Automated Prediction of Sepsis Onset Using Gradient Boosted Decision Trees

John Anda Du, Nadi Sadr, Philip de Chazal

Charles Perkins Centre and School of Electrical and Information Engineering,
The University of Sydney, Australia

Abstract

In this study, we developed an automatic algorithm that predicts onset of sepsis using hourly clinical data from patients in an ICU setting. We participated as team "Sepsyd" in the PhysioNet/Computing in Cardiology 2019 Challenge and were ranked 2nd with an official final test score of 0.345. Our developed system processed all the clinical input variables provided in the Challenge. We first applied a preprocessing step that applied a log transform to selected variables and imputed missing values of the variables. After preprocessing, a feature set was formed including the 40 preprocessed variables, 34 missing value flags, the changes in the time series in the vital signs variables and the variance of the vital signs variables. Following this, the features of the present hour were combined with the features of the past 5 to 8 hours of data. These combined features were then processed with a gradient boosting tree classifier to estimate the likelihood of a positive sepsis classification at each time step. We compared the utility score of a number of different system configurations using 3-fold cross validation on the training data. Our best system, assessed on the test set, used a maximum tree depth of 4, a look back of 5 hours, and processed the clinical input variables combined with the missing value flags.

1. Introduction

Sepsis is a critical health syndrome caused by infection which results in pathologic, physiologic, and biochemical abnormalities [1], [2]. It is a prevalent issue for critically ill patients, with a death rate greater than the combined breast and bowel cancer deaths [3]. A mortality rate of about 50% is reported in patients as a result of severe sepsis and sepsis shock, with average costs of about US$17B annually in the USA associated with sepsis [4]. Early detection of the syndrome affects the treatment procedure, potentially reducing the associated costs on healthcare system [3], [5] and is a vital key to improving the survival of sepsis patients. Developing automated systems are a key enabler for early detection [3], [6].

The clinical protocol for identifying sepsis involves detection of two or more of the following symptoms: body temperature outside of the range of 36-38℃, heart rate (>90bpm) and respiratory rate (>20 breath/min or $PaCO_2$ (<32mmHg) and blood cell count (>12,000 or <4000 cells/mm^3) or immature band forms (>10%) [3]. Automated systems for early diagnosis of sepsis using clinical- and laboratory-based data have been widely studied in the literature [7]–[11]. Through development of electronic health records and electronic surveillance systems for healthcare systems, a number of studies have evaluated models for automated sepsis detection and their effectiveness [7], [8], [10], [12], [13]. The studies mostly proposed automated systems of evaluating clinical data for prediction of sepsis using different machine learning algorithms such as support vector machine, k-nearest neighbor, decision trees, regression trees, random forests, logistic regression and lazy Bayesian rules [7], [10]–[12].

This study aimed to predict sepsis onset using a model developed from clinical data provided by the *PhysioNet Computing in Cardiology Challenge 2019* [14]. We examined the performance of different features and classifiers processing the clinical data and selected a high performing model for final evaluation on the unseen test data. In our proposed algorithm, we provide a model that uses all the clinical input data after preprocessing. The feature set contains the clinical data, as well as missing value flags, and the changes and variance of the vital sign variables. A gradient boosting classifier was used to estimate the likelihood of a sepsis at each time step. Weighted cross-entropy was used as the loss function. The clinical data and the different methods used for signal processing with the results and discussion are provided in the following sections.

2. Input Data

The dataset was provided by the *PhysioNet/ Computing in Cardiology Challenge 2019* [14]. It comprised of clinical data from ICU patients of three hospitals. The open access training data and associated sepsis labelling was from two hospitals (A and B) while the hidden test data contained data from the three hospitals (A, B and C). The clinical data included 8 vital signs, 26 laboratory and 6 demographic values. A timestamp of clinical data was recorded every hour but not all tests were

