Early Prediction of Sepsis via SMOTE Upsampling and Mutual Information Based Downsampling

Shiyu Liu, Ming Lun Ong, Kar Kin Mun, Jia Yao, and Mehul Motani

Department of Electrical and Computer Engineering
National University of Singapore, Singapore

Abstract

Sepsis is a life-threatening response to infection that can lead to tissue damage, organ failure and death. The early prediction of sepsis is important, as it reduces undesirable patient outcomes associated with late-stage septic shock. However, effective early prediction is challenging, because the data is often heavily imbalanced against positive sepsis diagnosis. If the class imbalance is not addressed, models trained will tend to overfit in favour of the majority class, leading to degraded performance on the minority class. In this paper, we suggest a two-step method which consists of a mutual information based downsampling algorithm and a Synthetic Minority Over-sampling Technique (SMOTE), in order to effectively perform early prediction of sepsis. Our team, Kent Ridge AI (ranked 77th), obtained a utility score of -0.164 on the full test set by using the proposed two-step method. Additionally, we report cross-validation results and identify several methods to improve performance.

1. Introduction

Sepsis is a life-threatening disease, claiming over 75 thousand lives every year in the United States [1], and is a leading cause of mortality amongst intensive-care unit (ICU) patients [2]. Early identification of sepsis amongst ICU patients is thus essential to prevent organ failure and death. However, effective early prediction of sepsis has proven to be difficult as sepsis biomarkers are not definitive, and often, patients at risk of sepsis have various disease complications [3]. As few patients are diagnosed with sepsis, a dataset of sepsis patients is often imbalanced. Within the PhysioNet Challenge 2019 dataset [4], only 7.98% of patients develop sepsis at one point or another. A classifier trained on an imbalanced dataset will be biased in favour of the majority class, causing misclassification of minority class samples [5].

In this paper, we propose a two-step method to address the imbalance in data, consisting of a mutual information-based downsampling algorithm to reduce the majority class, and the Synthetic Minority Over-sampling Technique (SMOTE) to increase the minority class. Using this two-step method along with a long short-term memory (LSTM) based neural network, our team, Kent Ridge AI (ranked 77th), obtained a utility score of -0.164 on the full test set. Additionally, we report cross-validation results and identify several methods to improve performance, such as including trend information.

2. Downsampling and Upsampling

Downsampling and upsampling are two common strategies employed to deal with imbalanced dataset. When data from the majority class is downsampled, a relatively smaller number of representative instances are selected. Downsampling is beneficial as it mitigates the overfitting effects, but excessive downsampling will cause a loss in useful information, and degrade the performance of the classifier [6]. Upsampling refers to the generation of synthetic samples from the minority class, to boost the number of samples so that the number of samples in minority class (after generation) is equal to the sample size of majority class. However, generating synthetic samples is difficult, and may cause overfitting as well if the generated samples are too similar to the original ones [7]. Ideally, both downsampling and upsampling methods should be used in tandem. In this section, we introduce a two-step method that maintains the balance between downsampling and upsampling methods.

2.1. Mutual Information Based Downsampling Algorithm

Mutual information (MI), a concept formed from information theory, can reliably quantify the dependency between random variables [8]. MI is a scaler quantity between two random variable, measuring the uncertainty of a random variable, given knowledge of another.

Our MI-based downsampling algorithm is used to select representative patients. Firstly, we assign a score to each

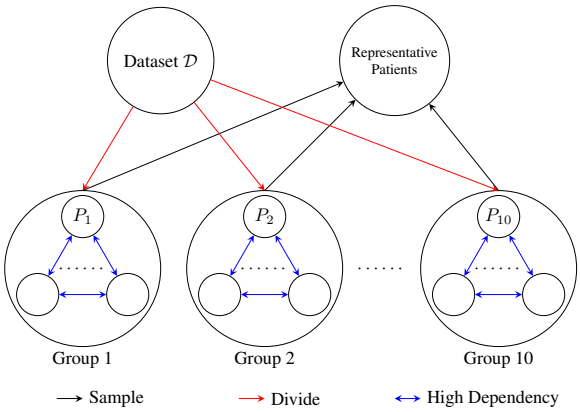

Figure 1. Illustration of Mutual Information based Downsampling Algorithm

patient, according to the following scoring function:

