Predicting Septic Shock Pre-Diagnosis in Patients with Sepsis

Joshua Krachman and Zachary Zarubin

***Abstract* — In this paper we will fit a logistic regression model to predict Septic Shock, compute a ROC to test the validity of our model, and plot a histogram of the Early Warning Time to see how effective our predictions are.**

I. INTRODUCTION

Sepsis is a disease that is caused by an immune system response in an attempt to fight an infection already present in the body. Many times, this is life-threatening, as the body releases harmful chemicals that can damage other tissues in various organ systems other than the one currently with the infection. This causes a chain reaction of malfunctions that can slow deteriorate the patient's health. Currently, there is a global need to be able to identify and predict Septic shock, which is the late-stage, more deadly version of Sepsis. There are over 5 million Sepsis-related deaths every year [1]. Early prediction of Septic Shock can lead to an earlier intervention, which decreases the likelihood of mortality, because every single hour a patient waits for treatment, there is an 8% increase in mortality [2]. Necessary treatments, such as antibiotics, fluids, and vasopressors, are usually administered following post-Septic Shock diagnosis. They are not commonly administered during Sepsis, as they can hurt the patient's normal cardiac activity. If we know a patient is going to go into Septic Shock beforehand, we can administer these necessities, since they will receive them anyways. This treatment will be more effective, since the Sepsis will not have reached as progressive of a stage at time of dose. By intervening early, we can save lives and hospitals can also save money by minimizing resources needed to treat patients and time patients spend in the ICU. Therefore, we present a logistic model where we can predict Septic Shock for patients with Sepsis before it is actually diagnosed.

II. TRAINING OUR MODEL

A. Pre-Processing

The data we used was already split up into a training and testing set. Most of the training data we received was pre-processed already to include 28 features (and one index column, which we deleted) from various time-steps of 6,737 patients diagnosed with Sepsis. At each timestep, a patient was given a classifier, 0 or 1, depending on if they were diagnosed with Septic Shock (1) or not (0) at that specific timestep. We put this info into a separate column vector and horizontally concatenated it to have a label array of 0s and 1s. We normalized every patient feature point by subtracting the mean from its feature column and dividing by the feature column's standard deviation at each metric. This ensured the weights of our logistic model would be of a comparable magnitude for comparison. The testing data was pre-processed in a near identical manner. The only difference in preprocessing of the test data was that we removed the patient information at each time step from the feature vector and saved this information to an array for future reference.

B. 10-Fold Validation

10-Fold validation was used in order to be able to justifiably feed a smaller chunk of training data into our Stochastic Gradient Descent. The general idea was to split our training data into 10 equal, stratified chunks (using Sklearn's Stratified K-fold), ensuring that each tenth of the dataset had 0s and 1s. Then we tested Chunk 1 on Chunks 2-10 as training data, tested Chunk 2 on Chunks 1 and 3-10 as training data, etc. until all chunks were used as testing data. At every single combination, we created several logistic models based off of different values of lambda (the constant used in the loss function to penalize use of unimportant features). Using Stochastic Gradient Descent, we figured out the optimal lambda value to minimize our loss function, thereby maximizing the likelihood of observing our data.

C. Stochastic Gradient Descent

Given classifiers 0 and 1, we wanted to maximize our specificity and sensitivity for predicting 1s (Septic Shock) at each patient time step that was in our data. To do this, we used the principle of maximum likelihood and chose a loss function, which we sought to minimize, to find this maximum likelihood of observing our data. As derived in the math, we add L1-regulation to our loss function to ensure that we penalized the excessive inclusion of unimportant features. This helped us prevent overfitting of our logistic model.

$$Consider\ \{(x_i, y_i)\}_{i=1}^{N}$$

$$f(x) = W^T X + B$$

$$\sigma(x) = \frac{1}{1+e^{-X}} = \frac{1}{1+e^{-(W^T+B)}}$$

$$P[Y=1|X] = \sigma\big(f(x)\big)$$

$$P[Y=0|X] = 1 - P[Y=1|X]$$

$$P_Y = \prod_{i=1}^{N} \sigma\big(f(x_i)\big)^{yi} \Big(1 - \sigma\big(f(x_i)\big)\Big)^{1-Y_i}$$

$$argmax_\theta\big(P_y\big) = argmax_\theta(\log(P_Y))$$

$$Leads\ to\ our\ loss\ function:$$

$$\min_\theta -\frac{1}{N}\log(P_Y) = -\frac{1}{N}\sum_{i=1}^{N}[Y_i \log\big(\sigma\big(f(x_i)\big)\big(1 - Y_i\big)\log\big(1 - \sigma\big(f(x_i)\big)\big)]$$

However, we add a "L1 regulation" term:

$$\lambda\|w\| = \lambda\sum_{i=1}^{28}|w_i|$$

Therefore, we wish to minimize the term

$$\sum_{i=1}^{N} L(f(x_i, y_i) + \lambda \|w\|^2$$

D. Removing Unnecessary Features

In order to find the best lambda to minimize our loss function and maximize our maximum likelihood, we first needed to determine the magnitude of our best lambda.