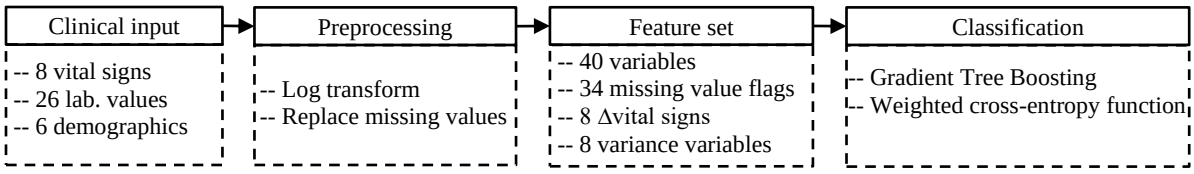

Figure 1. The block diagram of the proposed automated system for prediction of sepsis using clinical data.

performed every hour. When a test was not performed its missing value was recorded as 'NaN'. A sepsis label is provided at every hour timestamp. For patients who developed sepsis, the label was '0' up to 6 hours before sepsis and '1' otherwise. The details of the dataset can be found in the Challenge overview paper [14].

3. Methods

Figure 1 depicts our automatic system for sepsis prediction. It processes the clinical input data after preprocessing using methods including log-transformation of selected raw feature values and imputation of missing values. After preprocessing, a feature set is created for each hour of data. The feature set contained the preprocessed 40 variables from the clinical data, as well as 34 missing values flags, the difference of the vital signs variables from previous time steps and variance over the past six-hour period of the vital signs variables. The feature set of the present hour and the past six hours were then combined and passed to a gradient boosted Decision Tree classifier model. A weighted cross-entropy loss function was used. Our system outputs an estimated likelihood of a sepsis developing in the next 6 hours at each hourly time step.

3.1. Data preprocessing

The first preprocessing step involved applying a histogram correcting transform to the data. We first evaluated the distribution of the 40 clinical data values. We investigated the histogram of every clinical data and a log transform was applied to the clinical data values with an exponential or long-tailed distribution. As a result, the data more closely approximated a Gaussian distribution.

Following this, normalisation and missing values replacement was implemented. The data was normalised so that each variable have a mean of zero based on the available training data. After normalising, missing values were replaced by zero-values (i.e. mean values) until the first occurrence of a valid value. After the first valid value, variables were replaced with the most recent valid value.

3.2. Feature augmentation

After preprocessing, 40 preprocessed variables of the input series were used. They were combined with a 34-length mask vector for missing value flags. The mask was set to a value of one, if the original feature was present and zero, if the original feature was missing and thus imputed at that time step. The mask vector only covered the first 34 variables, as the 6 demographic variables were present at all time steps. This provided a core feature set containing 74 features.

The core feature set was supplemented with additional features including an 8-length delta (changes of the variables in time steps) and an 8-length variance vector. The 8-length delta was calculated by taking the difference of each vital sign variable from its previous time-step value, and the 8-length variance vectors were calculated by taking the variance of each vital sign over the past six hours. For each hour of data therefore, we had a vector of L=74, 82 or 90 features. Before applying the feature set to the classifier, the features of the present hour were appended with those of the past 3 to 9 (H) hours, and finally with zeros if the present hour was less than H. Thus, a vector of L*H features was created and applied to the classifier.

3.3. Performance measures

Performance was measured using a utility score defined by the Challenge organisers for each prediction [14]. The utility function rewards or penalises classifiers for their predictions within 12 hours before and 3 hours after sepsis onset time and was normalised as described in [14].

3.4. Loss function

The model was trained using a weighted cross-entropy function, with positive examples weighted 40:1 in order to reflect the weighting of the utility function, whereby false negatives were counted as -2 and false positives as -0.05. The loss function (L) for a patient was defined as follows.

$$L = -\frac{1}{T}\sum_{i=0}^{T} 40 * y_i * \log(\hat{y}_i) + (1 - y_i) * \log(1 - \hat{y}_i)$$

where T is the number of time steps recorded for a patient, y_i is the sepsis label output at time step i and $\hat{y}_i$ is the classifer output at time step i.

