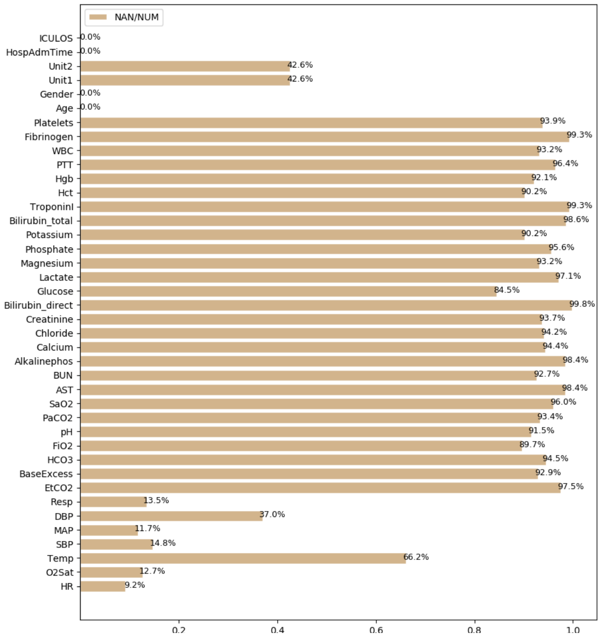

Figure 1. Data missing ratio.

formation. For compensation, we extract extra features to represent the temporal evolvement as much as possible.

The features include:

(1) Differential characteristics

For the original characteristic a, we derived the first order differential as:

$$g^{sp}_{a,t} = \begin{cases} a_t - a_{t-sp} & t \in [sp+1, N] \\ a_t - a_1 & t \in (\lfloor \frac{2\times sp}{3}, sp\rfloor] \\ NAN & t \in [1, \lfloor \frac{2\times sp}{3}\rfloor] \end{cases} \tag{2}$$

in which N denotes the number of the records in a file after NAN replacement, and the sp takes 1, 12, and 24 represents 1-hour, 12-hour, and 24-hour differential, respectively.

Subsequently, the second differential is derived as:

$$h^{sp}_{a,t} = \begin{cases} g^{sp}_{a,t} - g^{sp}_{a,t-sp} & t \in [sp+1, N] \\ g^{sp}_{a,t} - g^{sp}_{a,1} & t \in (\lfloor \frac{2\times sp}{3}, sp\rfloor] \\ NAN & t \in [1, \lfloor \frac{2\times sp}{3}\rfloor] \end{cases} \tag{3}$$

(2) Time-dependent statistic characteristics

We derived time-dependent mean, maximum, minimum and variance as:

$$m_{a,i} = \frac{\sum\limits_{j=1}^{i} a_j}{i}, \qquad i \in [1, N] \tag{4}$$

$$f^{max}_{a,i} = max\{a_1, a_2, \ldots, a_i\} \qquad i \in [1, N] \tag{5}$$

$$f^{min}_{a,i} = min\{a_1, a_2, \ldots, a_i\} \qquad i \in [1, N] \tag{6}$$

$$Var_{a,i} = \frac{\sum\limits_{j=1}^{i} (a_j - m_{a,i})^2}{i}, \qquad i \in [1, N] \tag{7}$$

(3) Variation coefficients

The basic variation coefficient formula is:

$$c^{sp}_{a,i} = \begin{cases} \frac{\sqrt{\frac{1}{sp}\sum\limits_{j=i-sp+1}^{i}(a_j - \mu_i^{sp})^2}}{\mu_i^{sp}} & t \in [sp+1, N] \\ NAN & t \in [1, sp) \end{cases} \tag{8}$$

where $\mu^{sp}_{a,i} = \frac{\sum\limits_{j=i-sp+1}^{i} a_j}{sp}$. We take sp 12 to derive 12-hour variation coefficient, and sp i to get time-dependent variation coefficient.

Furtherly, we drive a characteristic $c^{ex}_{a,i}$ as an exaggerate version of 12-hour variation coefficient:

$$c^{ex}_{a,i} = \begin{cases} max\{c^{12}_{a,1}, c^{12}_{a,2}, \ldots, c^{12}_{a,i}\} & c^{12}_{a,i} \geq M^{12}_{a,i} \\ min\{c^{12}_{a,1}, c^{12}_{a,2}, \ldots, c^{12}_{a,i}\} & c^{12}_{a,i} < M^{12}_{a,i} \end{cases} \tag{9}$$

in which $M^{12}_{a,i}$ is the median of set $\{c^{12}_{a,1}, c^{12}_{a,2}, \ldots, c^{12}_{a,i}\}$.

(4) Integration

We also adopt integration characteristic as:

$$p^{sp}_{a,i} = \begin{cases} \sum\limits_{j=i-sp+1}^{i} (a_j - \mu_a) & t \in [sp, N] \\ NAN & t \in [1, sp) \end{cases} \tag{10}$$

The sp can take i and 8.