$$J(P_i) = \sum_{P_j \in \mathcal{D}, P_j \neq P_i} I(P_i, P_j), \tag{1}$$

where P_i, $P_j \in$ dataset $\mathcal{D}$ represent for i_{th} and j_{th} patient, respectively. The data length of patients maybe different due to length of stay. Therefore, we estimate the MI via nearest-neighbor based MI estimators [9]. After scoring each patient, we divide all patients into $L = 10$ groups (of approximately equal size) based on the descending order of all scores. Therefore, patients with similar scores will be grouped together and patients within the same group will tend to be highly dependent on each other (see Figure 1). Random sampling from each group will select representative patients of each group. Random proportional sampling from all groups gives selective patients of the whole dataset.

2.2. SMOTE

SMOTE [10] is considered as an effective upsampling algorithm to generate synthetic samples. SMOTE firstly identifies the feature vector, its nearest neighbour and take the difference between two. Then it generates a new point on the line segment by adding the random number to feature vector (see Figure 2). Unlike making copies of existing samples, SMOTE learns the topological properties of the neighbourhood of points in the minority class. Therefore, the classifier trained on the synthetic data generated by SMOTE is less likely to overfit.

3. Performance Evaluation

3.1. Data Information and Preprocessing

The PhysioNet Computing in Cardiology Challenge 2019 dataset [4] contains the demographic, vital sign information and lab test results of 27148 patients, at regular 1-hour intervals. A large number of features have missing values, so we impute these corresponding features with the latest historical values. We shift the target class ahead by 6 hours, so that we can predict sepsis 12 hours early. Finally, we perform normalization on the dataset, so that the range of feature values is between 0 and 1.

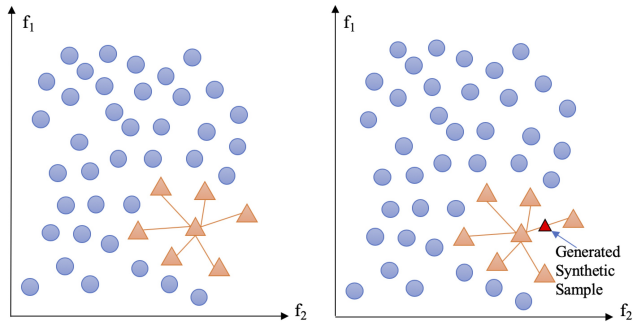

Figure 2. Illustration of Data before SMOTE (Left) and after SMOTE (Right) in 2D Space.

3.2. Benchmark Models

Four machine learning benchmark models are shortlisted and described below.

- The decision tree (DT) [11] is a tree-based algorithm for classification. Training data is repeated split according to splitting criteria (eg. entropy) with respect to an outcome variable, such that child nodes are more homogenous. Decision tree handles imbalance data classification problems well, and are benchmark models in many problems.
- The random forest (RF) [12] is an ensemble of decision trees. Bootstrapped samples of the training data are trained, and majority voting is performed for the final classification process. RF is cost-sensitive as it can incorporate class weights in the training process, penalising misclassification of minority class instances.
- The gradient boosting (GB) algorithm [13] minimizes loss in decision trees, by training instances in a sequential manner. The additional boosting step occurs to successively train on incorrectly classified examples, thus significantly improving the performance with respect to imbalanced data classifications [14, 15].
- The long short-term memory (LSTM) unit [16] is capable of learning information over a long sequence of temporal inputs. The unit utilises input, output and forget gates to store information over long time intervals, regulating the flow of information in and out of the cell. By overcoming the problem of vanishing or exploding gradients through memory blocks, the LSTM is designed for time-series data predictions.

	No D-sample		D-sample to 75%		D-sample to 50%		D-sample to 25%	
	F1	AUC-ROC	F1	AUC-ROC	F1	AUC-ROC	F1	AUC-ROC
Have U-sample	0.245	0.522	0.284	0.519	0.303	0.532	0.288	0.514
No U-sample	0.236	0.529	0.265	0.514	0.285	0.523	**0.317**	**0.535**

Table 1. Intermediate Training Performance of Decision Tree with and without down/up sampling method