To do this, we created a matrix that would store AUC values derived from the many logistic models we constructed from various combinations of training/testing sets (from within the overall training set only) and chosen lambda values. We created 10 rows in our AUC matrix to hold the AUCs derived at all 10 different combinations of testing/training data. We tested lambda values with various magnitudes between 10^{-6} *and* 10^{-1} at each testing and training data combination. This populated the columns of the AUC matrix at every single row. Once the AUC matrix was populated, we computed the mean of each column in the AUC matrix (corresponding to the mean AUC derived from using a particular magnitude of lambda over the various combinations of data). We put these mean AUC values into a one-dimensional array. The maximum of this one-dimensional array was then found, and the index of the location corresponded to the best magnitude of lambda to use.

Once the magnitude of the best lambda was determined, we needed to find an even more specific lambda. We performed the same steps mentioned above, using lambda values near the previous lambda magnitude identified, to compute an AUC matrix for specific lambda values. We then took the mean AUCs of each column and found the maximum of the mean AUCs. The index where this value was found corresponded to the best specific lambda value. We wanted to maximize AUC by using this specific lambda in the construction of our final prediction logistic model. The higher the AUC, the better our ratio of true positives to false positives was in the logistic model.

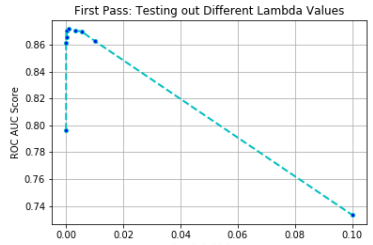


Fig. 1. Our first Stochastic Gradient Descent gave us an optimal lambda value equal to 0.001, as seen in the peak of this first pass graph.

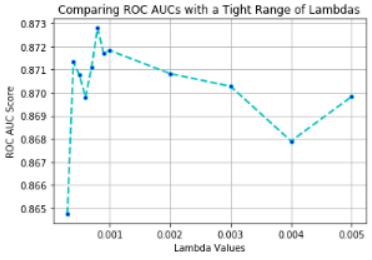


Fig. 2. Our second Stochastic Gradient Descent gave us a more specific lambda value equal to 0.0008, as seen in the peak of this graph.

E. Determining the Optimal Features

We constructed a logistic model using the specific lambda value we found and trained the logistic model on the entire training set feature matrix and training set labels. We then printed the coefficients of the logistic model to show which coefficient weights were set to zero. The features with coefficients set to 0 were determined by the model to be unimportant in classifying Septic shock. This ensured only the meaningful features in predicted Septic shock had weights that would be applied to the prediction score of every patient. Only having weights applied to the important features was important to avoid overfitting our model to the training set, which would cause low performance if the model was applied to another set of data for classification. The model had a specific lambda value to provide a penalty for times when excessive features were included in the model's prediction.

F. Finding a Threshold Value to Best Classify Predictions

After training our model on the entire training set with a specific lambda value, we were able to derive prediction scores of each patient at each time step of the training data. We used the predict_proba function, with the parameter being the training patient features, to derive prediction scores between 0 and 1 of the likelihood each patient was classified as a 0 and then as 1 at each time step. We extracted only the second column of this output, which corresponded to scores each patient was classified as a 1 at each time step. We did this to assign prediction labels based of the prediction score that showed how likely a patient was to be classified as having Septic Shock.

Once we had a prediction score of a patient classifying as a 1 at each timestep, we needed to find a threshold to classify these data points as 0 or 1. We used Sklearn's roc_curve function to evaluate true and false positive rates of classification (comparing predicted labels to actual labels) over many thresholds. We plotted these true positive and false positive metrics at every threshold to form a ROC curve. We then evaluated which point on the ROC curve was closest to the point (0,1). We did this by computing the sum of the squared distances from (0,1) on the ROC curve at every point. The point with the lowest sum of squared distance corresponded to the closest point to (0,1). The closest point to (0,1) provide the best true positive to false positive ratio for classification. We decided to use the threshold value that provided this point as the threshold value we would use for actual classification to get the best ratio of true positives to false positive in our actual predictions.

H. Classifying patients in testing set

Since we now had our optimal threshold and our specific lambda for constructing the best model with only the important features being weighted, we now were able to classify the patients in the test set at every time step.

We trained a new model on the entire training set, using the specific lambda value in the loss function of the model. We then calculated prediction scores at every time step of the test data set by using predict_proba on the model (extracting only second column) with the input as the testing feature matrix. We then assigned each patient at the timestep to a 0 or a 1 based on whether or not their prediction score was equal to or above our threshold.