(5) Kurtosis

The kurtosis can describe the steepness of the sample data distribution relative to the normal distribution, and index $Kts_{a,i}$ is defined as:

$$Kts_{a,i} = \frac{i\sum\limits_{j=1}^{i}(a_j - m_{a,i})^4}{[\sum\limits_{j=1}^{i}(a_j - m_{a,i})^2]^2} - 3, \quad i \in [1, N] \tag{11}$$

For space limit, we only introduce most important features. After all these processing, we obtain 968 derivative characteristics which extracted from first 34 columns in the raw data. We replace NAN in these characteristics with corresponding median.

2.2. Model selection and training

It has been widely accepted that ensemble learning can achieve better performances. Therefore, we tried three typical ensemble models, i.e. Random Forests (RF), XGBoost(XGB)[12] and LightGBM(LGB)[13] . No strict restrictions are imposed on these models, therefore, most of our characteristics can be input, no regarding they are categorical or numerical, Gaussian or non-Gaussian, et.

In addition, considering that the sample size of two categories of data is severely unbalanced, we employ EasyEnsemble[15] (EE) as our down-sampling algorithm. That is, we randomly divided the class of ‘regular’, which accounts for the majority in the training set, into T equal parts, and the sample size of each part is approximate to the sample size of the ‘high risk’ class. Based on these partitions, we trained T base learners. Then, we use bagging method to ensemble the output of the T learners as our final output.

Algorithm 1 The EasyEnsemble algorithm.

```
Input: $P$: A set of minor class examples, $N$: a set of major class examples, $T$: the number of subsets to be sampled from $N$,$|P| < |N|$
for each $i \in [1, T]$ do
    Randomly sample a subset $N_i \subset N$, $|N_i| = |P|$;
    Learn $H_i$ using $P$ and $N_i$, $H_i$ is an estimator to classify object.
end for
Output: An ensemble: $H(x) = f(H_i(x))$
```

2.3. Feature selection (FS)

In the training phase, we used features which combined raw data and derived characteristics as the model input, and those features need to be extracted in real-time when we apply the model to actual application. That means it will be heavily time-consuming. On the other hand, more features confront the model higher risk of overfitting. To alleviate the feature extraction burden as well as overfitting risk, we examine the importance ranks of all these features, and the most $\alpha\%$ important features will be selected as the input to train our model.

Algorithm 2 Feature Selection with EasyEnsemble.

```
Input: $P$: A set of minor class examples, $N$: a set of major class examples, $T$: the number of subsets to be sampled from $N$,$|P| < |N|$, $FS = \{\}$: feature selection set
for each $i \in [1, T]$ do
    Randomly sample a subset $N_i \subset N$, $|N_i| = |P|$;
    Learn $H_i$ using $P$ and $N_i$, $H_i$ is an estimator;
    Record feature importance in model, and write as $FS_i$;
    Pick out the top $\alpha\%$ importance features of $FS_i$,and write as $FS_i^{\alpha}$, default:$\alpha\% = 60\%$;
    then $FS = FS \cup FS_i^{\alpha}$;
end for
Output: The most importance features $FS$.
```

2.4. Model Structure

In our model structure, we first preprocess the raw data and use the EasyEnsemble[15] algorithm in combination with the LightGBM[13] classifier as the final classification model. In order to eliminate redundant features, we pretrained the classification model and used the feature importance function to sort the features and pick out the top $\alpha\%$ important features. In the end, Bayesian optimization is used to obtain better hyper-parameters.

3. Results

In the official test, njuedu which is our team name submitted an initial model version. We can see from the table 1 that the model obtained 0.282 U-Score[14] in full test set, and ranked 25th in all participants. We found that the U-Score[14] is more likely to have a higher score when the ROC score is higher in the same test set.

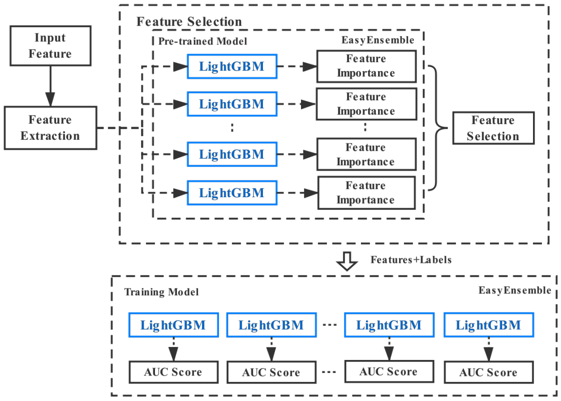

Figure 2. Model training process.

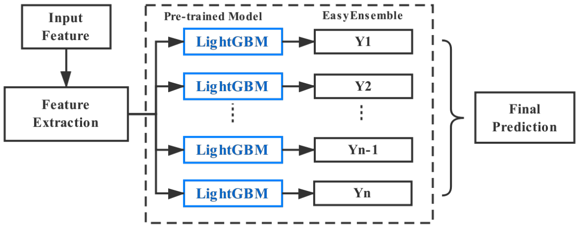

Figure 3. Model test process.