Table 1. The effect of input preprocessing on utility scores with H=5 hours combined

Preprocessing	Utility Score
No preprocessing	0.3931
Feature standardising	0.3926
Log transforming, feature standardising	0.3933

Table 2. A comparison of utility scores for different tree depths (D) using 40 training epochs

D	3	4	5
Utility score	0.3912	0.3933	0.3881

Table 3. The utility scores of different feature combinations

H	Base+Mask	Base+Mask+Delta	Base+Mask+Var	All
3	0.3927	0.3931	0.3946	0.3944
4	0.3936	0.3934	0.3950	0.3941
5	0.3933	0.3933	0.3947	0.3946
6	0.3932	0.3935	0.3956	**0.3959**
7	0.3913	0.3909	0.3947	0.3948
8	0.3935	0.3934	0.3943	0.3940
9	0.3911	0.3914	0.3948	0.3930

H: hours of combined features, **Base**: 40 normalised and transformed clinical features, **Mask**: 34 missing value flags, **Delta**: 8 changes in vital sign features, **Var:** 8 variance of vital sign features, **All:** Base+Mask+Delta+Var features

Table 4. The performance of the final system on the training and the full test sets

	10-fold cross-validation on train set						Official test set results			
	Train			Test						
	Full	Set A	Set B	Full	Set A	Set B	Full	Set A	Set B	Set C
Utility score	0.455	0.472	0.438	0.400	0.416	0.376	0.345	0.409	0.396	-0.042
Accuracy	0.861	0.822	0.901	0.858	0.819	0.899	-	0.819	0.901	0.785
F measure	0.147	0.144	0.152	0.133	0.132	0.136	-	0.131	0.142	0.050
AUROC	0.863	0.846	0.880	0.834	0.815	0.851	-	0.811	0.853	0.805
AUPRC	0.142	0.136	0.155	0.111	0.111	0.116	-	0.105	0.119	0.065

- Not reported in the official test set results. AUROC: Area under the receiver operator curve. AUPRC: Area under the precision recall curve.

3.5. Classifier

The classifier is a gradient boosting model implemented using the library Xgboost [15], [16], creating up to 30 decision trees. Models were tested and compared by means of a stratified 3-fold split, trained for a fixed 40 epochs with a learning rate of 0.2. Training for more than 40 epochs led to over-fitting and an overall decrease in validation set performance.

The final submitted model was trained by applying 10-fold cross validation to the data, using a learning rate of 0.1, and monitoring the AUPRC of the validation set of each fold for early stopping.

3.6. Performance estimation

Performance was estimated on 40,336 patients of the training set using 3-fold cross validation. The training set contained data from 2/3rd's of the patients with around 1 million time slices, each representing a time-step of data, and the resulting model was evaluated on the remaining 1/3rd of the patients.

4. Results

Our team name in the Challenge was "Sepsyd". We first evaluated the performance results of applying different preprocessing to the input data. Table 1 shows a model with H=5 trained with different combinations of input preprocessing. The results show that applying both log transform and normalisation outperformed other methods and was used for all following models.

Table 2 explores different tree depths. Using 3-fold cross validation, the best results were obtained with a maximum depth of 4. Table 3 shows the average cross-validated hold-out set performance of different models examined. In each case, inclusion of variance features improved the performance of the classifier, while 1-step deltas did not have much of an impact, and in some cases resulted in worse performance. We found that hold-out performance varied significantly based on how the data was shuffled and split into three folds, so three different

configurations were examined (created by shuffling using different random seeds) and the average performance of each model over these three configurations is calculated and reported. Table 4 shows the utility scores and other performance measures of our best submitted system.

5. Discussion and Conclusion

Table 3 shows that the highest performance for a classifier was obtained with H=6, ie the previous 6 hours included with the present hour, and the inclusion of the mask and variance vectors, but not the delta vectors. It obtained an average utility of 0.3981 on the holdout fold, after 3-fold validation. We were not able to submit this model for testing in the official phase of the competition.

A model with H=5 and base+mask vectors (obtaining a 3-fold cross-validated score of 0.3933 and 10-fold cross-validated score of 0.4004 during training), was trained using all available training data and submitted for final testing on the unseen test data. It achieved the second highest official score of the competition with a utility score of 0.345 on the full test set.

The same core model was used with additional features during the hackathon with team name "Sepsyd" and achieved the third highest official score of the hackathon with a utility score of 0.329. The additional features included sequential organ failure assessment (SOFA) [2], national early warning score (NEWS), acceleration in time variables, time differentials over multi-hour timescales. As the utility score was lower than our official score of 0.345, it suggests the new features don't enhance discrimination.