	No D-sample		D-sample to 75%		D-sample to 50%		D-sample to 25%	
	F1	AUC-ROC	F1	AUC-ROC	F1	AUC-ROC	F1	AUC-ROC
Have U-sample	0.186	0.513	0.235	0.541	**0.238**	**0.542**	0.224	0.533
No U-sample	0.153	0.531	0.101	0.516	0.131	0.522	0.158	0.526

Table 2. Intermediate Training Performance of Random Forest with and without down/up sampling method

	No D-sample		D-sample to 75%		D-sample to 50%		D-sample to 25%	
	F1	AUC-ROC	F1	AUC-ROC	F1	AUC-ROC	F1	AUC-ROC
Have U-sample	0.321	0.522	0.322	0.558	**0.341**	**0.563**	0.336	0.533
No U-sample	0.172	0.533	0.146	0.522	0.152	0.527	0.172	0.527

Table 3. Intermediate Training Performance of Gradient Boosting with and without down/up sampling method

	No D-sample		D-sample to 75%		D-sample to 50%		D-sample to 25%	
	F1	AUC-ROC	F1	AUC-ROC	F1	AUC-ROC	F1	AUC-ROC
Have U-sample	0.285	0.505	0.303	0.504	0.335	**0.527**	**0.356**	0.524
No U-sample	0.112	0.519	0.267	0.504	0.212	0.486	0.117	0.502

Table 4. Intermediate Training Performance of LSTM with and without down/up sampling method

3.3. Experiment Setup

The whole dataset $\mathcal{D} \in \mathbb{R}^{\mathbb{N}\times\mathbb{M}}$ ($\mathbb{N}$: number of samples; $\mathbb{M}$: number of features) is randomly split into two subsets: training dataset and test dataset at patient level. Namely that 80% of patients are used to train the classifier and the rest 20% of patients are used for evaluation. Then we perform MI based sampling algorithm on the majority class of training data to sample 75%, 50% and 25% of training dataset, respectively. After downsampling, we apply SMOTE to do upsampling on the downsampled training dataset to make sure that the number of samples in minority class and majority class are equal.

For decision tree (splitting criterion = entropy), gradient boosting (number of estimator = 100) and random forest (number of estimator = 45), we train them to perform 12-hour early prediction. During testing, the missing values of test data are imputed with the corresponding mean values obtained from training data and only the latest time step of test data is used for evaluation. For the LSTM based neural network, it consists of 6 hidden layers while first two layers are LSTM with 5 and 10 units, respectively. The rest four layers are dense layer with size = [256,128,64,32]. As LSTM requires three dimensional input to capture the temporal relationship, we reshape the training dataset into a three dimensional array $\in \mathbb{R}^{\mathbb{K}\times 3\times\mathbb{M}}$ via a sliding observation window of size equal to three (i.e., time step = 3). Moreover, the LSTM based neural network is trained for 300 epochs using Adam [17] with a learning rate of 0.001, batches of 128 and sigmoid activation function. During testing, the missing values of test data are imputed with the corresponding mean values obtained from training data and all past observations of each patient are used to perform classification task. Finally, AUC-ROC and F1 score [18] are used to evaluate the performance and both of them are averaged over 10 runs.

3.4. Performance Comparison

In this subsection, we report intermediate training performance in the form of F1 score and AUC-ROC values. The performance for various classifiers with and without down/up sampling are shown in Table 1 to Table 4. The best performance for each classifier is highlighted in red and bolded. In Table 1 to Table 4, the D-sample and U-sample represents MI based downsampling and SMOTE upsampling respectively. Using the SMOTE upsampling method improves the performance (compare row 1 to row 2 in Table 1 - Table 4), while applying MI based downsampling algorithm further boosts the performance. The best performance is observed using LSTM based neural network (25% downsampling rate & SMOTE upsamling), with an F1-score of 0.356 for 12-hour early prediction of sepsis. By using the proposed method together with LSTM based neural networks, our team, Kent Ridge AI, obtains a utility score of -0.047, -0.288, -0.361 on the test set A, B,

Prediction Horizon	12-Hour		9-Hour		6-Hour		3-Hour	
	F1	AUC-ROC	F1	AUC-ROC	F1	AUC-ROC	F1	AUC-ROC
LSTM (D-sample to 25%)	0.356	0.524	0.347	0.539	0.361	0.544	0.372	0.567

Table 5. Intermediate Training Performance of LSTM with different prediction horizons

C, respectively. Overall, our team (ranked 77th) obtains a utility score of -0.164 on the full test set.