I. Determining EWT times of patients with Septic Shock

EWT (Early warning time) is calculated by finding the difference between the time of prediction of Septic Shock provided by our model and the actual diagnosis of Septic Shock. Once we classified patients in the test set as 0 or 1 at all timesteps based on our model, we identified the indexes in the test set where patients were first diagnosed with Septic shock and the indexes where we first predicted these patients go into Septic Shock. We derived the times of diagnosis from the test.onsets dataset (onset of actual patients) and we derived the time we first predict Septic Shock for these patients from the timestamps.test data set (timestamps of all of test data). We took the difference between these two times for each patient that did in fact develop Septic Shock. If a prediction was not made for a patient that did in fact develop Septic Shock, the difference was set to 0. The differences of these times, or EWT, were plotted over a histogram on a log scale to show the distribution of early predictions we made.

J. Data Balancing

Without Data Balancing, our stochastic model performs poorer. This is due to having so many patient timestamps with Sepsis in comparison to patient timestamps that describe Septic Shock – the higher the occurrence rate is in a dataset, the easier for it is for a model to correctly make classifications. In this case the occurrence rate was low. We used data balancing to penalize in the loss function every incorrect identification of someone with Septic Shock. Theoretically, to maximize a threshold accuracy, we could diagnose no one with Septic Shock and still technically be accurate - data balancing prevented this.

III. RESULTS

A. Risk Factors

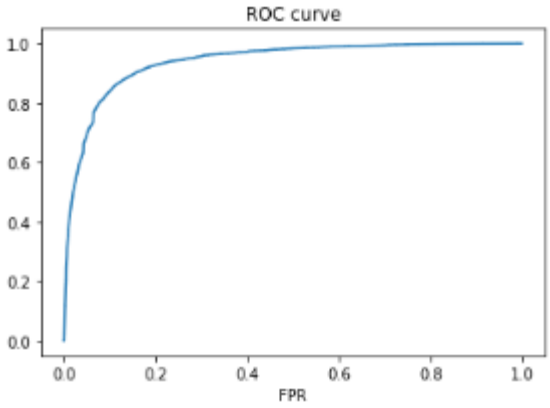
With this new, optimized lambda of 0.0008, we were able to narrow down our 28 features to 16 with associated weights. Below, we listed the results that overwhelmingly influenced the model.

FEATURES WITH NOTABLE WEIGHTS

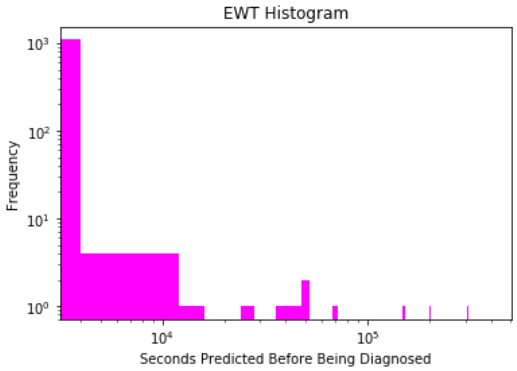
Features with Weights over 1	Weight
SBP	-2.319
GCS	-2.177
Lactate	1.781
Nervous SOFA	-1.874
Cardio SOFA	1.427

B. Receiver Operating Characteristic Curve (ROC)

Threshold with Least Error	True Positive Rate at Threshold	False Positive Rate at Threshold	Area Under the Curve
0.488	0.878	0.129	0.940



C. Histogram of the Early Warning Time (EWT)



V. CONCLUSIONS

We were able to correctly predict septic shock before diagnosis in 28 patients out of the 1114 patients that truly developed septic shock in the testing dataset. As shown in the histogram, we were able to predict the majority of these patients going into shock with an EWT between 0 and 10,000 seconds (around 2.7 hours). However, we were also able to predict some occurrences of shock with EWT around 50,000 seconds (around 14 hours) and few occurrences much in advance (over 28 hours in advance). Given these EWT times, a physician could realistically use our model to receive an alert and act for a patient that is likely to go into septic shock. The EWT times are large enough to give a physician enough time to make a decision whether to introduce a different treatment method before onset of septic shock. Given that the majority of our correct predictions were within 2.7 hours of septic shock, physicians would likely receive prediction alerts within 3 hours before shock. Our model could also provide early warning times around half a day before shock, so physicians may consider keeping a nurse next to the patient at all times for half a day when an alert is given. While our model does not achieve a suitable performance to be introduced clinically (it does not suitably predict enough patients that go into Septic shock), it does provide an example of how predicting septic shock in real time is feasible in the ICU.

REFERENCES

[1] Fleischmann C, Scherag A, Adhikari NKJ, Hartog CS, Tsaganos T, Schlattmann P, et al. *Assessment of global incidence and mortality of hospital-treated sepsis—current estimates and limitations.* Am J Respir Crit Care Med. 2016;193:253–272. doi: 10.1164/rccm.201504-0781OC.

[2] R. Liu, J. L. Greenstein, S. J. Granite, J. C. Fackler, M. M. Bembea, S. V. Sarma, and R. L. Winslow. *Data-driven discovery of a novel sepsis pre-shock state predicts impending septic shock in the icu.* Scientific Reports, 9(1):6145, 2019. doi: 10.1038/s41598-019-42637-5. URL https://doi.org/10.1038/ s41598-019-42637-5.