The score of our best model was 0.46 on training data, suggesting some overfitting in the model. Further improvements might be made by varying the hyper-parameters of the model, particularly parameters that effect generalisation such as regularisation. Another area to explore is the implementation of a custom loss function that approximates the utility score. Rubin *et al*. achieved 2nd place at hackathon and used a loss function that approximated the utility function by using custom time-dependent weights to the different error types [17]. Finally, inclusion of additional features (for example differentials calculated over a number of time slices) might also lead to gains in the performance by further exploiting any additional temporal information in the data.

References

[1] C. M. Torio and R. M. Andrews, "National Inpatient Hospital Costs: the Most Expensive Conditions by Payer, 2011," *Heal. Cost Util. Proj.*, vol. 31, no. 1, pp. 1–12, 2013.

[2] M. Singer *et al.*, "The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)," *JAMA*, vol. 315, no. 8, pp. 801, 2016

[3] H. McClelland and A. Moxon, "Early identification and treatment of sepsis.," *Nurs. Times*, vol. 110, no. 4, pp. 22–8, 2014.

[4] A. C. Derek, W. T. Linde-Zwirble, J. Lidicker, G. Clermont, J. Carcillo, and M. R. Pinsky, "Epidemiology of severe sepsis in the United States: Analysis of incidence, outcome, and associated costs of care," C*ritical Care Med.*, vol. 29, no. 7, pp. 1303–1310, 2001.

[5] A. Bravi, G. Green, A. Longtin, and A. J. E. Seely, "Monitoring and identification of sepsis development through a composite measure of heart rate variability," *PLoS One*, vol. 7, no. 9, pp. 1–7, 2012.

[6] E. Slade, P. S. Tamber, and J. L. Vincent, "The surviving sepsis campaign: Raising awareness to reduce mortality," *Crit. Care*, vol. 7, no. 1, pp. 1–2, 2003.

[7] J. S. de Bruin, W. Seeling, and C. Schuh, "Data use and effectiveness in electronic surveillance of healthcare associated infections in the 21st century: A systematic review," *J. Am. Med. Informatics Assoc.*, vol. 21, no. 5, pp. 942–951, 2014.

[8] R. Freeman, L. S. P. Moore, L. García Álvarez, A. Charlett, and A. Holmes, "Advances in electronic surveillance for healthcare-associated infections in the 21st Century: A systematic review," *J. Hosp. Infect.*, vol. 84, no. 2, pp. 106–119, 2013.

[9] L. A. Despins, "Automated detection of sepsis using electronic medical record data: A systematic review," *J. Healthc. Qual.*, vol. 39, no. 6, pp. 322–333, 2017.

[10] A. M. Sawyer *et al.*, "Implementation of a real-time computerized sepsis alert in nonintensive care unit patients," *Crit. Care Med.*, vol. 39, no. 3, pp. 469–473, 2011.

[11] S. Mani *et al.*, "Medical decision support using machine learning for early detection of late-onset neonatal sepsis," *J. Am. Med. Informatics Assoc.*, vol. 21, no. 2, pp. 326–336, 2014.

[12] B. N. Brandt *et al.*, "Identifying severe sepsis via electronic surveillance," *Am. J. Med. Qual.*, vol. 30, no. 6, pp. 559–565, 2015.

[13] M. Inada-Kim, B. Page, I. Maqsood, and C. Vincent, "Defining and measuring suspicion of sepsis: An analysis of routine data," *BMJ Open*, vol. 7, no. 6, pp. 1–7, 2017.

[14] M. Reyna *et al.*, "Early prediction of sepsis from clinical data: the PhysioNet/Computing in Cardiology Challenge 2019," *Crit. Med.*, In press, 2019.

[15] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Ann. Stat.*, pp. 1189–1232, 2001.

[16] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Ann. Stat.*, vol. 28, no. 2, pp. 337–407, 2000.

[17] J. Rubin, Y. Chang, S. Pravaneh, and G. Boverman, "A multi-task imputation and classification neural architecture for early prediction of sepsis from multivariate clinical time series," *Crit. Med. 2019*, In press, 2019.

nadi.sadr@sydney.edu.au.
Charles Perkins Centre and School of Electrical and Information Engineering. The University of Sydney, NSW 2006, Australia