Early Detection of Sepsis Using Ensemblers

Shailesh Nirgudkar [1], Tianyu Ding [1, 2]

[1] MathWorks, Natick, MA, USA
[2] Johns Hopkins University, Baltimore, MD, USA

Abstract

This paper describes a methodology to detect sepsis ahead of time by analyzing hourly patient records. The Physionet 2019 challenge consists of medical records of over 40,000 patients. Using imputation and weak ensembler technique to analyze these medical records and 3-fold validation, a model is created and validated internally. On a hidden test data set maintained by the organizers, the model obtained a utility score of 0.192. The utility score as defined by the organizers takes into account true positives, negatives and false alarms. Our team was Team_Tesseract and our overall ranking was 49 out of 79 officially ranked entries.

1. Introduction

Sepsis is a life threatening condition in which a person's immune response to infection can cause tissue damage, multiple organ failure, and even death [1]. If this condition is detected early enough, preventative steps can be taken to avoid a mortal outcome [2, 3]. Clinicians have revised the definition of sepsis in the hope that with it, relevant measurements can be recorded and an early diagnosis can be made. Still, the detection remains challenging because it is not possible to measure when the 'sepsis' starts. The onset can be observed only indirectly through vital signs, laboratory measurements, administration of antibiotics, and drawing of blood cultures for suspicion of infection. In emergency departments, there are established protocols for regular measurements; however, in normal hospital settings such regularity is not observed. Typically, the measurements are taken at irregular intervals and have missing information. Moreover, timely measurements are taken only in case of suspicion of disease. Prior art [4] has been able to predict sepsis ahead of time using machine learning techniques but with good, clean data. This database contains patient records only from one hospital between 2001 and 2007 and some of the features in this dataset were defined by ICD-9 (International Classification of Diseases, Ninth revision) codes, the sensitivity and specificity of these codes were very diagnosis-dependent. The techniques may not scale without further work. In real life, data available for analysis are often noisy because of the reasons mentioned earlier. The labeled dataset provided by Physionet 2019 challenge [5] has more than 90% values missing the laboratory measurements. Also the data are highly imbalanced meaning there are very few patients ($\approx$7.3%) having sepsis condition. The combination of high percentage of missing data and imbalance in class labels makes the problem of prediction challenging. In this paper, we describe an approach to preprocess the data and train ensemble learner on that data.

2. Methodology

Figure 1 shows the overview of implementation.

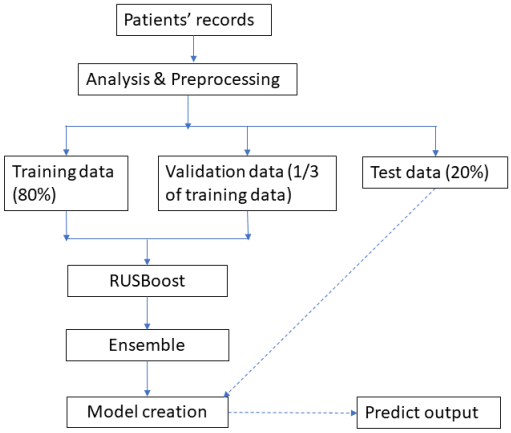

Figure 1. Overall framework of solution

5000 patient records were made available initially and then the organizers supplied two sets of approximately 20,000 records each. We combined all these records and this single pool formed our entire patient data set. All these patient records (about 45000) are analyzed for missing values. Such values are then replaced by using imputation. Once this is done, the records are divided into three buckets - training data constitutes 80% of total data set and test data constitutes remaining 20%. The training data is fur-

ther divided into validation data set using 3 fold validation. Only training data and validation data are used to create model. Ensemblers are used along with RUSBoost to obtain a model. Once model is created, test data is used to predict its output. Based on predicted output and true output, the hyper-parameters are tuned further so that loss between predicted output and true output is minimized. Following subsections describe these steps in details.

2.1. Feature Imputation

Each patient record consists of 40 features and a final label for each hour. Majority of features related to laboratory measurement contain over 90% NaN values. These missing values are not missing at random (MNAR) but because the measurements are generally taken twice a day. Data imputation is used to generate missing values. For each patient, values are imputed using linear imputation within that patient record. As an illustration, imputation of mean arterial pressure (MAP) for a patient record is shown in Figure 2. If the number of non-NaN values are less than 3, that feature record is treated as noisy and no imputation is attempted. At the end of this phase, there would still be some features containing NaN values. The simple imputation (of linearization within a patient record) is chosen over other complicated schemes because there is no satisfactory method when data are not missing at random (MNAR) and there are many features exhibiting the issue.