To examine the performance of different prediction horizons (e.g., 9-hour early prediction), we also evaluate the proposed two-step method via LSTM based neural network (25% downsampling rate & SMOTE upsampling) and the results are shown in Table 5. We observe that the performance roughly increases while we reduce the prediction horizon. We believe it is because that the relationship between past features and label are stronger as we reduce the prediction horizon, leading to better performance.

In Tables 1 - 5, we report the F1-Score and AUC-ROC for the algorithms tested. We note that the accuracies for the algorithms tested range from 0.57 to 0.63 in the experiments we conducted.

4. Reflections

In this paper, we suggest a two-step method to address the class imbalance issue in the given sepsis dataset. Our approach consists of a mutual information based downsampling algorithm and a SMOTE based upsampling. While we believe that this method has the potential to work in theory, we acknowledge that this approach did not bear fruit during the challenge itself. We suspect that the poor performance is connected to how we pre-processed the data and can be improved with proper feature engineering. Specifically, we propose the following improvements: (i) Using feature selection methods to select a relevant subset of features [19]; (ii) Using trend features to track the changes between feature values; (iii) Using a larger variety of imputation methods to treat missing values in the dataset.

Acknowledgments

This work was supported by the Singapore Ministry of Education under grants WBS R-263-000-D35-114 and WBS R-263-000-D64-114.

References

[1] Angus D, Linde-Zwirble W, Lidicker J, Clermont G, Carcillo J, Pinsky M. Epidemiology of severe sepsis in the united states: analysis of incidence, outcome, and associated costs of care. Crit Care Med 2001;29:1303–10.

[2] Singer M, Deutschman C, Seymour C, et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). JAMA 2016;315(8):801–10.

[3] Vincent JL. The clinical challenge of sepsis identification and monitoring. PLoS medicine 2016;13(5):e1002022.

[4] Reyna MA, Josef C, Jeter R, Shashikumar SP, M. Brandon Westover MB, Nemati S, Clifford GD, Sharma A. Early prediction of sepsis from clinical data: the PhysioNet/Computing in Cardiology Challenge 2019. Critical Care Medicine 2019;In Press.

[5] Shaza.M Abd E, Ajith A. A review of class imbalance problem. Journal of Network and Innovative Computing 2013; 1:332–340.

[6] Lunardon N, Menardi G, Torelli N. Rose: a package for binary imbalanced learning. R Journal 06 2014;6:79–89.

[7] Sun Y, Wong AKC, Kamel MS. Classification of imbalanced data: a review. IJPRAI 2009;23:687–719.

[8] Cover TM, Thomas JA. Elements of Information Theory, 2nd edition. John Wiley & Sons, 2006.

[9] Kraskov A, Stoegbauer H, Grassberger P. Estimating mutual information. Physical Review E69 019903 2004;.

[10] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. Smote: Synthetic minority over-sampling technique. J Artif Int Res June 2002;16(1):321–357. ISSN 1076-9757.

[11] Quinlan JR. Induction of decision trees. Mach Learn March 1986;1(1):81–106. ISSN 0885-6125.

[12] Breiman L. Random forests. Machine Learning 2001; 45(1):5–32.

[13] Friedman JH. Stochastic gradient boosting. Comput Stat Data Anal February 2002;38(4):367–378. ISSN 0167-9473.

[14] Brown I, Mues C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Systems with Applications 2012;39(3):3446 – 3453. ISSN 0957-4174.

[15] Burez J, den Poel DV. Handling class imbalance in customer churn prediction. Expert Systems with Applications 2009;36(3, Part 1):4626 – 4636. ISSN 0957-4174.

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

[17] Kingma D, Ba J. Adam: A method for stochastic optimization. Technical Report arXiv:1412.6980, ArXiV, 2014.

[18] Sokolova M, Japkowicz N, Szpakowicz S. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In Australasian joint conference on artificial intelligence. Springer, 2006; 1015–1021.

[19] Liu S, Yao J, Zhou C, Motani M. Suri: Feature selection based on unique relevant information for health data. In 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2018; 687–692.

Address for correspondence:

Liu Shiyu
4 Engineering Drive 3, E4-06-12,
Communication Lab, Singapore 117583
Shiyu_liu@u.nus.edu