Table 1. Official test scores

Test Set	ROC	PRC	ACC	F1	U-Score
A	0.798	0.097	0.835	0.136	0.401
B	0.746	0.066	0.912	0.122	0.278
C	0.716	0.047	0.765	0.039	-0.207
Full	N/A	N/A	N/A	N/A	0.282

Since in the earlier version model, we only used training set A to extract features and train our model, the performance of earlier versions on training set B was really unsatisfactory and we got a intermediate training U-Score[14] close to the official test set B results.In the later research, we will make full use of training sets A and B. We believe this will improve the performance of the model on test set B.

The former version we submitted used fewer characteristics, extracting the mean, extremum, differential characteristics, variation coefficient, integration characteristics, and other characteristics for the first 16 columns of the original data, and no feature selection, and we got 280 columns of derived characteristics. In the latter versions, we created the training set T_{AB} which we respectively extracted first 14000 data from the set A and the set B, using the rest of the data as validation set V_{AB}.This method is also called hold-out. We used the training set T_{AB} to minimize the performance difference on different data sets and used the first 8 columns, 16columns and 34 columns of the original data for feature extraction respectively. After feature extraction of the first 8/16/34 columns of raw data, we obtained 232/464/968 columns of derived characteristics. After feature selection, the number of derived characteristics was reduced to 156/312/616. The EasyEnsemble[15](EE) and LightGBM[13](LGB) combined models using the first 34 columns of data extraction features and using feature selection algorithm show optimal performance.Unfortunately, our highest score version ran timeout and could not get the official score.We also feel confused about that, because the time consumption is not a problem in our own test.All our models will be uploaded to github and challenge official site.

4. Conclusion

Combining time-dependent characteristics extraction and models like RF, XGBoost and LightGBM, we can get a not bad sepsis prediction. We believe that the temporal evolvement information can give us substantial clues to the development of sepsis.In the follow-up study, we will focus on solving the problem of program operation efficiency.

Acknowledgments

We acknowledge the Social Development Program of Primary Research & Development Plan in Jiangsu Province (BE2016733, BE2017679), foundation for the cooperation between school and enterprise of Nanjing City (201722002).

References

[1] Rivers, Emanuel, Nguyen, Bryant, et al. Early goal-directed therapy in the treatment of severe sepsis and septic shock. New England Journal of Medicine 2001;345(19):1368–1377. PMID: 11794169.

[2] Nguyen, Bryant H, Corbett, et al. Implementation of a bundle of quality indicators for the early management of severe sepsis and septic shock is associated with decreased mortality*. Critical Care Medicine 2007;35(4):1105–1112.

[3] Kumar, Anand, Roberts, et al. Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock*. Critical Care Medicine 2006;34(6).

[4] L. R, B.H. J. An introduction to hidden markov models. IEEE ASSP MAGAZINE Jan 1986;.

[5] Hochreiter S, Schmidhuber. J. Long short-term memory. Neural Computation Nov 1997;9(8):1735–1780.

[6] Lipton ZC, Kale DC, Elkan C, Wetzel RC. Learning to diagnose with LSTM recurrent neural networks. In 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings. 2016; URL http://arxiv.org/abs/1511.03677.

[7] Hyland SL, Hüser M, Lyu X, Faltys M, Merz T, Rätsch G. Predicting circulatory system deterioration in intensive care unit patients. In AIH@IJCAI. 2018; .

[8] Desautels T, Calvert J, Hoffman J, Jay M, Kerem Y, et al. Prediction of sepsis in the intensive care unit with minimal electronic health record data: a machine learning approach. JMIR medical informatics 2016;4(3).

[9] Mao Q, Jay M, Hoffman JL, et al. Multicentre validation of a sepsis prediction algorithm using only vital sign data in the emergency department, general ward and icu. BMJ Open Jan 2018;8(1):2044–6055.

[10] Kam HJ, Kim. HY. Learning representations for the early detection of sepsis with deep neural networks. Computers in biology and medicinen 2017;89:248–255.

[11] Calvert JS, Price DA, Chettipally UK, et al. A computational approach to early sepsis detection. Computers in Biology and Medicine 2016;74:69 – 73. ISSN 0010-4825.

[12] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16. New York, NY, USA: Association for Computing Machinery. ISBN 9781450342322, 2016; 785–794.

[13] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY. Lightgbm: A highly efficient gradient boosting decision tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510860964, 2017; 3149–3157.

[14] Reyna M, Josef C, Jeter R, SP.Shashikumar, et al. Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019. Critical Care Medicine ;in press.

[15] Liu X, Wu J, Zhou Z. Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems Man and Cybernetics Part B Cybernetics April 2009;39(2):539–550.

Address for correspondence:

Xiaolin Huang.
School of Electronic Science and Engineering, Nanjing University, 163 Xianlin Avenue, Nanjing, China, 210023.
xlhuang@nju.edu.cn