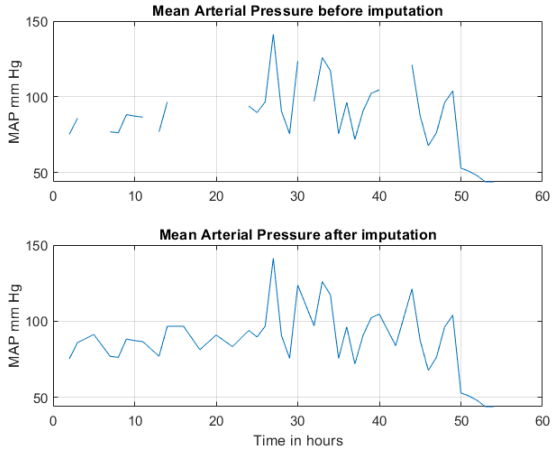

Figure 2. Imputation of MAP of a patient record

2.2. Feature Weights

As per [1], mean arterial pressure (MAP) and serum lactate level are important in identifying septic shock. This combination alone is associated with over 40% mortality rates in hospital. At least two of the respiratory rate, altered mentation and systolic blood pressure (SBP) determine quickSOFA (:Sequential [Sepsis-related] Organ Failure Assessment (SOFA)) score. quickSOFA is a practical way of identifying adult patients in out-of-hospital, emergency departments or general hospital wards who are likely to develop poor outcomes. So more weightage is given to features 'Lactate', 'MAP', 'HR' and 'Resp' by squaring them and replacing the original values with squared ones. Similarly according to literature [1], white blood cell count is also an important marker but in our experimentation, squaring it did not improve the utility score.

2.3. Model

Since the final training data set still contain missing values, surrogate trees are used, which can handle these values. MATLAB® release R2019a is used to implement the algorithm. The software sends the observation to left or right child node using the best surrogate predictor. Because of the imbalance in labels (sepsis vs non-sepsis) of training data, a variation of Boosting algorithm known as *Random UnderSampling Boost* (**RUSBoost**) [6] is used. It combines data sampling with AdaBoost algorithm. As the name implies, RUSBoost randomly undersamples the training data corresponding to the majority class. The main disadvantage of random undersampling, loss of information, is greatly overcome by combining it with boosting. RUSBoost automatically creates datasets where 35%, 50%, or 65% of the examples in the post-sampling dataset are minority class examples. The algorithm then reports the parameters that result in the best performance. This algorithm is simple and takes less training time and consumes less memory than its variation SMOTEBoost [7]. We tried to optimize various hyper-parameters such as learning rate, number of learning cycles and maximum number of splits. In addition to increasing the run-time significantly, the experimentation has shown us that the error loss is less but the model overfits the training data and gives poor result on test data. Hence to prevent the model from over-fitting, the number of learning cycles is limited to 200, learning rate is fixed to 0.1 and maximum number of splits is set to N. Here 'N' is sum of labeled output of training data.

3. Experimental results

All the experimentation is performed on the Physionet 2019 challenge dataset [5]. The dataset consists of over 45,000 patient records. Each patient record consists of hourly reading of vital signs, lab results and other demographics totalling 40 features. 5,000 records were provided initially and later on two sets of approximately 20,000 records were made available. All the three datasets were combined into a single one and then it was divided into a

training dataset (80%) and an internal test dataset (20%). We subdivided the training dataset into two parts: 1/3 is treated as cross-validation dataset and the remaining 2/3 is used as training dataset. The 3-fold validation scheme is chosen for efficiency. After model hyper-parameters are determined, the entire training dataset (36,000 records) is used to obtain the final model along with the utility score. The utility score is a specific metric devised by the challenge committee to detect usefulness of a given algorithm in predicting sepsis ahead of time. The metric rewards true early detection (true positives) and punishes false alarms (false positives) as well as failing to detect disease when it is present (false negatives).

The score could not be improved further because significant amount of data are missing. Also, the representation of classes is highly imbalanced. Only about 7% of patients are labeled as sepsis patients. We also observe that increasing number of weak learners or optimizing the hyper-parameters further overfits the training data and gives reduced results on hidden test data. Hence we limited values of hyper-parameters to pre-set values as described in earlier subsection. We could get classification loss minimized as shown in Figure 3.

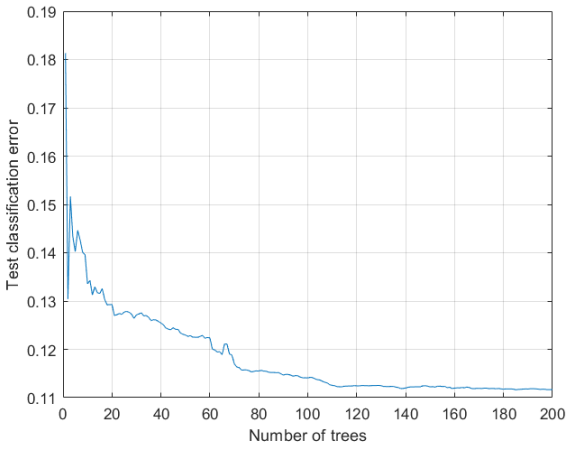

Figure 3. Classification loss for training data

This model is used to obtain utility score on the hidden dataset maintained by the Physionet organizers. On a hidden dataset maintained by organizers, a utility score of 0.192 was obtained. Other statistics on a test data maintained by Physionet is as follows:

Table 1. Performance score on individual test data set

Test set	AUROC	AUPRC	Accuracy	F-measure	Utility
Set A	0.579	0.057	0.929	0.171	0.274
Set B	0.621	0.038	0.928	0.120	0.233
Set C	0.626	0.014	0.765	0.036	-0.246

From the results, it is apparent that the model has worked satisfactorily for Set A and B but is not generalized enough. The organizers kept Set C completely isolated and it was not used during evaluation of submission phase. The model is giving negative utility score for Set C means it is predicting higher false negatives than true positives. Clearly records in Set C are sufficiently different than in Set A or B such that the trained model is under-fitting this data set. It will be instructive to know details of Set C so that root cause of under-fitting can be investigated.

4. Alternatives considered for improving score

The utility score obtained on internal test data as well as on hidden test data maintained by organizers is lower so various approaches are tried to improve it.

- Drop features which have high percentage of NaN values: This strategy resulted in lower utility score. Whatever small amount of information present in the feature is lost with this approach which adversely impacts the training of the model. Some teams adopted masking to deal with missing data. Here the additional feature indicates how long a feature value was available or valid. No imputation is done for missing values yet the results are better.
- Removal of outliers: From entire dataset, based on inter quartile range of a feature, outliers are removed. It resulted in lower utility score. If these extreme values would have been replaced by their maximum values (instead of complete removal), the results might have been better. However, the authors did not investigate the root cause for the issue.
- Feature creation: New features are created based on difference between hourly records of a feature. Since there is no new information this strategy did not result in improving the utility score. However, the strategy in itself is not wrong. There are other teams which created new features such as mean, standard deviation based on hourly data available till that time i.e. to calculate mean of a feature for 2nd hour, its 1st hour and 2nd hour recordings are considered, to calculate mean of a feature for 3rd hour, its 1st, 2nd and 3rd hour recordings are considered etc. This strategy seems to improve the utility score. Another team created various features by taking ratio of features with different exponents. Genetic algorithm is used to calculate values of these exponents.
- Imputation methods: Linear imputation is implemented within a patient record. Changing the imputation method to 'next' or 'previous' results in lower utility score. There were many teams which did not do imputation and relied on masking to handle missing data. This strategy, too, resulted in producing good utility score.

5. Conclusion

We, `Team_Tesseract`, propose a methodology to detect onset of sepsis ahead of time. The model is applied on Physionet 2019 challenge dataset.

On a hidden test data set maintained by challenge organizers, a utility score of 0.192 is obtained with an overall rank of 49 out of 79 officially ranked entries. The software is available as open source software under GNU license. We plan to refine strategy to handle missing data either by different imputation method [8] or by use of masking. We will also employ deep learning tools [9] which will help in improving classification accuracy. On a related but different note, if nursing assessments are available then Rothman Index [10] can be computed which is a better predictor in sepsis detection [11].

6. Conflict of interest statement

Shailesh Nirgudkar is employed at MathWorks and Tianyu Ding was an intern at MathWorks at the time of this work.

References

[1] Singer M, et al. The Third International Consensus Definitions For Sepsis And Septic Shock (Sepsis-3). Journal of American Medical Association 2016;.

[2] Kumar A, 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.

[3] Seymour C, et al. Time To Treatment And Mortality During Mandated Emergency Care For Sepsis. New England Journal of Medicine 2017;376:2235–2244.

[4] Henry K, et al. A Targeted Real-time Early Warning Score (TREWScore) For Septic Shock. Science Translational Medicine 2015;7.

[5] Reyna M, et al. Early Prediction Of Sepsis From Clinical Data: The PhysioNet Computing In Cardiology Challenge 2019. Critical Care Medicine In Press;.

[6] Seiffert C, et al. RUSBoost: Improving Classification Performance When Training Data is Skewed. IEEE Transactions on Systems Man and Cybernetics Part A Systems and Humans 2010;40:185–197.

[7] Chawla N, et al. SMOTE: Synthetic Minority Oversampling Technique. Journal of Ariticial Intelligence Research 2002;16:321–357.

[8] Camino R, et al. Improving Missing Data Imputation With Deep Generative Models. arXiv e prints 2019; arXiv:1902.10666.

[9] Lauritsen S, et al. Early Detection Of Sepsis Utilizing Deep Learning On Electronic Health Record Event Sequences. Dansk Tidsskrift for Akutmedicin 2019;2:39.

[10] Finley D, et al. Measuring The Modified Early Warning Score And The Rothman Index: Advantages Of Utilizing The Electronic Medical Record In An Early Warning System. Journal of Hospital Medicine 2013;9:116–119.

[11] Rothman M. The Rothman Index. Rothman Healthcare Corporation 2013;.

Address for correspondence:

Shailesh Nirgudkar
Control Design Automation, MathWorks,
1 Lakeside Campus Drive, Natick, MA 01760 USA.
shailesh.nirgudkar@gmail.com